

# KDA RIVET



**2009/2010 Gruppe 11**



# KDA RIVET



## Prosjektplan

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth



## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                              | 4  |
| 1.1   | Hensikt med dokumentet.....                    | 4  |
| 1.2   | Definisjoner, akronymer og forkortelser.....   | 4  |
| 1.3   | Dokumenthistorie.....                          | 4  |
| 1.4   | Ansvarlig.....                                 | 4  |
| 2     | Prosjektoversikt.....                          | 5  |
| 2.1   | Hensikten med prosjektet, mål og omfang.....   | 5  |
| 2.2   | Prosjektdokumentasjon som skal leveres.....    | 6  |
| 3     | Prosjektorganisasjon.....                      | 8  |
| 3.1   | Prosjektgruppen.....                           | 8  |
| 3.2   | Eksterne ressurser.....                        | 8  |
| 3.3   | Roller og ansvarsforhold.....                  | 8  |
| 3.3.1 | Administrative ansvarsområder.....             | 8  |
| 3.3.2 | Tekniske ansvarsområder.....                   | 9  |
| 3.3.3 | Ressurser.....                                 | 10 |
| 3.3.4 | Veiledere og sensor.....                       | 10 |
| 4     | Prosjektstyring.....                           | 12 |
| 4.1   | Kategorier.....                                | 12 |
| 4.2   | Aktiviteter.....                               | 12 |
| 4.3   | Gantt-diagrammer over fasene i prosjektet..... | 13 |
| 4.4   | Hovedmilepæler.....                            | 13 |
| 4.5   | Iterasjoner.....                               | 13 |
| 4.6   | Prosjektpressur.....                           | 13 |
| 4.6.1 | Menneskelige ressurser.....                    | 13 |
| 4.6.2 | Programvare.....                               | 13 |
| 4.6.3 | Maskinvare.....                                | 14 |
| 4.7   | Budsjett.....                                  | 14 |
| 4.8   | Prosjektovervåking og kontroll.....            | 14 |
| 4.8.1 | Møter.....                                     | 14 |
| 4.8.2 | Planer og rapporter.....                       | 14 |

|       |  |    |
|-------|--|----|
| 4.8.3 | Timelister.....                                | 14 |
| 4.9   | Kvalitetsstyring .....                         | 14 |
| 4.10  | Prosjektframføring.....                        | 15 |
| 4.11  | Dokumenter.....                                | 15 |
| 4.12  | Produkt .....                                  | 15 |
| 4.13  | Arbeidsmiljø.....                              | 15 |
| 4.14  | Backup .....                                   | 15 |
| 5     | Teknisk prosessplan .....                      | 16 |
| 5.1   | Programmeringsteknikker .....                  | 16 |
| 5.2   | Metoder, verktøy og teknikker.....             | 16 |
| 5.3   | Dokumentasjon og programmeringsstandarder..... | 16 |
| 5.3.1 | Kode.....                                      | 16 |
| 5.3.2 | Dokumentasjon .....                            | 17 |
| 6     | Gantt-diagram.....                             | 18 |
| 6.1   | Prosjektet.....                                | 18 |
| 6.2   | 1. iterasjon .....                             | 19 |
| 6.3   | 2. iterasjon .....                             | 20 |
| 6.4   | Iterasjon 3.....                               | 21 |
| 6.5   | Iterasjon 4.....                               | 22 |
| 6.6   | Iterasjon 5.....                               | 23 |
| 6.7   | Iterasjon 6.....                               | 24 |
| 6.8   | Iterasjon 7.....                               | 25 |
| 6.9   | Iterasjon 8.....                               | 26 |
| 6.10  | Iterasjon 9.....                               | 27 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Prosjektplanen er et dokument som forteller retningslinjene bak prosjektet og dets overordnede mål. Her skal all relevant informasjon samles og brukes av prosjektleder for å styre og planlegge prosjektet, samt fortelle prosjektdeltakerne om hva som skal gjøres og hvilke andre aktiviteter de er avhengige av.

Dokumentet opprettes under inception-fasen og oppdateres ved hver hovedmilepel.

## 1.2 Definisjoner, akronymer og forkortelser

**RIVET:** RWS In VBS2 Environment Testing

**VBS2:** Virtual Battle Space 2

**RWS:** Remote Weapon Station

**KDA:** Kongsberg Defence and Aerospace

**HiBu:** Høgskolen i Buskerud

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Ansvar                                |
|---------------|----------|--------------------------------|---------------------------------------|
| 2.1           | 10.05.10 | Oppdatert                      | Thomas Transeth                       |
| 2.98          | 19.05.10 | Gjennomgang av egne dokumenter | Thomas Transeth                       |
| 2.99          | 23.05.10 | Korrekturlest                  | Lone Knutsen og Arne Kristian Åmellem |
| 3.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                       |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Prosjektoversikt

---

### 2.1 Hensikten med prosjektet, mål og omfang

Oppgaven er gitt av KDA fordi de ønsker en studie av mulighetene til å integrere Protector simulatoren sin i VBS2.

Systemet som allerede finnes er en trener for RWS systemet Protector. Dette er en simulator som tilbyr de samme funksjonalitetene som det virkelige våpenet har. KDA leverer mobile klasseromsløsninger der en kan sette opp alle posisjonene som finnes på et kjøretøy, samt en instruktørplass. Hensikten med denne simulatoren er å trene elevene i krigssituasjoner i et uskarpt miljø.

KDA ønsker å oppdatere systemet sitt fordi den nåværende motoren de bruker begynner å bli utdatert, og ved en eventuell kundeforespørsel om å bruke VBS2 må de ha kunnskap om det er mulig. VBS2 er en simulatormotor som brukes av blant annet det britiske forsvaret for kampsimuleringer. Vi skal se på mulighetene til å implementere RWS delen av Protector simulatoren ved å lage en add-on til VBS2.

## 2.2 Prosjektdokumentasjon som skal leveres

| Dokumentnavn                          | Utarbeides i   |
|---------------------------------------|----------------|
| Analysedokument                       | 1-4. iterasjon |
| Brukermanual                          | 6-9. iterasjon |
| Designdokument                        | 3-8. iterasjon |
| Dokumentstandarder                    | 1-8. iterasjon |
| Forbedringsdokument                   | 9. iterasjon   |
| Fornøyelsesdokument                   | 3-9. iterasjon |
| Forstudierapport                      | 1. iterasjon   |
| Fraværsprotokoll                      | 1. iterasjon   |
| Gjennomgang av UML                    | 3-5. iterasjon |
| Implementasjonsdokument               | 4-8. iterasjon |
| Iterasjonsplan 1                      | 1. iterasjon   |
| Iterasjonsplan 2                      | 1. iterasjon   |
| Iterasjonsplan 3                      | 2. iterasjon   |
| Iterasjonsplan 4                      | 3. iterasjon   |
| Iterasjonsplan 5                      | 4. iterasjon   |
| Iterasjonsplan 6                      | 5. iterasjon   |
| Iterasjonsplan 7                      | 6. iterasjon   |
| Iterasjonsplan 8                      | 7. iterasjon   |
| Iterasjonsplan 9                      | 8. iterasjon   |
| Iterasjonsrapport 1                   | 9. iterasjon   |
| Iterasjonsrapport 2                   | 2. iterasjon   |
| Iterasjonsrapport 3                   | 3. iterasjon   |
| Iterasjonsrapport 4                   | 4. iterasjon   |
| Iterasjonsrapport 5                   | 5. iterasjon   |
| Iterasjonsrapport 6                   | 6. iterasjon   |
| Iterasjonsrapport 7                   | 7. iterasjon   |
| Iterasjonsrapport 8                   | 8. iterasjon   |
| Iterasjonsrapport 9                   | 9. iterasjon   |
| Iterasjonsutsettelsesdokument         | 3. iterasjon   |
| Kontaktinfo                           | 1-9. iterasjon |
| Kravdokument                          | 1-5. iterasjon |
| Kodestandard                          | 3-8. iterasjon |
| Kostnadsdokument                      | 1-9. iterasjon |
| Prosjektplan                          | 1-9. iterasjon |
| Prosjektrapport                       | 9. iterasjon   |
| Prototypedokument - dll               | 2-5. iterasjon |
| Prototypedokument - Joystick          | 5. iterasjon   |
| Prototypedokument - web               | 2-5. iterasjon |
| Prototypedokument - wxWidgets         | 2-5. iterasjon |
| Risikodokument                        | 1-8. iterasjon |
| Teknologidokument – ASI vs API        | 1-9. iterasjon |
| Teknologidokument – Brukergrensesnitt | 8-9. iterasjon |
| Teknologidokument – Servosimulering   | 8-9. iterasjon |



|                                   |                |
|-----------------------------------|----------------|
| Teknologidokument – VBS2          | 1-8. iterasjon |
| Teknologidokument – VBS2 - Addon  | 1-8. iterasjon |
| Teknologidokument – VBS2 - Script | 8-9. iterasjon |
| Teknologidokument – wxJoystick    | 7-8. iterasjon |
| Teknologidokument – WxWidgets     | 1-8. iterasjon |
| Testspesifikasjon                 | 1-8. iterasjon |
| Teststrategi                      | 1-8. iterasjon |
| Utvidelse av oppgaven             | 9. iterasjon   |

## 3 Prosjektorganisasjon

---

### 3.1 Prosjektgruppen

Prosjektgruppen har følgende fem medlemmer:

| Navn                  | Initialer | Alder | E-post                        | Telefon    |
|-----------------------|-----------|-------|-------------------------------|------------|
| Thomas Transeth       | TT        | 26    | thomas.transeth@gmail.com     | 986 97 853 |
| Ole-Martin Grøtterud  | OMG       | 21    | olemartin.grotterud@gmail.com | 991 64 935 |
| Lone Knutsen          | LK        | 21    | lone.knutsen@gmail.com        | 481 56 687 |
| Arne Kristian Åmellem | AKÅ       | 29    | selestcayal@hotmail.com       | 920 85 997 |
| Zoran Vukobratovic    | ZV        | 22    | bigz33@gmail.com              | 958 74 449 |

### 3.2 Eksterne ressurser

Som kommunikasjonskanal utad har vi opprettet følgende webside:

<http://project-rivet.com/>

Denne inneholder en beskrivelse av prosjektet og gruppen samt funksjonalitet der en kan logge seg på for å få tilgang til alle produserte dokumenter.

### 3.3 Roller og ansvarsforhold

- Thomas Transeth                      Ledelse og implementasjon
- Ole-Martin Grøtterud                Planlegging og analyse
- Lone Knutsen                            Dokument og krav
- Arne Kristian Åmellem                Web og design
- Zoran Vukobratovic                    Økonomi og test

#### 3.3.1 Administrative ansvarsområder

Alle gruppemedlemmene har minst ett administrativt ansvarsområde.

##### 3.3.1.1 Prosjektleder

Prosjektlederens ansvar er å styre prosjektet slik det holder seg innenfor rammene som prosjektplanen og HiBu definerer. Prosjektlederen har også det overordnede ansvaret for prosjektet og er gruppens kommunikasjonskanal mot eksternt veileder og sensor.

Ansvarlig: Thomas

##### 3.3.1.2 Planlegging

Planleggingsansvarlig har ansvaret for prosjektets iterasjonsplan og sammen med prosjektleder samt sørge for at denne blir utviklet innenfor prosjektets rammer.

Ansvarlig: Ole-Martin

### **3.3.1.3 Dokumentasjon**

Dokumentansvarlig har ansvaret for at gruppens dokumenter er strukturert riktig og likt, at det er laget dokumentmaler for alle offisielle dokumenter og ansvaret for at tidligere dokumenter blir oppdatert med nye, godtatte stiler.

Ansvarlig: Lone

### **3.3.1.4 Økonomi**

Økonomiansvarlig har ansvaret for å sette opp budsjett og holde regnskap for prosjektet. Personen skal passe på at innkjøp ikke overskrider de satte grensene og om budsjettet sprekker, og ta kontakt med arbeidsgiver for klare opp i dette.

Ansvarlig: Zoran

### **3.3.1.5 Backup**

Backupansvarlig har ansvaret for at alle dokumenter som er levert av gruppemedlemmer er forsvarlig lagret og sikkerhetskopierte.

Ansvarlig: Thomas

### **3.3.1.6 Web**

Webansvarlig har ansvaret for at gruppens webside er oppdatert med riktig informasjon og publikasjoner.

Ansvarlig: Arne Kristian

## **3.3.2 Tekniske ansvarsområder**

Alle gruppemedlemmene har minst ett teknisk ansvarsområde.

### **3.3.2.1 Analyse**

Analyseansvarlig har ansvaret for å analysere use-caser, analyseklasser, samt utvikle analysedokument.

Ansvarlig: Ole-Martin

### **3.3.2.2 Design**

Designansvarlig har ansvar for å lage en beskrivelse av systemet i planleggingsfasen gjengitt i klasser, objekter og tilstands/aktivitetsdiagrammer. Personen har også ansvar for at et designdokument blir produsert.

Ansvarlig: Arne Kristian

### **3.3.2.3 Krav**

Kravansvarlig har ansvar for å identifisere krav og use-caser, samt utvikle en kravspesifikasjon.

Ansvarlig: Lone

### **3.3.2.4 Implementasjon**

Implementasjonsansvarlig har ansvar for å utvikle et dokument for kodestandard og sørge for at all kode gruppen utvikler følger denne. Personen har også ansvar for at systemets ulike

delar kan settes sammen som beskrevet i respektive dokumenter, samt utvikle et implementasjonsdokument.

Ansvarlig: Thomas

### **3.3.2.5 Arbeidsstatistikk**

Arbeidsstatistikkansvarlig har ansvaret for at alt arbeid blir ført inn i lister slik at planleggingsansvarlig og prosjektleder kan følge med på tidsaspektet av arbeid som blir gjort.

Ansvarlig: Arne Kristian

### **3.3.2.6 Test**

Testansvarlig har ansvar for testingen av prosjektet ved å utarbeide teststrategi, testrapporter og testspesifikasjon. Personen skal også jobbe mot krav og hele tiden sørge for at disse er testbare, samt organisere kvalitetslesing av kode.

Ansvarlig: Zoran

## **3.3.3 Ressurser**

### **3.3.3.1 Kompetanse**

Prosjektgruppen har fordelt ansvarsområdene ut fra hvert medlems ønske, og egendefinerte evner. På denne måten har vi fastsatt at gruppemedlemmenes samlede kunnskaper er nok for å løse prosjektoppgaven. Prosjektleder har likevel mulighet til å bytte om på ansvarsområdene om det skulle vise seg nødvendig, i henhold til feilevaluering og dypere utfylling av risikoområder. Disse endringene skal ha minimal konsekvens for prosjektplanen.

### **3.3.3.2 Sykdom og permisjon**

Skulle det uforutsette inntre og et gruppemedlem skulle bli syk eller trenge permisjon vil prosjektgruppen som helhet utarbeide en ny ansvarsfordeling. Om frafallet er i så stor grad at selve prosjektutarbeidelsen står i fare, skal prosjektplanen endres i fullt samarbeid med ekstern veileder og HiBu.

### **3.3.3.3 Ferie/Eksamen**

All ferie og eksamener skal i meget god tid meldes fra om til prosjektleder slik at arbeidet kan tilpasses og ferdigstilles før oppholdet trer i kraft. Det er da gruppemedlemmets ansvar at arbeidet blir utført.

## **3.3.4 Veiledere og sensor**

### **3.3.4.1 Ekstern sensor**

Rune Ljøterud

### **3.3.4.2 Ekstern veileder**

Rune Ljøterud

### ***3.3.4.3 Intern sensor***

Olaf Hallan Graven

### ***3.3.4.4 Intern veileder***

Karoline Moholth



## 4 Prosjektstyring

---

### 4.1 Kategorier

| Nummer | Type                              |
|--------|-----------------------------------|
| 1xx    | Administrasjon og prosjektstyring |
| 2xx    | Research                          |
| 3xx    | Krav/analyse                      |
| 4xx    | Design                            |
| 5xx    | Implementasjon                    |
| 6xx    | Test                              |

### 4.2 Aktiviteter

| Nummer | Aktivitet                    |
|--------|------------------------------|
| 100    | Møter                        |
| 101    | Prosjektstyring              |
| 102    | Korrekturlesing              |
| 103    | Prosjektplan                 |
| 104    | Presentasjoner               |
| 105    | Dokumentstandard             |
| 106    | Økonomi                      |
| 107    | Diskusjon                    |
| 108    | Dokumentmaler                |
| 109    | Møterefertat                 |
| 110    | Møteinnkalling               |
| 111    | Bedriftsmøte                 |
| 112    | Webutvikling                 |
| 113    | Planlegging                  |
| 114    | Backup                       |
| 115    | Kvalitetssikring             |
| 116    | PR                           |
|        |                              |
| 200    | VBS2 funksjoner og bruk      |
| 201    | Trainer-funksjoner           |
| 202    | Interface joystick - trainer |
| 203    | Add-ons                      |
| 204    | Forskjeller VTK - Fusion     |
| 205    | WxWidgets                    |
|        |                              |
| 300    | Krav                         |
| 301    | Use case                     |
| 302    | Kravspesifikasjon            |
| 304    | Analyse                      |

| Nummer | Aktivitet             |
|--------|-----------------------|
| 305    | Forstudie             |
| 306    | Risikoanalyse         |
| 307    | Brukermanual          |
|        |                       |
| 400    | Klasser               |
| 401    | Designdokument        |
| 402    | Designdiagrammer      |
|        |                       |
| 500    | Kodestandard          |
| 501    | Prototype - dll       |
| 502    | Implementasjon        |
| 503    | Impl. dokumentasjon   |
| 504    | Testing av teknologi  |
| 505    | Prototype - wxWidgets |
|        |                       |
| 600    | Teststrategier        |
| 601    | Testspesifikasjon     |
| 602    | Testrapport           |
| 603    | Test                  |
| 604    | Kodeinspeksjon        |

### 4.3 Gantt-diagrammer over fasene i prosjektet

Til hver iterasjon blir det laget et gantt-diagram som viser tidsestimat av aktivitetene. Disse diagrammene blir lagt til som vedlegg.

### 4.4 Hovedmilepæler

- 1. Presentasjon 12.januar
- 2. Presentasjon 25.mars
- 3. Presentasjon 10.juni

### 4.5 Iterasjoner

Vi har valgt å bruke utviklingsstrategien Unified Process og det medfører at vi deler prosjektet opp i iterasjoner.

I starten av hver iterasjon blir det utviklet en detaljert iterasjonsplan som forteller hvilket arbeid og aktiviteter vi skal utføre innenfor perioden. Dette er ment som en sikring slik at vi alltid vet hva som skal gjøres og når. Et estimat av arbeidstimer som prosjektgruppen skal bruke i perioden blir også beskrevet her.

Etter endt iterasjon legges det ved en iterasjonsrapport som forklarer hva som faktisk har blitt gjort og det gjøres en sammenligning mot timelistene på hvor mange timer prosjektgruppen egentlig brukte.

### 4.6 Prosjektkressurser

#### 4.6.1 Menneskelige ressurser

Prosjektet har et estimert tidsforbruk på 2500 timer.

#### 4.6.2 Programvare

KDA har levert følgende programvare til prosjektgruppen:

- PROTECTOR treningssimulator  
Dette er simulatoren som skal bli implementert i VBS2.
- VBS2  
Dette er en treningssimulator som blir brukt til militære treningssituasjoner verden over.

### 4.6.3 Maskinvare

Siden PROTECTOR er et kommersielt produkt som KDA leverer ønsker de ikke at denne kommer ut for allmennheten. De har derfor valgt å utstyre oss med følgende maskinvare:

- Stasjonær PC
- 2 stk 19" skjerm
- Tastatur og mus
- Joystick

### 4.7 Budsjett

I følge prosjektkontrakten er KDA ansvarlige for å dekke eventuelle kostnader prosjektet skulle ha. Vi har derfor satt opp følgende budsjett:

| Type:                   | Totalt:            |
|-------------------------|--------------------|
| Fremføringsutgifter     | kr 1 500,00        |
| Reiseutgifter           | kr 2 000,00        |
| Administrative utgifter | kr 600,00          |
| <b>Sum</b>              | <b>kr 4 100,00</b> |

### 4.8 Prosjektovervåking og kontroll

#### 4.8.1 Møter

Prosjektgruppen har ukentlige møter (om ikke annet er spesifisert) med intern veileder. Her vil prosjektets framgang og kvalitet bli diskutert, og intern veileder stiller med rådføring og oppfølging.

Til disse møtene vil det alltid foreligge et oppfølgingsdokument.

Møter med ekstern veileder og arbeidsgiver vil bli avholdt om nødvendig, slik at prosjektet utvikler et produkt som er nærmest mulig spesifikasjonene.

#### 4.8.2 Planer og rapporter

Det er alltid dokumenter, planer og rapporter som blir utviklet i løpet av hver iterasjon. Dette gjør vi for å skape kontroll og flyt i prosjektet, samt god dokumentasjon av prosjektprosessen.

Følgende viktige dokumenter blir oppdatert for hver iterasjon:

- Prosjektplan og prosjektrapport
- Iterasjonsplan og iterasjonsrapport

#### 4.8.3 Timelister

For å få en best mulig gjengivelse av prosjektgruppens tidsforbruk, skal hvert gruppelem oppdatere timelisten sin i timelistesystemet for hver endte arbeidsdag.

### 4.9 Kvalitetsstyring

Kvalitet er noe prosjektgruppen har satt som et mål for prosjektet. Følgende vil bli gjort for å sikre dette:

#### **4.10 Prosjektframføring**

Kvalitetssikring av prosjektframføringen består i første omgang å dele ut et ark med spørsmål under første presentasjon. Dette dokumentet lar de som har hørt på presentasjonen evaluere hva de synes er bra og dårlig.

I etterkant vil det bli laget en analyserapport over hvordan publikum har evaluert framføringen. Gruppen vil da ta dette til rette og endre neste presentasjon med følge for å gjøre den bedre.

Se tilbakemeldingsrapport 1 og 2.

#### **4.11 Dokumenter**

For å kvalitetssikre prosjektgruppens dokumenter blir det til hver iterasjon korrekturlest av et annet gruppemedlem enn den som skrev det. På denne måten kan vi unngå dårlige setninger og feilstavinger.

#### **4.12 Produkt**

For å sikre at prosjektgruppen leverer et produkt som svarer mot arbeidsgivers forventninger vil det legges mye arbeid i testing. For mer informasjon se testdokumentene.

#### **4.13 Arbeidsmiljø**

Det er viktig at prosjektgruppen trives med å arbeide med prosjektet og derfor passes det på at vi belønner oss selv ved blant annet endt iterasjon. Se fornøydelsesdokument for mer informasjon.

#### **4.14 Backup**

For å sikre at prosjektgruppens dokumenter alltid er tilgjengelige og trygt lagret er det tatt i bruk et eksternt backup-system med navn Dropbox. Backupansvarlig har og ansvar for å ta ukentlige kopier av prosjektmappen. Disse vil bli lagret på en ekstert plassering som ikke er tilgjengelig for almenheten.

## 5 Teknisk prosessplan

---

### 5.1 Programmeringsteknikker

For å fullføre prosjektet på et softwaresystem har prosjektgruppen planlagt å bruke følgende språk:

| Navn        | Kunnskapsnivå |
|-------------|---------------|
| C++         | Høyt          |
| VBS2 script | Ingen         |

### 5.2 Metoder, verktøy og teknikker

Prosjektgruppen har valgt å bruke følgende metoder, verktøy og teknikker:

#### Utviklingsmetoder

- Rational Unified Process (RUP)
- Unified Modeling Language (UML)
- Scrum (ukentige møter)

#### Verktøy

- Microsoft Visual Studio 2008 SP1 (C++ programmering)
- Microsoft Office 2007 (dokument og rapport)
- Microsoft Project 2007 (Gantt-diagram)
- Dropbox (backup)
- wxWidgets
- VBS2 VTK
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop

Prosjektgruppen har valgt å ta i bruk en del av ad-hoc utviklingsmetoden Scrum, ved å starte dagen med en 15 minutters gjennomgang av hvilke arbeidsoppgaver som skal gjøres. Dette gir prosjektet en kvalitetskontroll ved at alle gruppemedlemmene vet hvor langt en er i prosjektet og hva som hele tiden gjøres

### 5.3 Dokumentasjon og programmeringsstandarder

Siden dette er et softwareprosjekt kommer prosjektgruppen til å produsere mye kode. Det er derfor viktig at dette blir gjort på en riktig og forståelig måte samt at dette blir dokumentert riktig.

#### 5.3.1 Kode

All kode som blir skrevet skal følge gruppens kodestandard. Dette er definert i kodestandarddokumentet som blir utviklet i iterasjon 2.

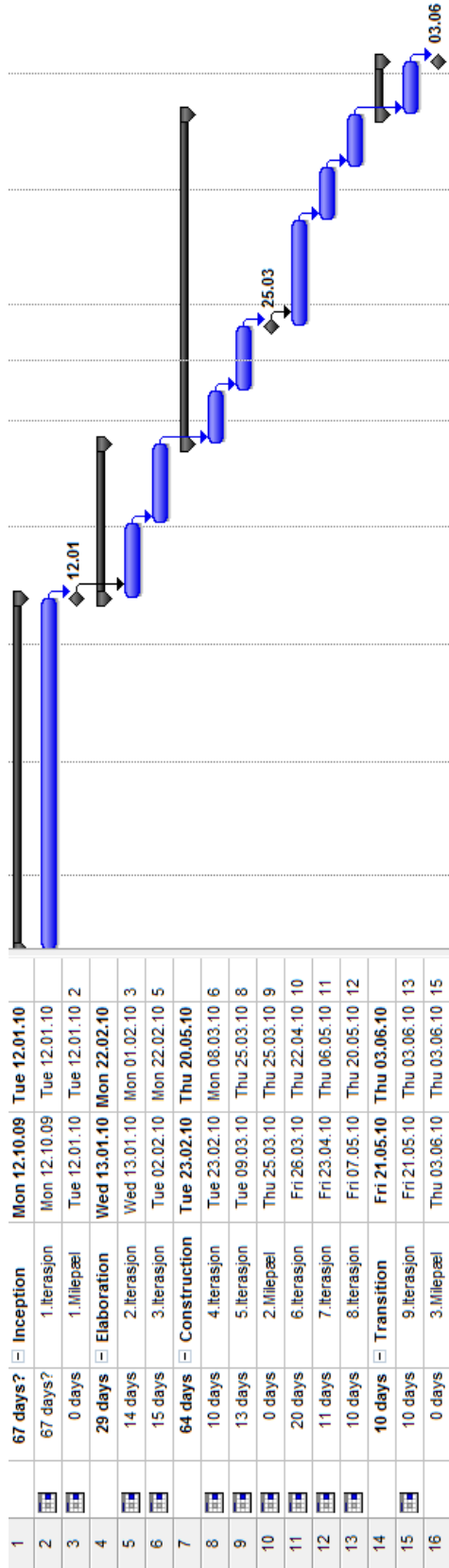


### 5.3.2 Dokumentasjon

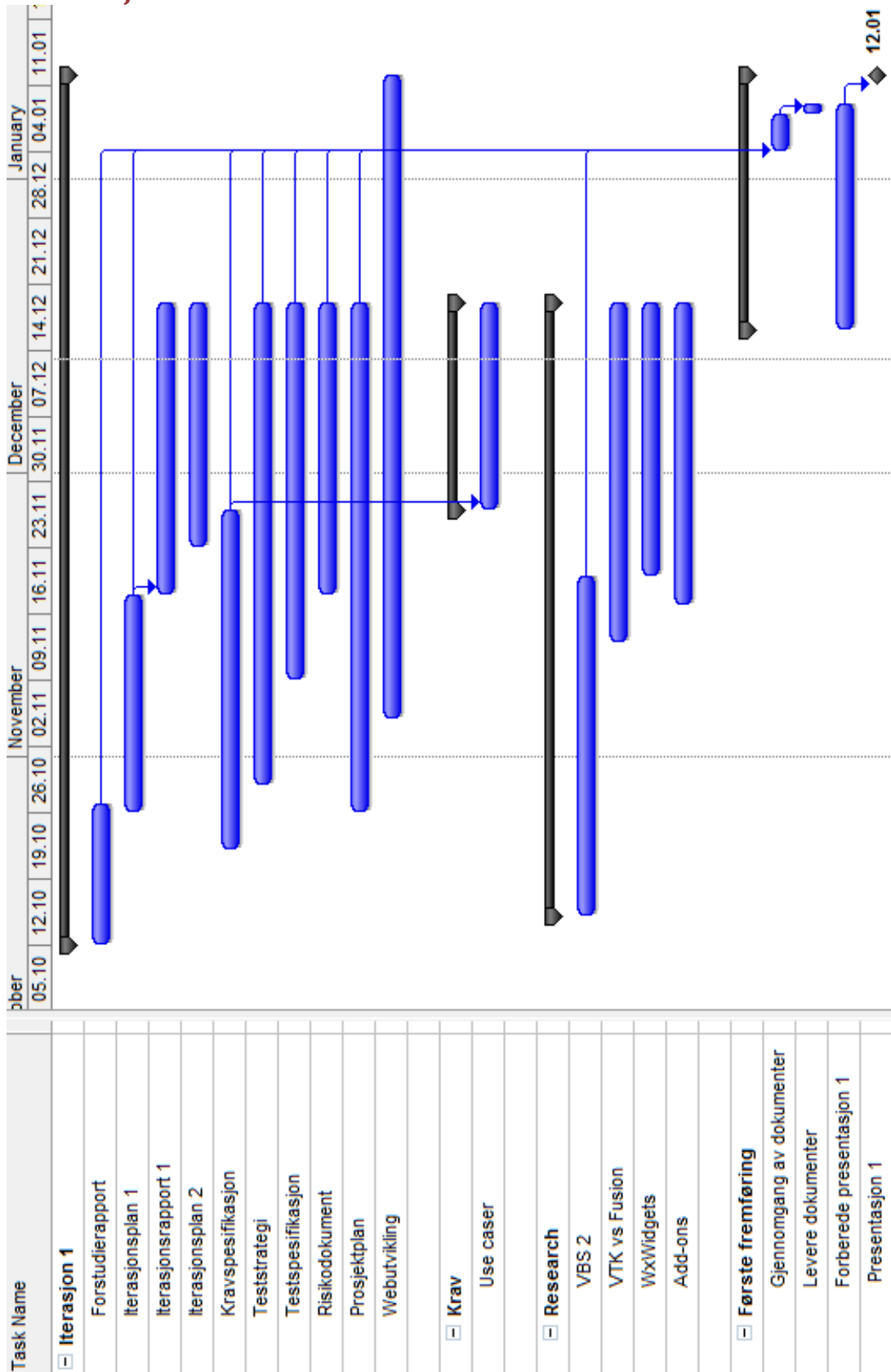
All kode skal dokumenteres og hvordan dette gjøres er beskrevet i kodenstandarddokumentet som blir utviklet i iterasjon 2. Dokumentasjonen skal også følge standarden som er satt i dokumentet dokumentstandarder.

## 6 Gantt-diagram

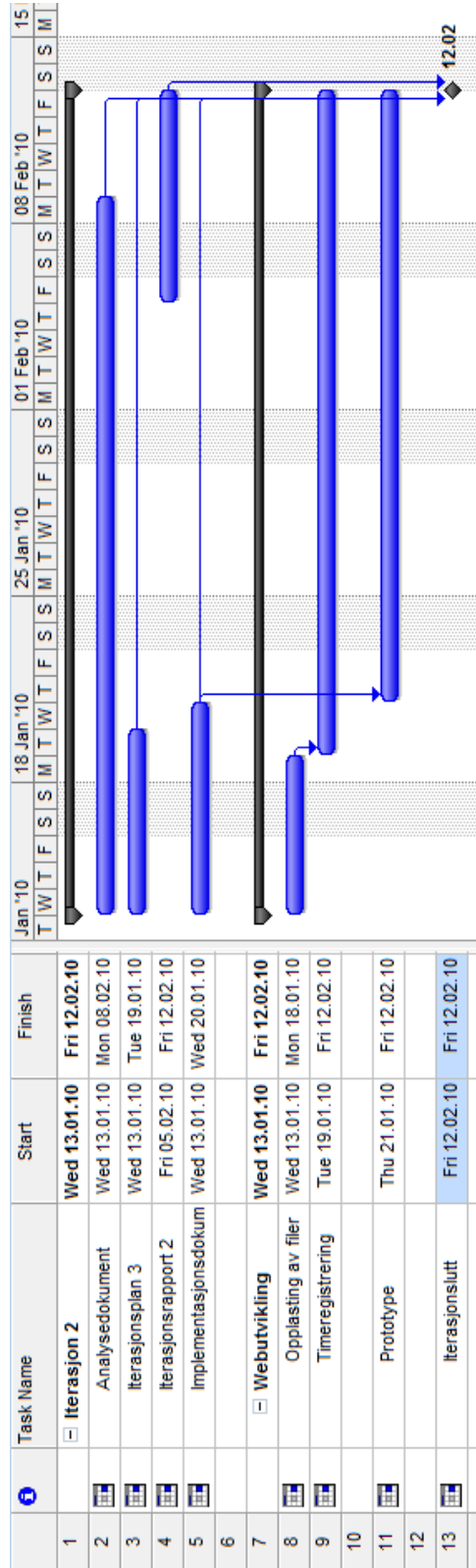
### 6.1 Prosjektet



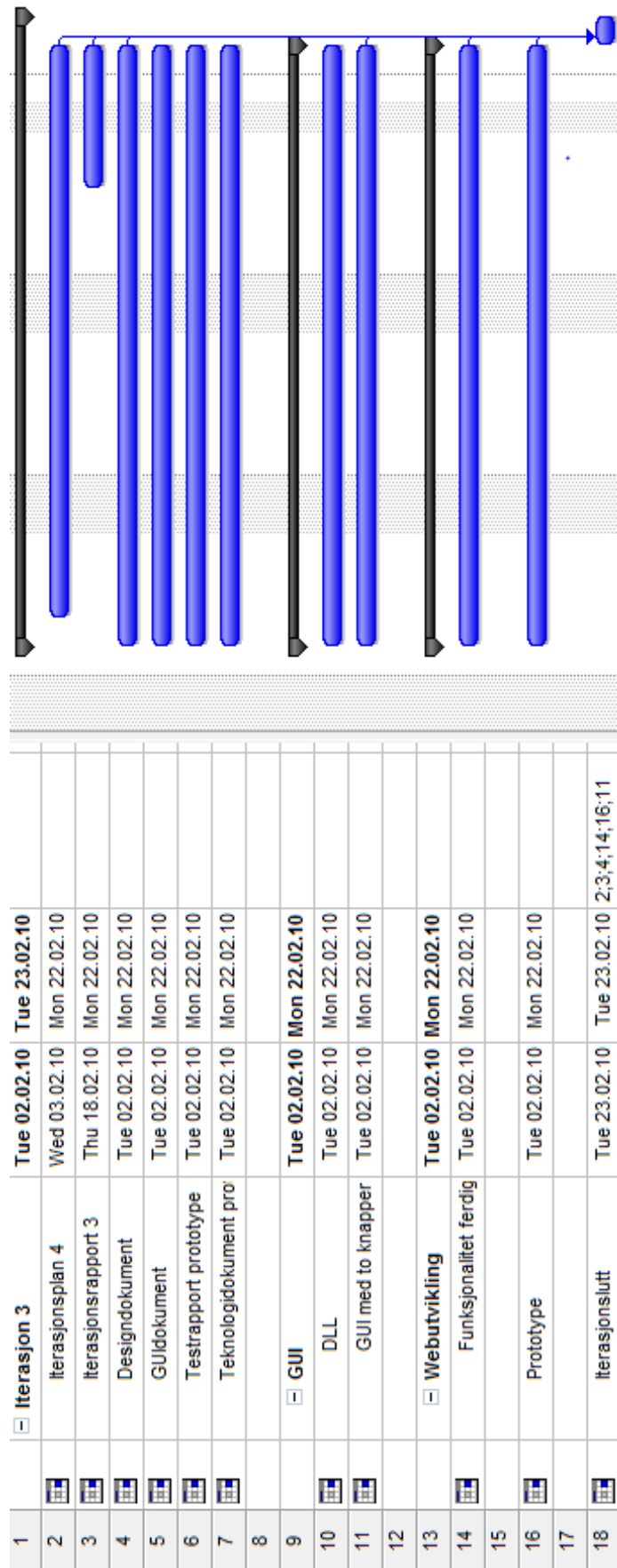
## 6.2 1. iterasjon



### 6.3 2. iterasjon

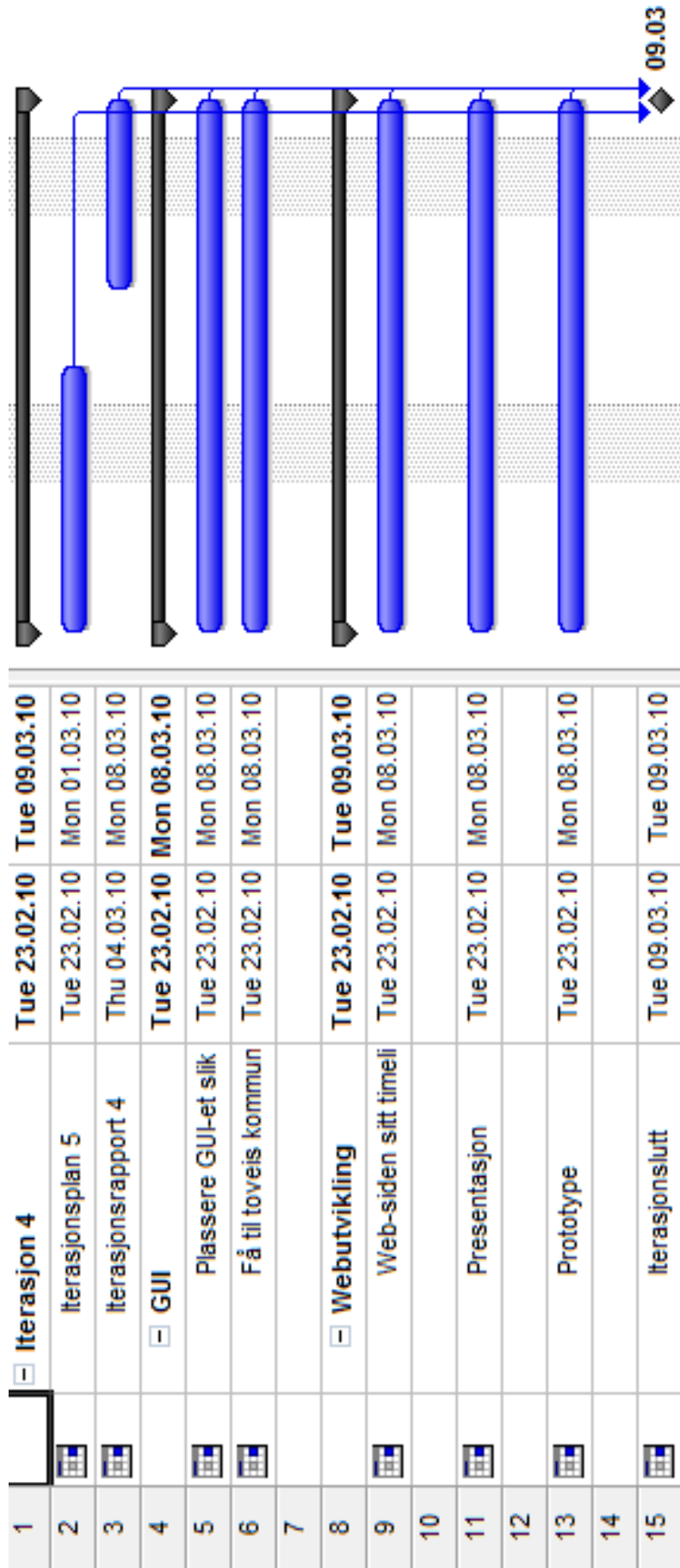


## 6.4 Iterasjon 3

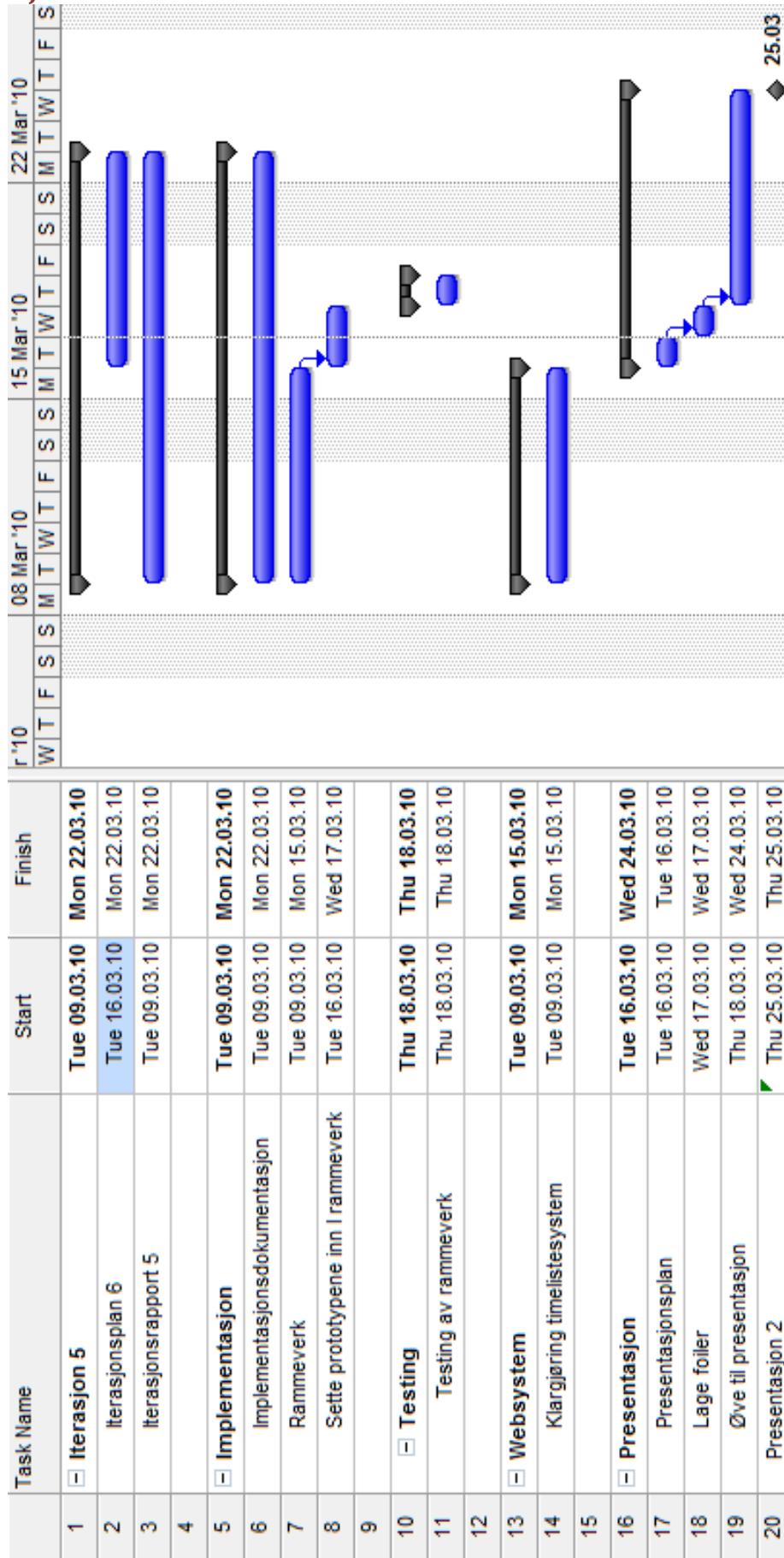




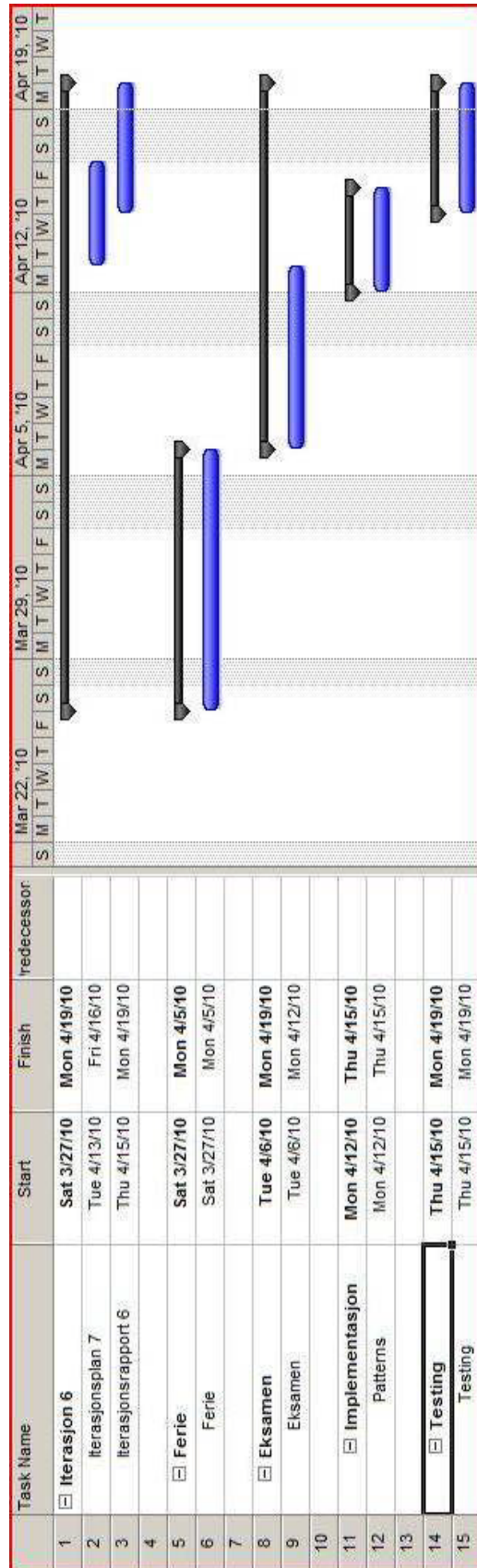
## 6.5 Iterasjon 4



## 6.6 Iterasjon 5



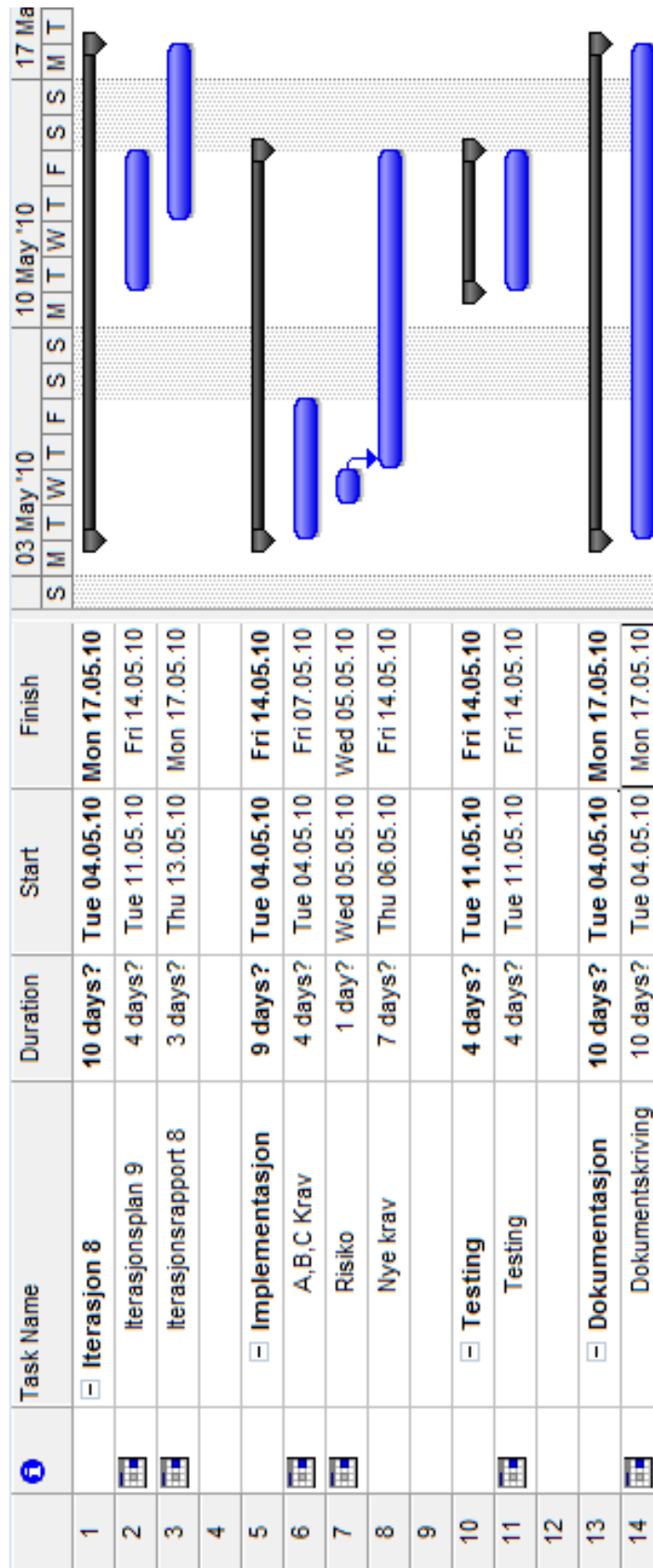
## 6.7 Iterasjon 6



## 6.8 Iterasjon 7



## 6.9 Iterasjon 8





# KDA RIVET



## Prosjektrapport

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....  | 4  |
| 1.1   | Hensikt med dokumentet.....                                  | 4  |
| 1.2   | Definisjoner, akronymer og forkortelser.....                 | 4  |
| 1.3   | Dokumenthistorie.....  | 4  |
| 1.4   | Ansvarlig .....  | 4  |
| 2     | Prosjektarbeidet.....  | 5  |
| 2.1   | Prosjektet.....  | 5  |
| 2.1.1 | Hvorfor vi valgte denne oppgaven.....                        | 5  |
| 2.1.2 | Oppdragsgiver .....  | 5  |
| 2.1.3 | Prosjektmodell og arbeidsfordeling.....                      | 5  |
| 2.1.4 | Måloppnåelse .....   | 6  |
| 2.2   | Timeforbruk.....   | 6  |
| 2.3   | Møter med intern veileder .....                              | 6  |
| 2.4   | Møter med ekstern veileder.....                              | 7  |
| 2.5   | Lokaler og tilgang.....                                      | 7  |
| 2.6   | Gruppemiljø og samarbeid .....                               | 7  |
| 2.7   | Kvalitetssikring.....  | 7  |
| 2.7.1 | Prosjektframføring .....                                     | 7  |
| 2.7.2 | Dokumenter .....   | 7  |
| 2.7.3 | Produkt.....   | 7  |
| 3     | Tekniske vurderinger og utfordringer.....                    | 8  |
| 3.1   | Simulatorsystem .....  | 8  |
| 3.1.1 | Bugs i simulatorsystemet .....                               | 8  |
| 3.2   | Utviklingen av prototyper.....                               | 8  |
| 3.3   | Oppretting av brukergrensesnittet.....                       | 8  |
| 3.4   | Utviklingen av et stabilt rammeverk .....                    | 9  |
| 3.5   | Videreutvikling av systemet .....                            | 9  |
| 4     | Gruppemedlemmenes egne kommentarer angående prosjektet ..... | 10 |
| 4.1   | Arne Kristian .....  | 10 |
| 4.2   | Lone .....   | 10 |





|     |                  |    |
|-----|------------------|----|
| 4.3 | Ole-Martin ..... | 10 |
| 4.4 | Thomas .....     | 11 |
| 4.5 | Zoran.....       | 11 |
| 5   | Konklusjon.....  | 12 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Dette dokumentet skal oppsummere prosjektet og gi en helhetlig tilbakemelding ut ifra hvordan prosjektmedlemmene har opplevd prosjektførløpet. Det blir her gitt en oppsummering over hva som har fungert og hva som kunne blitt gjort annerledes. Både tekniske, administrative og arbeidsmiljørelaterte temaer vil bli tatt opp.

## 1.2 Definisjoner, akronymer og forkortelser

**KDA:** Kongsberg Defence and Aerospace

**RWS:** Remote Weapon Station

**VBS:** Virtual Battle Space

**RUP:** Rational Unified Process

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                        | Redaktør                           |
|---------------|----------|----------------------------------|------------------------------------|
| 0.1           | 26.05.10 | Førsteutkast                     | Thomas Transeth                    |
| 0.1           | 27.05.10 | Videreutvikling                  | Thomas Transeth                    |
| 0.1           | 28.05.10 | Lagt til personlige evalueringer | Thomas Transeth                    |
| 0.99          | 28.05.10 | Korrekturlesing                  | Lone Knutsen<br>Zoran Vukobratovic |
| 1.0           | 28.05.10 | Klargjort for innlevering        | Thomas Transeth                    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Prosjektarbeidet

---

Vi, gruppemedlemmene i hovedprosjektgruppen KDA RIVET, har nå over en lengre periode jobbet sammen for å skape et produkt som er av en slik kvalitet en kan vente av profesjonelle aktører. Ved å ha en oppdragsgiver som venter å få tilbake et gjennomført system, har vi oppnådd mye. Da vi satt sammen gruppen og startet prosjektet for nesten ett år siden hadde vi liten anelse om de faktiske arbeidsmengdene og tekniske utfordringene som vi har støtt på. På denne måten har vi fått arbeide med teknologier vi ikke før har hatt kjennskap på, alle har fått testet sine administrative egenskaper og tilegnet seg nye og viktige egenskaper, både teknisk og sosialt.

### 2.1 Prosjektet

Vi valgte å kalle prosjektet vårt for KDA RIVET der RIVET står for "RWS In VBS2 Environment Testing". Oppgaven vår gikk ut på å lage og implementere brukergrensesnittet for kontrollenheten til KDA sin RWS-simulator inn i kamp- og treningssimulatoren VBS2. Simulatoren er en trener som blir brukt av mange lands militære avdelinger for å trene opp soldater i et sikkert og kontrollert miljø. Dette brukergrensesnittet, samt våpenfunksjonene skulle være så likt som mulig den eksisterende treneren.

#### 2.1.1 Hvorfor vi valgte denne oppgaven

Da vi stod ovenfor det å velge prosjektgruppe, fikk vi tidlig kontakt med KDA via allerede etablerte kontakter. Dette var oppnådd via skolens gode samarbeid med bedriften og kontaktordninger for studenter, og ikke minst via pappa'n til Lone. Vi fikk så flere tilbud, men ingen av dem var så godt mottatt hos gruppen som prosjektet vi bestemte oss for å jobbe med. Vi hadde også hørt mye godt om KDA som bedrift og visste at det ville gi en viss prestisje å ha prosjekt fra en så stor og veletablert bedrift.

#### 2.1.2 Oppdragsgiver

Oppdragsgiveren vår, Kongsberg Defence & Aerospace er en veletablert og anerkjent internasjonal teknologibedrift som tilhører Kongsberg Gruppen. KDA har som hovedområde å utvikle teknologi til forsvars- og romfartsteknologi, og er delt inn i flere spesialområder. Vi har i prosjektperioden jobbet for underavdelingen Defence Systems – Simulation & Training som baserer seg på å lage treningssimulatorer for eksisterende våpen og systemer.

#### 2.1.3 Prosjektmodell og arbeidsfordeling

Vi valgte tidlig å bruke prosjektmodellen RUP, en modell vi har vært borti via andre kurs i løpet av studietiden. Denne deles opp i fire faser: Inception, Elaboration, Construction og Transition. Disse fasene deler opp prosjektet i deler slik en kan konsentrere seg om de viktige elementene etter hvert som en beveger seg ut i prosjektiden.

Vi valgte denne modellen da den er godt dokumentert og testet for utviklingen av software-prosjekter. På denne måten visste vi hele tiden hva som var de viktige aktivitetene og kunne prioritere kritiske arbeidsoppgaver. Under er en forklaring på hvordan vi jobbet i de ulike fasene:

Første fase, Inception brukte vi til å dele opp arbeidsoppgavene og studere oppgaven i detalj slik at alle visste hva den dreide seg om. Vi skulle også sette opp alle kravene slik at vi hadde

en oversikt over hva som var forventet å bli gjort i løpet av prosjektet.

Andre fase, Elaboration, ble brukt til å analysere oppgaven, fylle ut mer krav og sette opp risikoer, samt utvikle en designmodell over hvordan prosjektet skulle utføres. Vi brukte også mye tid på å teste de ulike teknologiene vi skulle bruke samt lage prototyper for hver av disse.

Tredje fase, Construction ble brukt til å implementere teknologiene vi nå hadde blitt kjent med inn i et veldesignet system og det ble gjort kontinuerlig testing.

Fjerde fase, Transition, ble brukt i hovedsak til å fullføre oppgaven, teste den intenst og grundig, samt skrive ferdig all dokumentasjon.

## **2.1.4 Måloppnåelse**

### **2.1.4.1 Tekniske mål**

Når man får en arbeidsoppgave er så klart det viktigste målet å oppfylle alle krav som oppdragsgiveren stiller, og å utføre disse på en tilfredsstillende måte. Med god kontroll over egne kunnskaper utarbeidet vi en kravspesifikasjon sammen med oppdragsgiver der vi alltid hadde som mål å ferdigstille disse. Vi kan med god samvittighet si at alle krav som ble gitt har blitt innfridd!

For å vise at vi satte pris på muligheten vi fikk ved dette prosjektet tok vi også inn flere ønsker fra oppdragsgiver. Flere av disse ble implementert i systemet, men det viste seg også at noen ikke lot seg gjøre da simulatorsystemet ikke støttet disse.

### **2.1.4.2 Menneskelige ressurser og prosessrelaterte mål**

Alle i prosjektgruppen er studenter, noe som vil si at vi ikke er fullt utdannet og en ikke kan vente at vi er fullt utlært på alle punkter. Dette har gjenspeilet seg i kunnskapsmengden vi satt med i prosjektstart, men vi er heldigvis vant med å tilegne oss ny kunnskap og kompetansen vi manglet ble sakte, men sikkert økt til et nivå som gjorde oss i stand til å løse oppgaven. Så klart har vi gjort dårlige vurderinger underveis som har påvirket sluttresultatet og motivasjonen har til tider vært dårlig, men alt dette er viktig for å kunne videreutvikle personlige egenskaper. Ved å nå avslutte prosjektet og ved å ha jobbet oss fra start til mål, føler alle at dette er høyst nødvendig for å bli en god arbeidstaker og ingeniør.

## **2.2 Timeforbruk**

Noe av det som gruppen kanskje hadde minst erfaring med fra tidligere var det å estimere tidsforbruk. Vi merket spesielt dette i starten da estimatene ofte var ganske lang fra sannheten og ved å være fem ulike personer med ulik arbeidsevne og startsted merket vi raskt at dette var noe vi måtte lære gjennom å prøve. Etter hvert som prosjektet gikk og vi ble mer kjent med verktøyet vi brukte var det lettere å estimere tidsbruken. Vi er nok ingen mestere i dette enda, men ved å sette opp dagsmål og ukemål ble det lettere.

## **2.3 Møter med intern veileder**

Vi har under hele prosjektperioden passet på å holde ukentlige møter med intern veileder. Vi har hatt fast møtedag og tidspunkt, men pga uforutsette hendelser har ikke alltid møtet falt på samme dag. Samarbeidet med veileder ser vi på som positivt og vi har hatt god nytte av møtene for å kunne kontrollere prosessen og holde motivasjonen oppe.

## **2.4 Møter med ekstern veileder**

Vår eksterne veileder har til tider vært en travel mann, men han har alltid satt oss først. Siden KDA ofte har korte tidsfrister og det ikke alltid passer for en av partene har det ikke vært en fast tid for veiledermøter. Vi har likevel hatt ukentlig (til tider daglig) kontakt, og minst ett møte hver måned.

## **2.5 Lokaler og tilgang**

Det å ha en egen arbeidsplass, uforstyrret av andre er noe vi alle har sett som verdifullt. Da vi undertegnet kontrakten med KDA var en av avtalene at vi skulle ha et rom der vi kunne låse inn lånt utstyr, noe som førte til at vi fikk prosjekttrom samtidig som vi fikk låneutstyret. Siden vi kun hadde én lisens på simulatorprogrammet gjorde dette at mye av arbeidet måtte foregå i prosjekttrommet. Mange andre skoler har løsninger med døgnåpen tilgang for studenter, og dette er kanskje noe HiBu også bør revurdere. Det skal likevel nevnes at vi alltid har fått godkjent søknadene om helgejobbing og intern veileder har vært behjelpelig med å låse oss inn på skolen i disse periodene.

## **2.6 Gruppemiljø og samarbeid**

Da vi opprettet prosjektgruppen var det bevisst valgt medlemmer. Flesteparten av oss hadde allerede gode erfaringer med samarbeid og vi visste at vi fylte ut hverandre kunnskapsmessig. Vi tok også til oss ett gruppemedlem til da det viste seg at det var flere som ikke hadde fått gruppe enda.

Samarbeidet har gått noe opp og ned, og vi har lært hverandre å kjenne på helt nye måter, men dette har også ført til et nærmere vennskap mellom gruppemedlemmene. Vi ser på oss som en godt sammensveiset gruppe og vi har det ofte så moro at det høres langt bort i gangene.

## **2.7 Kvalitetssikring**

Kvalitet er noe prosjektgruppen har satt som et mål for prosjektet og det har blitt gjort en del tiltak for å sikre dette.

### **2.7.1 Prosjektframføring**

Under alle prosjektframføringene har det blitt lagt fram et tilbakemeldingsskjema som publikum har kunnet svare på. Disse bruker vi til å finne ut hva som har vært bra og dårlig under presentasjonen

### **2.7.2 Dokumenter**

For å sikre at dokumentene holder mål har de jevnlig blitt undersøkt og korrekturlest.

### **2.7.3 Produkt**

For at et produkt skal bli bra og tilnærmet feilfri må det testes. Vi har konstant testet systemet vårt og skrevet rapporter på dette.

## 3 Tekniske vurderinger og utfordringer

---

### 3.1 Simulatorsystem

Systemet vi brukte som grunnlag, VBS2, er et velutviklet kamp og trenings-system som er delt opp i ulike versjoner. Lisensen vi fikk var den versjonen som er ment på brukere av systemet, men som gir muligheter til å lage egendefinerte funksjoner og systemer ved hjelp av et innebygget grensesnitt for sending og mottakelse av skriptkommandoer. Ved ikke å ha direkte tilgang til systemets kjerne, men kun via predefinerte funksjonskall, ble oppgaven til tider vanskelig og utfordrende, men ved hardt arbeid og mye prøving og feiling har systemet utviklet seg til en løsning vi er stolte av.

Vi vil likevel på det sterkeste anbefale KDA å bruke VBS2 Fusion (utviklerversjonen) om de velger å videreutvikle systemet vårt, da dette gir langt flere, og ikke minst bedre, muligheter.

#### 3.1.1 Bugs i simulatorsystemet

Ett av de mer uforståelige og plagende feilene vi støtte på under funksjonsimplementeringen var feil i simulatorsystemet. Flere av de dokumenterte kommandoene virket ikke som de skulle og selv etter mye tid ved testing og, fjerning av kode ble det klart at dette ikke skyldes feil i systemet vårt, men i simulatorsystemet. Vi skal si vi var noe heldig her, for to måneder før prosjektslutt ble det gitt ut en oppdatering og bugfix til systemet. Vi fikk raskt kontakt med utviklerne som oppdaterte lisensnøkkelen vår (dette var inkludert i lisensavtalen). Dette løste de tekniske utfordringene som hadde oppstått.

### 3.2 Utviklingen av prototyper

Vi fant raskt ut at det å lage teknologidokumenter og prototyper for hver enkelt av teknologiene vi kom fram til som løsninger var veien å gå. Et gruppemedlem fikk først ansvaret for grundig å undersøke teknologien for så å skrive et dokument på denne. På denne måten slapp de resterende medlemmene å måtte lese seg opp på det samme, men da via allerede undersøkt funksjonalitet og eksempler på bruk var det lett å lage prototyper. Hver prototype hadde sitt særegne funksjonsområde og var kun ment til å teste én teknologi, men da så grundig at det var tydelig hvordan den virket og hvilke utfordringer en ville støte på. I implementasjonen ble all kunnskap tilegnet fra disse applikasjonene videreført og satt sammen.

### 3.3 Oppretting av brukergrensesnittet

Med en gang vi behersket teknologien bak overleggssystemet xwWidgets, ble brukergrensesnittet utviklet. Vi så på dette som det desidert viktigste elementet i prosjektet, da det ikke var rom for noen feil i dette. Brukergrensesnittet som skulle brukes er egentlig en digitalisering av et eksisterende hardware-system og om simulatoren ikke ser 100 % lik ut vil det være liten vits å trene med det. Gruppemedlemmer spesialiserte seg derfor på dette området og fikk senere størsteparten av ansvaret i utviklingen av grensesnittet. Dette har fungert veldig bra da nå alle elementene det består av fungerer på nøyaktig samme måte, noe som gjør det lett å implementere videre funksjonalitet.

### 3.4 Utviklingen av et stabilt rammeverk

Etter hvert som prototypene ble satt sammen fant vi raskt ut at vi måtte ha et system som lett kunne kommunisere sammen uten å måtte bruke for mye ressurser eller for mange kommunikasjonskanaler. Vi utarbeidet da et rammeverk som tok for seg de ulike komponentene i systemet og la et skille mellom dem. På denne måten ville all kommunikasjon mellom komponenter gå gjennom samme kanal og oppføre seg på nøyaktig samme måte. Som ved utviklingen av brukergrensesnittet ble dette også et spesialfelt for enkelte gruppe-medlemmer, noe som igjen førte til at systemet ble utviklet på en helhetlig måte.

### 3.5 Videreutvikling av systemet

Om vi skulle ha videreutviklet systemet vårt ville vi bli nødt til å utvikle nye krav med oppdragsgiver. Vi har også sett behovet for å oppgradere simulatoren til utviklerversjonen *Fusion* om dette hadde vært en realitet. utfordringene vi støttet på under utvidelsen av oppgaven var av den natur som den nåværende versjonen sannsynligvis ikke ville håndtert uten ekstremt mye triksing og billige, men tunge løsninger.

## 4 Gruppemedlemmenes egne kommentarer angående prosjektet

---

### 4.1 Arne Kristian

“Det å avslutte skolegangen med et prosjekt av en slik skala som man gjør på HiBu, er en unik og uvurderlig erfaring for en ingeniørstudent. Det gir en god innsikt i hva det vil si å jobbe som ingeniør, noe som gir oss et fortrinn fremfor andre studenter som ikke har denne muligheten. Min personlige kunnskap om prosjektarbeid, samt programmering har vokst en god del ved å jobbe med dette prosjektet og dette har gitt meg mye god kunnskap jeg vil ta med meg videre.

Min rolle i prosjektgruppen som design og web-ansvarlig ga meg et godt innblikk i hvordan en utviklingsprosess fungerer. Fra det å designe selve systemet, til å produsere noen hjelpesystemer til internt bruk. Den største utfordringen vil jeg nok si var å designe et system som skulle være god og solid nok til at vi kunne benytte dette i vårt arbeid videre. Men godt samarbeid med analyseansvarlig og hardt arbeid, løste vi den utfordringen på en god måte. Jeg føler at vi har et fullverdig og godt design. Det var også veldig gøy og implementere det andre brukergrensesnittet vi la til som en ekstraraksjon på systemet. Websiden har det vært mye frem og tilbake med, men vi har et fungerende timelistesystem som vi har benyttet oss av.”

### 4.2 Lone

“I løpet av prosjektet synes jeg at jeg har lært mye om hvordan man faktisk går frem for å gjennomføre et prosjekt i praksis, og ikke bare i teorien. I tillegg så synes jeg det har vært veldig spennende å være med på et prosjekt fra begynnelse til slutt, hele veien fra avdekking av kravene til overlevering av ferdig system. Man får et annet syn på prosjektarbeid, både med tanke på det tekniske og ikke minst det teammessige.

Selv synes jeg at jeg har lært mye av å ha ansvaret for utvikling og oppbygging av brukergrensesnittet. Selv om vi fikk spesifikk beskjed om hvordan brukergrensesnittet skulle se ut, så var det ikke gitt hvordan det skulle bygges opp rent teknisk. Ei heller hvilken teknologi som skulle brukes, utover at vi skulle kode i C++. En annen ting som var spennende å være med på var å avdekke kravene til systemet, for dersom man skal kunne gjennomføre et prosjekt tilfredsstillende så må man ha et mål å jobbe mot.”

### 4.3 Ole-Martin

“Jeg syntes prosjektet har vært utrolig givende og lærerikt og i løpet av det siste året har jeg fått et helt nytt syn på hva det vil si å være dataingeniør. Å finne våre egne løsninger og ta disse i bruk har vært det morsomste og ikke minst mest lærerike. Det å se hvordan andre personer rundt seg jobber og deres tankegang rundt et slikt prosjekt har også vært interessant, og ikke minst vist meg hvor givende det er å få flere synspunkter.

Rollene mine som planleggings og analyse ansvarlig i dette prosjektet har passet meg utmerket, og jeg føler jeg har gjort en god jobb med det. Planleggingen av prosjektet har vært viktig og det at vi delte opp i mange iterasjoner har vært veldig positivt. Analysen var mye vanskeligere i starten, men etter hvert følte jeg at jeg fikk virkelig tak på det og fikk gjort



en god jobb med det. I tillegg har jeg fått laget mange av de tekniske løsningene vi har tatt i bruk, noe som har gitt meg ett mye bedre løsningsorientert syn enn jeg hadde fra før.”

#### **4.4 Thomas**

“Dette prosjektet har vært en fantastisk måte å lære hvordan en planlegger og utvikler et profesjonelt system. Ved å ta på seg en så stor og krevende oppgave vet en at en må jobbe hardt for å få et godt resultat og en må raskt tilegne seg ikke bare nye kunnskaper, men også de sosiale egenskapene det kreves å jobbe i en gruppe med mange sterke personligheter.

Da jeg tok på meg ansvaret som prosjektleder for denne gruppen hadde jeg ingen anelse om hvor mye jobb det faktisk innebar, men på denne måten jeg kunne utviklet lederegenskapene mine på en effektiv måte og jeg anger ikke et sekund på dette valget.

Ved å være ansvarlig for selve implementasjonsdelen har jeg også sett hvor viktig det er og raskt lage et rammeverk som enkelt gir muligheten for å videreutvikle systemet, og alt dette er kunnskaper jeg vil ta med meg videre.”

#### **4.5 Zoran**

“De siste 8 månedene har vært ganske annerledes. Å ha et prosjekt som går over såpass lang tid har vært en ny opplevelse for meg, og det at gruppen har kommet seg gjennom det hele uten store uenigheter var helt knall! I starten var det ganske rolig, men vi har klart å holde oss i et jevnt tempo slik at vi kom oss i mål i tide.

I starten av prosjektet var testingen veldig dokumentorientert, og lite praktisk. Etterhvert som vi bygde systemet og det utviklet seg stadig raskere ble det en utfordring å avtale tider til testing uten at det gikk ut over tiden til implementasjonen. Med god planlegging så fikk vi ordnet tidsluker for testing og bugfixs. Alt i alt har dette prosjektet vært kjempegøy å være med på! Jeg føler at jeg har lært utrolig mye angående kvalitetssikring og testing, noe jeg gjerne vil ta med meg ut i arbeidslivet.”

## 5 Konklusjon

---

Vi i KDA RIVET er fornøyd med resultatene vi har oppnådd i prosjektet, og alle er enige i at dette har vært en positiv og lærerik prosess som vi uten tvil vil dra nytte av videre. Gjennom prosjektiden har vi støtt på mange utfordringer, og en del utfordringer som til dels har vært svært kompliserte, men ingen ting som ikke har vært mulige å løse. Og ikke minst så har vi ikke vært borti noe som vi har kunnet se på som et problem. Vi har hatt ett motto gjennom hele prosjektet, og det er: "Det finnes ikke problemer, kun utfordringer".

Vi har hatt en bratt læringskurve, og vi har måttet tenke mye utenfor boksen på en del av utfordringene vi har støtt på. Ettersom vi i tillegg har jobbet med teknologi som det finnes lite norsk kompetanse på, så har det vært lite hjelp å få, og det har gått med mye tid på å tråle forumet til Bohemia Interactive etter svar.

Vi føler oss alle nå mer rustede for å gå ut som nyutdannede ingeniører nå enn vi følte oss for 6 måneder siden. Vi har fått bevist for oss selv at vi både kan være med på å produsere profesjonelle systemer, samt å være med på å administrere en slik prosess. Vi kan herved, med god samvittighet, si at dette prosjektet har vært vellykket.

# KDA RIVET



## Analysedokument

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                            | 4  |
| 1.1   | Hensikt med dokumentet.....                  | 4  |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 4  |
| 1.3   | Dokumenthistorie.....                        | 4  |
| 1.4   | Ansvarlig.....                               | 4  |
| 2     | Innledning.....                              | 5  |
| 3     | Use-case analyse.....                        | 6  |
| 3.1   | UC1 – Bemanne systemet.....                  | 7  |
| 3.1.1 | Kommunikasjonsdiagram.....                   | 7  |
| 3.1.2 | Klassediagram.....                           | 7  |
| 3.1.3 | Sekvensdiagram.....                          | 7  |
| 3.2   | UC2 – Sikte.....                             | 8  |
| 3.2.1 | Kommunikasjonsdiagram.....                   | 8  |
| 3.2.2 | Klassediagram.....                           | 8  |
| 3.2.3 | Sekvensdiagram.....                          | 9  |
| 3.3   | UC3 – Armering på/av.....                    | 9  |
| 3.3.1 | Kommunikasjonsdiagram.....                   | 9  |
| 3.3.2 | Klassediagram.....                           | 10 |
| 3.3.3 | Sekvensdiagram.....                          | 10 |
| 3.3.4 | Tilstandsdiagram.....                        | 10 |
| 3.4   | UC4 – Skyte.....                             | 11 |
| 3.4.1 | Kommunikasjonsdiagram.....                   | 11 |
| 3.4.2 | Klassediagram.....                           | 12 |
| 3.4.3 | Sekvensdiagram.....                          | 12 |
| 3.4.4 | Aktivitetsdiagram.....                       | 13 |
| 3.5   | UC5 – Velge skuddhastighet.....              | 14 |
| 3.5.1 | Kommunikasjonsdiagram.....                   | 14 |
| 3.5.2 | Klassediagram.....                           | 14 |
| 3.5.3 | Sekvensdiagram.....                          | 15 |
| 3.5.4 | Tilstandsdiagram.....                        | 15 |



|       |  |    |
|-------|--|----|
| 3.6   | UC6 – Slå på/av nattmodus .....        | 16 |
| 3.6.1 | Kommunikasjonsdiagram.....             | 16 |
| 3.6.2 | Klassediagram.....                     | 16 |
| 3.6.3 | Sekvensdiagram .....                   | 17 |
| 3.6.4 | Tilstandsdiagram .....                 | 17 |
| 3.7   | UC7 – Zoom .....                       | 18 |
| 3.7.1 | Kommunikasjonsdiagram.....             | 18 |
| 3.7.2 | Klassediagram.....                     | 18 |
| 3.7.3 | Sekvensdiagram .....                   | 19 |
| 3.8   | UC9 – Sikte_sensitivitet_joystick..... | 20 |
| 3.8.1 | Kommunikasjonsdiagram.....             | 20 |
| 3.8.2 | Klassediagram.....                     | 20 |
| 3.8.3 | Sekvensdiagram .....                   | 21 |
| 4     | Rapport på analyseklassene .....       | 22 |
| 4.1.1 | Sensitivitet.....                      | 22 |
| 4.1.2 | Input device handler .....             | 22 |
| 4.1.3 | DLL.....                               | 22 |
| 4.1.4 | RIVET Core .....                       | 23 |
| 4.1.5 | Firing Mode .....                      | 23 |
| 4.1.6 | Variable-Storage.....                  | 24 |
| 4.1.7 | NightVision .....                      | 25 |
| 4.1.8 | Armeringsbryter .....                  | 25 |
| 4.1.9 | Zoom.....                              | 25 |
| 5     | Referanser.....                        | 26 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dette dokumentet er å analysere og utdype use-casene på et lavere nivå enn tidligere. Det skal inneholde en oversikt over hvordan systemet skal se ut når utvikleren skal arbeide med systemet.

## 1.2 Definisjoner, akronymer og forkortelser

**UML:** Unified Modeling Language

**UP:** Unified Process

**VBS2:** Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                           |
|---------------|----------|--------------------------------|------------------------------------|
| 1.98          | 21.05.10 | Gjennomgang av egne dokumenter | Ole-Martin Grøtterud               |
| 1.99          | 23.05.10 | Korrekturlest                  | Lone Knutsen og<br>Thomas Transeth |
| 2.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                    |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Innledning

---

I sin helhet vil dokumentet gi en foreslått struktur for systemet, og ta for seg en del av systemets utfordringer på ett høyere nivå, uten å gå veldig dypt inn i det. Det vil gå igjennom de kravene som tidligere er gjennomgått i kravdokumentet og utdype de fra en utviklers synspunkt.

Der hvor kravdokumentet så på systemet fra utsiden, vil analysedokumentet ta for seg systemet fra innsiden. For hver use-case skal realisasjonen være forklart ved et kommunikasjonsdiagram med analyseklasser, samt senere et klassediagram, sekvensdiagram og flere diagrammer om nødvendig. [1]

I diagrammene vil VBS2 objektene være i en mørkere farge for å vise til at de verken blir laget eller endret av oss. Det er viktig å få med seg at analysedokumentet ikke vil ta for seg alle utfordringene lenger nede i systemet, men abstraherer bort en del ting som bør håndteres i design. [1]

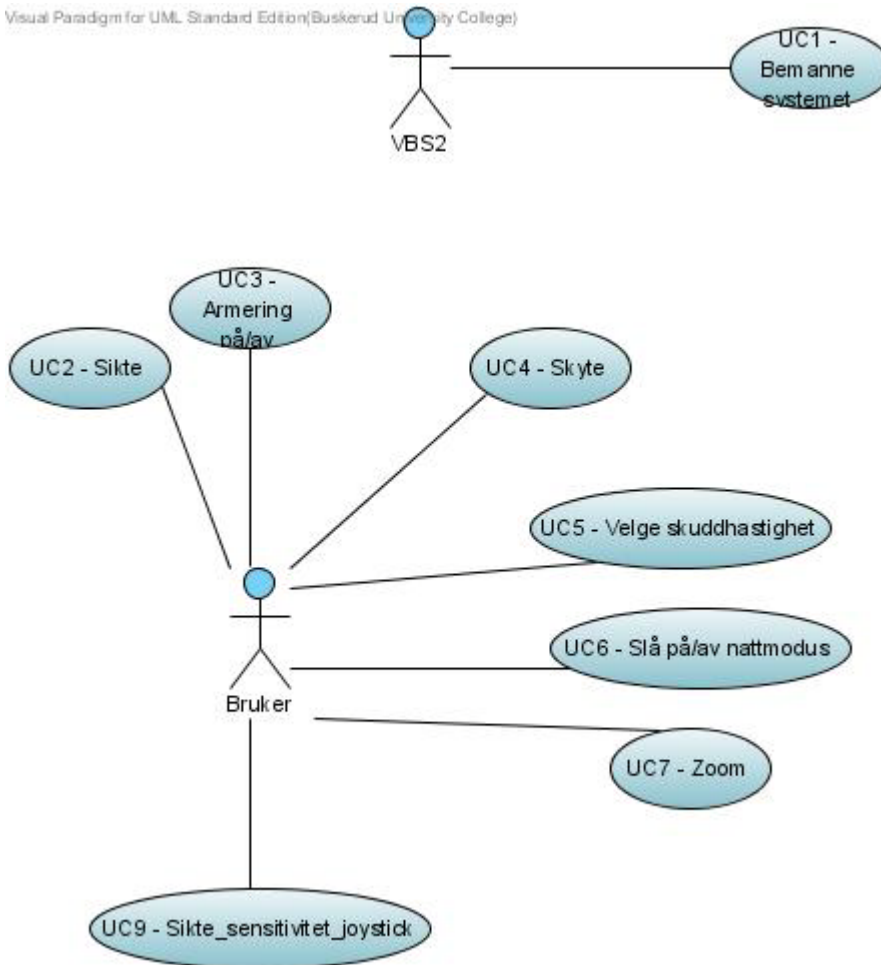
Eksempelvis om lagringsklassen er en C++ klasse eller en database. Alle diagrammene i dette dokumentet er laget på grunnlag av Unified Modelling Language. Det anbefales at leseren har kunnskaper om dette på forhånd, men det vil være forklart i dokumentet "Gjennomgang av UML". Det forventes også at leseren har kjennskap til kravdokumentet.

Som begrunnet i "Iterasjonsutsettelsesrapport"-dokumentet, vil analysedokumentet ikke lenger bli oppdatert etter elaboration-fasen. Alle analyseendringer som eventuelt blir foretatt på ett senere tidspunkt, vil bli håndtert i designdokumentet.

### 3 Use-case analyse

---

Visual Paradigm for UML Standard Edition (Buskerud University College)



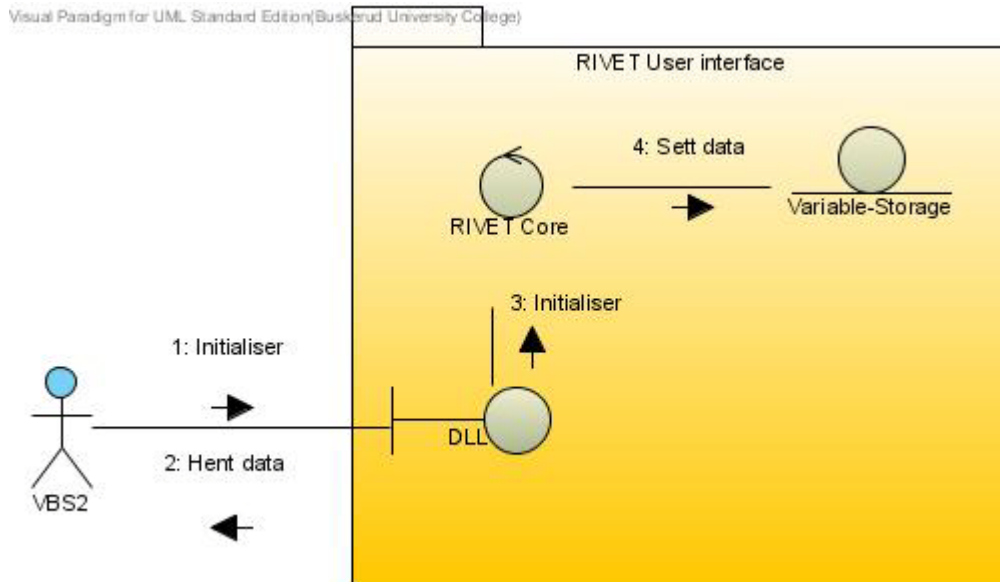
Her er en oversikt over de use-casene vi har kommet opp med til nå, som beskrevet i kravdokumentet. Analysen for hver enkelt use-case vil følge i de neste avsnittene.



### 3.1 UC1 - Bemanne systemet

For å bemanne systemet må VBS2 sende en kommando til brukergrensesnittet vårt som forteller at det skal starte. Brukergrensesnittet skal da hente informasjon fra VBS2 om nåværende status og sende dette videre til kontrollklassen. Denne vil så lagre informasjonen og initialisere brukergrensesnittet.

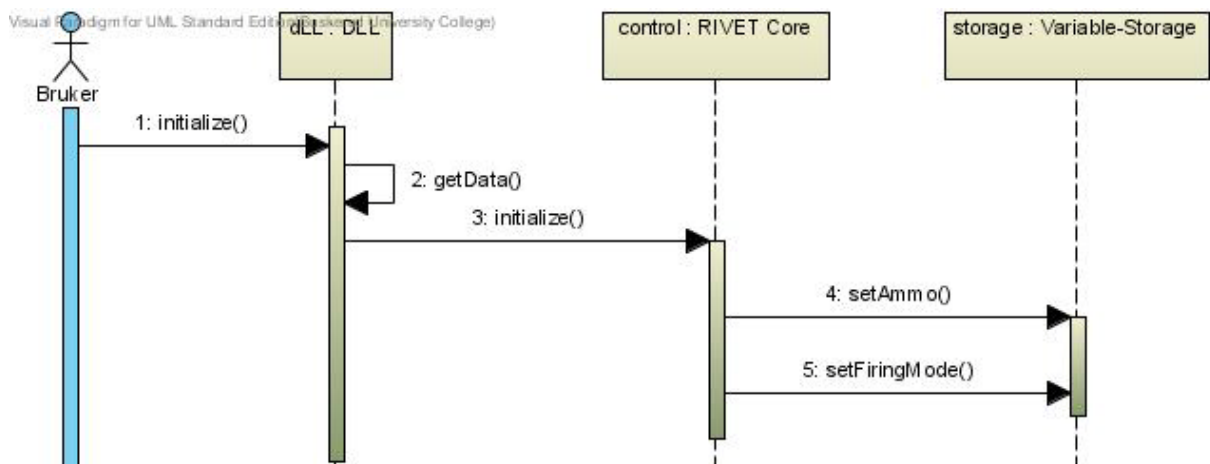
#### 3.1.1 Kommunikasjonsdiagram



#### 3.1.2 Klassediagram



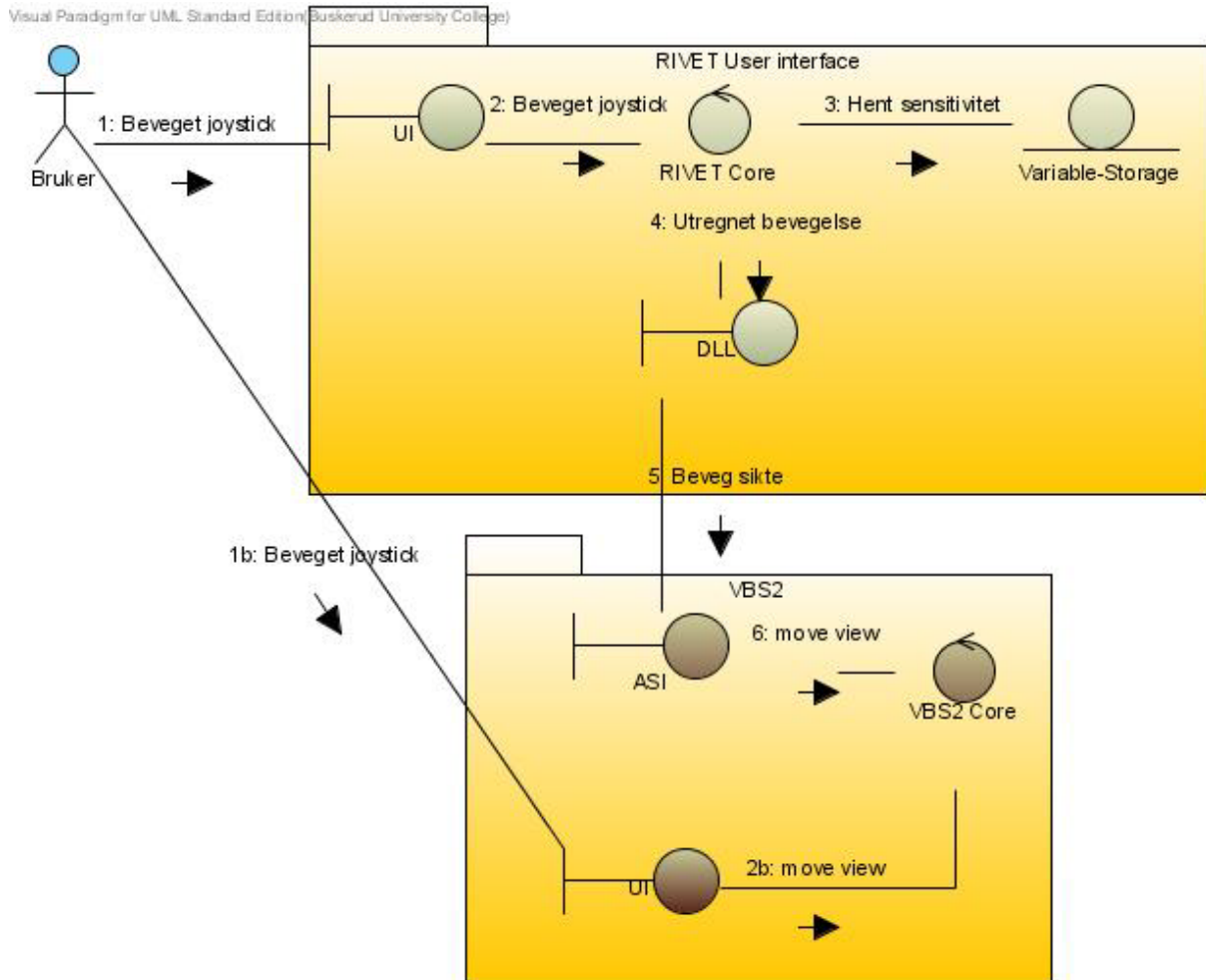
#### 3.1.3 Sekvensdiagram



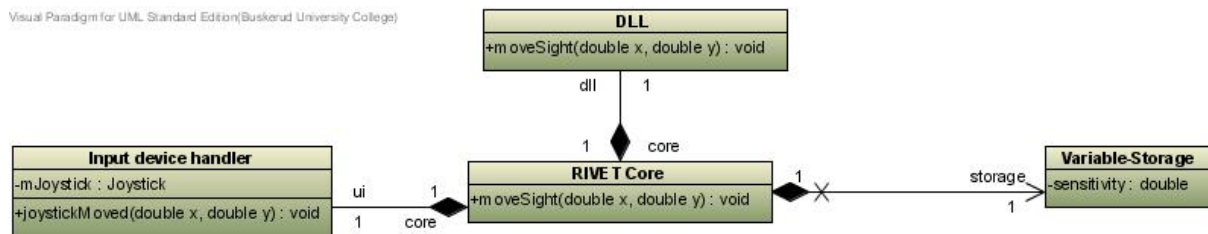
### 3.2 UC2 – Sikte

Her skal systemet vårt motta joystick bevegelser fra brukeren. Denne vil så bli sendt til kontrollklassen som multipliserer den med sensitiviteten og videreformidle dette til VBS2. Alternativt kan joystickbevegelserne sendes rett til VBS2. VBS2 vil så foreta justeringene på sikte ut i fra inputen den får.

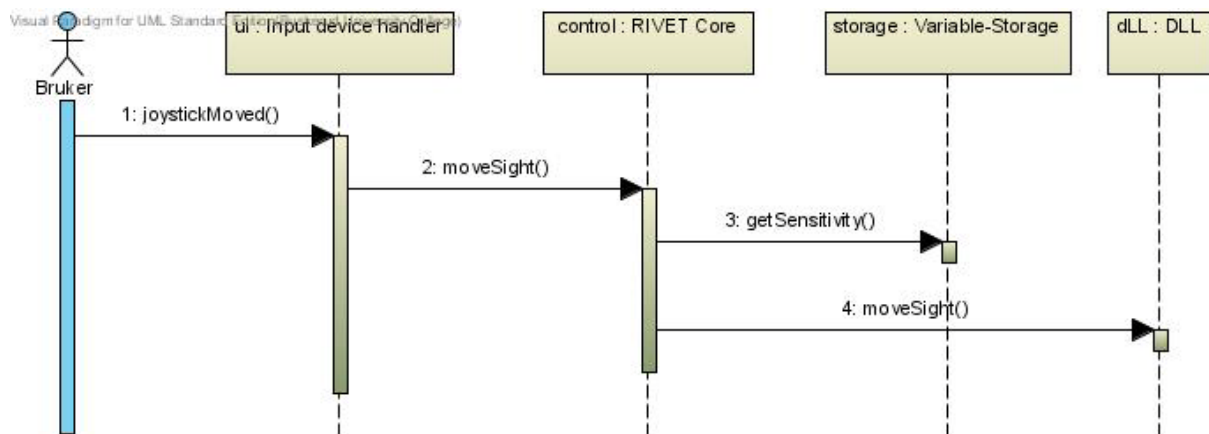
#### 3.2.1 Kommunikasjonsdiagram



#### 3.2.2 Klassediagram



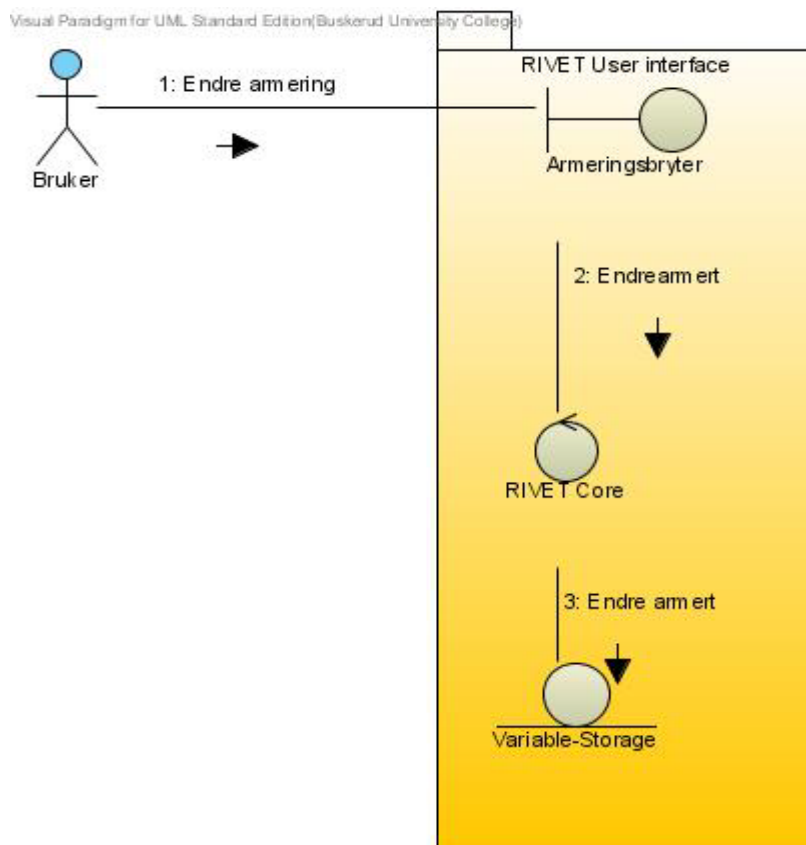
### 3.2.3 Sekvensdiagram



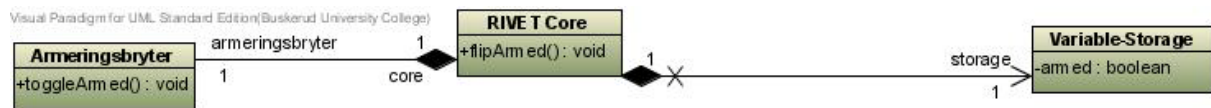
### 3.3 UC3 - Armering på/av

Ved å trykke på armeringsbryteren skal det sendes en beskjed til kontrollklassen, som vil skru armert av hvis den var på, eller på hvis den var av.

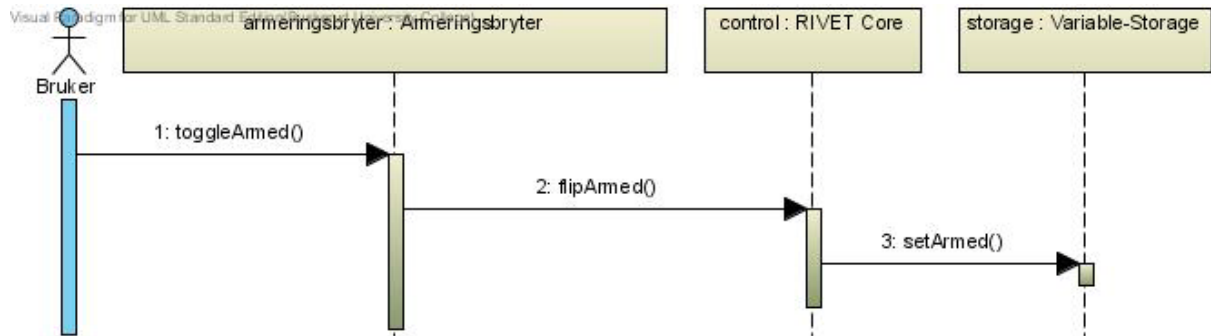
#### 3.3.1 Kommunikasjonsdiagram



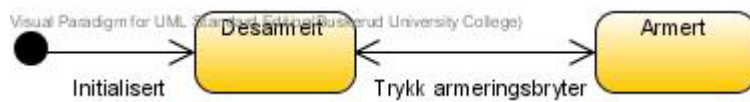
### 3.3.2 Klassediagram



### 3.3.3 Sekvensdiagram



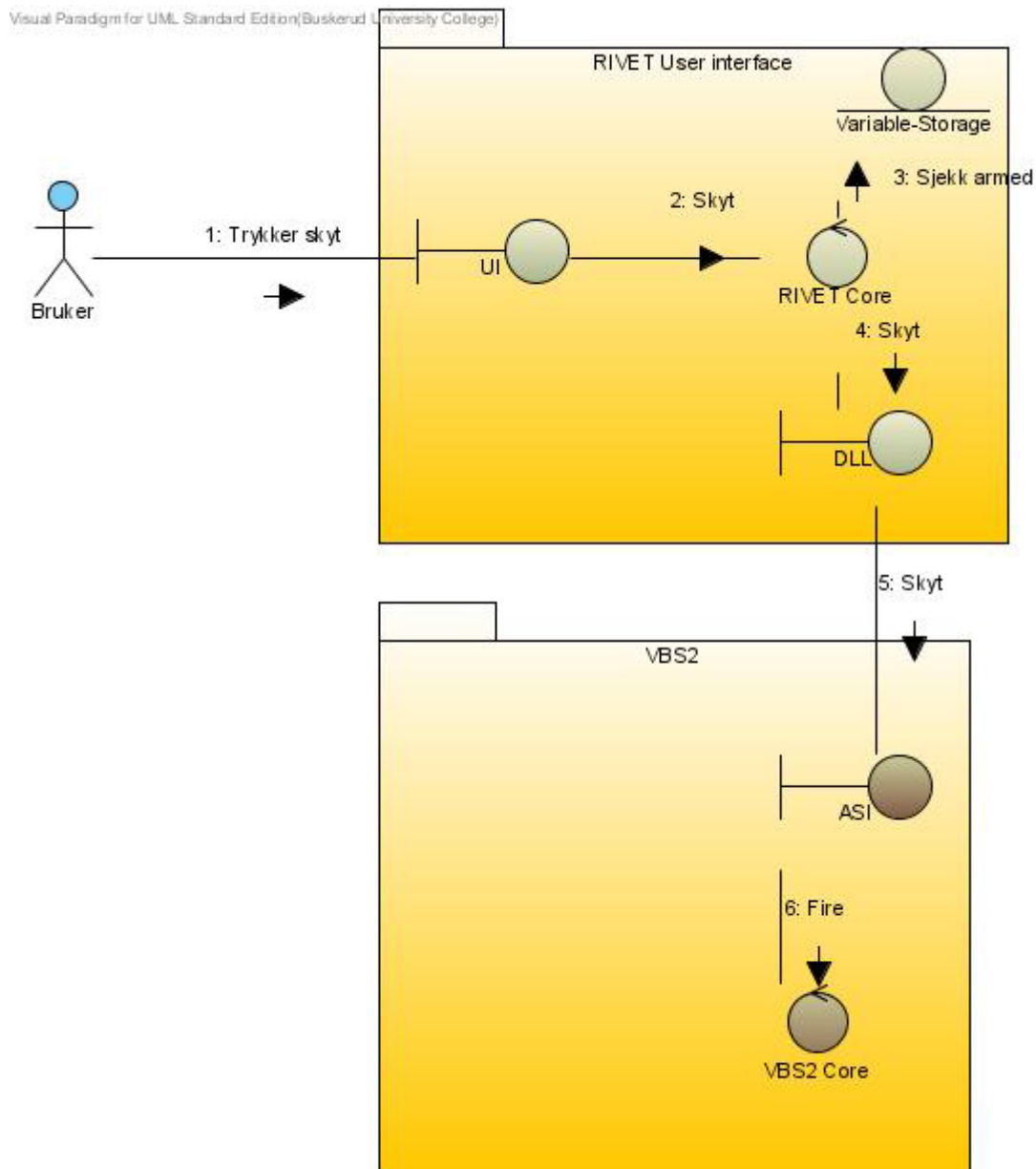
### 3.3.4 Tilstandsdiagram



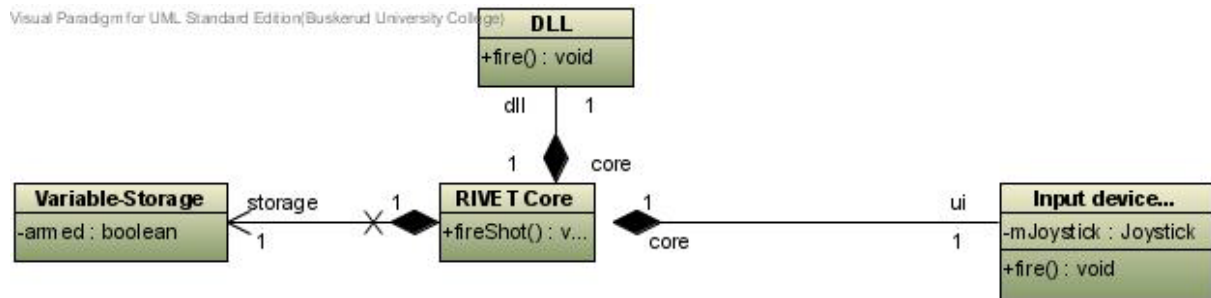
### 3.4 UC4 - Skyte

Ved å trykke på skyteknappen skal systemet motta dette og videreformidle det til kontrollklassen. Denne vil så se om systemet er armert, og hvis det er, sende kommandoen videre til VBS2 som vil avfyre våpenet.

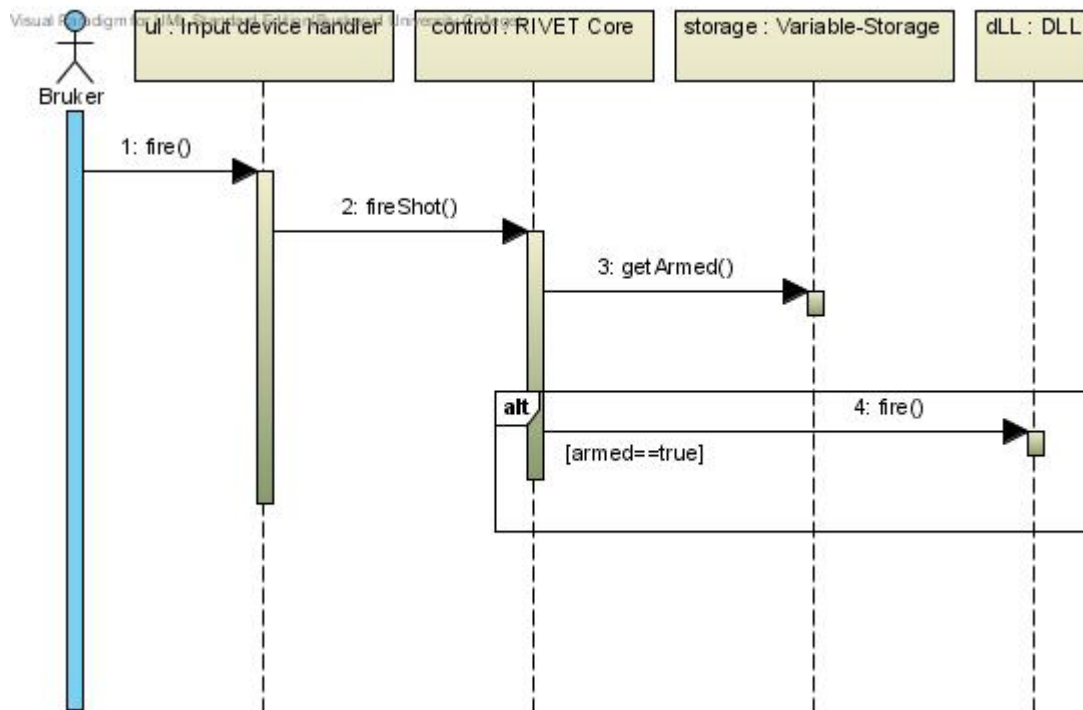
#### 3.4.1 Kommunikasjonsdiagram



### 3.4.2 Klassediagram

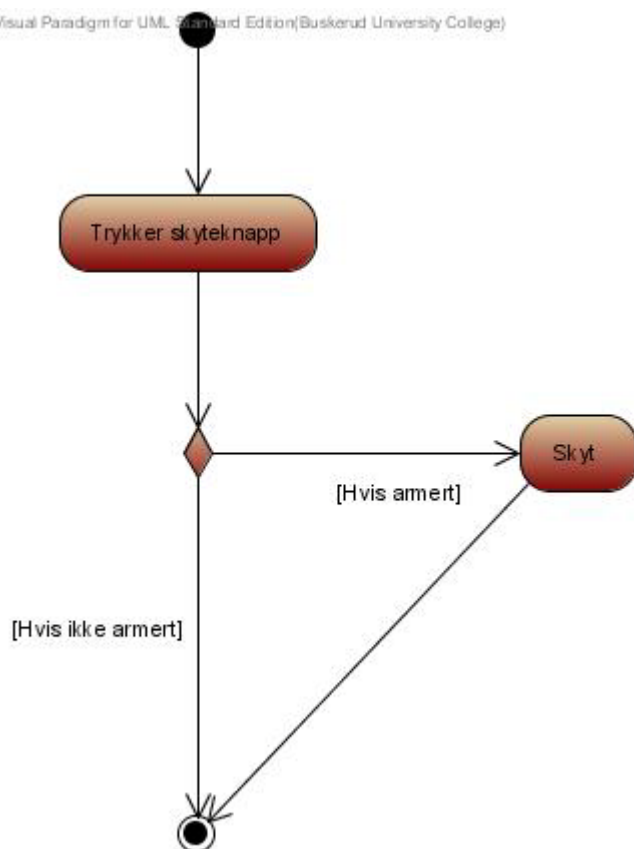


### 3.4.3 Sekvensdiagram



### 3.4.4 Aktivitetsdiagram

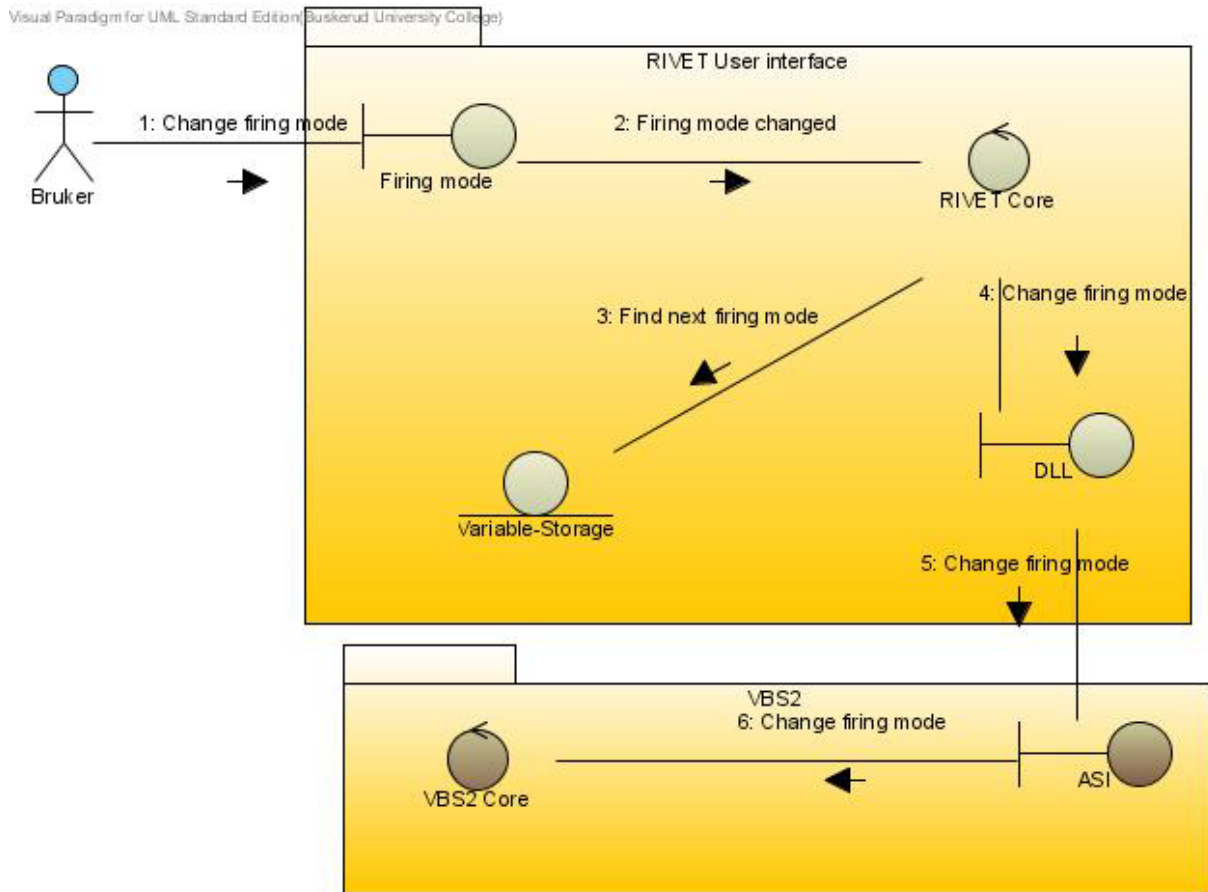
Visual Paradigm for UML Standard Edition (Buskerud University College)



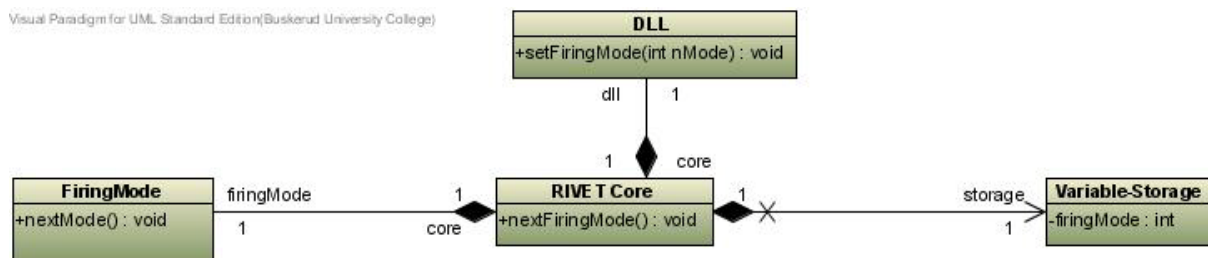
### 3.5 UC5 - Velge skuddhastighet

Ved å trykke på neste skuddhastighet skal systemet videreformidle dette til kontrollklassen. Denne vil så hente ut neste skuddhastighet fra lageret og videreformidle hvilken skuddhastighet som skal velges til VBS2.

#### 3.5.1 Kommunikasjonsdiagram

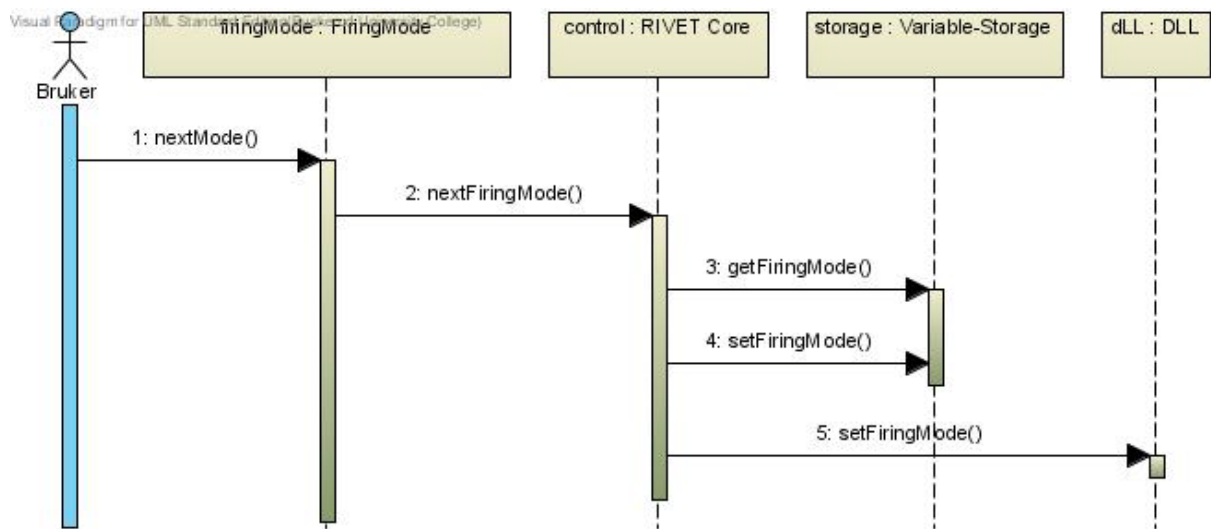


#### 3.5.2 Klassediagram

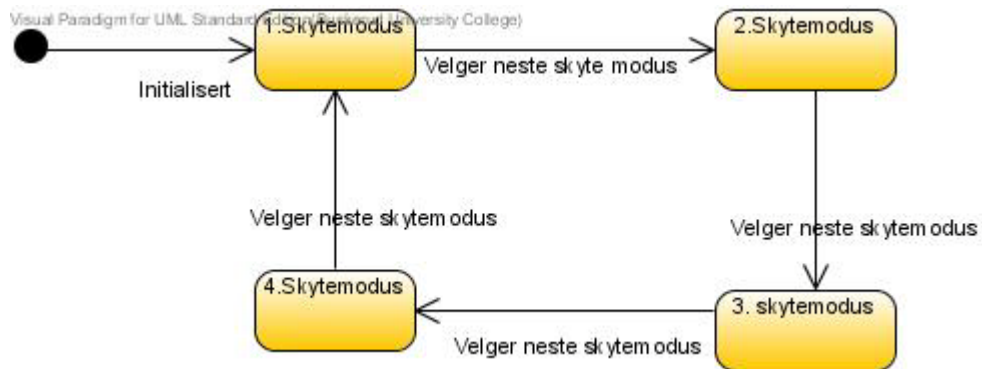




### 3.5.3 Sekvensdiagram



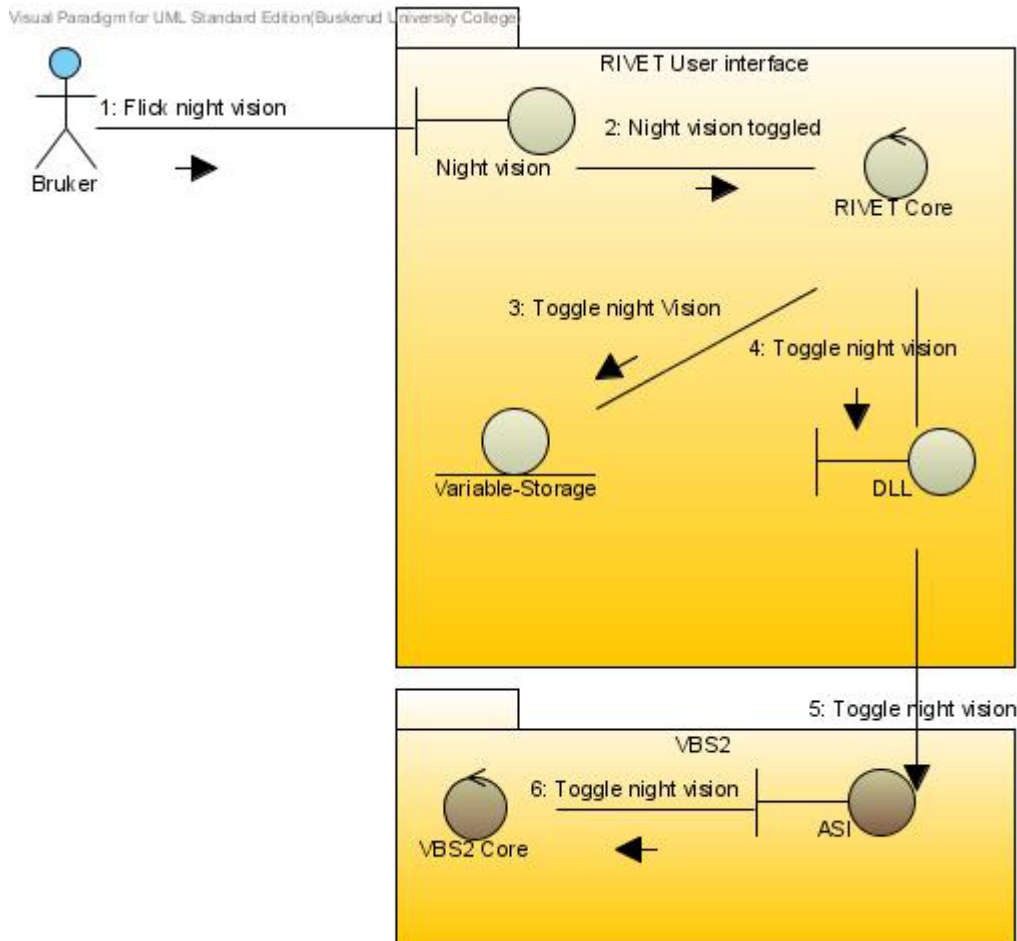
### 3.5.4 Tilstandsdiagram



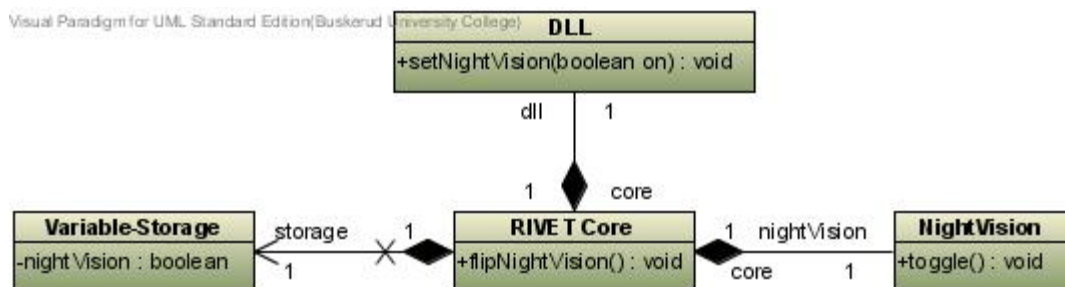
### 3.6 UC6 – Slå på/av nattmodus

Når brukeren slår nattmodus av eller på, skal systemet viderefremde at nattmodus har blitt endret til kontrollklassen. Denne vil så endre den lagrede informasjonen om nattmodus og sende dette videre til VBS2 som slår nattmodusen av eller på.

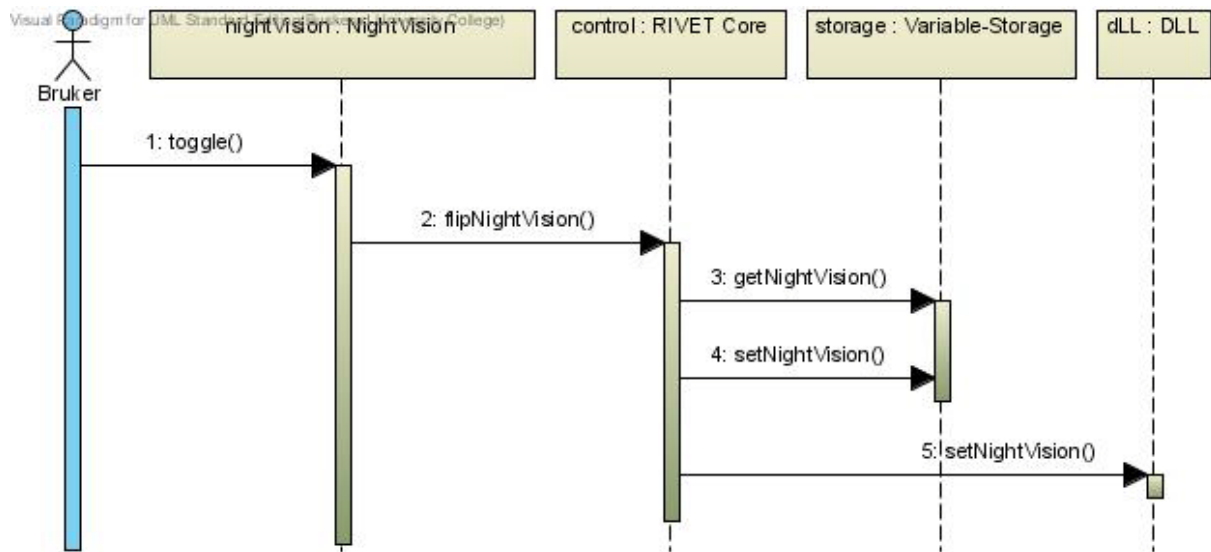
#### 3.6.1 Kommunikasjonsdiagram



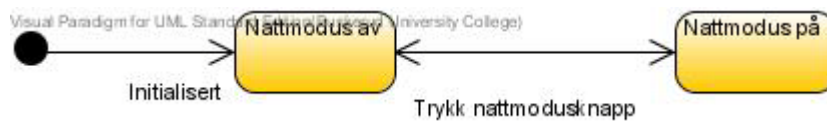
#### 3.6.2 Klassediagram



### 3.6.3 Sekvensdiagram



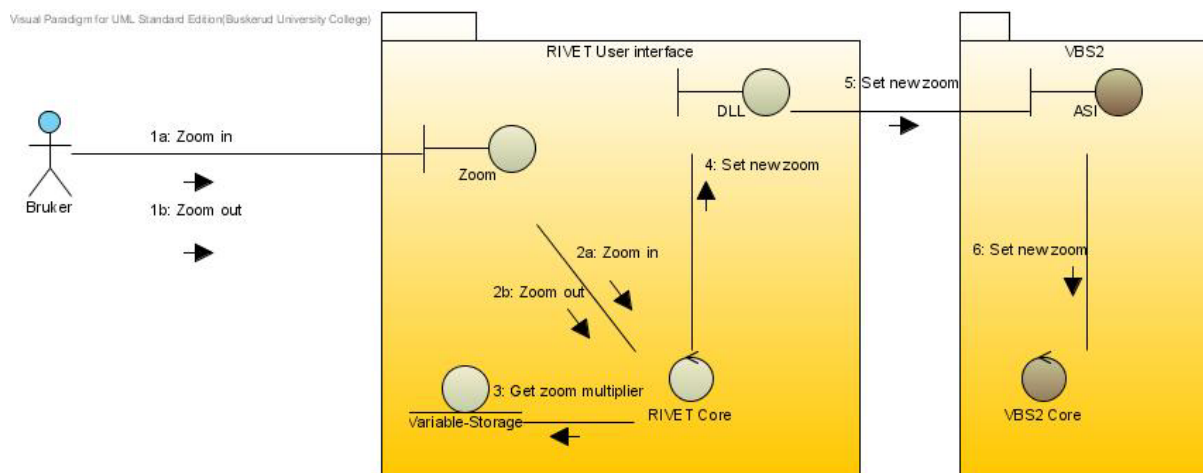
### 3.6.4 Tilstandsdiagram



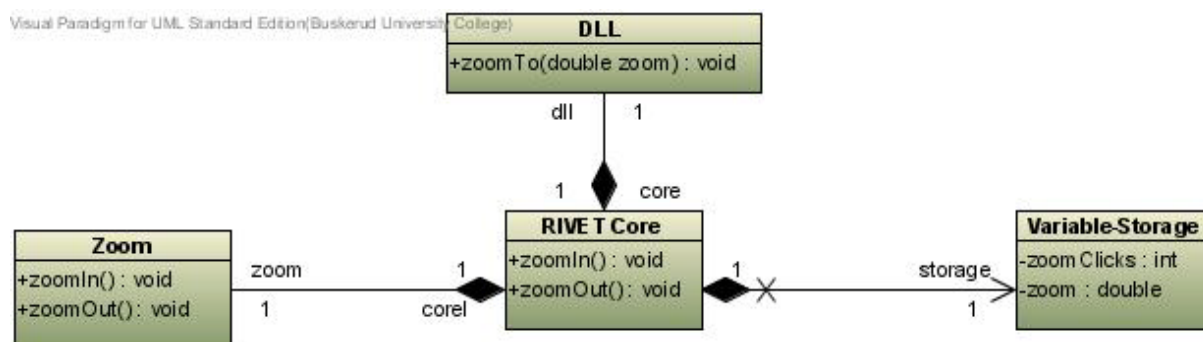
### 3.7 UC7 - Zoom

Når brukeren trykker zoom inn eller zoom ut skal den respektive kommandoen bli sendt til kontrollklassen. Denne vil så gange på en zoom faktor, som er hvor langt den skal zoome per trykk. Deretter vil den legge til eller trekke fra dette tallet fra nåværende lagret zoom og sende det nye zoom nivået til VBS2.

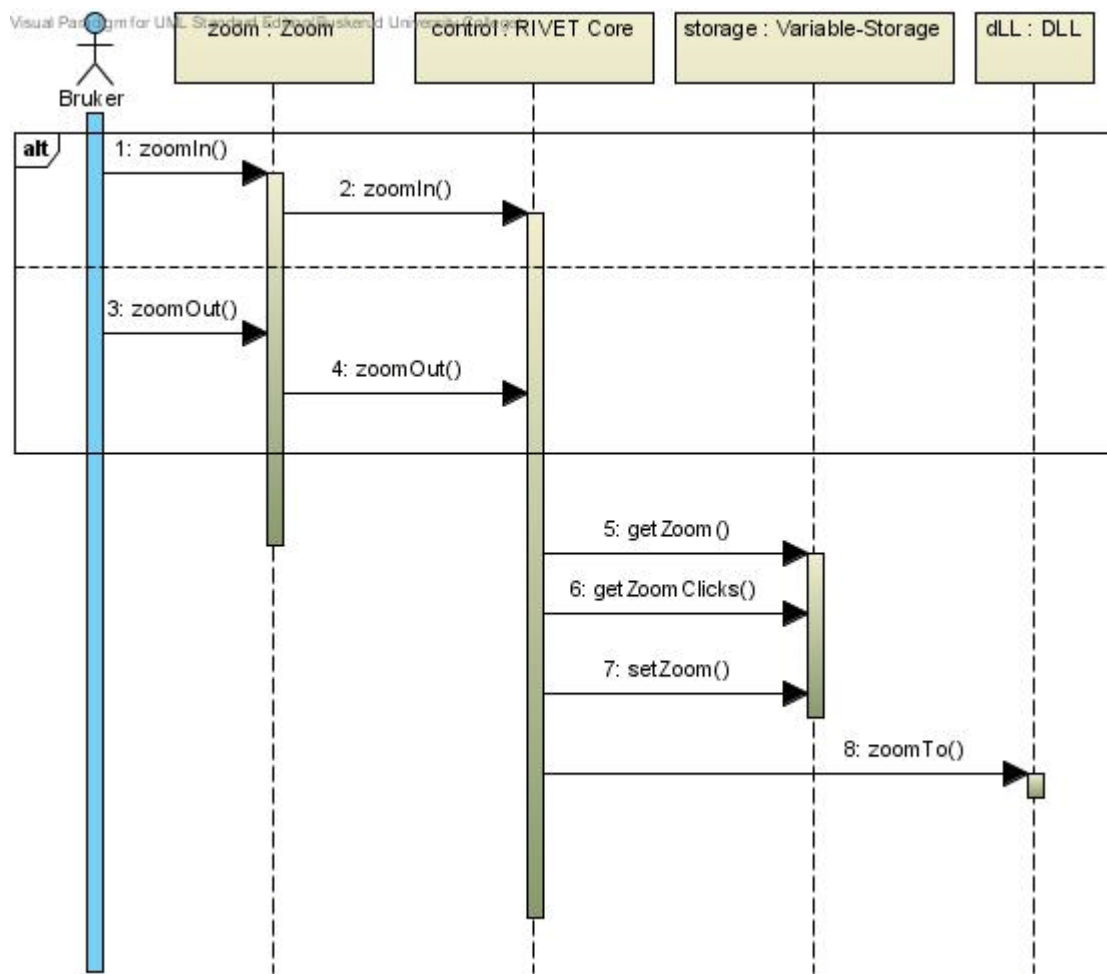
#### 3.7.1 Kommunikasjonsdiagram



#### 3.7.2 Klassediagram



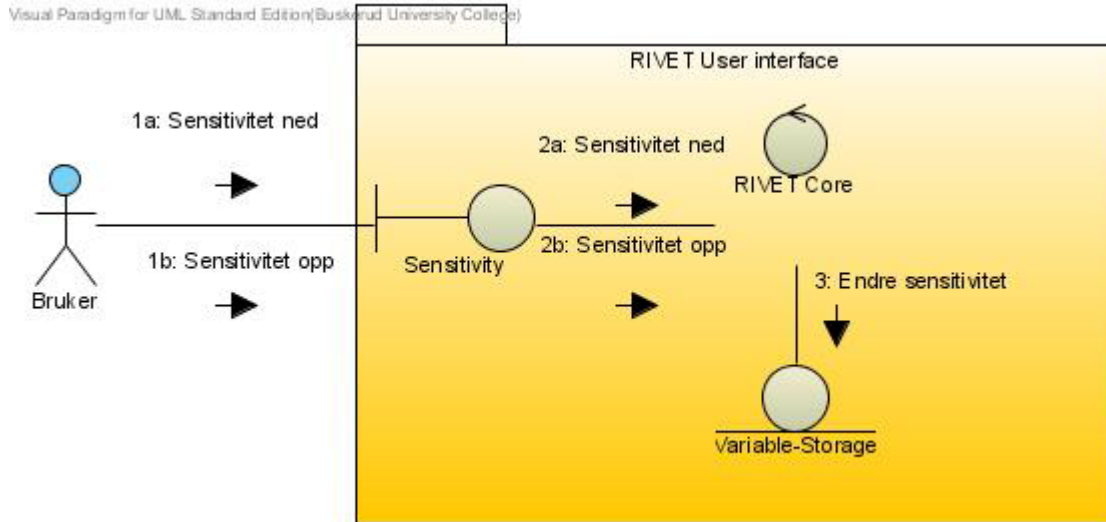
### 3.7.3 Sekvensdiagram



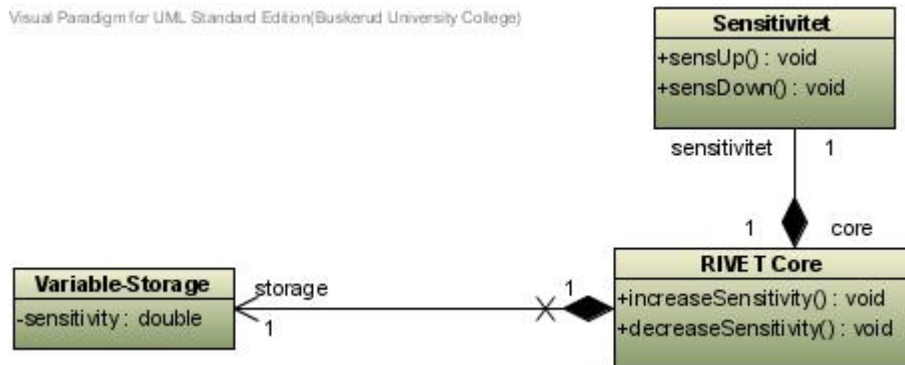
### 3.8 UC9 – Sikte\_sensitivitet\_joystick

Brukeren skal kunne endre sensitiviteten opp eller ned som ønskelig. Systemet skal da sende respektive kommando til kontrollklassen som så endrer sensitiviteten i lageret.

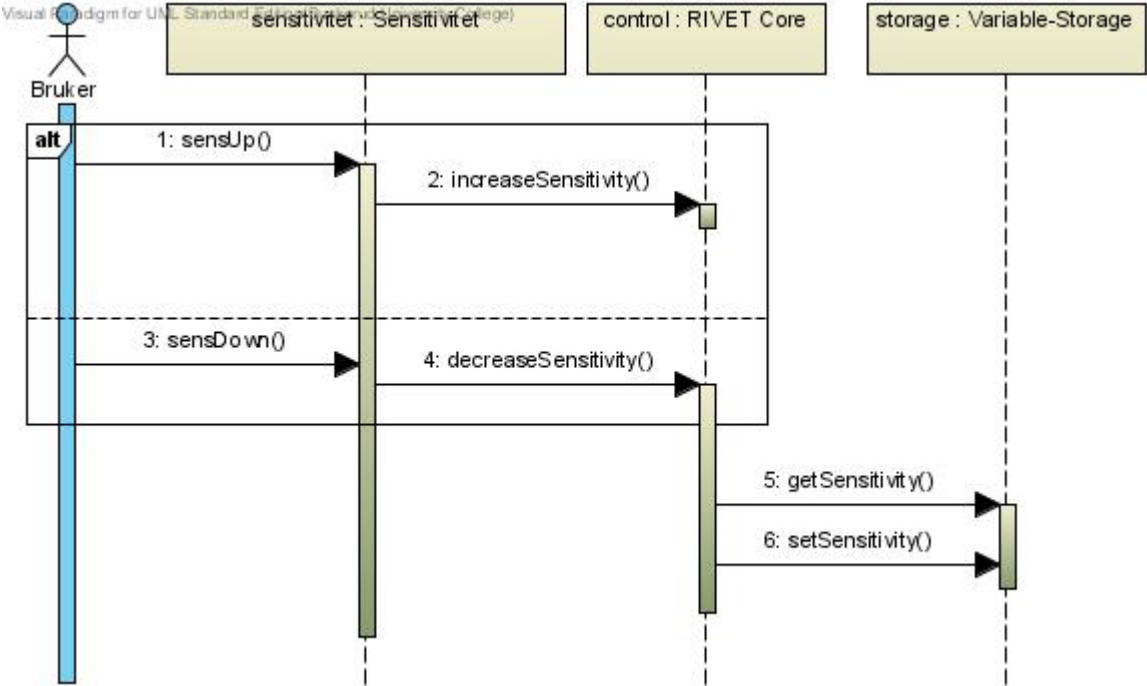
#### 3.8.1 Kommunikasjonsdiagram



#### 3.8.2 Klassediagram



### 3.8.3 Sekvensdiagram



## 4 Rapport på analyseklassene

Under følger en rapport på alle analyseklassene med en beskrivelse av alle funksjonene.

### 4.1.1 Sensitivitet

#### 4.1.1.1 Operations

```
public sensUp() : void
```

Blir kalt når bruker trykker på sensitivitet opp, sier ifra til Control klassen.

```
public sensDown() : void
```

Blir kalt når bruker trykker på sensitivitet ned, sier ifra til Control klassen.

### 4.1.2 Input device handler

#### 4.1.2.1 Attributes

```
private mJoystick : Joystick
```

|        |       |        |       |
|--------|-------|--------|-------|
| Getter | false | Setter | false |
|--------|-------|--------|-------|

|         |       |
|---------|-------|
| Derived | false |
|---------|-------|

|             |      |
|-------------|------|
| Aggregation | None |
|-------------|------|

#### 4.1.2.2 Operations

```
public joystickMoved(double x, double y) : void
```

Blir kalt når bruker beveger joysticken, sier ifra til Control klassen.

```
public fire() : void
```

Blir kalt når bruker trykker på skyteknappen, sier ifra til Control klassen.

### 4.1.3 DLL

#### 4.1.3.1 Operations

```
public initialize() : void
```

Blir kalt av VBS2 for å initialisere brukergrensesnittet.

```
public getData() : string
```

Henter informasjon fra VBS2 og får det returnert som string.

```
public setFiringMode(int nMode) : void
```

Forteller VBS2 hvilken skytemodus den skal skifte til.

```
public setNightVision(booleen on) : void
```

Forteller VBS2 at den skal skru av eller på nattmodus.

```
public zoomTo(double zoom) : void
```



Forteller VBS2 hvilket zoom-nivå den skal gå til.

```
public moveSight(double x, double y) : void
```

Forteller VBS2 hvor den skal flytte siktet.

```
public fire() : void
```

Forteller VBS2 at den skal skyte.

#### 4.1.4 RIVET Core

##### 4.1.4.1 Operations

```
public initialize(int ammo, int fMode) : void
```

Legger inn informasjonen i lageret og synliggjør brukergrensesnittet på skjermen.

```
public flipArmed() : void
```

Endrer armed verdien i lageret til det motsatte.

```
public nextFiringMode() : void
```

Henter neste skytemodus fra lageret og sender det til DLL klassen.

```
public flipNightVision() : void
```

Endrer nightvision verdien i lageret til det motsatte og sender det til DLL klassen.

```
public zoomIn() : void
```

Henter zoomClicks fra lageret og legger det til zoom verdien. Sender så den nye zoom verdien til DLL klassen.

```
public zoomOut() : void
```

Henter zoomClicks fra lageret og trekker fra zoom verdien. Sender så den nye zoom verdien til DLL klassen.

```
public fireShot() : void
```

Sjekker armed verdien i storage. Hvis den er true bruker den fired funksjonen i DLL klassen.

```
public moveSight(double x, double y) : void
```

Tar imot bevegelsesverdiene fra joysticken, ganger det med sensitiviteten fra storage og sender de nye verdiene til DLL klassen.

```
public increaseSensitivity() : void
```

Øker verdien av sensitivitet i storage.

```
public decreaseSensitivity() : void
```

Senker verdien av sensitivitet i storage.

#### 4.1.5 Firing Mode

##### 4.1.5.1 Operations

```
public nextMode() : void
```

Blir kalt når bruker trykker neste skytemodus, sier ifra til Control klassen

## 4.1.6 Variable-Storage

### 4.1.6.1 Attributes

| <b>private ammo : int</b> |       |        |      |
|---------------------------|-------|--------|------|
| Getter                    | true  | Setter | true |
| Derived                   | false |        |      |
| Aggregation               | None  |        |      |

| <b>private firingMode : int</b> |       |        |      |
|---------------------------------|-------|--------|------|
| Getter                          | true  | Setter | true |
| Derived                         | false |        |      |
| Aggregation                     | None  |        |      |

| <b>private armed : boolean</b> |       |        |      |
|--------------------------------|-------|--------|------|
| Getter                         | true  | Setter | true |
| Derived                        | false |        |      |
| Aggregation                    | None  |        |      |

| <b>private nightVision : boolean</b> |       |        |      |
|--------------------------------------|-------|--------|------|
| Getter                               | true  | Setter | true |
| Derived                              | false |        |      |
| Aggregation                          | None  |        |      |

| <b>private zoomClicks : int</b> |       |        |      |
|---------------------------------|-------|--------|------|
| Getter                          | true  | Setter | true |
| Derived                         | false |        |      |
| Aggregation                     | None  |        |      |

| <b>private zoom : double</b> |       |        |      |
|------------------------------|-------|--------|------|
| Getter                       | true  | Setter | true |
| Derived                      | false |        |      |
| Aggregation                  | None  |        |      |

| <b>private sensitivity : double</b> |       |        |      |
|-------------------------------------|-------|--------|------|
| Getter                              | true  | Setter | true |
| Derived                             | false |        |      |
| Aggregation                         | None  |        |      |

## 4.1.7 NightVision

### 4.1.7.1 Operations

```
public toggle() : void
```

Blir kalt når bruker trykker på nightvision knappen, sier ifra til Control klassen.

## 4.1.8 Armeringsbryter

### 4.1.8.1 Operations

```
public toggleArmed() : void
```

Blir kalt når bruker trykker på armed bryteren, sier ifra til Control klassen.

## 4.1.9 Zoom

### 4.1.9.1 Operations

```
public zoomIn() : void
```

Blir kalt når bruker trykker på zoom inn, sier ifra til Control klassen.

```
public zoomOut() : void
```

Blir kalt når bruker trykker på zoom out, sier ifra til Control klassen.

## 5 Referanser

[1] I. Jacobson, G. Booch og J.Rumbaugh, The Unified Software Development Process, Addison-Wesley, 2000, side 173-214.

# KDA RIVET



## Brukermanual

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøtterud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                                | 3  |
| 1.1   | Hensikt med dokumentet.....                      | 3  |
| 1.2   | Definisjoner, akronymer og forkortelser.....     | 3  |
| 1.3   | Dokumenthistorie.....                            | 3  |
| 1.4   | Ansvarlig.....                                   | 3  |
| 2     | Installasjon av RIVET-plugin.....                | 4  |
| 2.1   | Hvordan installere RIVET-plugin.....             | 4  |
| 3     | Hvordan lage eget scenario for RIVET-plugin..... | 5  |
| 3.1   | VBS2 mission editor.....                         | 5  |
| 3.2   | Navnsetting på biler og nøkkelpersoner.....      | 5  |
| 3.3   | Intel og trigger verdier.....                    | 6  |
| 4     | Kontroller.....                                  | 7  |
| 4.1   | VBS2 kontroller.....                             | 7  |
| 4.2   | RIVET-Plugin kontroller.....                     | 8  |
| 4.2.1 | Sikte:.....                                      | 8  |
| 4.2.2 | Skyte:.....                                      | 8  |
| 4.2.3 | Zoom:.....                                       | 8  |
| 4.2.4 | GUI funksjoner.....                              | 9  |
| 5     | RIVET-plugin funksjonalitet.....                 | 11 |
| 5.1   | Bemanning av system.....                         | 11 |
| 5.2   | Armering.....                                    | 13 |
| 5.3   | Skuddhastighet.....                              | 14 |
| 5.4   | Zoom.....  | 14 |
| 5.5   | Nattmodus og Thermal vision.....                 | 14 |
| 5.6   | Sensitivitet.....                                | 14 |
| 5.7   | Granatkaster.....                                | 14 |
| 5.8   | Meny.....  | 15 |
| 5.9   | På/Av knapp.....                                 | 15 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en innføring i hvordan produktet vårt kan tas i bruk

## 1.2 Definisjoner, akronymer og forkortelser

**VBS2:** Virtual Battle Space 2

**GUI:** Graphical User Interface

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                                   |
|---------------|----------|---------------------------|--|
| 0.1           | 23.04.10 | Førsteutkast              | Ole-Martin Grøtterud                       |
| 0.2           | 24.04.10 | Videreutvikling           | Zoran Vukobratovic                         |
| 0.99          | 27.05.10 | Korrekturlest             | Ole-Martin Grøtterud<br>og Thomas Transeth |
| 1.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                            |

## 1.4 Ansvarlig

Zoran Vukobratovic

## 2 Installasjon av RIVET-plugin

---

### 2.1 Hvordan installere RIVET-plugin

Plasser RIVET-plugin CD-en inn i CD-ROM stasjonen på datamaskinen. Når installeringen starter opp vil du bli spurt om hvor den skal installeres. Her velger du den mappen som VBS2 er installert i, f.eks C:\Bohemia Interactive\VBS2 VTK 1.3 (hvis installeringen ikke starter automatisk, dobbelklikk CD-ROM stasjonen, deretter dobbelklikk "setup.exe").





## 3 Hvordan lage eget scenario for RIVET-plugin

---

Det første som kommer opp når VBS2 startes er hovedmenyen. For å begynne å bruke treneren trengs det først ha et scenario. Velg **Mission Editor** for å begynne å lage et scenario.

Her må det velges et kart, eller område som scenarioet skal spilles ut på. Gi scenarioet et navn, f.eks MittScenario1.

### 3.1 VBS2 mission editor

Det neste steget er å sette sammen et scenario. Nå vises det et topografisk kart over området som er valgt og her skal det plasseres ut objekter som f.eks mennesker, dyr, kjøretøy og hus.

### 3.2 Navnsetting på biler og nøkkelpersoner

Alle objekter burde ha navn på seg, men for vårt system er det følgende navn som **MÅ** være med.

Hovedpersonen, figuren som brukeren vil kontrollere må hete "*Per*".

Kjøretøy med RWS montert på seg må ha navn som kontrollerer hvilken type RWS den har montert. Hvis kjøretøyet har en RWS- CROWS skal kjøretøyet hete "*bilenbrom*".

Hvis kjøretøyet har en RWS-BLOCK2 montert, må det hete "*pellemilitarbil*".

Disse restriksjonene kontrollerer triggerer for hvilke GUI som skal komme opp, og når de skal komme opp.

### 3.3 Intel og trigger verdier

Alle objekter kan ha en trigger på seg, triggere kan utløses ved at visse objekter kommer inn i området som triggeren dekker, de kan også trigges ved hjelp av andre triggere og tidsur.

I tillegg til den navnsettingen som må være til stede er det også intel og 3 triggere som **MÅ** være med. bilenbrom bruker en CROWS, mens pellemilitarbil har BLOCK2. Dersom kun en av bilene brukes, fjernes aktiveringstriggeren for den bilen og dens del i condition delen av deaktiveringstriggeren.

INTEL:

Initialization statements: inTank = false;ret = pluginFunction ["VBSPlugin", "0"] ;

#### **bilenbrom aktiveringstrigger settes opp med:**

Condition: (gunner bilenbrom) == Per && !inTank

On activation: inTank = true; ret = pluginFunction ["VBSPlugin", "1,1"]; Per switchCamera "Gunner";

#### **pellemilitarbil aktiveringstrigger settes opp med:**

Condition: (gunner pellemilitarbil) == Per && !inTank

On activate: inTank = true; ret = pluginFunction ["VBSPlugin", "1,2"]; Per switchCamera "Gunner";

#### **Trigger for deaktivering av GUI for begge biler settes opp med:**

Condition: (gunner bilenbrom) != Per && (gunner pellemilitarbil) != Per && inTank

On activation: inTank = false; ret = pluginFunction ["VBSPlugin", "2"]; Menu = "Menu0"; nul=[] execVM "ole.sqf";

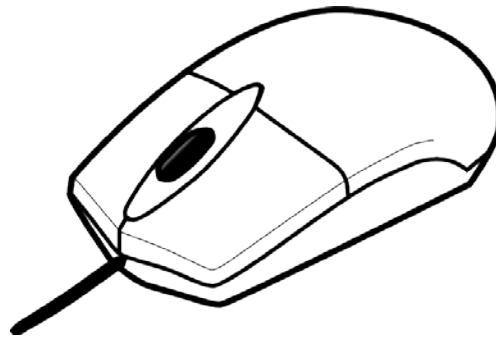
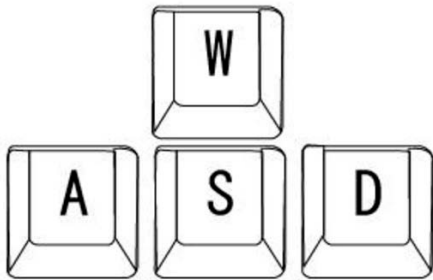
## 4 Kontroller

---

### 4.1 VBS2 kontroller

Bevegelse og sikte:

For å bevege på figuren i VBS2 skal det brukes tastatur og mus. WASD tastene styrer hvilken retning figuren løper i, og musen styrer siktet. For å snu figuren rundt må siktet bevegges i den retningen du vil snu mot.



Skyte:

Venstre museknapp styrer avfiring av våpen.



Bruke objekter i VBS2:

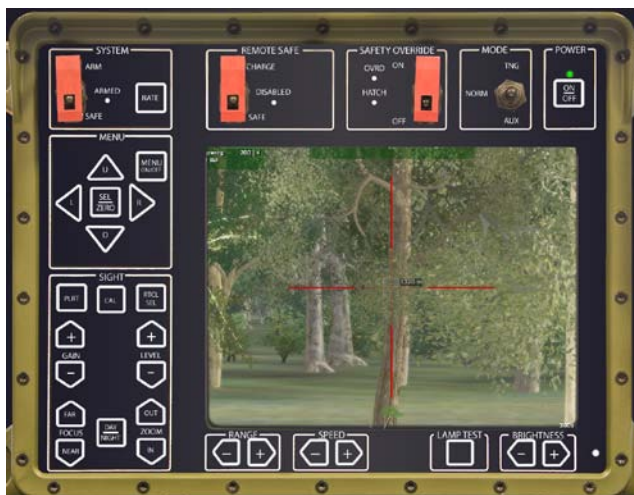
Objekter i VBS2, blandt annet kjøretøy, dører, innretninger osv. Kan brukes ved å trykke på U på tastaturet.



## 4.2 RIVET-Plugin kontroller

### 4.2.1 Sikte:

For å bevege på siktet til RWS-en er det bare å bevege joysticken i den ønskelige retningen. Ved å bevege joysticken venstre og høyre vil siktet følge med. Når joysticken peker forover vil siktet gå ned, og når joysticken peker bakover vil siktet gå opp.



### 4.2.2 Skyte:

For å avfyre RWS våpenet må det være armert (se kap. 5.3, Armering). Etter at våpenet er armert vil det være klart til avfiring ved at trigger knappen på joysticken trykkes inn.



### 4.2.3 Zoom:

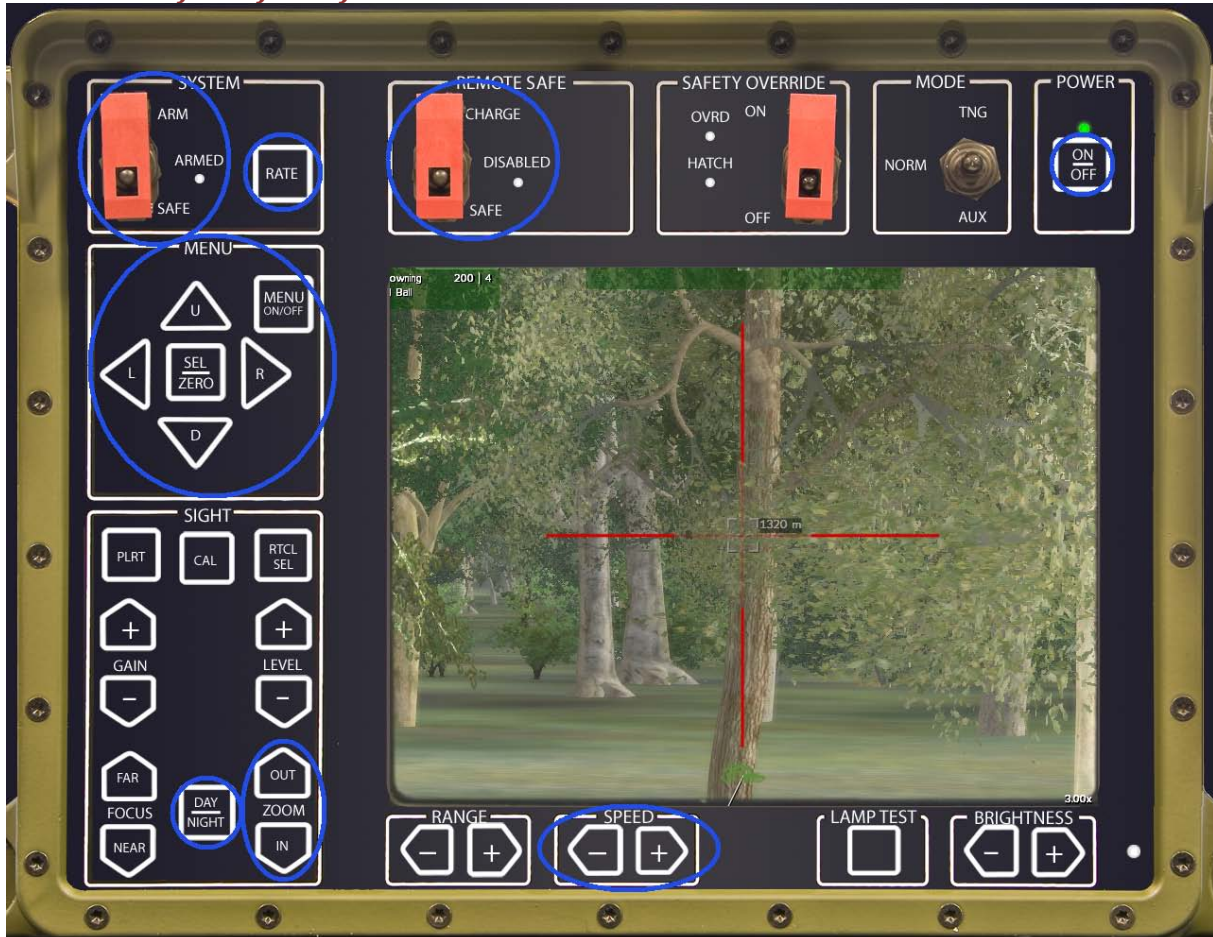
For å bruke zoom funksjonen kan Hat Switch-en brukes i tillegg til zoom knappene (se kap. 5.4) på brukergrensesnittet. Når denne trykkes forover vil RWS-en zoome inn ett hakk, og når den blir trykket bakover vil det bli zoomet ut ett hakk.



#### 4.2.4 GUI funksjoner

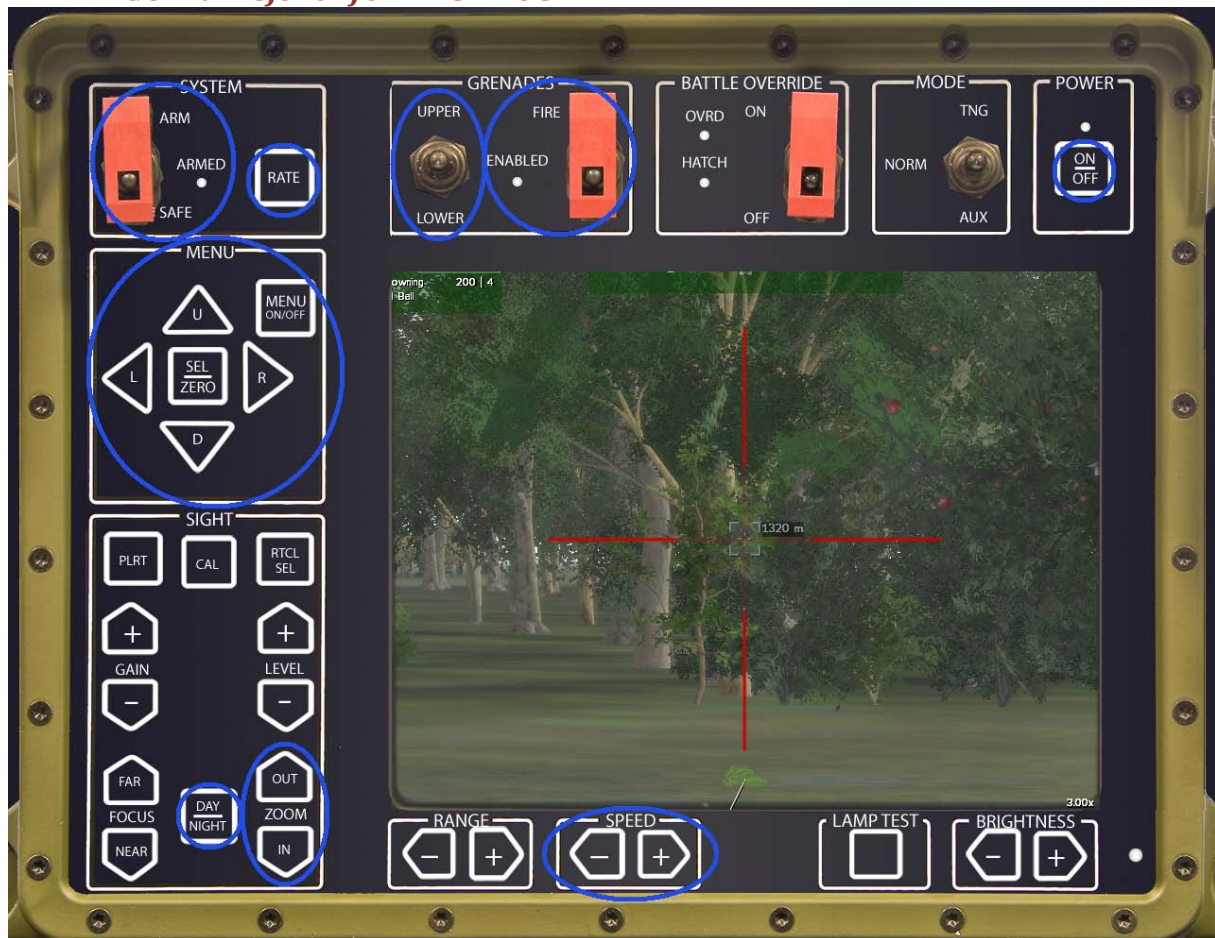
Alle GUI funksjoner velges med musen. Noen knapper skal bare trykkes på, mens for andre skal den venstre musetasten trykkes inn og musen skal dras opp eller ned.

##### 4.2.4.1 GUI funksjoner for RWS-CROWS





#### 4.2.4.2 GUI Funksjoner for RWS-BLOCK2



## 5 RIVET-plugin funksjonalitet

---

### 5.1 Bemanning av system

Når et scenario starter opp vil brukeren starte med å kontrollere den virtuelle representasjonen av seg selv. I vårt system vil brukeren starte rett ved siden av et kjøretøy med RWS montert på.

Brukeren skal navigere seg frem til kjøretøyet og trykke på bruk-knappen ( U ) på tastaturet, da kommer følgende bilde opp som viser forskjellige plasseringer som kan bemannes i kjøretøyet.





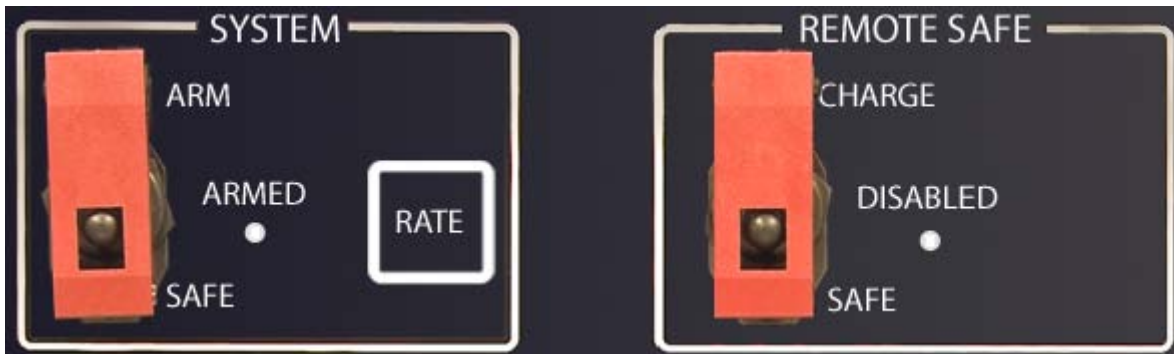
Brukeren skal bruke musen til å trykke på "Get in as gunner" posisjonen som er merket på bildet under.



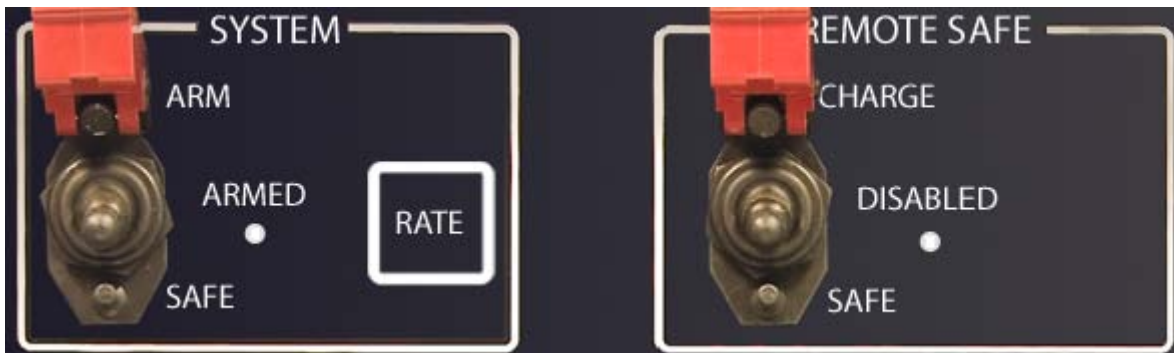


## 5.2 Armering

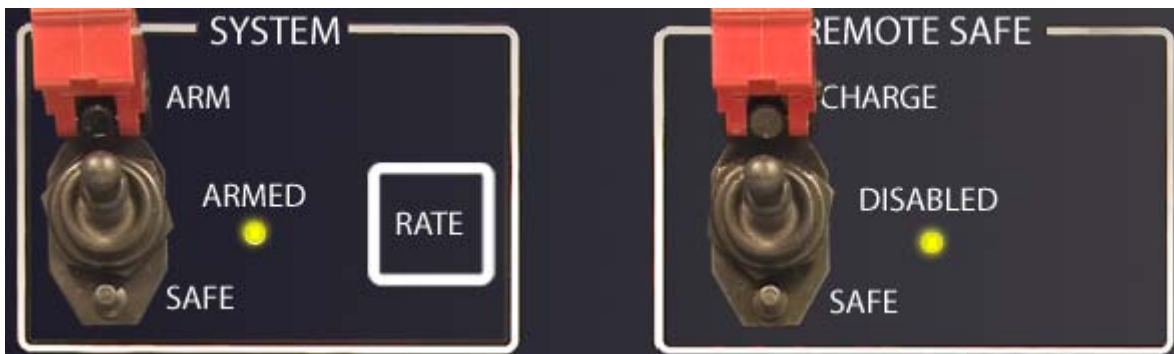
For å armere våpenet må arming switch og remote safe switch brukes. Den røde sikkerhetsvippen må trykkes og dras oppover



Når sikkerhetsvippen er åpnet kan selve bryteren trykkes på for å armere våpenet.

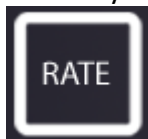


Når armeringsbryterne står som vis under er systemet armert og klart til å avfyres.



### 5.3 Skuddhastighet

Skuddhastighet (Rate of fire) velges med knappen RATE. Denne bytter mellom 3 forskjellige modus og starter alltid på singelskudd. Når knappen trykkes byttes det over til 5 skudd serie, neste trykk aktiverer full automat, og neste trykk aktiverer singelskudd igjen.



### 5.4 Zoom

Zoom knappene styrer zoom nivået. Zoom nivåene varierer fra 1x til 34x. Trykk på "IN" for å zoome inn ett hakk, og "OUT" for å zoome ut ett hakk.



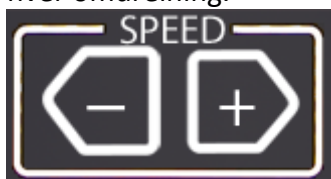
### 5.5 Nattmodus og Thermal vision

DAY/NIGHT knappen styrer hvilken optikk som skal vises på skjermen. I tillegg til den vanlige optikken er RWS-en utstyrt med lysforsterkende og varmefølsom optikk. Ved å trykke på knappen byttes optikken mellom dag, natt og varmefølsom visning på siktet.



### 5.6 Sensitivitet

Sensitiviteten på joysticken styres med SPEED knappene. Her ligger det 4 nivåer som styrer hvor raskt RWS-en skal bevege på seg. Maks hastighet er 15rpm, det tar altså 4 sekunder for hver omdreining.

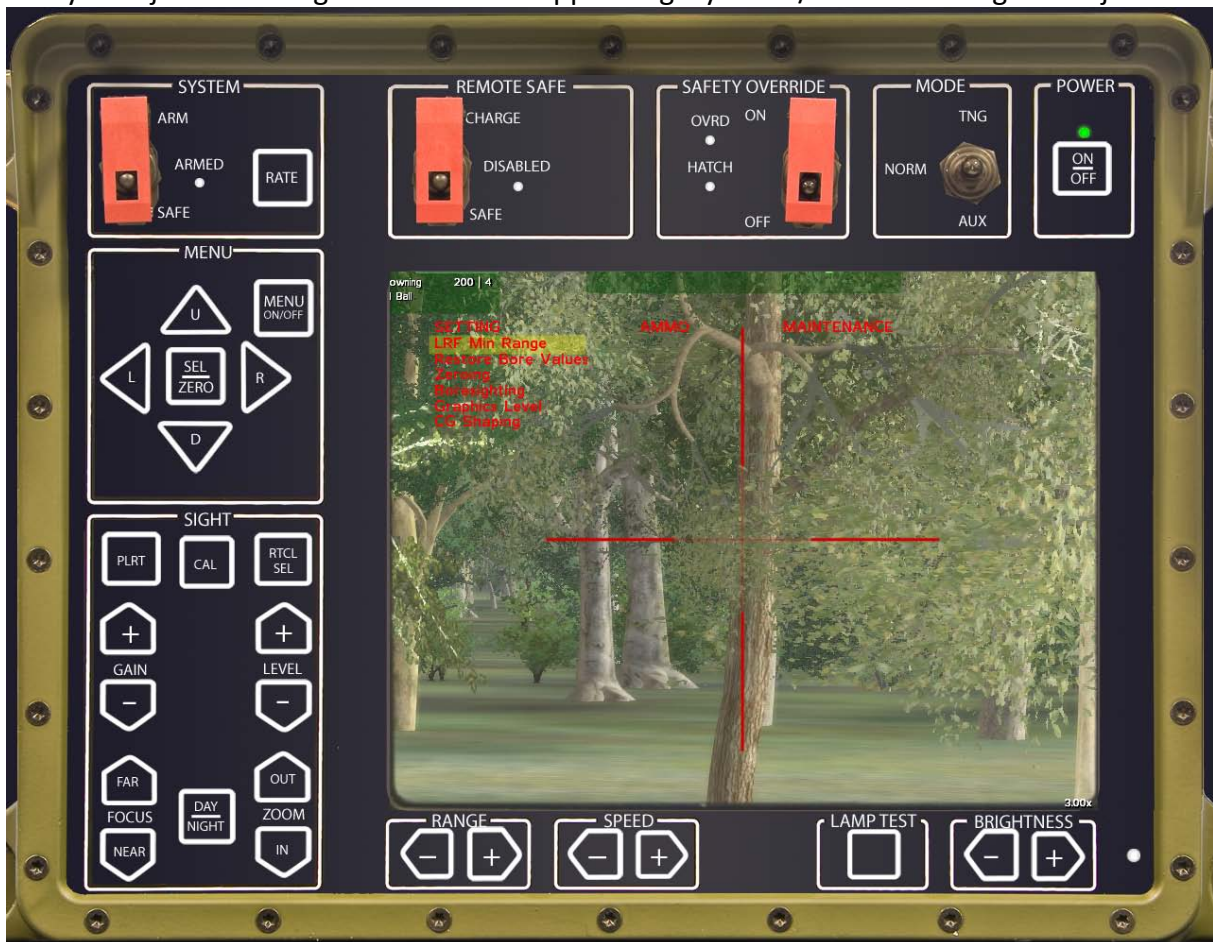


### 5.7 Granatkaster

For å bruke granatkasteren på RWS-CROWS skal denne armeres først. Hver RWS er utstyrt med 2 ladninger, upper og lower. For å armere granatkasteren må en av disse velges først. Når øvre eller nedre ladning er valgt kan sikkerhetsvippen slås opp, og ved å trykke på avfyringsbryteren vil ladningen bli skutt ut i retningen der siktet peker. Merk at granatkasteren ikke kan lades om.

## 5.8 Meny

Menyen slås på og av ved å trykke på MENU ON/OFF knappen, her ligger alle menyfunksjonene. Naviger med UDLR knappene og trykk SEL/ZERO for å velge funksjoner.



## 5.9 På/Av knapp

For å avslutte bruken av RWS-en og forlate kjøretøyet skal ON/OFF knappen trykkes på, når dette er gjort vil figuren forlate kjøretøyet og brukeren får ta kontroll over den med alle VBS2 kommandoene igjen.



# KDA RIVET



## Designdokument

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                            | 6  |
| 1.1   | Hensikt med dokumentet.....                  | 6  |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 6  |
| 1.3   | Dokumenthistorie.....                        | 6  |
| 1.4   | Ansvarlig.....                               | 6  |
| 2     | Innledning.....                              | 7  |
| 2.1   | Oppbygning av dokumentet.....                | 7  |
| 3     | Arkitektur.....                              | 8  |
| 4     | Use Case Design Oversikt.....                | 9  |
| 4.1   | Use case Design.....                         | 9  |
| 4.2   | UC1 – Bemanne systemet.....                  | 10 |
| 4.2.1 | Kommunikasjonsdiagram.....                   | 10 |
| 4.2.2 | Klassediagram.....                           | 10 |
| 4.2.3 | Sekvensdiagram.....                          | 11 |
| 4.2.4 | Aktivitetsdiagram.....                       | 11 |
| 4.3   | UC2 – Sikte.....                             | 12 |
| 4.3.1 | Kommunikasjonsdiagram.....                   | 12 |
| 4.3.2 | Klassediagram.....                           | 12 |
| 4.3.3 | Sekvensdiagram.....                          | 13 |
| 4.3.4 | Aktivitetsdiagram.....                       | 13 |
| 4.4   | UC3 – Armering på/av.....                    | 14 |
| 4.4.1 | Kommunikasjonsdiagram.....                   | 14 |
| 4.4.2 | Klassediagram.....                           | 14 |
| 4.4.3 | Sekvensdiagram.....                          | 15 |
| 4.4.4 | Aktivitetsdiagram.....                       | 15 |
| 4.4.5 | Tilstandsdiagram.....                        | 16 |
| 4.5   | UC4 – Skyte.....                             | 17 |
| 4.5.1 | Kommunikasjonsdiagram.....                   | 17 |
| 4.5.2 | Klassediagram.....                           | 18 |
| 4.5.3 | Sekvensdiagram.....                          | 18 |

|  |                                 |    |
|--|---------------------------------|----|
| 4.5.4                                  | Aktivitetsdiagram .....         | 19 |
| 4.6                                    | UC5 – Velge skuddhastighet..... | 20 |
| 4.6.1                                  | Kommunikasjonsdiagram.....      | 20 |
| 4.6.2                                  | Klassediagram.....              | 20 |
| 4.6.3                                  | Sekvensdiagram .....            | 21 |
| 4.6.4                                  | Tilstandsdiagram .....          | 21 |
| 4.7                                    | UC6 – Slå på/av nattmodus ..... | 22 |
| 4.8                                    | Kommunikasjonsdiagram .....     | 22 |
| 4.8.1                                  | Klassediagram.....              | 22 |
| 4.8.2                                  | Sekvensdiagram .....            | 23 |
| 4.8.3                                  | Statediagram .....              | 23 |
| 4.9                                    | UC7 – Zoom .....                | 24 |
| 4.9.1                                  | Kommunikasjonsdiagram.....      | 24 |
| 4.9.2                                  | Klassediagram.....              | 24 |
| 4.9.3                                  | Sekvensdiagram .....            | 25 |
| 4.9.4                                  | Aktivitetsdiagram .....         | 25 |
| UC9 – Sikte_sensitivitet_joystick..... |                                 | 26 |
| 4.9.5                                  | Kommunikasjonsdiagram.....      | 26 |
| 4.9.6                                  | Klassediagram.....              | 26 |
| 4.9.7                                  | Sekvensdiagram .....            | 27 |
| 4.9.8                                  | Aktivitetsdiagram .....         | 27 |
| 5                                      | Klasseoversikt.....             | 28 |
| 6                                      | Komponent - RIVET CORE .....    | 29 |
| 6.1                                    | Klasse – CORE .....             | 30 |
| 6.1.1                                  | Metoder.....                    | 30 |
| 6.1.2                                  | Kommunikasjon - CORE.....       | 31 |
| 7                                      | Komponent – RIVET LAGER.....    | 32 |
| 7.1                                    | Klasse – VariableStorage .....  | 32 |
| 7.1.1                                  | Attributter .....               | 32 |
| 7.1.2                                  | Metoder.....                    | 32 |
| 8                                      | Komponent – RIVET GUI .....     | 33 |
| 8.1                                    | Klasse - Gui_Facade .....       | 34 |
| 8.1.1                                  | Metoder.....                    | 34 |



|        |                                    |    |
|--------|------------------------------------|----|
| 8.1.2  | Kommunikasjon – Gui_Facade .....   | 35 |
| 8.2    | Klasse – Background .....          | 36 |
| 8.2.1  | Attributter .....                  | 36 |
| 8.2.2  | Metoder.....                       | 36 |
| 8.3    | Klasse – ArmingSwitch .....        | 37 |
| 8.3.1  | Attributter .....                  | 37 |
| 8.3.2  | Metoder.....                       | 37 |
| 8.3.3  | Kommunikasjon – ArmingSwitch ..... | 38 |
| 8.4    | Klasse – FiringMode.....           | 39 |
| 8.4.1  | Attributter .....                  | 39 |
| 8.4.2  | Metoder.....                       | 39 |
| 8.4.3  | Kommunikasjon – FiringMode .....   | 40 |
| 8.5    | Klasse – NightVision.....          | 41 |
| 8.5.1  | Attributter .....                  | 41 |
| 8.5.2  | Metoder.....                       | 41 |
| 8.5.3  | Kommunikasjon – NightVision .....  | 41 |
| 8.6    | Klasse – Zoom .....                | 42 |
| 8.6.1  | Attributter .....                  | 42 |
| 8.6.2  | Metoder.....                       | 42 |
| 8.6.3  | Kommunikasjon – Zoom.....          | 42 |
| 8.7    | Klasse – Sensitivity.....          | 44 |
| 8.7.1  | Attributter .....                  | 44 |
| 8.7.2  | Metoder.....                       | 44 |
| 8.7.3  | Kommunikasjon – Sensitivity .....  | 45 |
| 8.8    | Klasse – OnOFF .....               | 46 |
| 8.8.1  | Attributter .....                  | 46 |
| 8.8.2  | Metoder.....                       | 46 |
| 8.9    | Klasse – GrenadeArmer .....        | 47 |
| 8.9.1  | Attributter .....                  | 47 |
| 8.9.2  | Metoder.....                       | 47 |
| 8.9.3  | Kommunikasjon – GrenadeArmer.....  | 48 |
| 8.10   | Klasse – GrenadeFlipper .....      | 49 |
| 8.10.1 | Attributter .....                  | 49 |



|        |   |    |
|--------|---|----|
| 8.10.2 | Metoder.....                                | 49 |
| 8.10.3 | Kommunikasjon – GrenadeFlipper.....         | 50 |
| 8.11   | Klasse – BattleOverride .....               | 51 |
| 8.11.1 | Attributter .....                           | 51 |
| 8.11.2 | Metoder.....                                | 52 |
| 8.11.3 | Kommunikasjon - BattleOverride.....         | 52 |
| 8.12   | Klasse – BattleOverrideFlipper .....        | 53 |
| 8.12.1 | Attributter .....                           | 53 |
| 8.12.2 | Metoder.....                                | 53 |
| 8.12.3 | Kommunikasjon – BattleOverrideFlipper.....  | 54 |
| 8.13   | Klasser som ikke er planlagt utviklet ..... | 55 |
| 9      | Komponent – RIVET TOLK.....                 | 56 |
| 9.1    | Klasse – Vbs2 Interface.....                | 56 |
| 9.1.1  | Metoder.....                                | 56 |
| 9.1.2  | Kommunikasjon – VBS2Interface.....          | 57 |
| 10     | Referanser .....                            | 58 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med designdokumentet er å videreutvikle analysen, til den grad at vi vet hvordan vi skal implementere systemet. Dokumentet vil inneholde en oversikt over systemets arkitektur og spesifikk oppbygning av komponenter. Dette vil gjøres ved hjelp av klasse-, sekvens- og aktivitetsdiagrammer. Disse diagrammene følger UML-standarden. Vi vil også oppdatere designet under implementasjonen, da det mest sannsynlig vil endres.

## 1.2 Definisjoner, akronymer og forkortelser

**DLL:** Dynamic-link Library

**UM:** Unified Modeling Language

**VBS2:** Virtual Battle Space 2

**ASI:** Application Scripting Interface

**RIVET:** RWS In VBS2 Environment Testing

**CORE:** Command Operation Retrieval Entity

**TOLK:** Two-way Organized Language Kit

**LAGER:** Long Abbreviation That Isn't Quite Right

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                              | Redaktør                              |
|---------------|----------|--|---------------------------------------|
| 1.1           | 19.04.10 | Oppdatere                              | Arne Kr.                              |
| 1.2           | 23.04.10 | Dokumentstandard                       | Lone                                  |
| 1.3           | 24.05.10 | Oppdatert analyse og designdiagramer   | Ole-Martin Grøtterud                  |
| 1.4           | 26.05.10 | Lagt til flere klasser og forklaringer | Arne Kristian Åmellem                 |
| 1.99          | 27.05.10 | Korrekturlesing                        | Zoran Vukobratovic og Thomas Transeth |
| 2.0           | 28.05.10 | Klargjort for innlevering              | Thomas Transeth                       |

## 1.4 Ansvarlig

Arne Kristian Åmellem

## 2 Innledning

---

Basert på Analysen og erfaringer vi har gjort oss, har vi laget et mer detaljert design. Vi har av designarbeid gjort en mengde vurderinger og tatt mange valg, disse blir presentert og begrunnet i dette dokumentet. Disse valgene vil danne grunnlaget for implementasjonen. Underveis i implementasjonen vil arbeidet dokumenteres i teknologidokumenter. Til slutt vil det ligge ved et implementasjonsdokument som sammen med teknologidokumentene (og andre dokumenter) vil ta over for designdokumentet og beskrive hva sluttresultatet ble.

### 2.1 Oppbygning av dokumentet

*Vi har valgt å ha følgende oppbygning av dokumentet:*

Først skal vi vise hvordan systemet er oppdelt (kapittel 3).

Videre tar vi for oss designet for alle Use Casene og deres diagrammer (Kapittel 4).

Deretter vil vi ta hver komponent for seg. Vi starter med en kort beskrivelse av hovedfunksjonaliteten til komponentene ved hjelp av klassediagrammer. Videre vil vi beskrive hver klasse og hvordan disse jobber (Kapittel 5-8). Til slutt for hver klasse beskriver vi kommunikasjonen fra den aktuelle klassen.

### 3 Arkitektur

Dette kapitlet beskriver koblingen mellom alle programvare-komponentene i prosjektet.

Ut ifra analysen har vi kommet fram til fire hovedkomponenter i systemet.

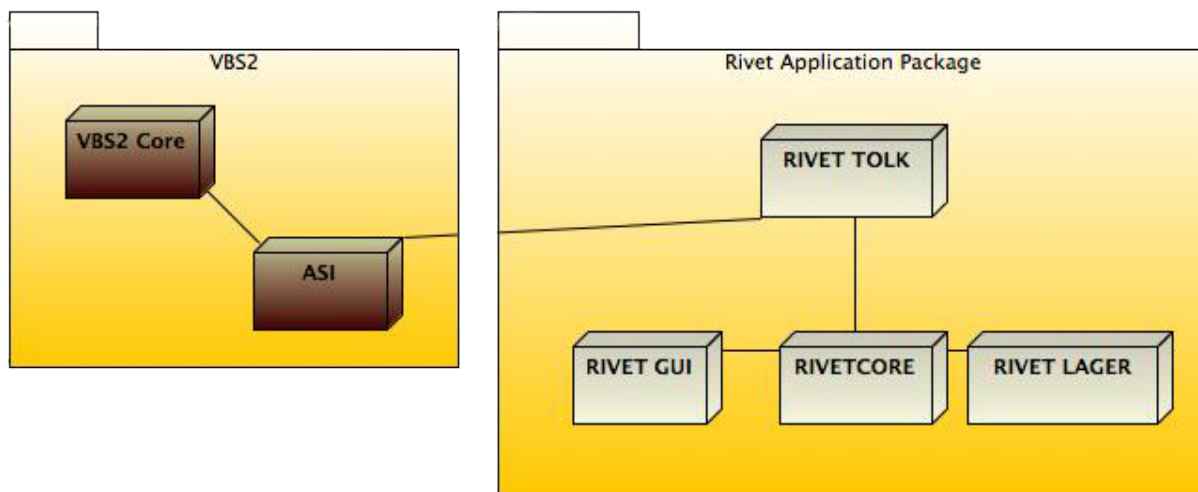
- Komponent – RIVET CORE (se kap. 6)
- Komponent – RIVET LAGER (se kap. 7)
- Komponent – RIVET GUI (se kap. 8)
- Komponent – RIVET TOLK (se kap. 9)

Disse vil komponentene vil kjøre separat, men kommunisere sammen og samarbeide som beskrevet senere. Det er en stor fordel at hver komponent kan til en hvis grad implementeres og testes hver for seg.

Nedenfor er det et diagram over de komponentene vi har kommet frem til.

Til venstre(VBS2) vises de to hovedkomponentene til simuleringsmiljøet vi skal benytte oss av. VBS2 Core er grafikk motoren og den funksjonalitet. ASI er den modulen som vi skal kommunisere med får å oppnå ønsket resultat.

Til høye(Rivet Application Package) er de pakkene vi skal lage får at KDA skal kunne bruke VBS som en plattform til deres simulator trener.

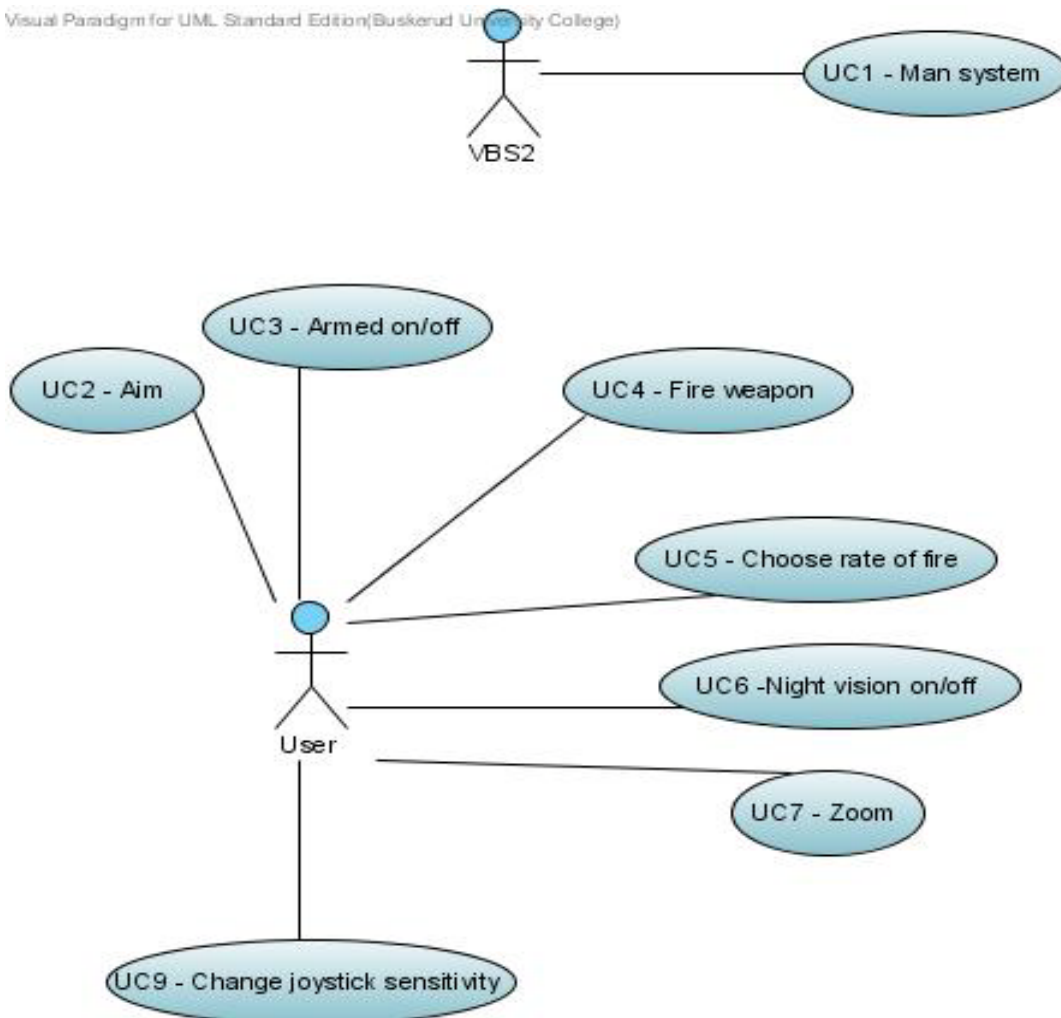


## 4 Use Case Design Oversikt

---

### 4.1 Use case Design

Visual Paradigm for UML Standard Edition (Buskerud University College)

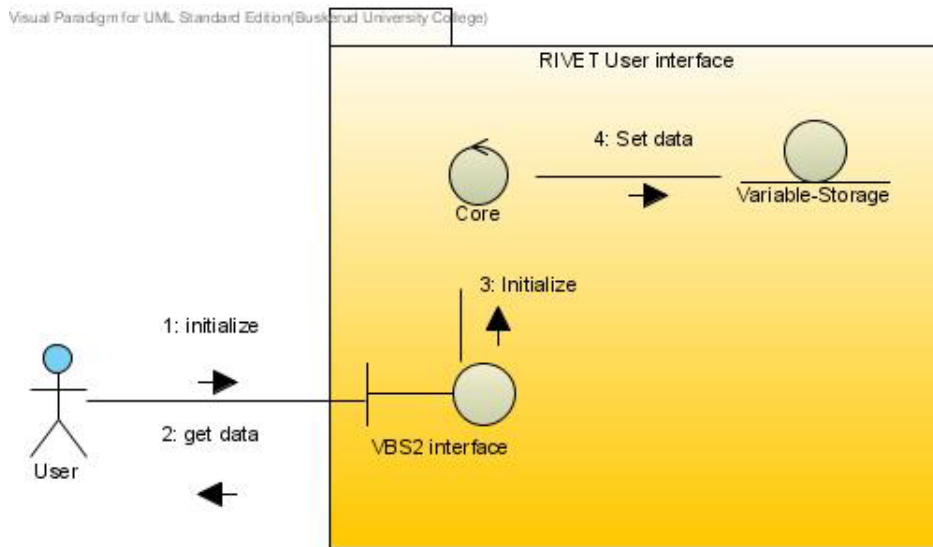


Dette er en oversikt over de use casene vi har kommet opp med til nå, som beskrevet i kravdokumentet. Designet for hver enkelt use case vil følge i de neste avsnittene.

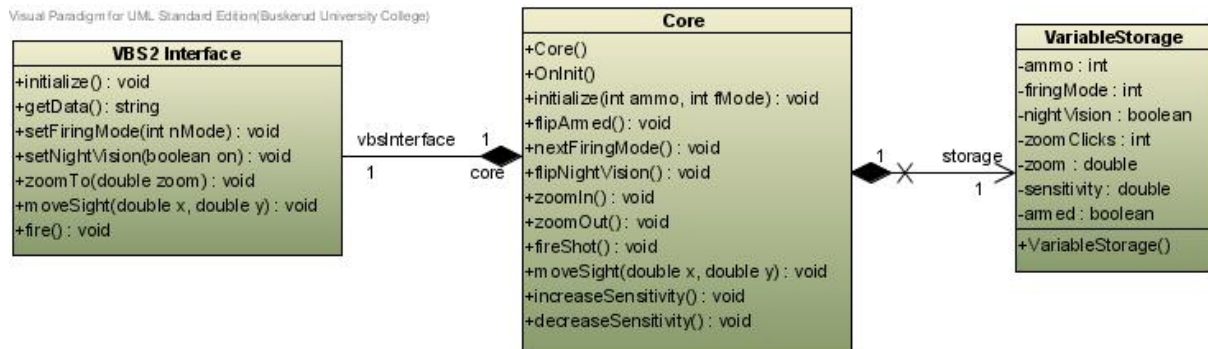
## 4.2 UC1 – Bemanne systemet

For å bemanne systemet må VBS2 sende en kommando til brukergrensesnittet vårt som forteller at det skal starte. Brukergrensesnittet skal da hente informasjon fra VBS2 om nåværende status og sende dette videre til kontrollklassen. Denne vil så lagre informasjonen og initialisere brukergrensesnittet.

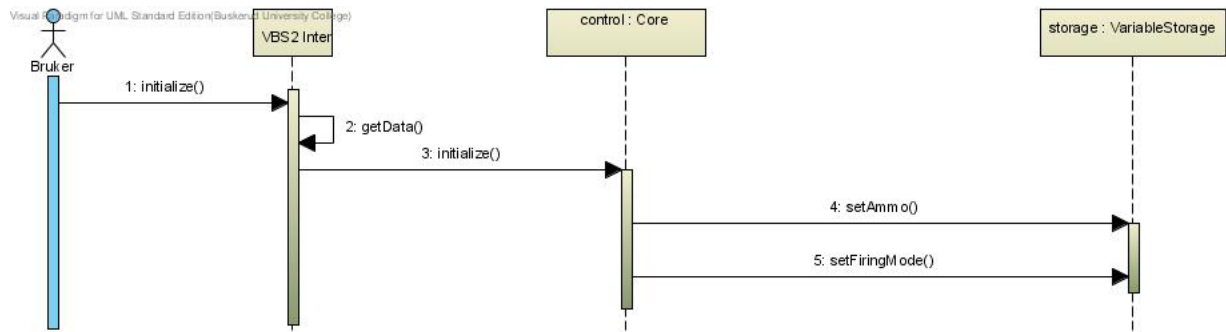
### 4.2.1 Kommunikasjonsdiagram



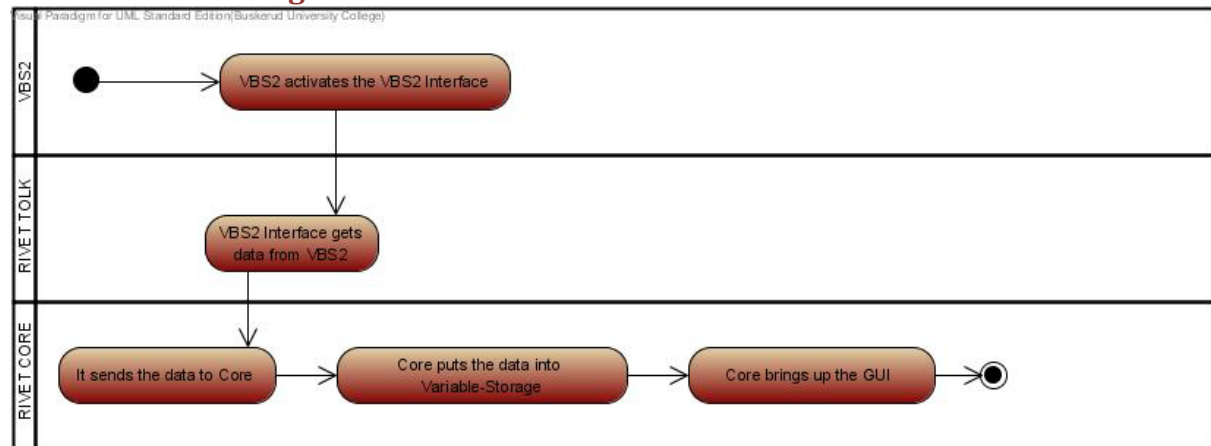
### 4.2.2 Klassediagram



### 4.2.3 Sekvensdiagram



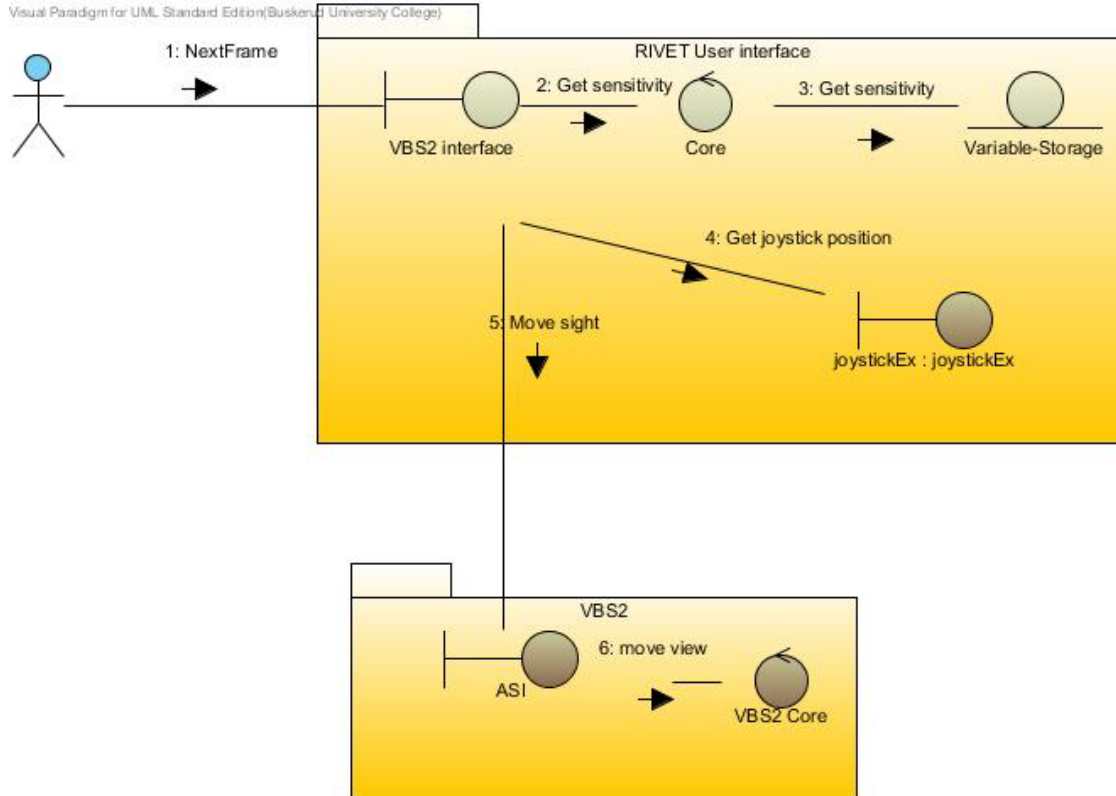
### 4.2.4 Aktivitetsdiagram



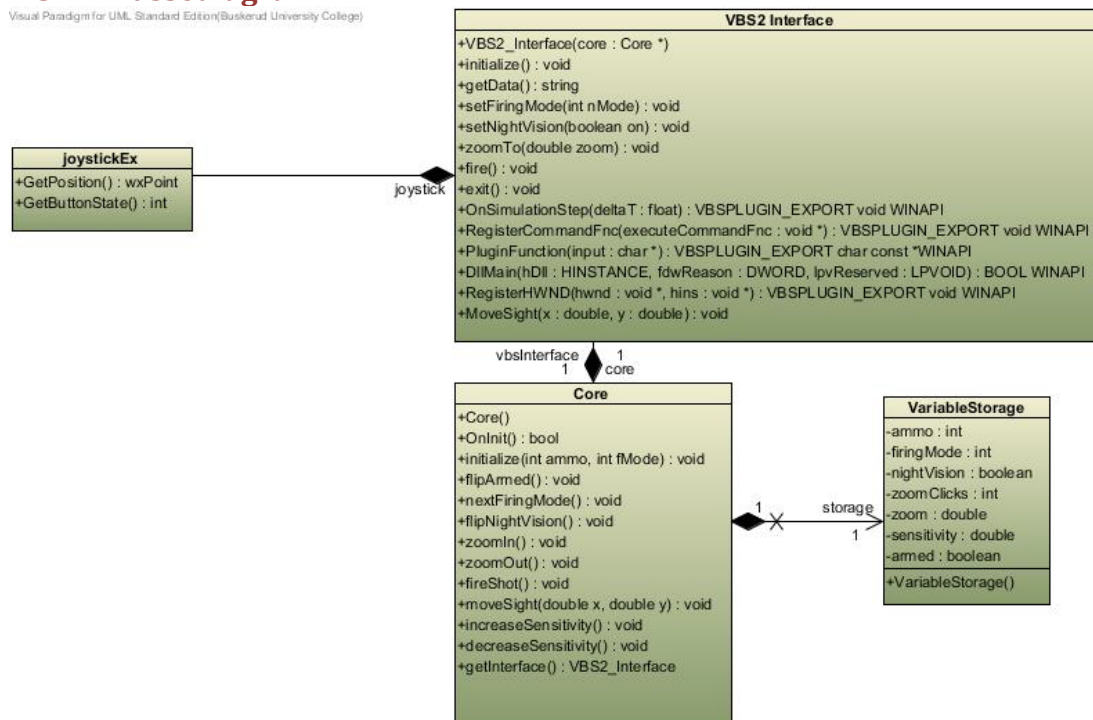
### 4.3 UC2 - Sikte

Her skal systemet vårt motta joystick-bevegelser fra brukeren. Denne vil så bli sendt til CORE-klassen som multipliserer den med sensitiviteten og videreformidler dette til VBS2.

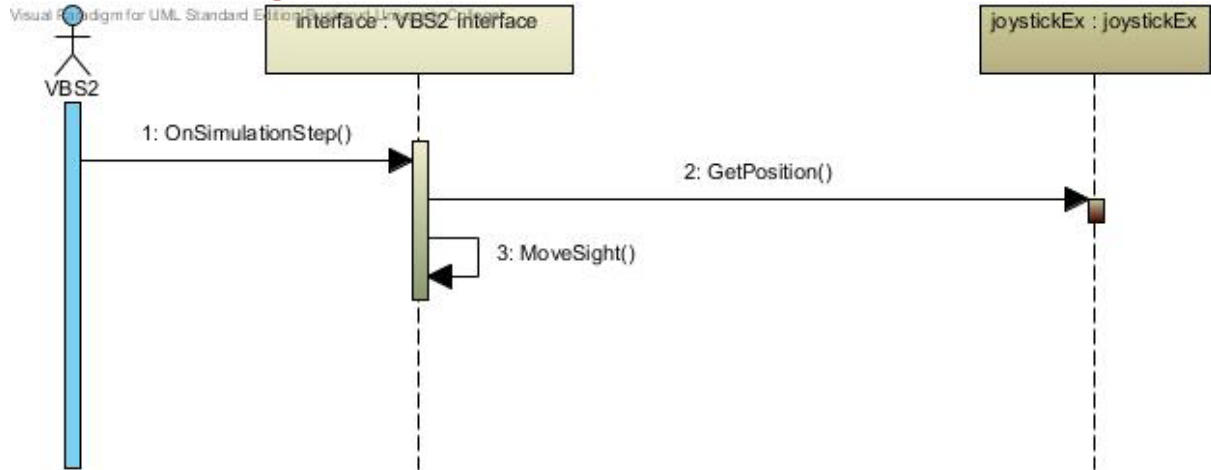
#### 4.3.1 Kommunikasjonsdiagram



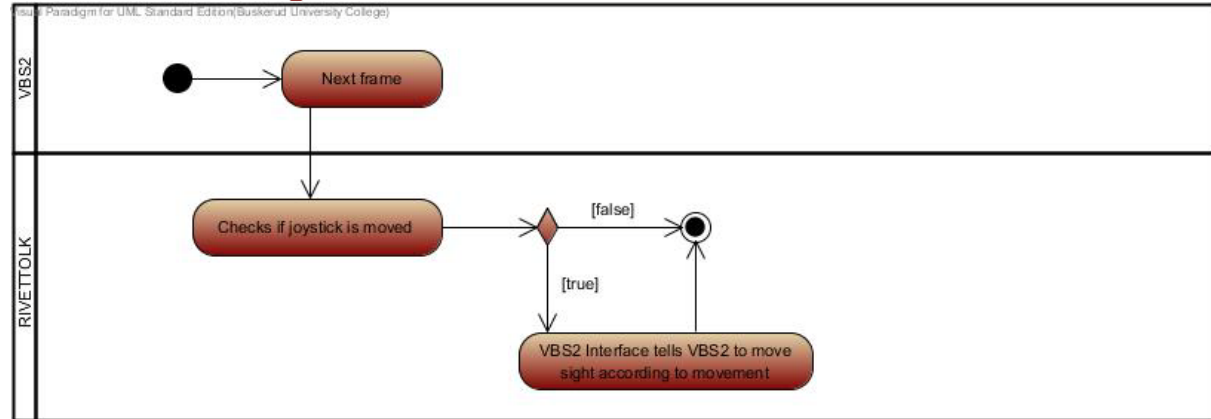
#### 4.3.2 Klassediagram



### 4.3.3 Sekvensdiagram



### 4.3.4 Aktivitetsdiagram

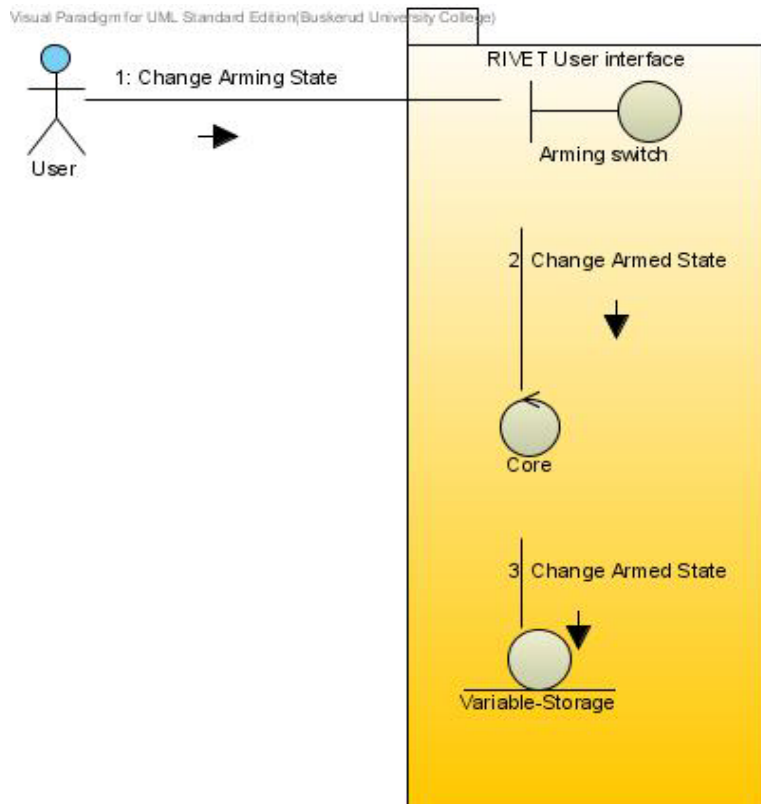




## 4.4 UC3 – Armering på/av

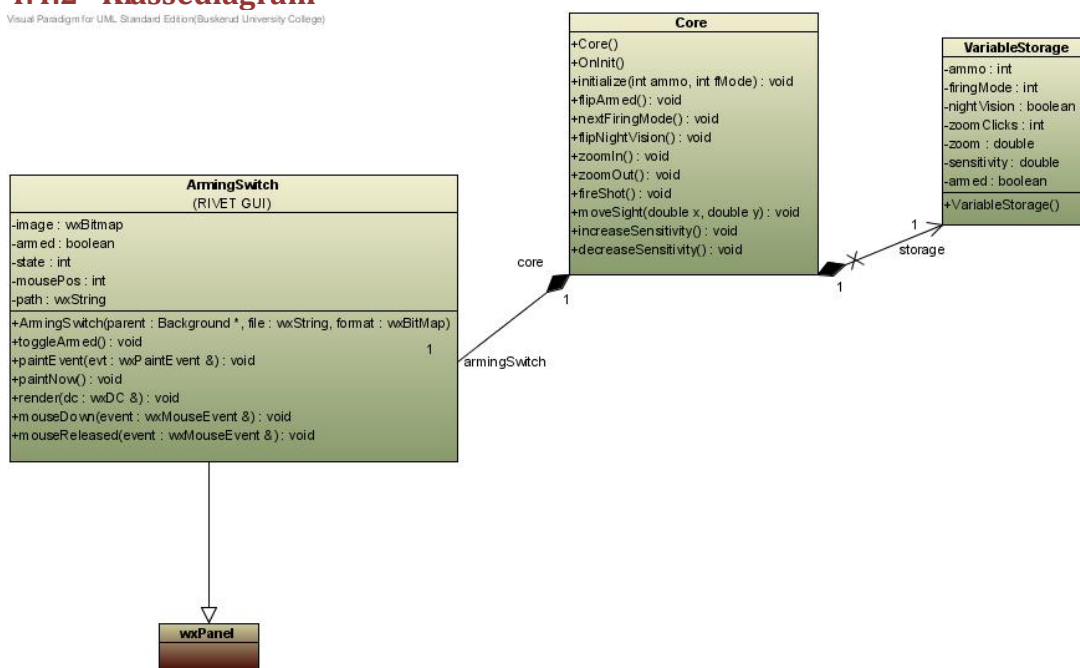
Ved å trykke på armeringsbryteren skal det sendes en beskjed til kontrollklassen, som vil skru armert av hvis den var på, eller på hvis den var av.

### 4.4.1 Kommunikasjonsdiagram

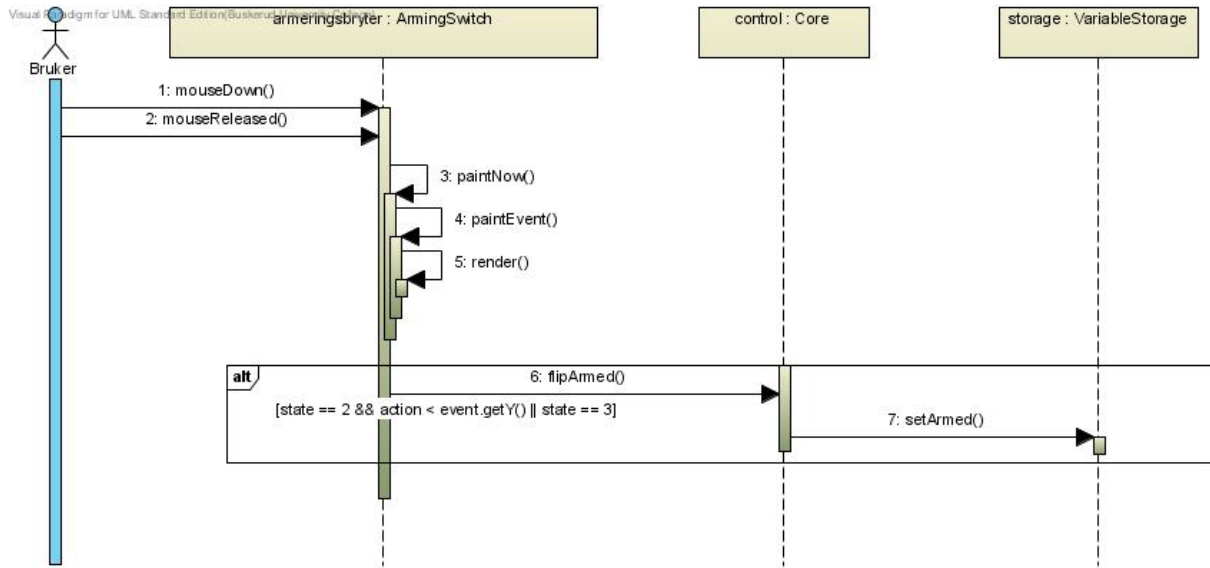


### 4.4.2 Klassediagram

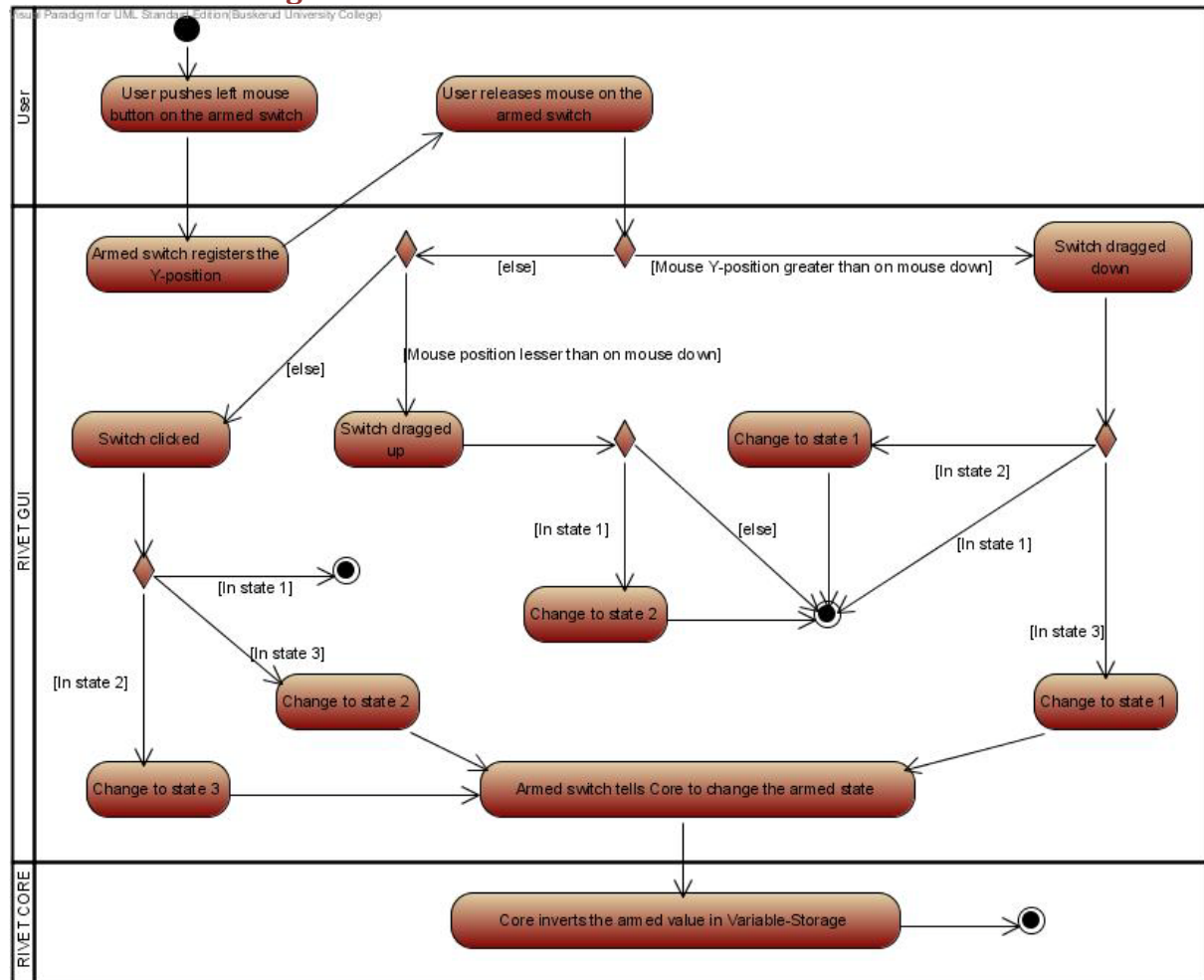
Visual Paradigm for UML Standard Edition (Buskerud University College)



### 4.4.3 Sekvensdiagram

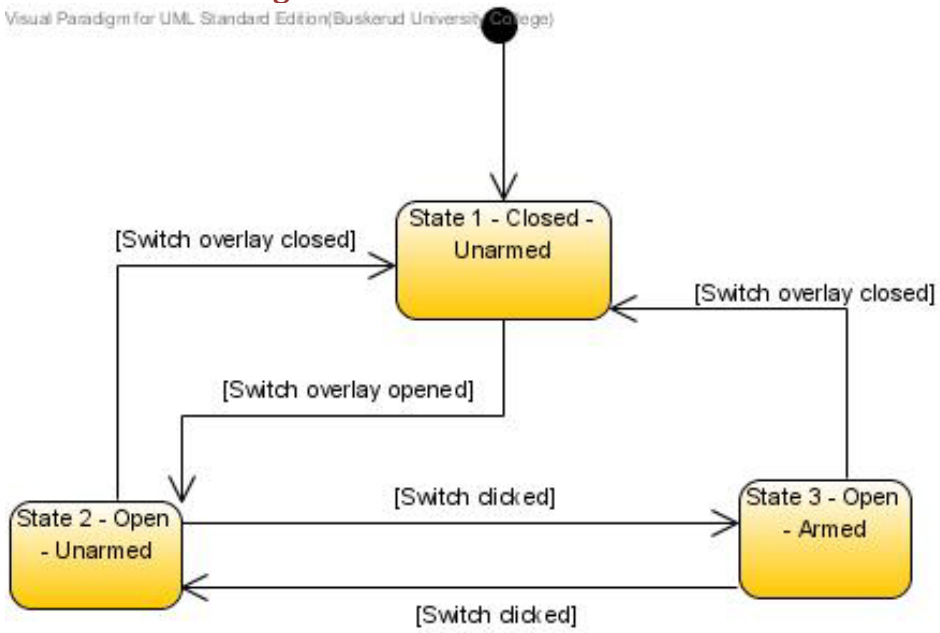


### 4.4.4 Aktivitetsdiagram



## 4.4.5 Tilstandsdiagram

Visual Paradigm for UML Standard Edition (Buskerud University College)

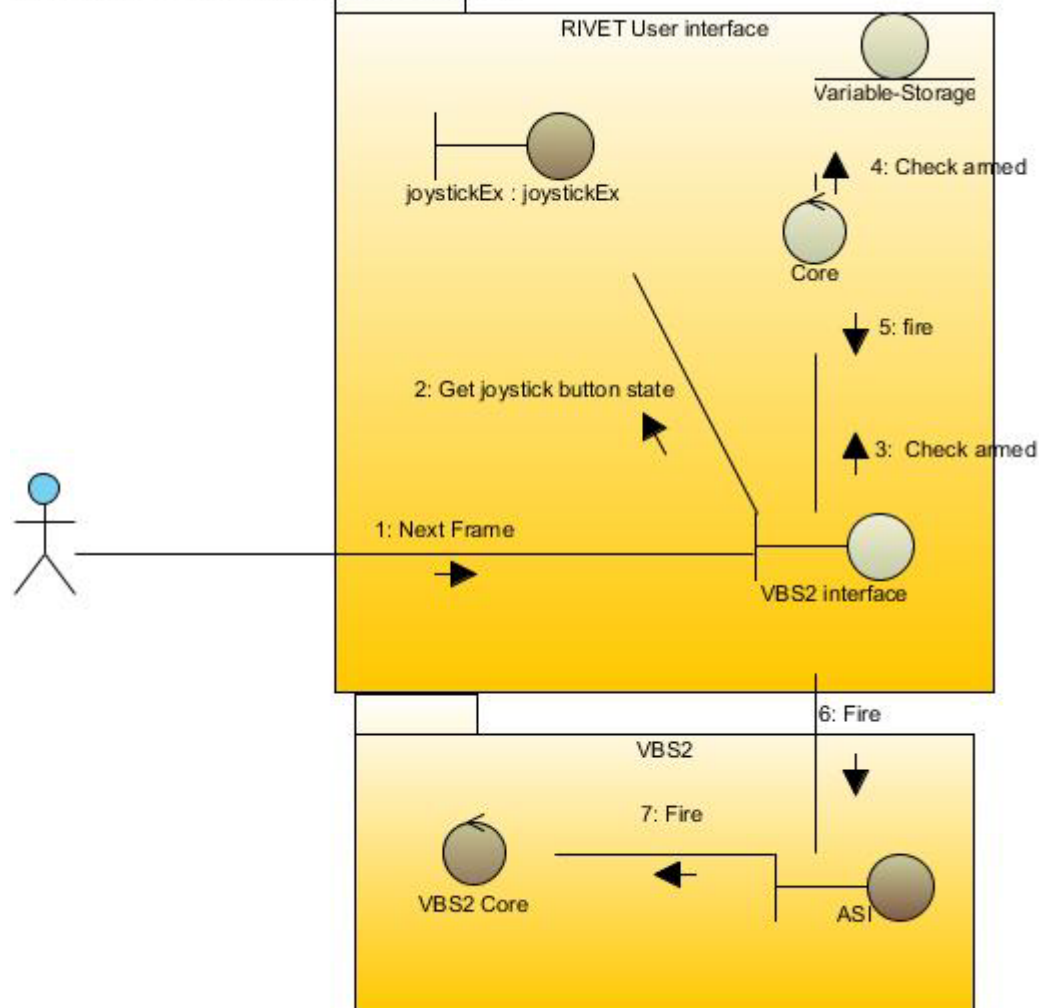


## 4.5 UC4 - Skyte

Ved å trykke på skyteknappen skal systemet motta dette og formidle det videre til kontrollklassen. Denne vil så se om systemet er armert, og hvis det er, sende kommandoen videre til VBS2 som vil avfyre våpenet.

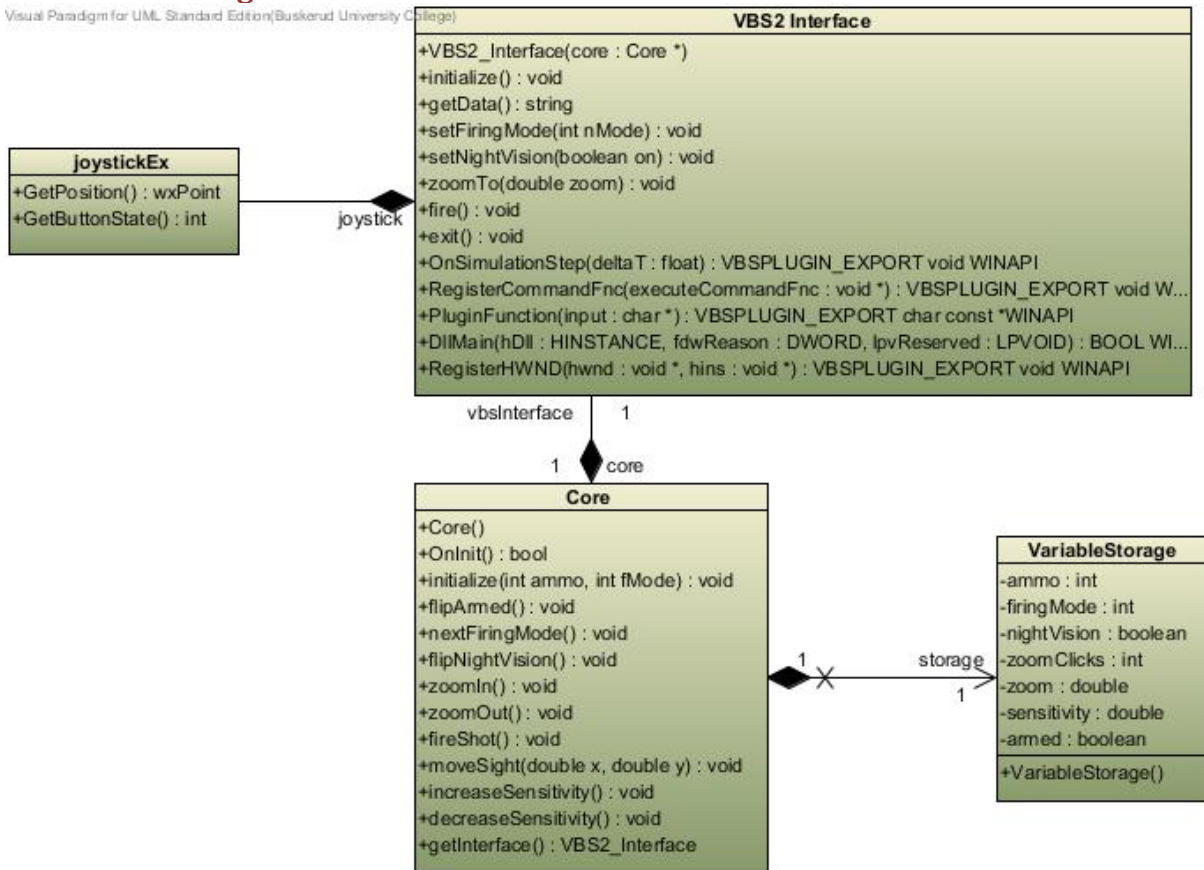
### 4.5.1 Kommunikasjonsdiagram

Visual Paradigm for UML Standard Edition (Baskerud University College)



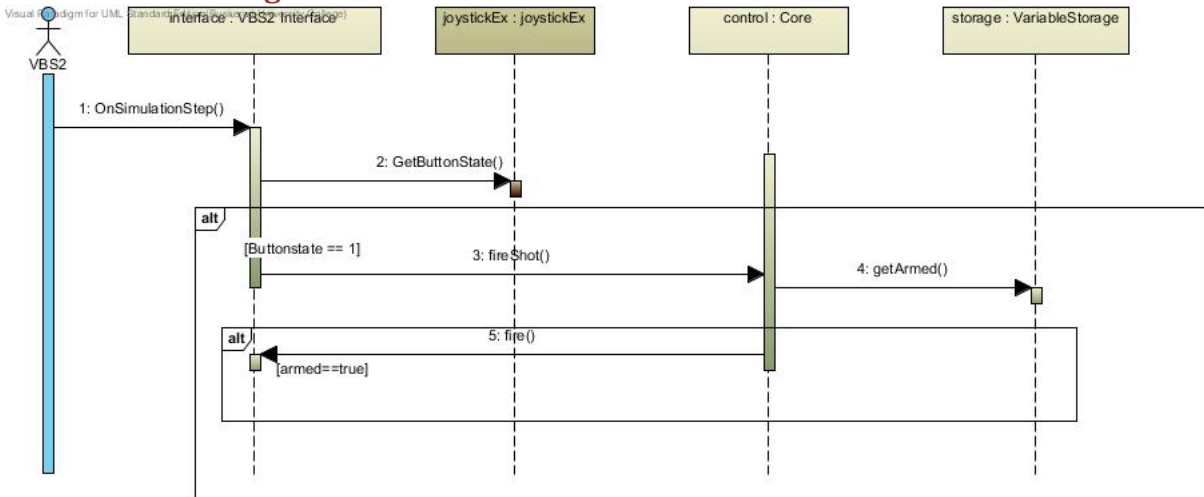
## 4.5.2 Klassediagram

Visual Paradigm for UML Standard Edition (Buskerud University College)

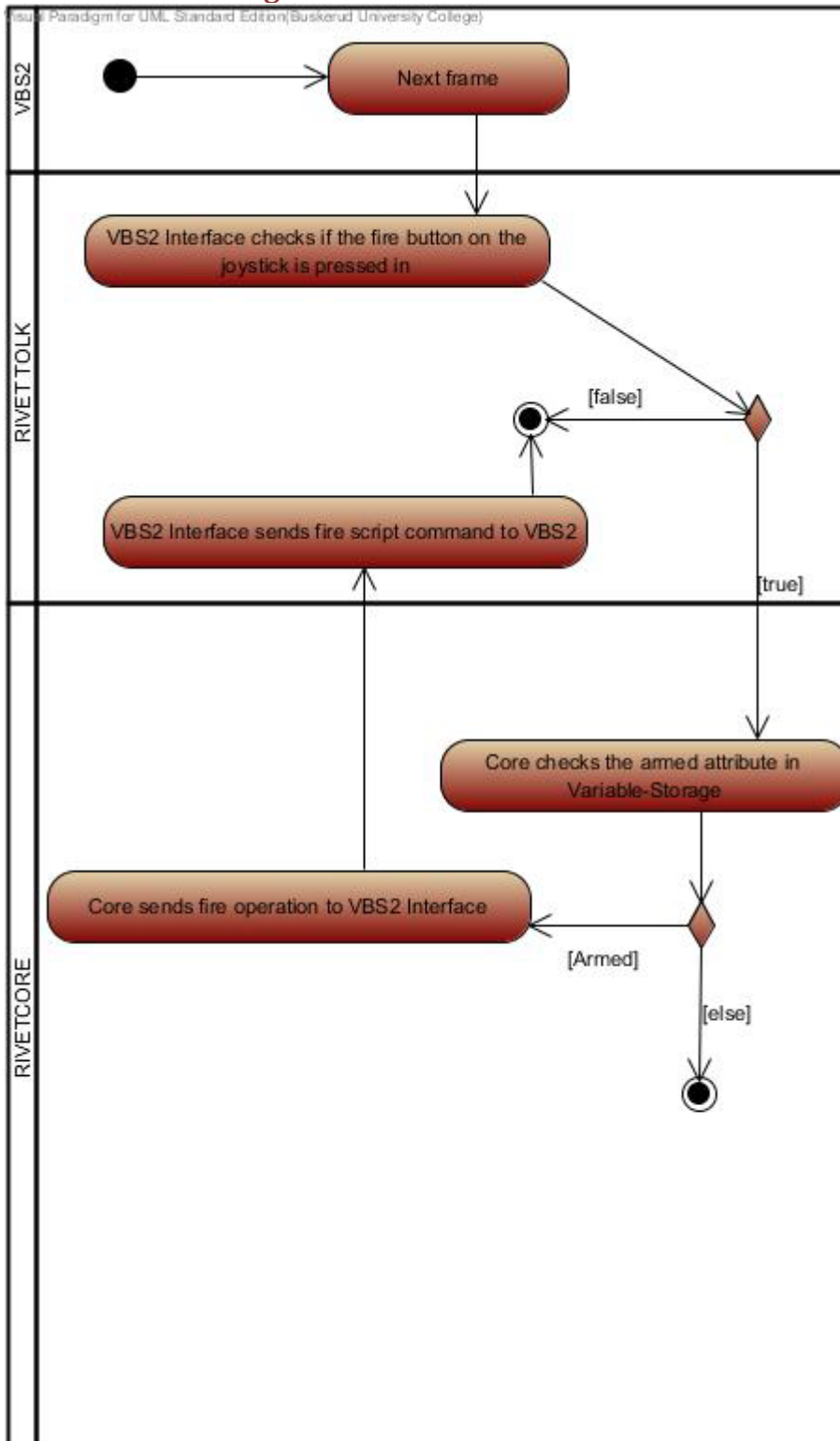


## 4.5.3 Sekvensdiagram

Visual Paradigm for UML Standard Edition (Buskerud University College)



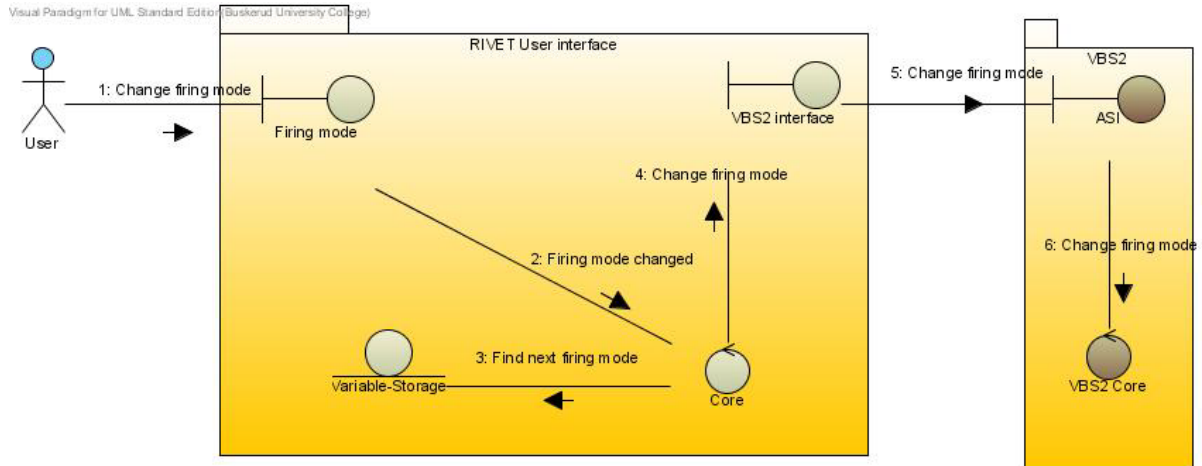
#### 4.5.4 Aktivitetsdiagram



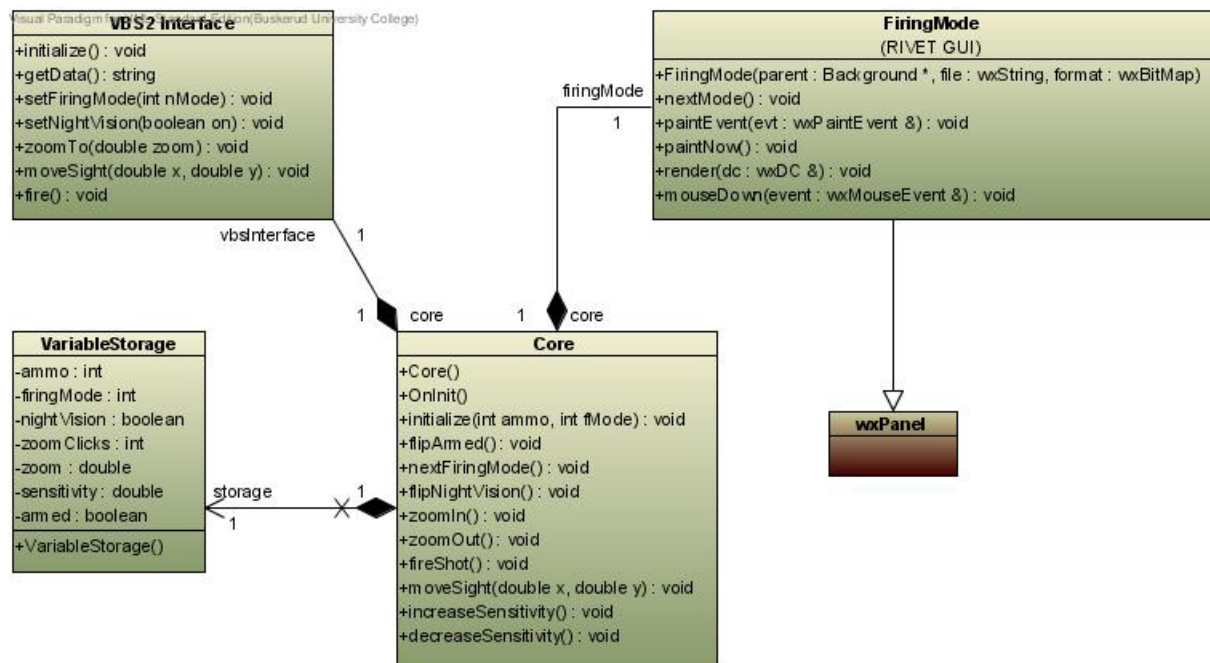
## 4.6 UC5 – Velge skuddhastighet

Ved å trykke på neste skuddhastighet skal systemet viderefremidle dette til kontrollklassen. Denne vil så hente ut neste skuddhastighet fra lageret og viderefremidle hvilken skuddhastighet som skal velges til VBS2.

### 4.6.1 Kommunikasjonsdiagram

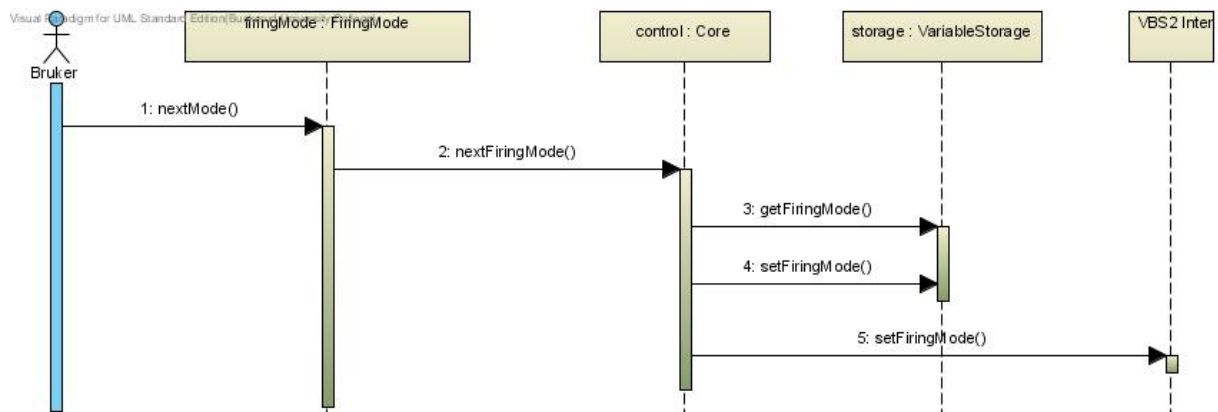


### 4.6.2 Klassediagram

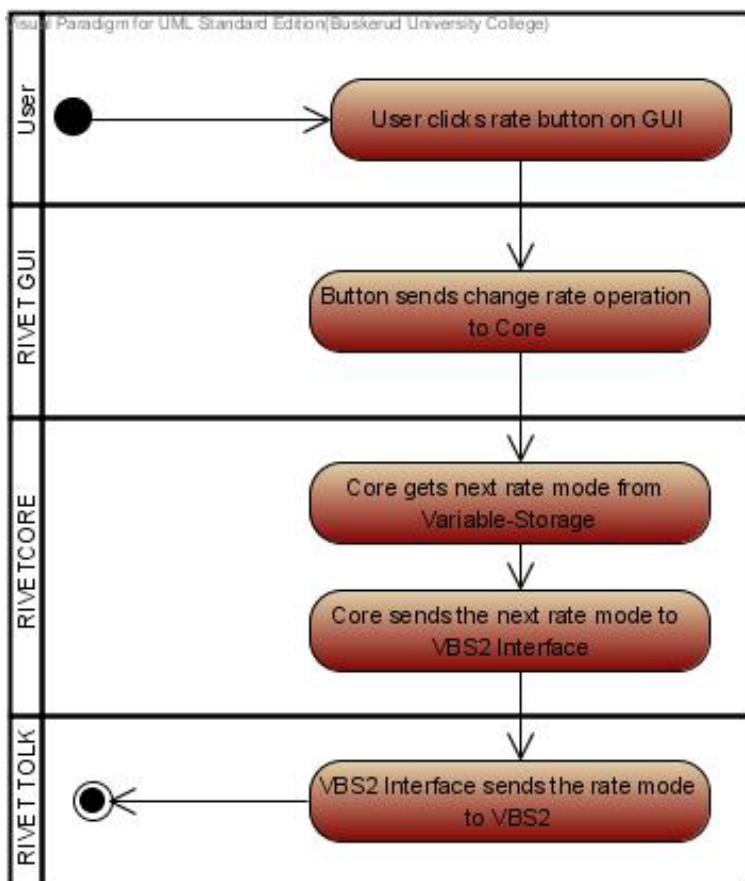




### 4.6.3 Sekvensdiagram



### 4.6.4 Tilstandsdiagram

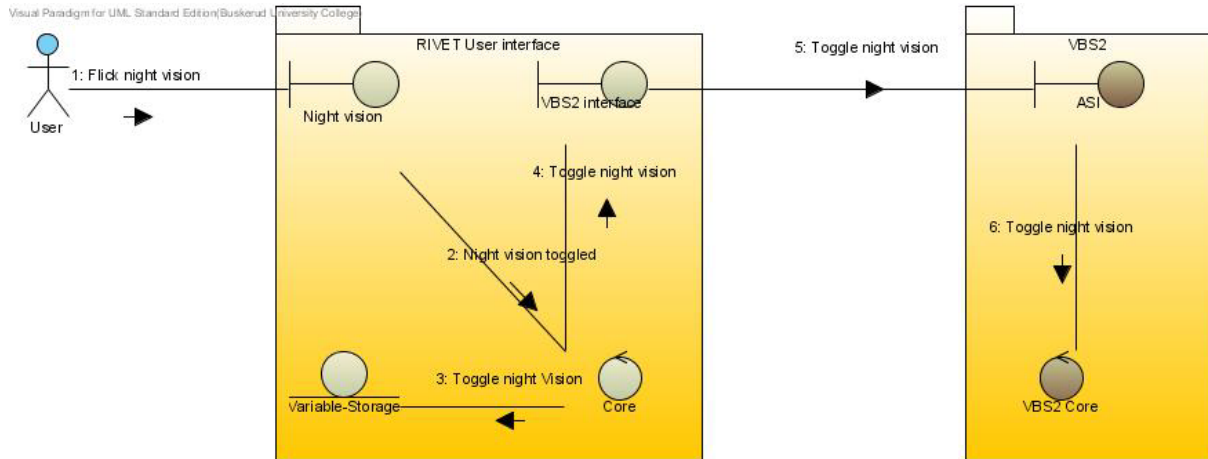




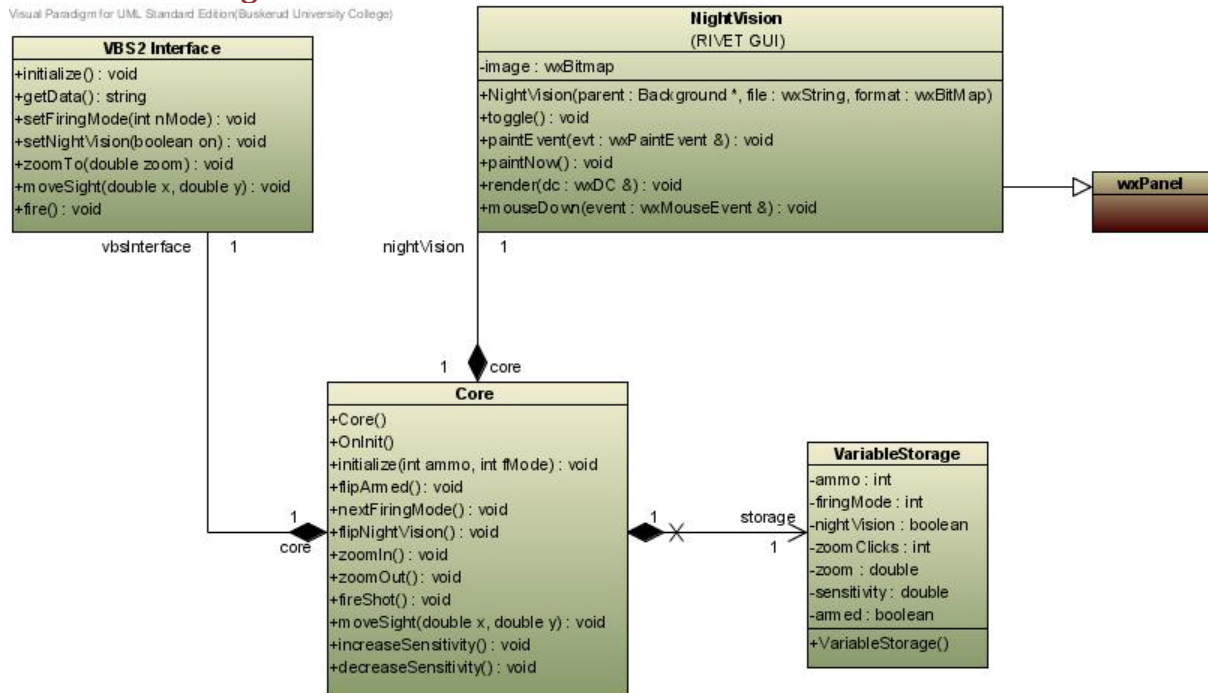
## 4.7 UC6 – Slå på/av nattmodus

Når brukeren slår nattmodus av eller på skal systemet viderefremde at nattmodus har blitt endret til kontrollklassen. Denne vil så endre den lagrede informasjonen om nattmodus og sende dette videre til VBS2 som slår nattmodusen av eller på.

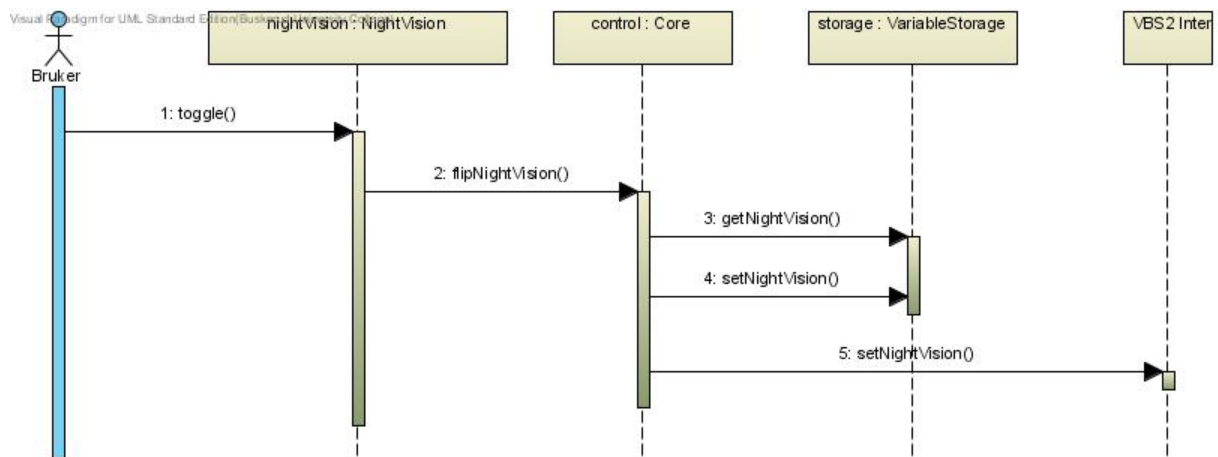
## 4.8 Kommunikasjonsdiagram



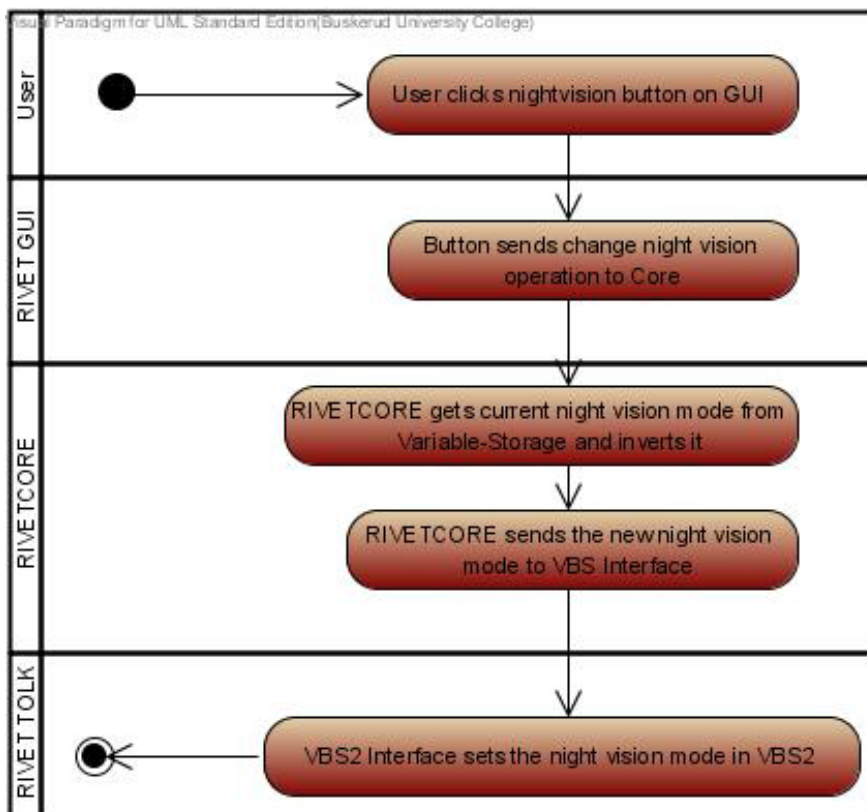
### 4.8.1 Klassediagram



## 4.8.2 Sekvensdiagram



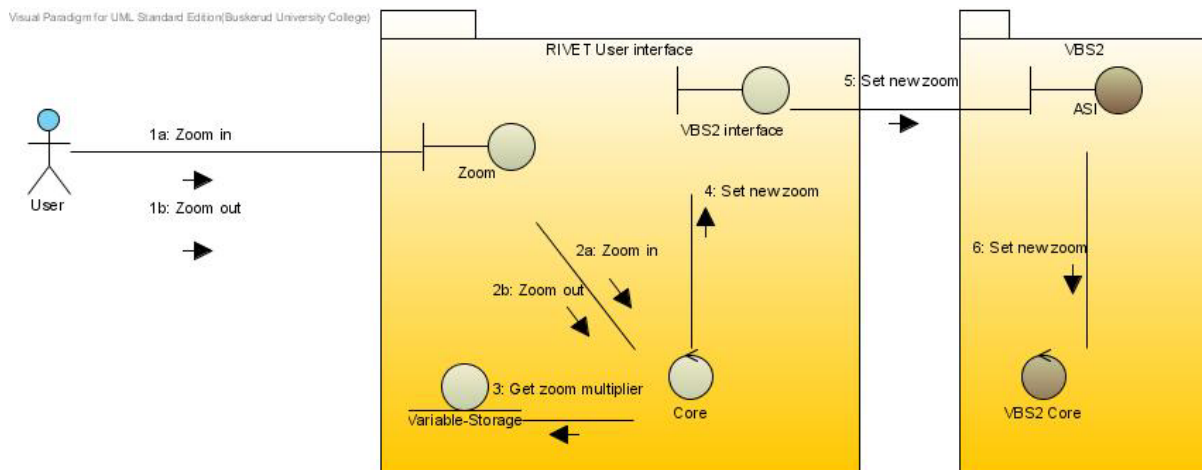
## 4.8.3 Statediagram



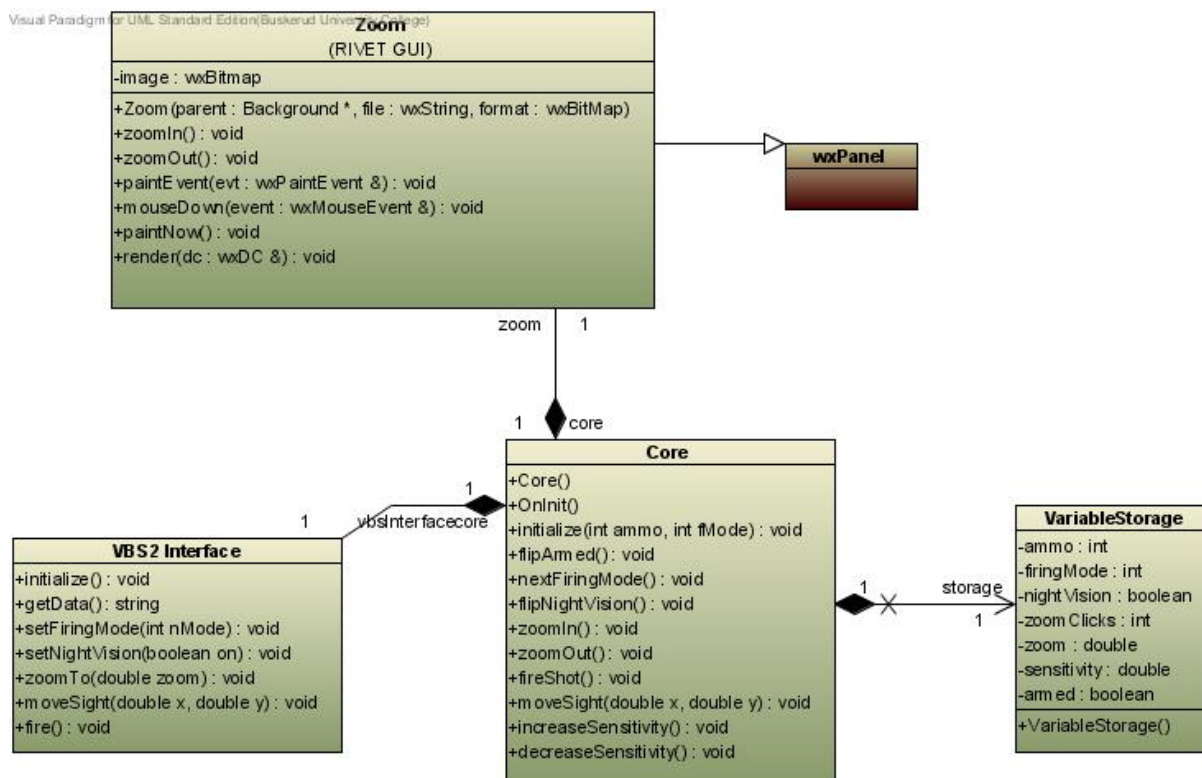
## 4.9 UC7 - Zoom

Når brukeren trykker zoom inn eller zoom ut skal den respektive kommandoen bli sendt til kontrollklassen. Denne vil så gange på en zoom-faktor, som er hvor langt den skal zoome per trykk. Deretter vil den legge til eller trekke fra dette tallet fra nåværende lagret zoom og sende det nye zoom-nivået til VBS2. Brukeren kan også zoome med joysticken.

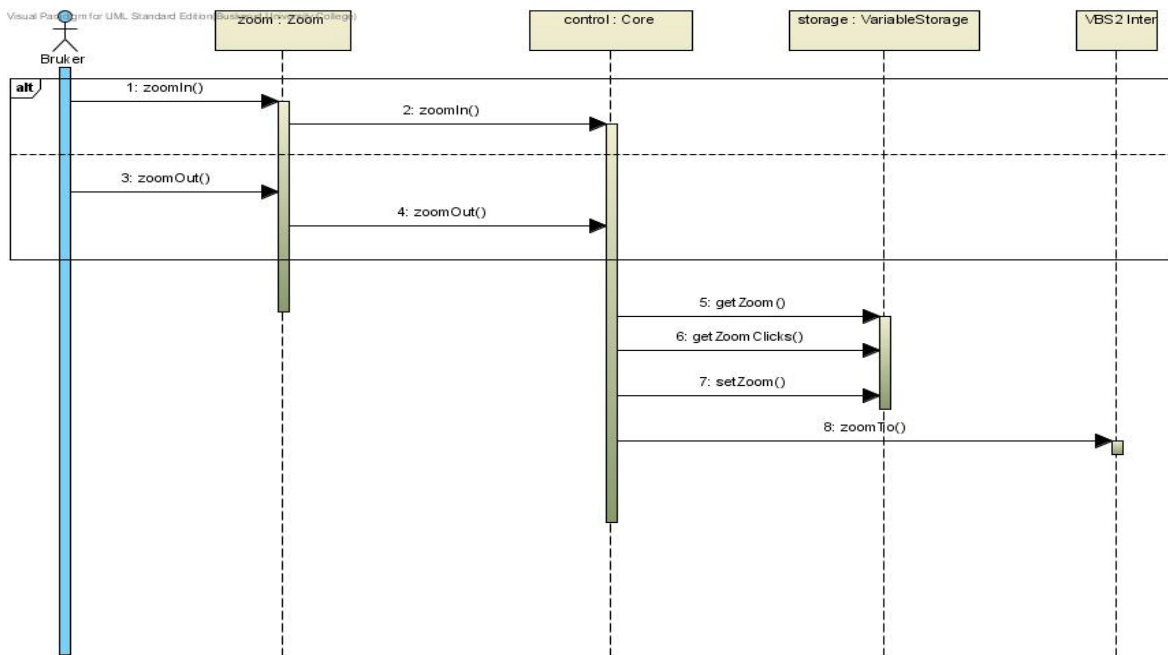
### 4.9.1 Kommunikasjonsdiagram



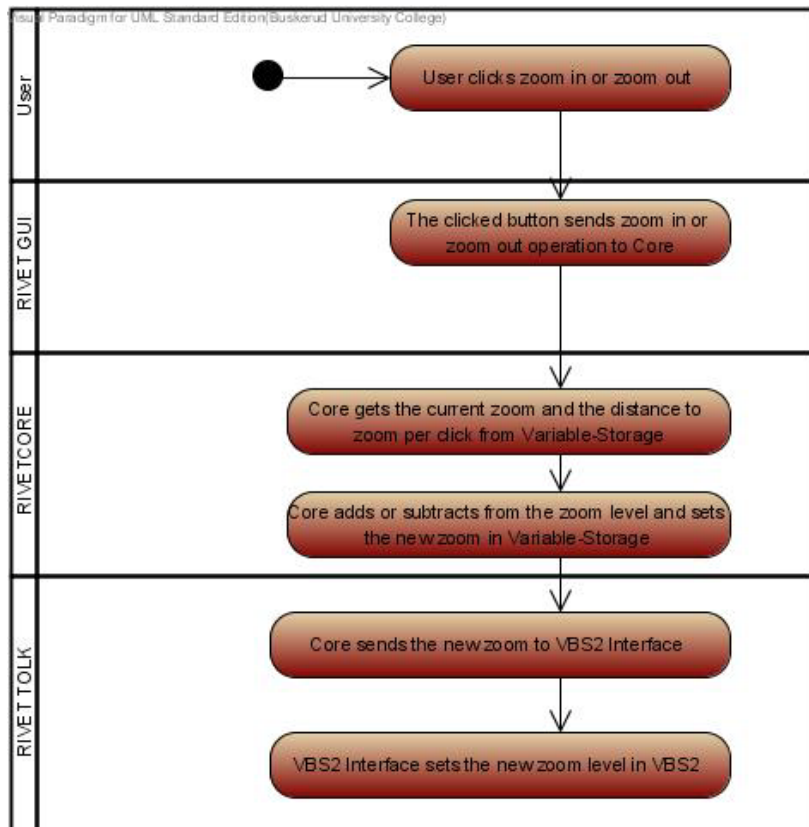
### 4.9.2 Klassediagram



### 4.9.3 Sekvensdiagram



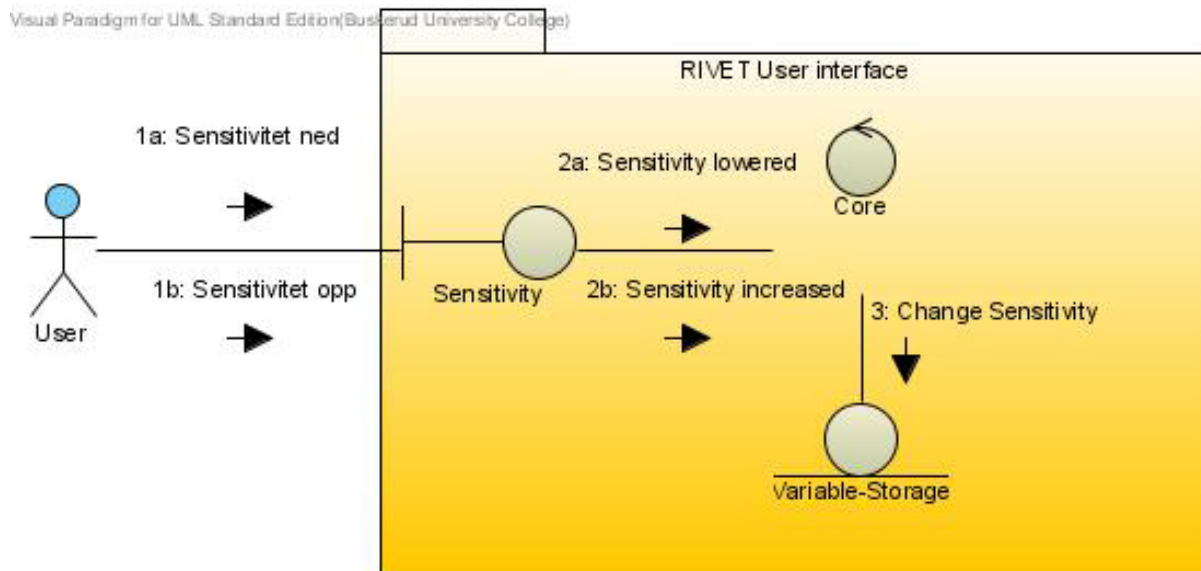
### 4.9.4 Aktivitetsdiagram



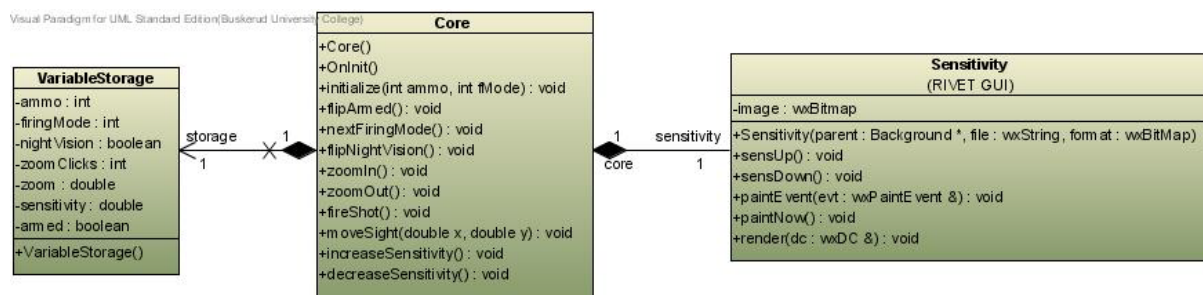
## UC9 – Sikte\_sensitivitet\_joystick

Brukeren skal kunne endre sensitiviteten opp eller ned som ønskelig. Systemet skal da sende respektive kommando til kontrollklassen som så endrer sensitiviteten i lageret.

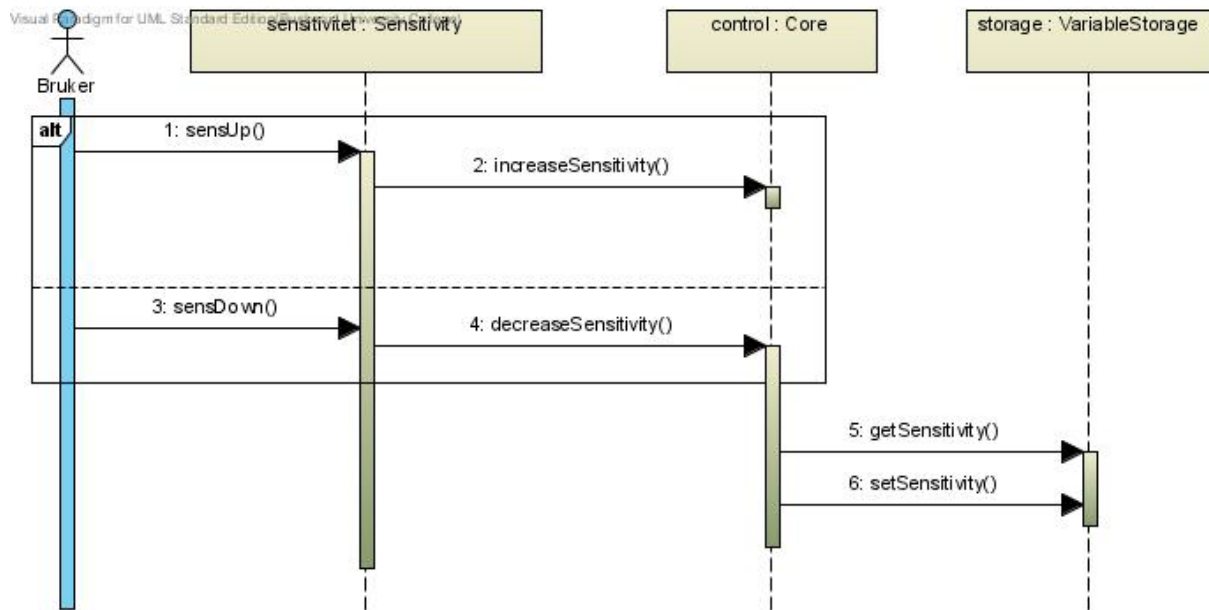
### 4.9.5 Kommunikasjonsdiagram



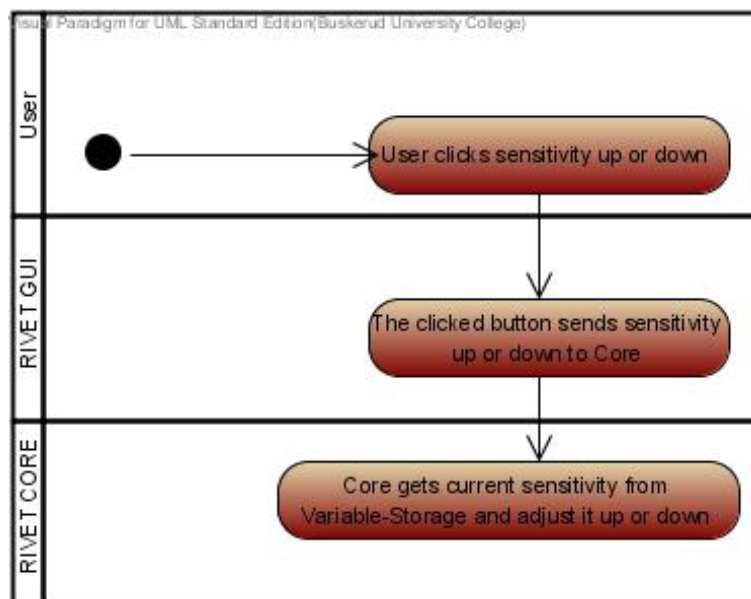
### 4.9.6 Klassediagram



## 4.9.7 Sekvensdiagram



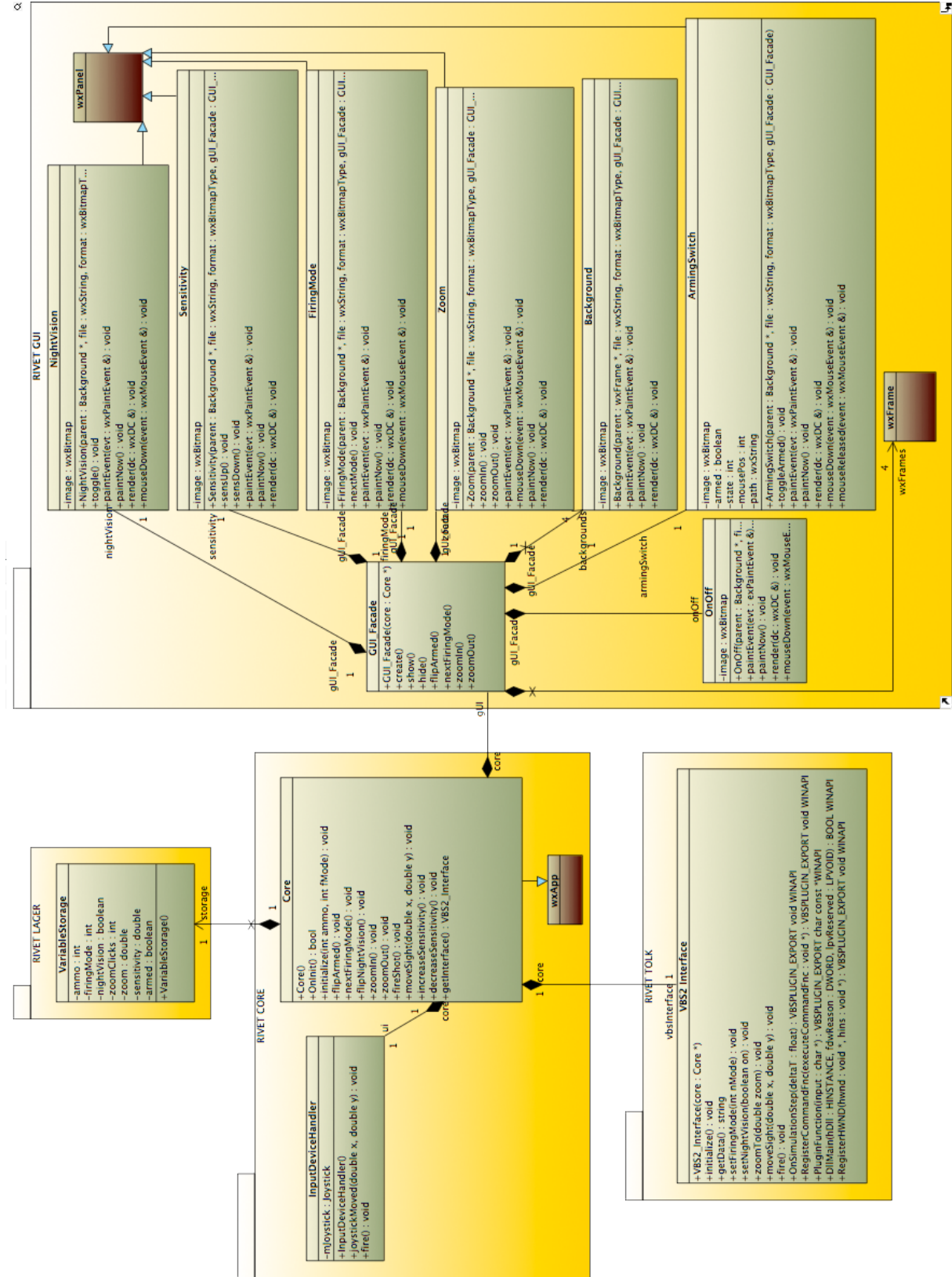
## 4.9.8 Aktivitetsdiagram





# 5 Klasseoversikt

Dette er et oversikts klassediagram for hele RIVET systemet. Som vist i deployment-diagrammet under, er systemet delt opp i 4 komponenter. I dette diagrammet har vi lagt inn rammer for de samme grupperingene slik at det skal være lettere å se sammenhengen.



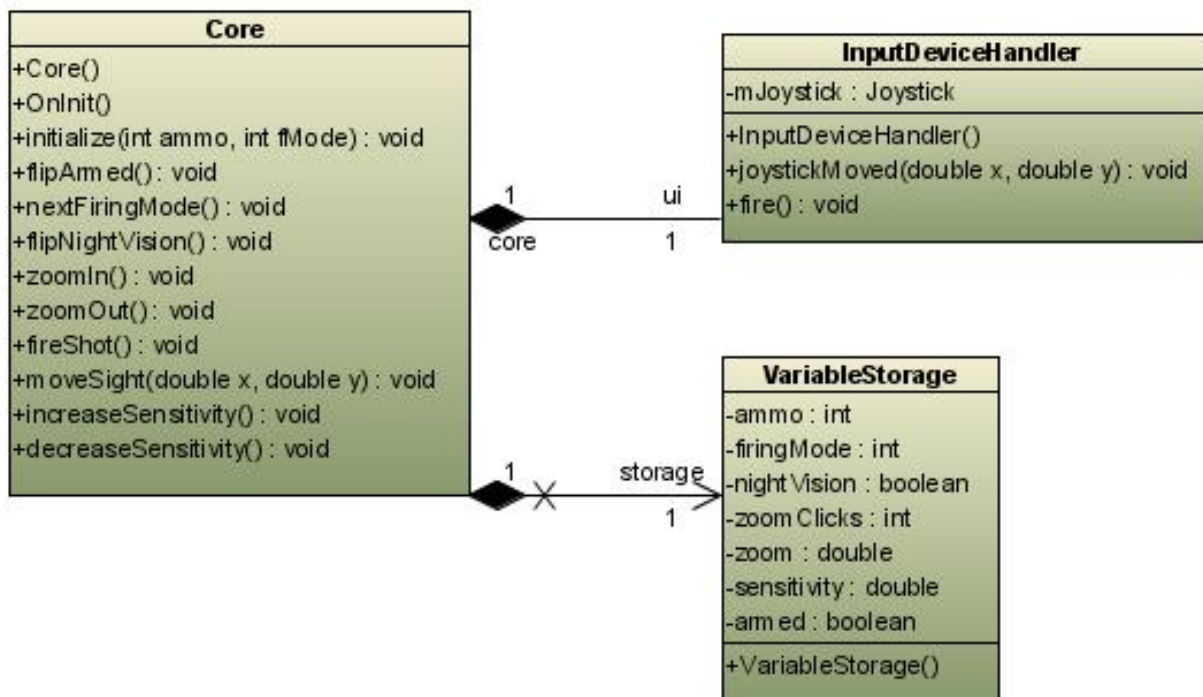
## 6 Komponent - RIVET CORE

RIVET CORE består hovedsakelig av tre viktige komponenter. Selve RIVET CORE, inputDeviceHandler og VariableStorage.

InputDeviceHandler er et objekt som skal ta seg av all input til systemet som kommer fra eksterne enheter, som joystick.

VariableStorage er en komponent som har i hovedsak én oppgave, nemlig det og holde alle variablene programmet trenger å huske på.

Til slutt er kommer RIVET CORE som starter opp og styrer InputDeviceHandler. I tillegg vil den få ansvaret for å starte alle komponentene i Rivet Application Package og VariableStorage. På denne måten vil systemet startes fra ett punkt, som er bra siden dette systemet blir en plugin til VBS2 og derfor ikke trenger og kunne startes på egenhånd.





## 6.1 Klasse - CORE

RIVET CORE har, som navnet tilsier som oppgave å styre hele Rivet Application Package. denne starter med å hente alle variabler den trenger i VariableStorage og deretter opprette resten av systemet. VariableStorage inneholder oppstartsinnstillinger til systemet, som våpentype, antall skudd, nattmodus av/på. Dette trenger RIVET CORE for at den skal kunne fungere korrekt.

```
Visual Paradigm for UML - Standard Edition (Buskerud Univ)
Core
+Core()
+OnInit()
+initialize(int ammo, int fMode) : void
+flipArmed() : void
+nextFiringMode() : void
+flipNightVision() : void
+zoomIn() : void
+zoomOut() : void
+fireShot() : void
+moveSight(double x, double y) : void
+increaseSensitivity() : void
+decreaseSensitivity() : void
```

### 6.1.1 Metoder

| Navn                 | Beskrivelse   |
|----------------------|---|
| +CORE                | Konstruktøren oppretter de objekter og setter de verdier som trengs.  |
| +OnInit              | Denne metoden oppretter samtlige objekter i Rivet application package.  |
| +Initialize          | Denne funksjonens eneste oppgave er å sette mengde ammunisjon og hvilke skuddhastighet som skal brukes.                                       |
| +FlipArmed           | Dette er en funksjon som sjekker og varsler om armerings bryter er av eller på slik at ved å armere våpenet på gui, så skjer det i VBS2 også. |
| +NextFiringMode      | Dette er en funksjon som gir beskjed videre til VBS2 når man har valgt neste skuddvalg på GUI'et.   |
| +FlipNightVision     | Denne funksjonen sørger for at valget om å skru på nattsikte blir overført til vbs2, slik at det trer i kraft.                                |
| +ZoomIn              | Dette er en funksjon som kontrollerer at brukeren kan zoome innover.  |
| +ZoomOut             | Dette er en funksjon som kontrollerer at brukeren kan zoome utover.   |
| +FireShoot           | Dette er en funksjon som lar brukeren skyte med våpenet, dette skjer ved at bruker klemmer på avtrekkeren (joystick).                         |
| +MoveSight           | Dette er en funksjon som lar brukeren flytte på siktet ved å rotere på våpenet.   |
| +IncreaseSensitivity | Dette er en funksjon som gjør det mulig å øke sensitiviteten på joysticken. Dette gjør at det beveger seg fortere.                            |

|                      |   |
|----------------------|---|
| +DecreaseSensitivity | Dette er en funksjon som gjør det mulig å senke sensitiviteten på joysticken. Dette gjør at det beveger seg saktere |
|----------------------|---|

### 6.1.2 Kommunikasjon - CORE

|        |  |
|--------|--|
| Fra    | CORE   |
| Til    | VariableStorage  |
| Info   | Hver eneste variabel i VariableStorage blir manipulert av CORE i form av set metoder. CORE kan også kalle på VariableStorages get metoder får å innhente de variablene som trengs. |
| Format | Metode kall  |

|        |   |
|--------|---|
| Fra    | CORE  |
| Til    | VBS2 Interface  |
| Info   | CORE kaller en rekke metoder i VBS2 Interface klassen som blir kalt opp. Disse metodene er vårt systems kall på funksjonalitet i VBS2 |
| Format | Metode kall   |

|        |   |
|--------|---|
| Fra    | CORE  |
| Til    | wxFrame   |
| Info   | CORE kaller setSize metodene til wxFrame får å sette ønsket størrelse på brukergrensesnittet. |
| Format | Metodekall  |

## 7 Komponent – RIVET LAGER

### 7.1 Klasse – VariableStorage

Denne klassen har en oppgave nemlig å ta vare på de forskjellige variablene som vi ønsker å lagre. Disse variablene ar alt fra ammunisjon, avfyringsmodus, nattsikte, armering av/på.

| VariableStorage         |
|-------------------------|
| -ammo : int             |
| -firingMode : int       |
| -night Vision : boolean |
| -zoom Clicks : int      |
| -zoom : double          |
| -sensitivity : double   |
| -armed : boolean        |
| +VariableStorage()      |

#### 7.1.1 Attributter

| Navn                 | Beskrivelse  |
|----------------------|--|
| -int ammo            | Denne variabelen inneholder antall skudd i våpenet.  |
| - int firingMode     | Denne variabelen inneholder en tallverdi som indikerer hvilken skuddmodussystemet er i.  |
| -boolean nightVision | Denne variabelen holder styr på om nattsiktet er aktivt eller ei.  |
| -int zoomClicks      | Denne variabelen inneholder en tallverdi som bestemmer hvor mange zoom-steg brukeren kan gjøre. Slik at man ikke kan zoome evig. |
| -double zoom         | Denne variabelen holder verdien hvor langt hvert zoom er.  |
| -double sensitivity  | Denne variabelen inneholder en tallverdi som indikerer hvor sensitive kontrollene er.  |
| -boolean armed       | Denne variabelen holder verdien om systemet er armert eller ei.  |

#### 7.1.2 Metoder

| Navn             | Beskrivelse                              |
|------------------|--|
| +VariableStorage | Konstruktøren initialiserer alle verdier |



## 8.1 Klasse - Gui\_Facade

Denne klassen har som oppgave å "samle" alle koblingene i gui-delen av systemet. Alle kall i gui-delen routes gjennom Gui\_Facade til eller fra GUI klassene. Dette er for at vi skal begrense antall koblinger til Core. Dette er et kjent pattern som forklart ovenfor.

| GUI_Facade                        |
|-----------------------------------|
| +GUI_Facade(core : Core *)        |
| +Create()                         |
| +ShowGrenadeGUI()                 |
| +ShowM2GUI()                      |
| +Hide()                           |
| +Exit()                           |
| +FlipArmed()                      |
| +VisionMode()                     |
| +ZoomIn()                         |
| +ZoomOut()                        |
| +GrenadeState(grenadeState : int) |
| +GrenadeFire()                    |
| +ShowMenu(menuEvent : int)        |

### 8.1.1 Metoder

| Navn            | Beskrivelse   |
|-----------------|---|
| +Gui_Facade     | Konstruktøren tar imot peker fra Core og lagrer denne.  |
| +Create         | Denne metoden oppretter begge GUI'ene og alt som hører med disse.   |
| +ShowGrenadeGUI | Denne metoden gjør Block2 GUI'et synlig når spiller entrer et kjøretøy med den RWS'en                     |
| +ShowM2GUI      | Denne metoden gjør CROWS GUI'et synlig når spiller entrer et kjøretøy med den RWS'en                      |
| +Hide           | Denne Metoden gjemmer alle GUI komponenter.   |
| +Exit           | Denne metoden gjemmer GUI og spilleren går ut av kjøretøyet. Viduet omstilles til riktig størrelse.       |
| +FlipArmed      | Denne metoden viderefører Armeringsbryterkallet til CORE.   |
| +VisionMode     | Denne metoden viderefører VisionModekallet til CORE.  |
| +ZoomIn         | Denne metoden viderefører ZoomInkallet til CORE.  |
| +ZoomOut        | Denne metoden viderefører ZoomOutkallet til CORE.   |
| +GrenadeState   | Denne metoden sjekker hvilken tilstand granatarmeringsbryteren står i og sender tilstand videre til CORE. |
| +GrenadeFire    | Denne metoden viderefører GrenadeFire kallet til CORE.  |
| +ShowMenu       | Denne metoden viderefører kallet om å vise menyen på skjermen.  |

### 8.1.2 Kommunikasjon - Gui\_Facade

|        |  |
|--------|--|
| Fra    | GuiFacade  |
| Til    | Core   |
| Info   | Alle kall routes igjennom Gui_Facade, derfor kommunikasjon begge veier |
| Format | Metodekall   |

|        |  |
|--------|--|
| Fra    | Gui_Facade   |
| Til    | Alle Guiklasser  |
| Info   | Alle kall routes igjennom Gui_Facade, derfor kommunikasjon begge veier |
| Format | Metodekall   |

## 8.2 Klasse - Background

Background klassen har som oppgave å vise og plassere bakgrunnen av GUI'et. Med dette mener vi panelet

som bryterne sitter på og alle statiske GUI komponenter. Klassen arver fra wxFrame.

| Background  |
|---|
| -image : wxBitmap   |
| +Background(parent : wxFrame *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void |

### 8.2.1 Attributter

| Navn            | Beskrivelse  |
|-----------------|--|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap. |

### 8.2.2 Metoder

| Navn        | Beskrivelse  |
|-------------|--|
| +Background | Konstruktøren tar imot peker fra wxFrame, Gui_Facade, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                      |
| +PaintEvent | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.  |
| +PaintNow   | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.                |
| +Render     | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge. |

## 8.3 Klasse – ArmingSwitch

Denne klassen arver fra wxPanel. Den har som oppgave å styre armeringsbryteren.

Bryteren kan man flippe opp

beskyttelsesdekselet og slå bryteren av og på.

Beskyttelsesdekselet må man klikke og dra

opp eller ned for at den skal endre tilstand mens selve armeringsbryteren under dekselet kan bare klikkes på. Dette gjør at det hele virker mer realistisk.

| ArmingSwitch  |
|---|
| -image : wxBitmap<br>-armed : boolean<br>-state : int<br>-mousePos : int<br>-path : wxString  |
| +ArmingSwitch(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+toggleArmed() : void<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void<br>+mouseDown(event : wxMouseEvent &) : void<br>+mouseReleased(event : wxMouseEvent &) : void |

### 8.3.1 Attributter

| Navn            | Beskrivelse  |
|-----------------|--|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap.                                   |
| -boolean armed  | Holder tilstanden på våpenet, om det er armet eller ei.                              |
| -int state      | Holder orden på hvilken tilstand vippebryteren er i, inklusivt beskyttelsesdekselet. |
| -int mousePos   | Denne variabelen holder rede på hvor musepekeren er på skjermen.                     |
| -wxString path  | Inneholder en string.  |

### 8.3.2 Metoder

| Navn          | Beskrivelse   |
|---------------|---|
| +ArmingSwitch | Konstruktøren tar imot peker fra Gui_Facade, background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.    |
| +ToggleArmed  | Denne metoden lar brukeren manipulere om våpenet er armet eller ikke.   |
| +PaintEvent   | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.   |
| +PaintNow     | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt. |
| +Render       | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner,                         |



|                |   |
|----------------|---|
|                | og denne kan samarbeide med dem begge   |
| +MouseDown     | Denne metoden detekterer om museknappen blir presset ned, eller skjermen blir trykt på der man har touchscreen.         |
| +MouseReleased | Denne metoden detekterer om museknappen blir sluppet opp, eller der en finger blir sluppet opp der man har touchscreen. |

### 8.3.3 Kommunikasjon - ArmingSwitch

|        |  |
|--------|--|
| Fra    | ArmingSwitch   |
| Til    | GuiFacade  |
| Info   | Ved å manipulere Armeringsbryteren vil metoden toggleArmed bli kalt og denne kommuniserer med CORE slik at systemet kan endre våpenet til armert eller sikret. |
| Format | Metodekall   |

## 8.4 Klasse - FiringMode

Denne Klassen arver fra wxPanel. Den har som oppgave å vise brukergrensesnittet for "firingMode" og på den måte la brukeren av systemet velge mellom de forskjellige skuddhastighetene og moduser systemet har å by på.

| FiringMode  |
|---|
| -image : wxBitmap   |
| +FiringMode(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +nextMode() : void  |
| +paintEvent(evt : wxPaintEvent &) : void  |
| +paintNow() : void  |
| +render(dc : wxDC &) : void   |
| +mouseDown(event : wxMouseEvent &) : void   |

### 8.4.1 Attributter

| Navn            | Beskrivelse  |
|-----------------|--|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap. |

### 8.4.2 Metoder

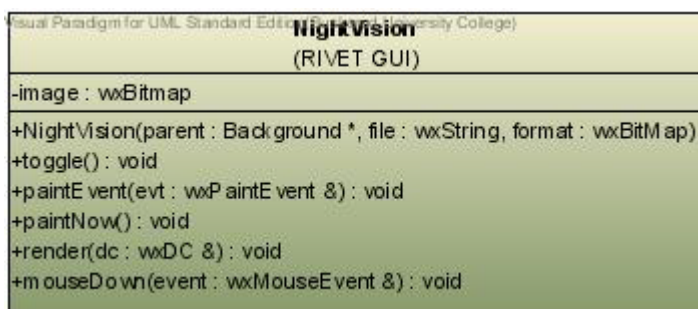
| Navn        | Beskrivelse  |
|-------------|--|
| +FiringMode | Konstruktøren tar imot peker fra Gui_Facade, background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                   |
| +NextMode   | Denne metoden kalles når brukeren trykker på musa på rett sted, det vil si firingmode knappen.   |
| +PaintEvent | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.  |
| +PaintNow   | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.                |
| +Render     | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge. |
| +MouseDown  | Denne metoden detekterer om museknappen blir presset ned, eller skjermen blir trykt på der man har touchscreen.  |

### 8.4.3 Kommunikasjon - FiringMode

|        |   |
|--------|---|
| Fra    | FiringMode  |
| Til    | GuiFacade   |
| Info   | Ved å trykke på knappen "Rate" vil nextMode metode kallet signalisere til CORE den skal bytte til neste skuddmodus. CORE tar seg av resten. |
| Format | Metodekall  |

## 8.5 Klasse - NightVision

Denne klassen arver fra wxPanel. Den har som oppgave å vise det brukeren ser som et menyvalg for å skru av og på nattkamera. Ved å trykke på knappen skifter systemet mellom disse to modusene.



### 8.5.1 Attributter

| Navn            | Beskrivelse  |
|-----------------|--|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap. |

### 8.5.2 Metoder

| Navn         | Beskrivelse  |
|--------------|--|
| +Nightvision | Konstruktøren tar imot peker fra Gui_Facade, background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                   |
| +Toggle      | Denne metoden lar bruker bytte mellom nattmodus og vanlig kameramodus.   |
| +PaintEvent  | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.  |
| +PaintNow    | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.                |
| +Render      | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge. |
| +MouseDown   | Denne metoden detekterer om museknappen blir presset ned, eller skjermen blir trykt på der man har touchscreen.  |

### 8.5.3 Kommunikasjon - NightVision

|        |  |
|--------|--|
| Fra    | NightVision  |
| Til    | GuiFacade  |
| Info   | Ved å trykke på knappen "Day/Night" vil toggle metodekallet signalisere til CORE at den skal bytte til neste skuddmodus. CORE tar seg av resten. |
| Format | Metodekall   |

## 8.6 Klasse - Zoom

Denne klassen arver fra wxPanel. Den har som oppgave å vise brukeren zoom-grensesnittet og gjøre at brukeren kan zoome inn og ut.

Dette er representert med to knapper på brukergrensesnittet.

| Zoom  |
|---|
| image : wxBitmap  |
| +Zoom(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +zoomIn() : void  |
| +zoomOut() : void   |
| +paintEvent(evt : wxPaintEvent &) : void  |
| +mouseDown(event : wxMouseEvent &) : void   |
| +paintNow() : void  |
| +render(dc : wxDC &) : void   |

### 8.6.1 Attributter

| Navn            | Beskrivelse  |
|-----------------|--|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap. |

### 8.6.2 Metoder

| Navn        | Beskrivelse   |
|-------------|---|
| +Zoom       | Konstruktøren tar imot peker fra Gui_Facade, background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                  |
| +ZoomIn     | Lar bruker zoome inn stegvis.   |
| +ZoomOut    | Lar bruker zoome ut stegvis.  |
| +MouseDown  | Denne metoden detekterer om museknappen blir trykket ned, eller skjermen blir trykt på der man har touchscreen.   |
| +PaintEvent | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.   |
| +PaintNow   | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.               |
| +Render     | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge |

### 8.6.3 Kommunikasjon - Zoom

|      |  |
|------|--|
| Fra  | Zoom   |
| Til  | GuiFacade  |
| Info | I tillegg til zoom på joystick så kan man også zoome ved hjelp av knapper på |

|        |   |
|--------|---|
|        | brukergrensesnittet. Ved å trykke på disse kjøres toggle metoden og CORE får signal om å skifte mellom natt og dag modus. |
| Format | Metodekall  |

## 8.7 Klasse - Sensitivity

Denne klassen arver fra klassen wxPanel. Den har som oppgave å vise sensitivitetskontrollen på bruker grensesnittet. Brukeren kan her stille inn om sensitiviteten skal justeres opp eller ned.

| Sensitivity  |
|--|
| -image : wxBitmap  |
| +Sensitivity(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +sensUp() : void   |
| +sensDown() : void   |
| +paintEvent(evt : wxPaintEvent &) : void   |
| +paintNow() : void   |
| +render(dc : wxDC &) : void  |

### 8.7.1 Attributter

| Navn            | Beskrivelse  |
|-----------------|--|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap. |

### 8.7.2 Metoder

| Navn         | Beskrivelse  |
|--------------|--|
| +Sensitivity | Konstruktøren tar imot peker fra Gui_Facade, background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                   |
| +SensUp      | Lar bruker få øke sensitiviteten ved å kjøre denne metoden ved interaksjon med brukergrensesnittet.  |
| +SensDown    | Lar bruker få senke sensitiviteten ved å kjøre denne metoden ved interaksjon med brukergrensesnittet.  |
| +PaintEvent  | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.  |
| +PaintNow    | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.                |
| +Render      | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge. |

### 8.7.3 Kommunikasjon – Sensitivity

|        |   |
|--------|---|
| Fra    | Sensitivity   |
| Til    | GuiFacade   |
| Info   | Det er mulig å justere sensitiviteten på joysticken i systemet. Dette gjøres ved å trykke på knappene relatert til disse. Da kjøres sensUp eller sensDown avhengig av hvilken knapp man trykker på. Disse metodene gir CORE signal om å sette sensitiviteten opp eller ned. |
| Format | Metodekall  |



## 8.8 Klasse - OnOFF

Denne klassens oppgave er å vise av og på knappen på GUI'et. Den skal også kalle Exit metoden når spiller trykker på av-knappen.

| OnOff  |
|--|
| -image : wxBitmap  |
| +OnOff(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +paintEvent(evt : wxPaintEvent &) : void   |
| +paintNow() : void   |
| +render(dc : wxDC &) : void  |
| +mouseDown(event : wxMouseEvent &) : void  |

### 8.8.1 Attributter

| Navn            | Beskrivelse  |
|-----------------|--|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap. |

### 8.8.2 Metoder

| Navn        | Beskrivelse   |
|-------------|---|
| +OnOff      | Konstruktøren tar imot peker fra Gui_Facade, background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                  |
| +MouseDown  | Denne metoden detekterer at museknappen trykkes ned.  |
| +PaintEvent | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.   |
| +PaintNow   | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.               |
| +Render     | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge |

## 8.9 Klasse – GrenadeArmer

Denne klassen arver fra wxPanel. Den har som oppgave og vise armeringsbryteren for granatkasteren. Klassen inneholder også logikk for grafikk styring for dette elementet.

Armeringsbryteren har

3 tilsander som forteller resten av systemet om det er øvre eller nedre granatrekke som er valgt. Ved midtre valg er ikke granatavfiring mulig.

| GrenadeArmer  |
|---|
| -image : wxBitmap   |
| -path : wxString  |
| -counter : int  |
| -action : int   |
| -armed : boolean  |
| +GrenadeArmer(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +paintEvent(evt : wxPaintEvent &) : void  |
| +paintNow() : void  |
| +render(dc : wxDC &) : void   |
| +mouseDown(event : wxMouseEvent &) : void   |
| +mouseReleased(event : wxMouseEvent &) : void   |

### 8.9.1 Attributter

| Navn            | Beskrivelse   |
|-----------------|---|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap.                  |
| -boolean armed  | Holder tilstanden på våpenet, om det er armet eller ei.             |
| -int counter    | Holder orden på hvilken tilstand vippebryteren er i. (3 tilstander) |
| -int action     | Denne variabelen fanger opp hvor på skjermen musen er i Y-retning.  |
| -wxString path  | Inneholder en string.   |

### 8.9.2 Metoder

| Navn          | Beskrivelse   |
|---------------|---|
| +GrenadeArmer | Konstruktøren tar imot peker fra background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                              |
| +MouseRelease | Denne metoden detekterer at museknappen slippes opp.  |
| +MouseDown    | Denne metoden detekterer at museknappen trykkes ned.  |
| +PaintEvent   | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.   |
| +PaintNow     | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.               |
| +Render       | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge |

### 8.9.3 Kommunikasjon - GrenadeArmer

|        |   |
|--------|---|
| Fra    | GrenadeArmer                                    |
| Til    | GuiFacade                                       |
| Info   | Det er mulig å armere granatkasterne på våpenet |
| Format | Metodekall                                      |

## 8.10 Klasse - GrenadeFlipper

Denne klassen arver fra wxPanel. Denne klassen har som oppgave å vise vippebryteren som brukes i avfiring av granater. Bryteren har en sikkerhetsklaff som ligger over selve avfiringbryteren. Klassen har logikk for å håndtere grafikk på riktig måte.

| GrenadeFlipper   |
|--|
| -image : wxBitmap<br>-path : wxString<br>-counter : int<br>-action : int<br>-fire : bool   |
| +GrenadeFlipper(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void<br>+mouseDown(event : wxMouseEvent &) : void<br>+mouseReleased(event : wxMouseEvent &) : void |

### 8.10.1 Attributter

| Navn            | Beskrivelse   |
|-----------------|---|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap.                  |
| -boolean armed  | Holder tilstanden på våpenet, om det er armet eller ei.             |
| -int counter    | Holder orden på hvilken tilstand vippebryteren er i. (3 tilstander) |
| -int action     | Denne variabelen fanger opp hvor på skjermen musen er i Y-retning.  |
| -wxString path  | Inneholder en string.   |

### 8.10.2 Metoder

| Navn            | Beskrivelse   |
|-----------------|---|
| +GrenadeFlipper | Konstruktøren tar imot peker fra background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                              |
| +MouseRelease   | Denne metoden detekterer at museknappen slippes opp.  |
| +MouseDown      | Denne metoden detekterer at museknappen trykkes ned.  |
| +PaintEvent     | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.   |
| +PaintNow       | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.               |
| +Render         | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge |

### 8.10.3 Kommunikasjon - GrenadeFlipper

|        |                                 |
|--------|---------------------------------|
| Fra    | GrenadeFlipper                  |
| Til    | GuiFacade                       |
| Info   | Det er mulig å avfyre granater. |
| Format | Metodekall                      |

## 8.11 Klasse - BattleOverride

Denne klassen arver fra wxPanel. Klassen har som oppgave å vise vippebryteren som har 3 tilstander. Klassen inneholder logikk for å håndtere grafikk på rett måte. Det er ikke planlagt noe funksjonalitet på dette valget.

| BattleOverride  |
|---|
| -image : wxBitmap   |
| -path : wxString  |
| -counter : int  |
| -action : int   |
| -armed : boolean  |
| +BattleOverride(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +paintEvent(evt : wxPaintEvent &) : void  |
| +paintNow() : void  |
| +render(dc : wxDC &) : void   |
| +mouseDown(event : wxMouseEvent &) : void   |
| +mouseReleased(event : wxMouseEvent &) : void   |

### 8.11.1 Attributter

| Navn            | Beskrivelse   |
|-----------------|---|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap.                  |
| -boolean armed  | Holder tilstanden på våpenet, om det er armet eller ei.             |
| -int counter    | Holder orden på hvilken tilstand vippebryteren er i. (3 tilstander) |
| -int action     | Denne variabelen fanger opp hvor på skjermen musen er i Y-retning.  |
| -wxString path  | Inneholder en string.   |

### 8.11.2 Metoder

| Navn            | Beskrivelse   |
|-----------------|---|
| +BattleOverride | Konstruktøren tar imot peker fra background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                              |
| +MouseRelease   | Denne metoden detekterer at museknappen slippes opp.  |
| +MouseDown      | Denne metoden detekterer at museknappen trykkes ned.  |
| +PaintEvent     | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.   |
| +PaintNow       | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.               |
| +Render         | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge |

### 8.11.3 Kommunikasjon - BattleOverride

|        |                               |
|--------|-------------------------------|
| Fra    | GrenadeBattleOverride         |
| Til    | GuiFacade                     |
| Info   | Ingen funksjonalitet planlagt |
| Format | Metodekall                    |

## 8.12 Klasse - BattleOverrideFlipper

Denne klassen arver fra wxPanel. Klassen har som oppgave å vise

sikkerhetsbryteren som har 2 tilstander.

Klassen inneholder logikk for å håndtere

grafikk på rett måte. Det er ikke planlagt noe funksjonalitet på dette valget.

| BattleOverrideFlipper  |
|--|
| -image : wxBitmap  |
| -path : wxString   |
| -counter : int   |
| -action : int  |
| -armed : boolean   |
| +BattleOverrideFlipper(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +paintEvent(evt : wxPaintEvent &) : void   |
| +paintNow() : void   |
| +render(dc : wxDC &) : void  |
| +mouseDown(event : wxMouseEvent &) : void  |
| +mouseReleased(event : wxMouseEvent &) : void  |

### 8.12.1 Attributter

| Navn            | Beskrivelse   |
|-----------------|---|
| -wxBitmap image | Holder referansen til et objekt av typen wxBitmap.                  |
| -boolean armed  | Holder tilstanden på våpenet, om det er armet eller ei.             |
| -int counter    | Holder orden på hvilken tilstand vippebryteren er i. (3 tilstander) |
| -int action     | Denne variabelen fanger opp hvor på skjermen musen er i Y-retning.  |
| -wxString path  | Inneholder en string.   |

### 8.12.2 Metoder

| Navn                   | Beskrivelse   |
|------------------------|---|
| +BattleOverrideFlipper | Konstruktøren tar imot peker fra background, en wxString og en wxBitmap og lagrer denne. Setter også startverdier på klassens variabler.                              |
| +MouseRelease          | Denne metoden detekterer at museknappen slippes opp.  |
| +MouseDown             | Denne metoden detekterer at museknappen trykkes ned.  |
| +PaintEvent            | Denne metoden kalles av wxWidgets når panelet trenger å oppdateres.   |
| +PaintNow              | Denne metoden kan også brukes til å oppdatere et panel, når som helst. Men ved å bruke denne ofte kan man risikere at vinduer ikke oppfører seg ideelt.               |
| +Render                | Denne metoden står for selve rendringen. Den ligger i en egen metode fordi det er mulig å bruke 2 forskjellige Draw funksjoner, og denne kan samarbeide med dem begge |



### 8.12.3 Kommunikasjon – BattleOverrideFlipper

|        |                               |
|--------|-------------------------------|
| Fra    | BattleOverrideFlipper         |
| Til    | GuiFacade                     |
| Info   | Ingen funksjonalitet planlagt |
| Format | Metodekall                    |

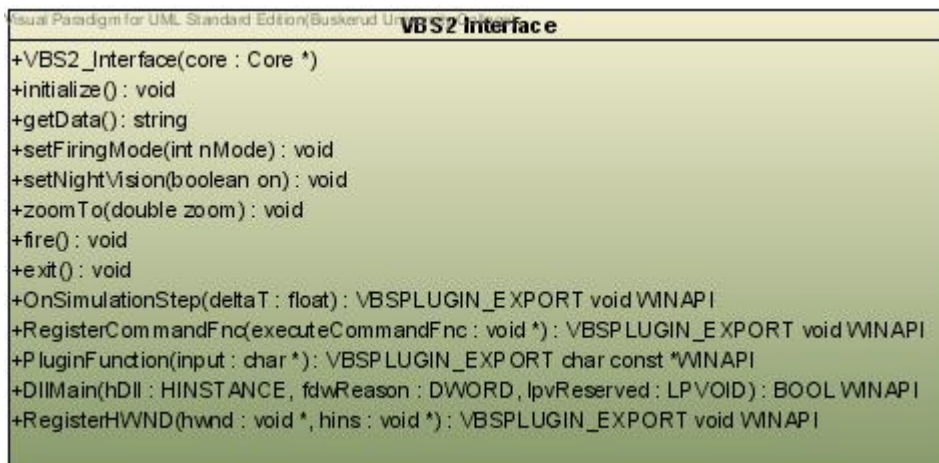
### 8.13 Klasser som ikke er planlagt utviklet

Disse klassene er ikke tatt med siden dem ikke er planlagt utviklet. Felles for alle klassene er at dem kun brukes til å vise noen GUI-komponenter slik at utseendet blir korrekt. Vi valgte å legge dem som egne klasser slik at dem enkelt skal kunne implementeres med funksjonalitet på et senere tidspunkt, om dette er ønskelig for oppdragsgiver.

- Klasse – ReticleSelect
- Klasse – Range
- Klasse – Focus
- Klasse – Polarity
- Klasse – Lamptest
- Klasse – Gain
- Klasse – Level
- Klasse – Brightness
- Klasse – Calibration

## 9 Komponent – RIVET TOLK

Rivet Swapper består av en klasse som heter VBS2 Interface. Denne komponenten har som jobb å kommunisere med VBS2 sin Application Scripting Interface(ASI).



### 9.1 Klasse – Vbs2 Interface

Denne klassen har som oppgave å knytte VBS2 og RIVET systemet sammen. Denne modulen skal tolke det som mottas fra VBS2. Siden vår versjon av VBS2 kun sender tekststrenger ut må disse håndteres og tolkes. Ut ifra dette vil forskjellige kall bli gjort. Kommunikasjonen fra Rivet Application Package til VBS2 skjer også via denne klassen, men med kommandoer.

#### 9.1.1 Metoder

| Navn            | Beskrivelse   |
|-----------------|---|
| +Initialize     | Denne metoden initialiserer kommunikasjonen mellom VBS2 og vårt system.   |
| +Exit           | Denne metoden skjuler brukergrensesnittet.  |
| +ActionData     | Denne metoden tar imot en string fra metodekallet pluginFunction i VBS2. Denne stringen tolkes og ut ifra hva stringen inneholder så vil forskjellige kall til Core skje. |
| +SetFiringMode  | Denne metoden mottar et heltall og dette tallet vil fortelle hvilken firingmode systemet ber VBS2 bytte til.  |
| +SetNightVision | Denne metoden mottar en boolean verdi og ut ifra det ber VBS2 aktivere natt modus eller deaktivere nattmodus.   |
| +ZoomTo         | Denne metoden mottar en tall verdi og ber VBS2 zoome inn eller ut avhengig av tallverdi og posisjon.  |
| +MoveSight      | Denne metoden mottar koordinater og ber deretter VBS2 beveger siktet i den retting.   |
| +Fire           | Denne metoden forteller VBS2 at brukeren av RIVET systemet ønsker å skyte.  |

|                     |  |
|---------------------|--|
| +OnSimulationStep   | Denne metoden kjøres hver eneste frame. Ved hjelp av denne metoden kan vi for eksempel sette været til å skifte fra fint til regn. Alt ettersom vi ønsker det. |
| +RegisterCommandFnc | Denne metoden definerer hvilken type ExecureCommandType og hvor tilbakekall funksjonen skal lagres.  |
| +PluginFunction     | Denne metoden kalles hver gang VBS2 kjører en scriptkommando. Slike scriptkommandoer kan kjøres når som helt fra en trigger.                                   |
| +DllMain            | Denne metoden er det som kalles om Dll'en skal kobles til eller fra.   |
| +RegisterHWND       | Denne metoden sender VBS2 sin "window handle" og instans id, som en plugin trenger om den skal lage en grafisk brukergrensesnitt.                              |

### 9.1.2 Kommunikasjon - VBS2Interface

|        |   |
|--------|---|
| Fra    | VBS2Interface   |
| Til    | CORE  |
| Info   | VBS2Interface er den klassen som står for oversetting mellom VBS2 og RIVET. |
| Format | Metodekall  |

## 10 Referanser

---

[1] Erich Gamma, Richard Helm, Ralph Johnson og John M. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. ISBN-13: 978-0201633610

# KDA RIVET



## Dokumentstandard

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....                            | 3 |
| 1.1   | Hensikt med dokumentet.....                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3   | Dokumenthistorie.....                        | 3 |
| 1.4   | Ansvarlig .....                              | 3 |
| 2     | Overskrift 1.....                            | 4 |
| 2.1   | Overskrift 2 .....                           | 4 |
| 2.1.1 | Overskrift 3.....                            | 4 |
| 3     | Tabeller .....                               | 5 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en innføring i hvordan dokumentene våre skal se ut. Dette vil si layout, skriftstørrelse og type, farger, bruk av overskrifter, forside, tabeller, headere og footere osv.

## 1.2 Definisjoner, akronymer og forkortelser

Her skal alle forkortelser og definisjoner inn

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                                |
|---------------|----------|--------------------------------|---|
| 0.1           | 28.02.10 | Førsteutkast                   | Ole-Martin Grøtterud                    |
| 0.98          | 19.05.10 | Gjennomgang av egne dokumenter | Lone Knutsen                            |
| 0.99          | 27.05.10 | Korrekturlest                  | Ole-Martin Grøtterud og Thomas Transeth |
| 1.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                         |

## 1.4 Ansvarlig

Lone Knutsen



## 2 Overskrift 1

---

Overskrift 1 skal være slik som vist over. Alle hovedavsnitt skal på ny side.

### 2.1 Overskrift 2

Som over

#### 2.1.1 Overskrift 3

Som over

##### 2.1.1.1 Overskrift 4

Som over

#### Annet:

- Brødtekst, footere og annet skal være i Calibri 12.
- Brødtekst skal starte innat margen. Kun avsnitt og punktlister skal ha innrykk.

### 3 Tabeller

---

Tabeller skal se ut som under:

| Utkast nummer | Dato     | Aktivitet                      | Redaktør             |
|---------------|----------|--------------------------------|----------------------|
| 0.1           | 28.02.10 | Førsteutkast                   | Ole-Martin Grøtterud |
| 0.98          | 19.05.10 | Gjennomgang av egne dokumenter | Lone Knutsen         |

# KDA RIVET



## Forbedringsdokument

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Kjente forbedringsmuligheter.....            | 4 |
| 2.1 | Pekerfeil når VBS2 avsluttes.....            | 4 |
| 2.2 | Granat avfyringsbryter .....                 | 4 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en oversikt over kjente forbedringer som kan gjøres i systemet.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS2:** Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                                 |
|---------------|----------|---------------------------|--|
| 0.1           | 27.05.10 | Førsteutkast              | Ole-Martin Grøtterud                     |
| 0.99          | 27.05.10 | Korrekurlest              | Arne Kristian Åmellem<br>og Lone Knutsen |
| 1.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                          |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Kjente forbedringsmuligheter

---

### 2.1 Pekerfeil når VBS2 avsluttes

**Beskrivelse:** Når brukeren går ut av VBS2 etter å ha brukt RIVET-plugin, kommer det opp en "unable to write"- feilmelding til en minne adresse.

**Mulig årsak:** En dekonstruktør kan forsøke å slette en peker som ikke er opprettet eller allerede slettet. Foretok en rask gjennomgang av dekonstruktørene, men fant ikke feilen.

### 2.2 Granat avfyrringsbryter

**Beskrivelse:** Granat avfyrringsbryteren går ikke tilbake igjen slik som på den originale treneren. Den må trekkes tilbake manuelt.

**Mulig årsak:** Vi kan ikke bruke "sleep"-funksjonen for å få bryteren til å sette seg selv tilbake. Vi har da valgt at bryteren må trekkes manuelt tilbake, i stedet for at den ikke bytter bilde.

### 2.3 Siktet beveges til 0,0 eller sist posisjon når man går inn i RWS

**Beskrivelse:** Når man går inn i RWS og beveger joystick første gang vil den automatisk peke mot nord +- joystickbevegelsen. Hvis man går ut av en RWS og inn i en annen vil siktet fortsette fra der man gikk ut fra den andre idet man beveger joystick +- joystickbevegelsen.

**Mulig årsak:** Skriptspråket mangler muligheter for å hente ut retningen på RWS stasjonen slik at vi kunne satt den idet vi gikk inn i bilen.

# KDA RIVET



## Fornøydelsesdokument

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                          | 3 |
| 1.1 | Hensikt med dokumentet.....                | 3 |
| 1.2 | Definisjoner og forkortelser .....         | 3 |
| 1.3 | Dokumenthistorie.....                      | 3 |
| 1.4 | Ansvarlig .....                            | 3 |
| 2   | Oversikt over foreslåtte aktiviteter ..... | 4 |
| 3   | Planlagte og gjennomførte aktiviteter..... | 5 |





# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Dokumentet gir en oversikt over sosiale teambuildings-aktiviteter og planlegger dem, samt å rapportere gjennomførte hendelser.

## 1.2 Definisjoner og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                       | Redaktør                           |
|---------------|----------|---------------------------------|------------------------------------|
| 1.1           | 23.04.10 | Oppdatert iht dokumentastandard | Lone Knutsen                       |
| 1.2           | 24.04.10 | Lagt til aktivitet A3           | Ole-Martin Grøtterud               |
| 1.98          | 19.05.10 | Gjennomgang av egne dokumenter  | Ole-Martin Grøtterud               |
| 1.99          | 23.05.10 | Korrekturlest                   | Lone Knutsen og<br>Thomas Transeth |
| 2.0           | 28.05.10 | Klargjort for innlevering       | Thomas Transeth                    |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Oversikt over foreslåtte aktiviteter

Under følger en oversikt over aktiviteter vi har foreslått, med de fordeller, ulemper og risikoer som følger med.

|                     |                         |
|---------------------|-------------------------|
| <b>Aktivetsnavn</b> | <b>Bowling</b>          |
| Fordeler            | Alle kan delta          |
| Ulemper             | Lone er en dårlig taper |
| Risikoer            | R6, R8, R10             |

|                     |                                |
|---------------------|--------------------------------|
| <b>Aktivetsnavn</b> | <b>Spekeskinke og drikke</b>   |
| Fordeler            | Bedre bekjentskap og mye morro |
| Ulemper             | Dagen derpå                    |
| Risikoer            | R6                             |

|                     |                               |
|---------------------|-------------------------------|
| <b>Aktivetsnavn</b> | <b>Xbox kveld</b>             |
| Fordeler            | Billig og enkel morro         |
| Ulemper             | Lone og Ole er dårlige tapere |
| Risikoer            |                               |

|                     |  |
|---------------------|--|
| <b>Aktivetsnavn</b> | <b>Moralsk støtte når Lone skal flytte</b> |
| Fordeler            | Gøy å se andre arbeide                     |
| Ulemper             | Langt unna                                 |
| Risikoer            |  |

|                     |                              |
|---------------------|------------------------------|
| <b>Aktivetsnavn</b> | <b>Filmkveld</b>             |
| Fordeler            | Billig morro                 |
| Ulemper             | Ikke sosialt nok på egenhånd |
| Risikoer            |                              |

|                     |  |
|---------------------|--|
| <b>Aktivetsnavn</b> | <b>Lunch</b>                               |
| Fordeler            | Alle samles og det kan gjennomføres fast   |
| Ulemper             | Dyrt for de som glemmer å ta med seg niste |
| Risikoer            | R6   |

|                     |                               |
|---------------------|-------------------------------|
| <b>Aktivetsnavn</b> | <b>Gå ut på byen</b>          |
| Fordeler            | Fancy drinker                 |
| Ulemper             | Dyrt og AK møter gamle kjente |
| Risikoer            | R6, R7, R8                    |

### 3 Planlagte og gjennomførte aktiviteter

|                     |   |
|---------------------|---|
| <b>AktivitetsNr</b> | AX  |
| <b>Aktivitet</b>    | Lunch   |
| <b>Dato</b>         | 04.03.2010  |
| <b>Tidspunkt</b>    | 12:00-12:30 tirsdag, onsdag og torsdag  |
| <b>Deltakere</b>    | Alle som er tilstede.   |
| <b>Rapport</b>      | Gruppen vedtok at på tirsdager, onsdager og torsdager er det feller lunch på skolen. Aktiviteten får nummer X som følge at det skjer 3 ganger i uken og ikke er ment for å bli referert til som en enkelt hendelse. |
| <b>Konklusjon</b>   | Lunchen fungerer som en fin avkobling fra arbeidet og det er safari i kantinen med statsvitere, økonomer og optikere rundtom.   |

|                     |  |
|---------------------|--|
| <b>AktivitetsNr</b> | A1   |
| <b>Aktivitet</b>    | Spillkveld med spekeskinke og drikke   |
| <b>Dato</b>         | 22.02.2010   |
| <b>Tidspunkt</b>    | 18:00-24:00  |
| <b>Deltakere</b>    | Arne Kristian, Thomas, Ole-Martin og Zoran   |
| <b>Rapport</b>      | Lone måtte desverre melde frafall, men resten av gruppen klarte å kose seg, til tross for stort savn. Kvelden ble begynt med Xbox 360 og Castle Crashers, så videre på Tekken 6 og Dead or Alive 4. Rundt klokken 22:30 gikk vi over på Munchkin og ble sittende med det resten av kvelden med stor suksess. I alt ble det konsumert 1.5kg spekeskinke, 1 pose potetgull, en bolle smash, ett dusin tyrkisk pepper, noen håndfuller med nøtter og rosiner, et ukjent antall bokser drikke, en stor flaske vin, og diverse fra andre flasker. |
| <b>Konklusjon</b>   | Formen var fin og vi koste oss. Dagen derpå var alle trøtte som ett resultat av vår suksess.   |

|                     |   |
|---------------------|---|
| <b>AktivitetsNr</b> | A2  |
| <b>Aktivitet</b>    | Bowling   |
| <b>Dato</b>         | 04.03.2010  |
| <b>Tidspunkt</b>    | 18:00   |
| <b>Deltakere</b>    | Prosjektgruppene RIVET og SCRAT   |
| <b>Rapport</b>      | Alle gruppelemmene fra begge grupper møtte opp som planlagt klokken 18 på bowlinga. Zoran, som aldri hadde prøvd å bowle før, tok spillet raskt, og opparbeidet seg raskt en innovativ sideswing teknikk. Etter to runder sto begge lag med én seier hver, og ble enige om en omkamp på diskobanen i nær fremtid. I løpet av de to rundene hadde gruppene opparbeidet seg en statistikk på 13 splits, 24 strikes, 39 spares og 114 kast i renna til sammen. |
| <b>Konklusjon</b>   | Begge gruppene hadde det bra og lagmoralen ble styrket. Alle burde øve på Zorans sideswing til nestegang!   |

|                     |   |
|---------------------|---|
| <b>AktivitetsNr</b> | A3  |
| <b>Aktivitet</b>    | Spillkveld med spekeskinke og drikke  |
| <b>Dato</b>         | 21.04.2010  |
| <b>Tidspunkt</b>    | 18:00-02:00   |
| <b>Deltakere</b>    | Prosjektgruppene RIVET  |
| <b>Rapport</b>      | Denne gangen hadde vi fullt oppmøte med alle medlemmene, dobbelt så mye spekeskinke og enda bedre humør med oss. Vi gjorde som sist og spilte castle crashers på xbox'en, men passet på å ta pauser for å konsumere spekeskinke og drikke. Det ble i alt spist 2,5kg spekeskinke, 1 bolle potetgull, ½ pakke saltstenger og laget ett 3meter tårn av bokser+ en kleshenger. |
| <b>Konklusjon</b>   | Vi føler oss sikre på at vi skal klare å runde castle crashers nestegang, og planlegger allerede neste spillekveld så snart lommeboken kan takle det.   |

# KDA RIVET



## Forstudierapport

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....  | 3  |
| 1.1   | Hensikt med dokumentet.....                                    | 3  |
| 1.2   | Definisjoner, akronymer og forkortelser.....                   | 3  |
| 1.3   | Dokumenthistorie.....  | 3  |
| 1.4   | Ansvarlig.....   | 3  |
| 2     | Presentasjoner.....  | 4  |
| 2.1   | Presentasjon av prosjektgruppa (styrker og svakheter).....     | 4  |
| 2.1.1 | Arne Kristian Åmelle.....                                      | 4  |
| 2.1.2 | Lone Knutsen.....  | 4  |
| 2.1.3 | Ole-Martin Grøtterud.....                                      | 4  |
| 2.1.4 | Thomas Transeth.....   | 4  |
| 2.1.5 | Zoran Vukobratovic.....  | 5  |
| 2.2   | Presentasjon av KDA.....                                       | 5  |
| 3     | Beskrivelse av prosjektomgivelser og relaterte prosjekter..... | 6  |
| 4     | En funksjonell beskrivelse av målet med prosjektet.....        | 7  |
| 4.1   | Prioritet 1.....   | 7  |
| 4.2   | Prioritet 2.....   | 8  |
| 4.3   | Prioritet 3.....   | 8  |
| 5     | Kostnader.....   | 9  |
| 6     | Analyse av problemer og begrensninger.....                     | 10 |
| 7     | Utviklingsmetode.....  | 11 |
| 8     | Aktiviteter.....   | 12 |
| 8.1   | Aktivitetsgrener:.....   | 12 |
| 8.2   | Aktiviteter:.....  | 12 |
| 9     | Tidsangivelser og milepæler.....                               | 13 |
| 10    | Ressurser.....   | 14 |
| 10.1  | Menneskelige ressurser:.....                                   | 14 |
| 10.2  | Software-ressurser.....  | 14 |
| 10.3  | Hardware-ressurser.....  | 14 |
| 11    | Arbeid frem til prosjektplan foreligger.....                   | 15 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å få en grov oversikt av prosjektet, og å presentere alle parter. Det skal også legge grunnlag for videre arbeid og utførelse av prosjektet.

## 1.2 Definisjoner, akronymer og forkortelser

**KDA:** Kongsberg Defence & Aerospace

**HiBu:** Høgskolen i Buskerud

**RWS:** Remote Weapon System

**VBS:** Virtual Battle Space

**MS:** Microsoft

**RUP:** Rational Unified Process

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet        | Redaktør           |
|---------------|----------|------------------|--------------------|
| 2.99          | 31.05.10 | Korrekturlesning | Zoran Vukobratovic |
| 3.0           | 31.05.10 | Klargjort        | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Presentasjoner

---

### 2.1 Presentasjon av prosjektgruppa (styrker og svakheter)

Her er en presentasjon av gruppemedlemmene, samt deres styrker og svakheter.

#### 2.1.1 Arne Kristian Åmellem

- 28 åring fra Nes kommune i Hallingdal. Studerer bachelor i Simulering og Spill Utvikling.
- *Styrker:* Flink med mennesker og åpen for nytenking. Engasjert og ivrig når jeg arbeider med noe som interesserer meg. Ikke redd for å skjære igjennom og ta en avgjørelse om nødvendig. Erfaring som leder fra arbeidslivet.
- *Svakheter:* Kan fort bli avsporet, og litt for snill til tider.

#### 2.1.2 Lone Knutsen

- 21-åring fra Hvitvingfoss. 3.års dataingeniørstudent på HiBu med spesialisering innen Embedded systems.
- *Styrker:* Flink til å få folk til å jobbe og til å overholde frister i prosjektarbeid. Er ikke redd for å skjære igjennom og ta avgjørelser dersom det er nødvendig. Opptatt av detaljer, og er tildels ganske pirkete.
- *Svakheter:* Har en tendens til å drøye jobbing til siste liten dersom det er noe jeg synes er uinteressant. Kan til tider bli litt sjefete.

#### 2.1.3 Ole-Martin Grøtterud

- 21-åring fra Nedre Eiker, går 3.året dataingeniør med spesialisering innen simulering og spill.
- *Styrker:* Flink til å finne egne løsninger, overholder frister, selvmotivert og humørfyllt.
- *Svakheter:* Lett distraherert, kan spore av i perioder.

#### 2.1.4 Thomas Transeth

- Kort beskrivelse: 25 år og fra Bergen som er på sitt siste år i bachelorgraden embedded systems.
- *Styrker:* Har bred kunnskap innenfor flere programmeringsspråk og synes det er underholdende å skrive algoritmer. Synes også at kodestandarder er noe en skal følge.
- *Svakheter:* Kan bruke litt tid på å finne den beste løsningen og finner i mellomtiden på mye rart.



### 2.1.5 Zoran Vukobratovic

- Kort beskrivelse: 21-åring fra Lier i Buskerud som går siste året på Bachelor studiene på HiBu.
- *Styrker:* Kan til tider sette meg veldig inn i oppgaver og være virkelig produktiv. Finner matte og fysikk veldig underholdende, og er veldig nysgjerrig.
- *Svakheter:* Kan til tider sette meg inn i feil ting og sløse masse tid. Sporer av mer enn jeg ønsker. Liker å utsette ting.

### 2.2 Presentasjon av KDA

Kongsberg Defence & Aerospace (KDA) er Norges største leverandør av forsvar og romfartsrelaterte systemer. Porteføljen består av produkter og systemer for kommando og kontroll, våpenstyring, overvåkning, kommunikasjonsløsninger og missiler. Kongsberg Defence & Aerospace lager også avanserte komposittprodukter for fly og helikoptermarkedet.

### 3 Beskrivelse av prosjektomgivelser og relaterte prosjekter

---

Oppgaven er gitt av KDA fordi de ønsker en studie av mulighetene til å integrere Protector simulatoren sin i VBS 2 simuleringsverdenen.

Systemet som allerede finnes er en trener for RWS systemet Protector. Dette er en simulator som tilbyr de samme funksjonalitetene som det virkelige våpenet har. KDA leverer mobile klasseroms løsninger der en kan sette opp alle posisjonene som finnes på et kjøretøy samt en instruktørplass. Hensikten med denne simulatoren er å trene elevene i krigssituasjoner i et uskarpt miljø.

De ønsker å oppdatere systemet sitt fordi den nåværende motoren de bruker begynner å bli utdatert, og ved en kundeforespørsel om å bruke VBS 2 må de ha kunnskap om det er mulig. VBS 2 er en simulatormotor som brukes av blant annet det Amerikanske forsvaret for kampsimuleringer. Vi skal se på mulighetene til å implementere RWS delen av Protector simulatoren, og kartlegge hva som er lett og vanskelig.

## 4 En funksjonell beskrivelse av målet med prosjektet

Formålet med dette prosjektet er å utvikle KDA sin Remote Weapon System(RWS) trener i Virtual Battle Space 2(VBS2). KDA ønsker å kartlegge vanskelighetsgraden og hvor tidkrevende det vil være å utvikle RWS trener i denne teknologien. Slik at de har en formening om utviklingsprosessen i tilfelle en kunde skulle ønske denne teknologien spesifikt.

Treneren som gruppen utvikler skal være så virkelig simulering som mulig. I tillegg er det ønsket at gruppen skal legge sitt eget preg på treneren, samt gjøre den morsommere. Slik at brukerne av treneren skal ønske å spille mer.

KDA ønsket at vi tok utgangspunkt i treneren de leverer til Kuwait, CROWS.

Vi har fått et bilde av hvordan det grafiske brukergrensesnittet ser ut, og vi skal bruke det samme utseende. *Dette er det korrekte utseende for treneren:*



Figur 1: Trainer

Det er forskjellige prioriteringer av hva vi må få til utafra hva som er viktig for å få en funksjonell trener.

### 4.1 Prioritet 1.

- Første vi skal få på plass er utseende på GUI'et. Det vil si knappene på rett sted. På dette brukergrensesnittet er det et vindu som viser bildet fra kameraet, eller selve simuleringsverdenen. Dette bildet skal oppdatere seg korrekt, og ha et korssikte.

- Klare å skyte
- Lage omgivelser til å trene i.

## 4.2 Prioritet 2.

- Armeringsbryteren skal implementeres slik at man kan sikre våpenet og armere det.
- Sikte funksjoner. Vi må implementere funksjonaliteten til kamerat. Dette vil si nattkamera, kalibrering (blur effekt som blir fjernet),
- Menysystemet skal implementeres. Funksjonaliteten skal implementeres på et senere stadium.
- Lysjusteringer, gain og zoom

## 4.3 Prioritet 3.

- Implementasjon av noen menyfunksjoner
- Safety override funksjonalitet og Remote safe
- Implementasjon av No fire Zone's.

Dette er en røff prioriterings liste som dekker mesteparten av de målene KDA ønsker at vi skal kikke på. Om dette viser seg å være lettere enn forventet, er det andre utvidelser vi kan gjøre. KDA ønsker at vi skal lage en selvstendig trener, uten å tenke på klasserom og instruktørfunksjonalitet. Senere om vi har tid, kan vi se på disse funksjonene.

## 5 Kostnader

---

- KDA prosjekt-PC med joystick, levert av KDA. ingen reel kostnad
- VBS lisens.
- Fremføringer: Kaffe-85, te-30, brus-50, frukt-100 per presentasjon
- Kopieringskostnader estimeres til rundt 400,-
- Reisekostnader er ingen, da bedriften er lokal.

## 6 Analyse av problemer og begrensninger

---

*Er det mulig å gjennomføre prosjektet?*

- Vi har ikke en full lisens av VBS2, og vi har ingen informasjon om den lisensen vi får, vil være nok til å få gjort alt vi har bestemt skal gjøres.  
Det kan fort vise seg at vi trenger en annen lisens for å fullføre oppgaven, noe som må tas opp med oppdragsgiver, og så må vi foreta et valg om vi skal fortsette eller ikke.
- Gruppen har sett på en demo av VBS2, men vi har ikke fått vite hva slags programmeringsspråk VBS2 bruker.
- Vi har kun én maskin med VBS2 programvare på, noe som begrenser tiden hver enkelt person får, før noen andre gjerne vil bruke samme PC.  
Gruppen må planlegge en tidstabell og forholde seg til den for å få løst dette problemet.

## 7 Utviklingsmetode

---

Gruppen har studert flere utviklingsmetoder og kommet fram til at Rational Unified Process er den som passer best. Dette er en metode som beskriver hvordan en kan fordele oppgaver og ansvar i en utviklingsprosess.

Prosjektet har som krav å dokumenteres skikkelig. Når en skal få en oversikt over hvordan en skal programmere, er det viktig å beskrive dette på en god måte. Ved å bruke RUP sikrer vi at mange av dokumentasjonskravene er dekket og testet, da kravspesifikasjonene allerede er definert.

Prosjektet har en viktig forstudie som må gjøres før mange av aktivitetene kan defineres og dette passer godt inn under innledningsfasen som er definert i RUP.

Før denne oppgaven ble gitt hadde vi allerede en idé på hvilket ansvarsområde de forskjellige gruppe-medlemmene kunne tenke seg, og dette kan ledes inn til begrepet "workers" der hver aktivitet får sin ansvarsperson.

RUP er også utviklet spesielt med tanke på utvikling av software og vårt prosjekt har full hovedvekt på dette.

## 8 Aktiviteter

---

Dette er de forløpige aktivitetene og aktivitetsgrenene vi har. Det vil bli lagt til flere aktiviteter etter hvert, men da i ett annet dokument sammen med timelister.

### 8.1 Aktivitetsgrenener:

| Nummer | Type                              |
|--------|-----------------------------------|
| 1xx    | Administrasjon og prosjektstyring |
| 2xx    | Research                          |
| 3xx    | Krav/analyse                      |
| 4xx    | Design                            |
| 5xx    | Implementasjon                    |
| 6xx    | Test                              |

### 8.2 Aktiviteter:

- 100: gruppemøte
- 101: prosjektstyring
- 102: korrekturlesing
- 103: prosjektplan
- 104: presentasjoner
  
- 200: VBS2 funksjoner og bruk
- 201: Trainer-funksjoner
- 202: interface mellom joystick og trainer.
  
- 300: krav
- 301: use case
- 302: kravspesifikasjon
- 304: analyse
  
- 400: klasser
- 401: designdokument
- 402: designdiagrammer
  
- 500: kodestandard
- 501: prototype
- 502: implementasjon
  
- 600: teststrategier
- 601: testspesifikasjon
- 602: testrapport
- 603: test



## 9 Tidsangivelser og milepæler

---

I prosjektet så har vi tre hovedmilepæler. Nemlig presentasjonene.

- 1. Presentasjon -1-2 uke i januar
- 2. Presentasjon - 12. mars - 13. april
- 3. Presentasjon - Begynnelsen av juni

I tillegg så skal all dokumentasjon være levert innen 29. Mai 2010.

## 10 Ressurser

---

Vi har i hovedsak tre forskjellige ressurstyper. Menneskelige, software og hardware.

### 10.1 Menneskelige ressurser:

- 5 stk a ca 500 arbeidstimer. Totalt ca 2500 timer.

### 10.2 Software-ressurser

- VBS2 (Virtual battle space2)
- VisualStudio/netbeans (C++ programmering)
- MS office
- Ms project
- Google docs

### 10.3 Hardware-ressurser

- Stasjonær pc
- Joystick
- Private bærbare

## 11 Arbeid frem til prosjektplan foreligger

---

- Kravspek
- Testspek
- Utkast til brukermanual

# KDA RIVET



## Fraværsprotokoll

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....  | 3 |
| 1.1 | Hensikt med dokumentet.....                                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser.....                 | 3 |
| 1.3 | Dokumenthistorie.....  | 3 |
| 1.4 | Ansvarlig .....  | 3 |
| 2   | Komme for sent til avtalte møter: .....                      | 4 |
| 2.1 | 5-15 min.....  | 4 |
| 2.2 | 15min til slutten av arbeidsdagen .....                      | 4 |
| 2.3 | Fravær på grunn av andre avtaler .....                       | 4 |
| 3   | Fravær på grunn av sykdom.....                               | 5 |
| 3.1 | Sykdom enkeltdager .....                                     | 5 |
| 3.2 | Sykdom over to sammenhengende arbeidsdager .....             | 5 |
| 3.3 | Sykdom ved tre eller flere sammenhengende arbeidsdager ..... | 5 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dette dokumentet er å gjøre det klart for gruppemedlemmene hvordan en skal håndtere fravær.

## 1.2 Definisjoner, akronymer og forkortelser

Rislunsj: Piano Rislunsj (riskrem)

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                                    |
|---------------|----------|--------------------------------|---|
| 1.98          | 19.05.10 | Gjennomgang av egne dokumenter | Thomas Transeth                             |
| 1.99          | 28.05.10 | Korrekturlesing                | Arne Kristian Åmellem og Zoran vukobratovic |
| 2.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                             |

## 1.4 Ansvarlig

Thomas Transeth

## **2 Komme for sent til avtalte møter:**

---

### **2.1 5-15 min**

Det skal gis beskjed til gruppeleder om mulig. Dersom gruppeleder har gitt beskjed om fravær skal planleggingsansvarlig kontaktes.

Det skal kjøpes Rislunsj til alle gruppemedlemmene til neste avtalte møte.

### **2.2 15min til slutten av arbeidsdagen**

Det *skal* gis beskjed til gruppeleder. Dersom gruppeleder har gitt beskjed om fravær skal planleggingsansvarlig kontaktes.

Det skal kjøpes Rislunsj til alle gruppemedlemmene til neste avtalte møte.

### **2.3 Fravær på grunn av andre avtaler**

Gruppeleder skal ha skriftlig beskjed på mail innen 24 timer før gitt fravær. Grunnen for fraværet skal legges ved.

## **3 Fravær på grunn av sykdom**

---

### **3.1 Sykdom enkeltdager**

Gruppeleder skal ha beskjed senest 15min før arbeidsdager starter.

### **3.2 Sykdom over to sammenhengende arbeidsdager**

Om en fremdeles føler seg syk dagen før andre arbeidsdag, skal gruppeleder ha beskjed innen kvelden dagen før.

### **3.3 Sykdom ved tre eller flere sammenhengende arbeidsdager**

Om en fremdeles er syk ved tredje arbeidsdag, skal gruppeleder ha beskjed innen kvelden dagen før, men gruppemedlemmet skal også på møtedagen skaffe seg sykemelding.



# KDA RIVET



## Gjennomgang av UML

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....                            | 3 |
| 1.1   | Hensikt med dokumentet.....                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3   | Dokumenthistorie.....                        | 3 |
| 1.4   | Ansvarlig.....                               | 3 |
| 2     | Unified Modeling Language.....               | 4 |
| 2.1   | Fargevalg.....                               | 4 |
| 2.2   | Use-case Diagram.....                        | 4 |
| 2.3   | Kommunikasjonsdiagram.....                   | 5 |
| 2.3.1 | Stereotype klasser.....                      | 5 |
| 2.3.2 | Boundary.....                                | 5 |
| 2.3.3 | Control.....                                 | 5 |
| 2.3.4 | Entity.....                                  | 5 |
| 2.4   | Klassediagram.....                           | 5 |
| 2.5   | Sekvensdiagram.....                          | 6 |
| 2.6   | Aktivitetsdiagram.....                       | 6 |
| 2.7   | Tilstandsdiagram.....                        | 7 |
| 3     | Referanser.....                              | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi leseren en gjennomgang av de UML diagrammer som blir brukt i analysedokumentet og designdokumentet.

## 1.2 Definisjoner, akronymer og forkortelser

**UML:** Unified Modelling Language

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                                |
|---------------|----------|--------------------------------|---|
| 1.98          | 19.05.10 | Gjennomgang av egne dokumenter | Ole-Martin Grøtterud                    |
| 1.99          | 24.05.10 | Korrekturlest                  | Arne Kristian Åmellem<br>& Lone Knutsen |
| 2.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                         |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Unified Modeling Language

---

For å visualisere hvilke komponenter som skal være med i systemet, forholdet mellom disse og programflyten, bruker vi UML. I følgende underkapitler vil alle UML diagrammene som er brukt i dette dokumentet bli kort forklart.

### 2.1 Fargevalg

**Bruker** – Brukeren av systemet er i alle våre diagrammer blå. Dette har vi valg på grunn av et sitat fra sangen ” *Jeg har sett en ordentlig negermann*” hvor det i siste vers synges: ” Jeg synes mennesker skulle males blå for da blir de veldig morsomme å titte på”.

**Packages** – Alle packages i systemet vårt er gule.

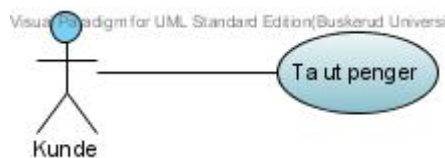
**Klasser** – Klassene våre som skal lages og endres av oss er lyse grønne, mens klasser fra VBS2 eller wxWidgets, vil være røde for å vise at det er ferdiglagde klasser.

**Use-caser** – Use-caser i diagrammene vil være lyseblå.

**Aktiviteter** – Aktiviteter er røde.

**Tilstander** – Tilstander vil være gule.

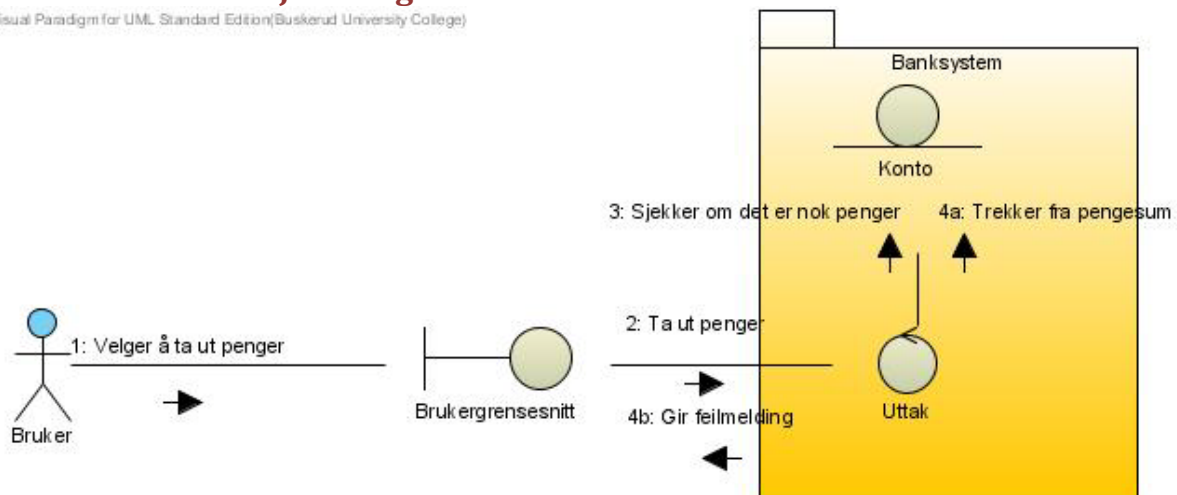
### 2.2 Use-case Diagram



Et use case diagram, er en oversikt over alle interaksjoner som skal kunne utføres på systemet. Det kan være interaksjoner utført av en person, en annen del av systemet eller et annet system. I eksempelet over skal kunden kunne ta ut penger fra systemet.

## 2.3 Kommunikasjonsdiagram

Visual Paradigm for UML Standard Edition (Buskerud University College)



Ett kommunikasjonsdiagram er en oversikt over hvilke klasser som kommuniserer med hverandre. Den vil vise flyten for hver av use casene, men bruker ord som forklaring for hvert steg i flyten, i stedet for metode og funksjonsnavn.

Packages som ligger rundt klassene, forteller at klassene tilhører samme komponent av systemet, eller er ett eget system.

### 2.3.1 Stereotype klasser

Visual Paradigm for UML Standard Edition (Buskerud University College)



I analysedokumentet vil klasser i kommunikasjonsdiagrammet være representert ved analyseklasser. Dette er stereotyper av klasser som er standardisert i UML, for at utviklere skal kunne skille mellom de ulike utfordringene i klassene.[1]

### 2.3.2 Boundary

Boundary klasser er klasser som gir kommunikasjon mellom systemet og utvendige aktører. De er representert som vist til venstre på tegningen over.

### 2.3.3 Control

Control klasser er klasser som tar seg av logikk og kalkulasjoner og håndterer dynamiske endringer i programmet. De er representert som vist i midten på tegningen over.

### 2.3.4 Entity

Entity klasser inneholder informasjon som skal huskes av programmet. De er representert som vist til høyre på tegningen over.

## 2.4 Klassediagram

Visual Paradigm for UML Standard Edition (Buskerud University College)

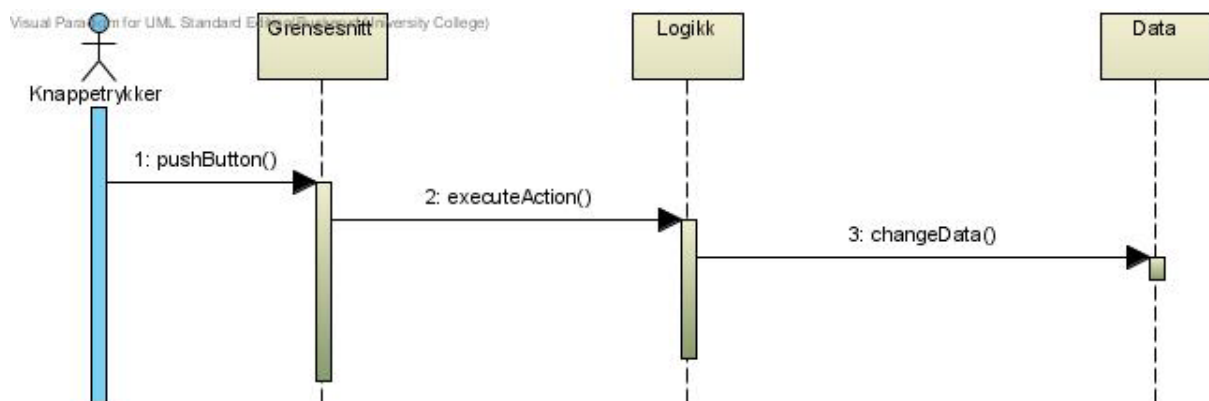


Ett klassediagram forklarer forholdet mellom klassene og hvilke metoder og attributter de inneholder. I diagrammet over ser vi ett panel som har fra 1 til mange knapper, mens knappen(e) kun har ett panel. På streken mellom dem er det verken pil eller kryss, som betyr

at kommunikasjonen er toveis. Den svarte diamanten til venstre indikerer panelet vårt består av blant annet knapper. Videre sier diagrammet at Panel klassen har en public funksjon ved navn reset som returnerer void, mens Knapper klassen har en private attributt ved navn lys som er en bool, og en tennLys funksjon.

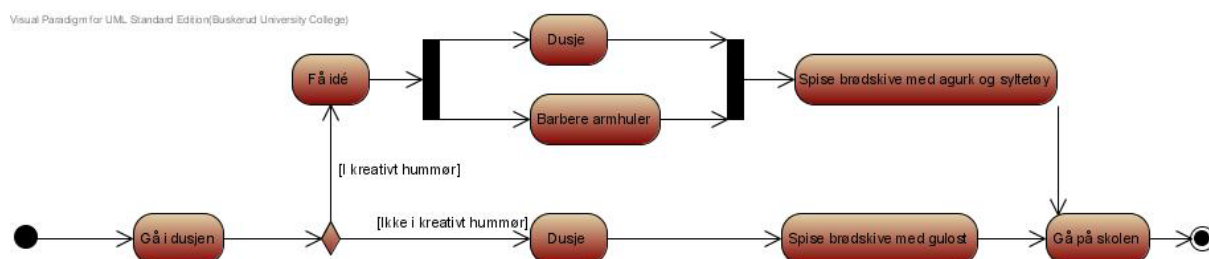
I designdokumentet vil også klassediagrammene inneholde packages(se kommunikasjonsdiagram)

## 2.5 Sekvensdiagram



Sekvensdiagrammet er en oversikt over hendelsesforløpet til en use-case. Den vil vise hvilke deler av programmet som kommuniserer med hverandre og rekkefølgen på kommunikasjonen for de forskjellige use casene. I motsetning til kommunikasjonsdiagrammet viser sekvensdiagrammet hvilke metoder og funksjoner som blir kalt.

## 2.6 Aktivitetsdiagram



Ett aktivitetsdiagram viser trinnvis hendelsesforløpet for en situasjon og kontrollflyten. Den kan inneholde avgjørelser(diamant) som vil sende flyten i forskjellige retninger, avhengig av attributter. Det kan også inneholde forks(svart rektangel) som betyr at hendelser skjer i parallell.

I designdiagrammet vil aktivitetsdiagrammet i tillegg være delt opp i såkalte "Swimlanes". Det vil si at det er horisontale grupper som skiller mellom aktiviteter utført av ulike packages.

## 2.7 Tilstandsdiagram



Et tilstandsdiagram beskriver oppførselen til systemet eller en del av det, og inneholder de forskjellige tilstandene et objekt kan ha. Pilene beskriver hvordan man kan komme fra en tilstand til en annen

### 3 Referanser

---

[1] I. Jacobson, G. Booch og J.Rumbaugh, The Unified Software Development Process, Addison-Wesley, 2000, side 173-214.





# KDA RIVET



## Implementasjons- dokument

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....  | 5  |
| 1.1   | Hensikt med dokumentet.....                              | 5  |
| 1.2   | Definisjoner, akronymer og forkortelser.....             | 5  |
| 1.3   | Dokumenthistorie.....                                    | 5  |
| 1.4   | Ansvarlig.....   | 5  |
| 2     | Innledning.....  | 6  |
| 2.1   | Oppbygning av dokumentet.....                            | 6  |
| 2.1.1 | Arkitektur.....  | 6  |
| 2.1.2 | Komponenter.....   | 6  |
| 2.1.3 | Klasseoversikt.....                                      | 6  |
| 2.2   | Oppbygging av implementasjonen.....                      | 6  |
| 2.3   | Bruk av namespace som struktur.....                      | 6  |
| 3     | Arkitektur.....  | 7  |
| 3.1   | Kommunikasjon mellom simulatorsystem og applikasjon..... | 7  |
| 4     | Komponenter.....   | 8  |
| 4.1   | RIVET TOLK.....  | 8  |
| 4.1.1 | Oppbygging.....  | 8  |
| 4.1.2 | Endringer fra design.....                                | 8  |
| 4.2   | RIVET CORE.....  | 9  |
| 4.2.1 | Oppbygging.....  | 9  |
| 4.2.2 | Endringer fra design.....                                | 9  |
| 4.3   | RIVET LAGER.....   | 10 |
| 4.3.1 | Oppbygging.....  | 11 |
| 4.3.2 | Endringer fra design.....                                | 11 |
| 4.4   | RIVET GUI.....   | 12 |
| 4.4.1 | Oppbygging.....  | 13 |
| 4.4.2 | Endringer fra design.....                                | 15 |
| 5     | Klasseoversikt.....                                      | 16 |
| 5.1   | GLOBAL.....  | 16 |
| 5.1.1 | OnSimulationStep().....                                  | 16 |



|        |                          |    |
|--------|--------------------------|----|
| 5.1.2  | PluginFunction()         | 16 |
| 5.1.3  | ExecuteCommand()         | 16 |
| 5.1.4  | Funksjoner               | 16 |
| 5.2    | Klasse – VBS2Interface   | 17 |
| 5.2.1  | Funksjoner               | 17 |
| 5.3    | Klasse – Core            | 18 |
| 5.3.1  | Funksjoner               | 18 |
| 5.4    | Klasse – VariableStorage | 20 |
| 5.4.1  | Funksjoner               | 21 |
| 5.5    | Klasse – GuiFacade       | 22 |
| 5.5.1  | Funksjoner               | 22 |
| 5.6    | ArmingSwitch             | 23 |
| 5.6.1  | Funksjoner               | 23 |
| 5.7    | Background               | 23 |
| 5.7.1  | Funksjoner               | 23 |
| 5.8    | FiringMode               | 24 |
| 5.8.1  | Funksjoner               | 24 |
| 5.9    | GrenadeArmer             | 24 |
| 5.9.1  | Funksjoner               | 25 |
| 5.10   | GrenadeFlipper           | 25 |
| 5.10.1 | Funksjoner               | 25 |
| 5.11   | Menu                     | 26 |
| 5.11.1 | Funksjoner               | 26 |
| 5.12   | NightVision              | 26 |
| 5.12.1 | Funksjoner               | 26 |
| 5.13   | OnOff                    | 27 |
| 5.13.1 | Funksjoner               | 27 |
| 5.14   | RemoteSafe               | 28 |
| 5.14.1 | Funksjoner               | 28 |
| 5.15   | Sensitivity              | 28 |
| 5.15.1 | Funksjoner               | 28 |
| 5.16   | Zoom                     | 29 |
| 5.16.1 | Funksjoner               | 29 |



|        |  |    |
|--------|--|----|
| 6      | Løsninger og kodeeksempler .....           | 30 |
| 6.1    | UC1 – Bemanne systemet.....                | 30 |
| 6.2    | UC2 – Sikte.....                           | 30 |
| 6.3    | UC3 – Armering på/av .....                 | 30 |
| 6.4    | UC4 – Skyte.....                           | 31 |
| 6.5    | UC5 – Velge skuddhastighet.....            | 31 |
| 6.6    | Simulere servoer.....                      | 31 |
| 6.7    | UC7 – Zoom .....                           | 31 |
| 6.8    | UC8 – Slå av/på nattmodus .....            | 32 |
| 6.9    | UC9 – Sikte: sensitivitet på joystick..... | 32 |
| 6.10   | Ekstrafunksjoner .....                     | 32 |
| 6.10.1 | Menysystem .....                           | 32 |
| 6.11   | Mulighet for å velge FCU .....             | 34 |
| 6.12   | Referanser .....                           | 35 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Dette dokumentet skal gi leseren en forståelse på hvordan implementasjonen faktisk foregikk. Tabeller og diagrammer stammer hovedsakelig fra designdokumentet, men i stedet for å vise det samme igjen, er dette dokumentet oppdatert etter hvordan systemet ble kodet.

## 1.2 Definisjoner, akronymer og forkortelser

**DLL:** Dynamic-link Library

**UML:** Unified Modeling Language

**VBS2:** Virtual Battle Space 2

**RWS:** Remote Weapon Station

**FCU:** Field Command Unit

**ASI:** Application Scripting Interface

**RIVET:** Rws In VBS2 Environment Testing

**CORE:** Command, Operation and Retrival Entity

**TOLK:** Two-way Organized Language Kit

**LAGER:** Long Abbreviation That Isn't Quite Right

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                           |
|---------------|----------|---------------------------|------------------------------------|
| 1.1           | 19.04.10 | Videreutviklet kapittel 4 | Thomas Transeth                    |
| 1.99          | 27.05.10 | Korrekturlest             | Lone Knutsen og Zoran Vukobratovic |
| 2.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Innledning

---

### 2.1 Oppbygning av dokumentet

For uten om innledingen består dette dokumentet av tre deler:

#### 2.1.1 Arkitektur

Denne delen gir en forståelse på hvordan systemet er bygget opp på en overordnet måte og det blir her forklart hvordan de ulike hovedsystemene er koblet sammen og kommuniserer.

#### 2.1.2 Komponenter

Denne delen gir en oversikt over de ulike komponentene systemet er satt opp av og de viktigste systemene i disse.

#### 2.1.3 Klasseoversikt

Denne delen gir en dypere beskrivelse på hvordan hver klasse er bygget opp og gir en beskrivelse på klassens hovedmål og viktigste kode.

### 2.2 Oppbygging av implementasjonen

Implementasjonen er i denne tidlige iterasjonen av Construction hovedsaklig bygget på "Prototype – dll" og "Prototype – wxWidgets" (se egne dokumenter). Ved å sette sammen disse to testsystemene kom vi fram til et system som hadde et riktig utseende og med kommunikasjon både fra kode til VBS2 og motsatt, men gruppen syntes ikke dette var en god nok løsning for oppgaven, og vi valgte derfor å sette hele systemet inn i et mer stabilt rammeverk. På denne måten sikrer vi en stabil arkitektur som er styrt av et gjennomtenkt og strukturert system.

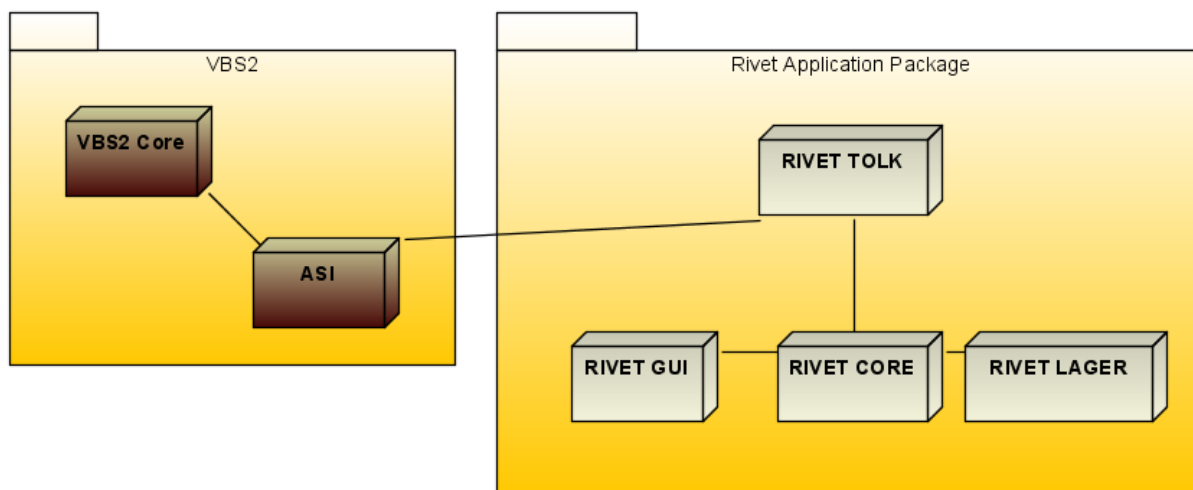
### 2.3 Bruk av namespace som struktur

For å kunne dele systemet inn i en oversiktelig struktur velger vi å bruke "pakker" der klasser med samme overordnede egenskap samles. Siden implementasjonen skrives i C++ har vi tatt i bruk "namespaces". Denne strukturen er godt egnet til å gruppere relatert data [1], og på denne måten sikrer vi at alle komponenter blir referert til hvor de hører hjemme.

### 3 Arkitektur

---

Det fulle systemet består av to hoveddeler, der den ene er systemet VBS2 driver og den andre applikasjonen som prosjektet baserer seg på.



#### 3.1 Kommunikasjon mellom simulatorsystem og applikasjon

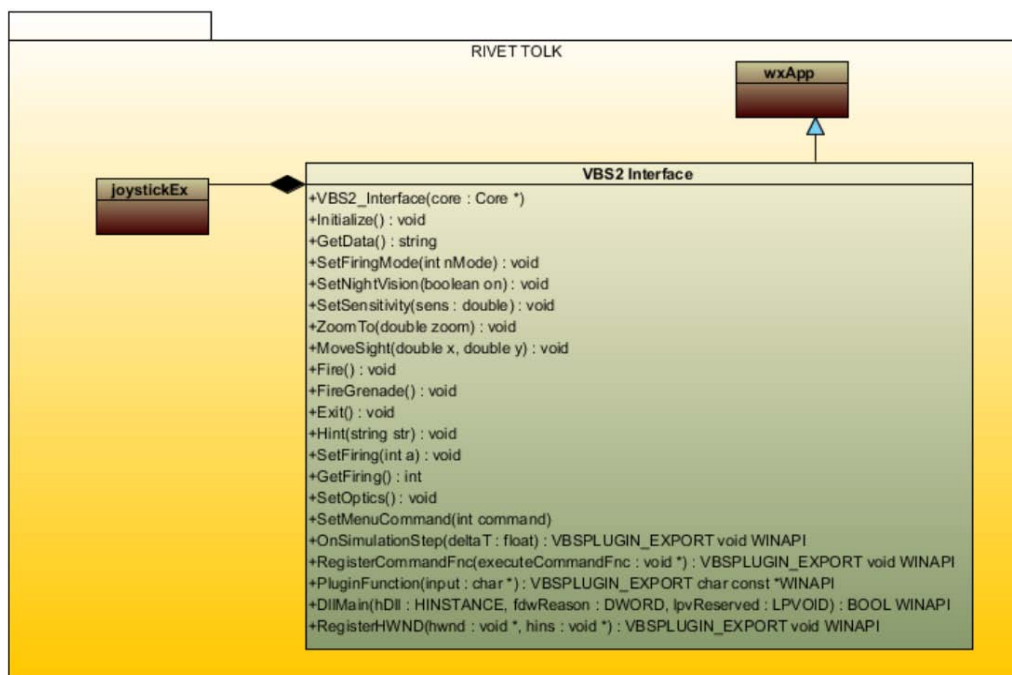
For å kunne kommunisere med VBS2 må kommunikasjonen gå gjennom det predefinerte grensesnittet ASI og dette gjøres ved hjelp av DLL-filer, brukt som en plugin. Ved å legge disse filene inn i en predefinert mappe vil VBS2 være i stand til å bruke kode som gruppen ønsker skal kjøres.

For å forenkle kommunikasjonen har Bohemia Interactive, skaperne av VBS2, satt opp fire funksjoner som tar seg av kommunikasjonen mellom sitt eget script og brukerdefinert kode. Mer om dette kan leses i kapittel 4 og 5, RIVET TOLK og GLOBAL

## 4 Komponenter

### 4.1 RIVET TOLK

Denne komponenten består av kommunikasjonen mellom kjernene av applikasjonen og fungerer som et interface mot VBS2.



#### 4.1.1 Oppbygging

Komponenten består av to klasser, VBS2Interface og joystickEx, men inneholder også de fire kommunikasjonsfunksjonene til VBS2.

##### 4.1.1.1 VBS2Interface

Når systemet starter opp vil VBS2 gi et funksjonskall til denne klassen via GLOBAL. Den bygger og starter den egendefinerte koden, først via RIVET CORE som videre oppretter objektet av denne klassen.

##### 4.1.1.2 joystickEx

Denne klassen ble lagt til systemet da den svarte til akkurat den funksjonaliteten systemet trengte for å implementere et skikkelig joystick-oppsett. Denne klassen er en del av kildekode tilgjengelig i 3.partssystemet wxWidgets.

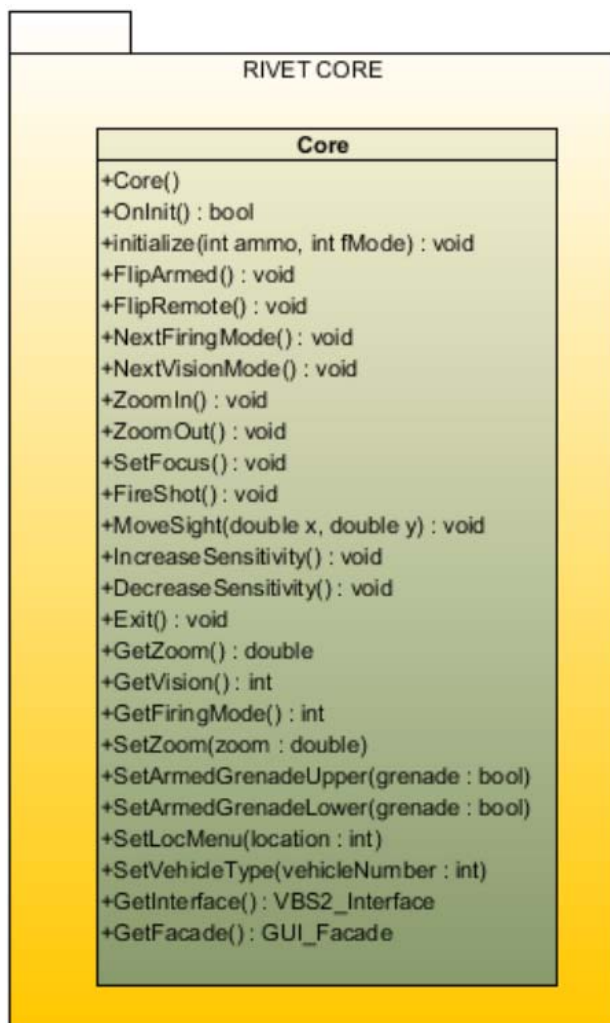
#### 4.1.2 Endringer fra design

Denne komponenten er relativ lik fra den vi utarbeidet i designet, da uansett hva som skal gjøres må gjennom kommunikasjonsmetodene til VBS2. Klassen joystickEx ble lagt til og det er lagt til noen ekstra funksjoner da oppgaven ble videreutviklet etter ønske fra oppdragsgiver.



## 4.2 RIVET CORE

Dette er kontrollkomponenten i systemet og all oppbygging, systemfunksjoner og kommunikasjon går gjennom denne.



### 4.2.1 Oppbygging

Komponenten består av én klasse, Core.

#### 4.2.1.1 Core

Når VBS2 starter Core vil alle resterende klasser bli opprettet og alle disse kontrolleres fra denne klassen. For å sørge for at all drift av systemet er samlet på ett sted, vil alt prosessering av data skje her og all kommunikasjon mellom klasser (sett bort fra GuiFacade; se RIVET GUI) går gjennom Core.

### 4.2.2 Endringer fra design

Tidligere i designet var det til stedet en klasse InputDeviceHandler som skulle ta seg av kontrollen av joystick, men denne ble tatt bort da joystickEx ble implementert (se RIVET TOLK::joystickEx)

### 4.3 RIVET LAGER

Denne komponenten lagrer alle verdier og nivåer systemfunksjonene trenger.



### **4.3.1 Oppbygging**

Komponenten består av én klasse, VariableStorage.

#### **4.3.1.1 VariableStorage**

Flere funksjoner i systemet har et behov for å lagre data og denne klassen har ansvaret for alltid å ha oppdaterte verdier. Den består kun av get()- og set()-funksjoner.

### **4.3.2 Endringer fra design**

Det er ikke endret noe i denne klassen bort sett fra flere get()- og set()-funksjoner etter hvert som det oppsto flere variabler.

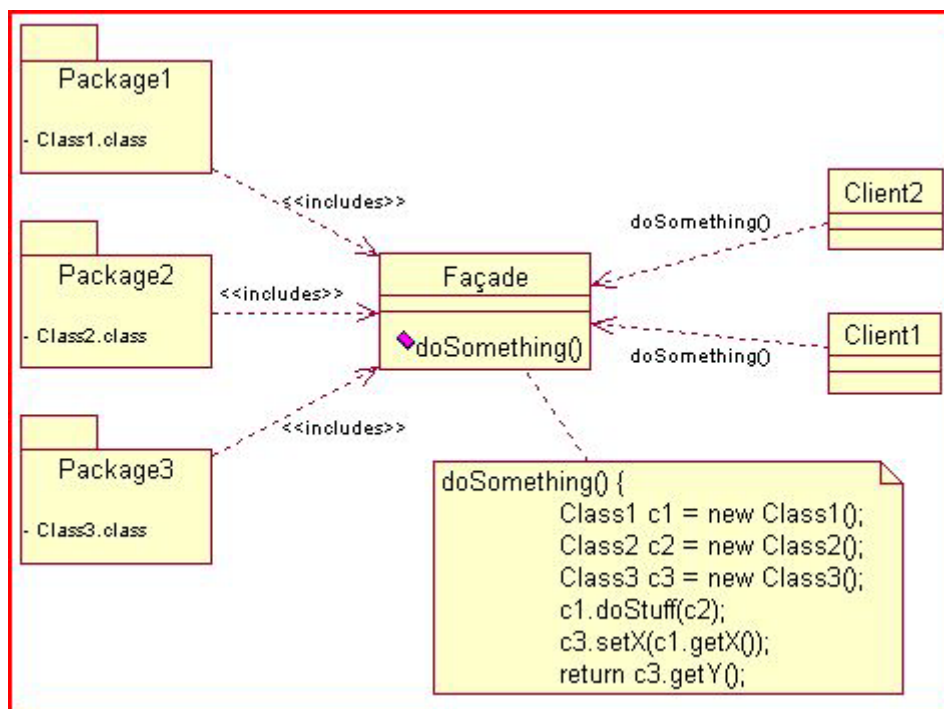


#### 4.4.1 Oppbygging

Komponenten består av 23 klasser; se under. Hver klasse inneholder en grafisk representasjon av knapper og brytere og ivaretar tilhørende brukerinteraksjon.

##### 4.4.1.1 GuiFacade

For å få et mer standardisert system har vi valgt å bruke design pattern "Facade"[2] som kontrollobjekt for den grafiske biten av systemet. Denne er et bindeledd mellom alle GUI-bitene og Core. På denne måten slipper en å opprette hver minste bit i Core og avlaster denne.



##### 4.4.1.1.1 Motivasjonen for å bruke Facade:

Vi har gjort dette for å tilby en forenklet, forbedret og mer objektorientert interface til at subsystem eller delsystem som er veldig komplekst (eller mer komplekst enn vi ønsker). Facade tillater også gjenbruk av verdifulle undersystemer uten at dette nødvendigvis har en direkte kobling som passer.

I RIVET-systemet ønsker vi å redusere koblingene mellom RIVET GUI og CORE. Vi føler at dette pattern-et er det riktige å bruke her.

I tillegg oppretter denne klassen også alle "frames" som GUI-et legges på, samt velger hvilket system som skal vises i henhold til hvilket kjøretøy som blir bemannet.

##### 4.4.1.2 ArmingSwitch

Bryter for armering av våpenet.

##### 4.4.1.3 Background

Elementene GUI-et tegnes på.

##### 4.4.1.4 FiringMode

Knapp for skifting av skytemodus og skuddhastighet.

#### **4.4.1.5 GrenadeArmer**

Bryter for å armere granatkasteren.

#### **4.4.1.6 GrenadeFlipper**

Bryter for å avfyre granatkasteren.

#### **4.4.1.7 Menu**

Knappesett for å slå på menysystemet og navigering i denne.

#### **4.4.1.8 NightVision**

Knapp for å bytte mellom kameratypene dag, lysforsterkende og infrarødt.

#### **4.4.1.9 OnOff**

Knapp for å slå av og på systemet.

#### **4.4.1.10 RemoteSafe**

Bryter for å mekanisk sikre våpenet.

#### **4.4.1.11 Sensitivity**

Knapper for å justere sensitiviteten på joystick.

#### **4.4.1.12 Zoom**

Knapper for å justere på zoom-nivået.

#### **4.4.1.13 Brightness**

Knapp for justering av lysstyrken på FCU. *Ikke implementert*

#### **4.4.1.14 BtlOverride**

Bryter for overstyring av sikkerhetssoner. *Ikke implementert*

#### **4.4.1.15 Calibration**

Knapp for kalibrering av systemet. *Ikke implementert*

#### **4.4.1.16 Focus**

Knapper for å justere fokus på FCU. *Ikke implementert*

#### **4.4.1.17 Gain**

Knapp for justering av kalibrering av lysstyrken på FCU for å øke kontrastene. *Ikke implementert*

#### **4.4.1.18 LampTest**

Knapp for å teste om statuslamper virker. *Ikke implementert*

#### **4.4.1.19 Level**

Knapp for justering av kalibrering av lysstyrken(temperatur) på FCU for å øke kontrastene. *Ikke implementert*

#### **4.4.1.20 Polarity**

Knapp for å invertere fargene på FCU. *Ikke implementert*

#### **4.4.1.21 Range**

Knapp for manuell justering av avstanden slik siktet og våpen er kalibrert på flere avstander. *Ikke implementert*

#### **4.4.1.22 ReticleSelect**

Knapp for å bytte siktekors (crosshair). *Ikke implementert*

#### **4.4.2 Endringer fra design**

I det første designet var det ikke planlagt å bruke en kontrollklasse for GUI, men etter som systemet vokste og det oppstod flere og flere klasser var dette et naturlig valg.

Vi valgte også å implementere alle knappegruppene fra FCU i egne klasser slik at knapper som direkte har noe med hverandre ligger sammen. Vi har også opprettet samtlige knapper som finnes for å lette videre arbeid med implementasjon.



## 5 Klasseoversikt

### 5.1 GLOBAL

Denne klassen er eneste delen av applikasjonen som kommuniserer med VBS2. Det er spesielt tre funksjoner som spesielt er verdt å beskrive:

| GLOBAL  |
|---|
| +OnSimulationStep(deltaT : float) : VBSPLUGIN_EXPORT void WINAPI                  |
| +RegisterCommandFnc(executeCommandFnc : void *) : VBSPLUGIN_EXPORT void WINAPI    |
| +PluginFunction(input : char *) : VBSPLUGIN_EXPORT char const *WINAPI             |
| +DllMain(hDll : HINSTANCE, fdwReason : DWORD, lpvReserved : LPVOID) : BOOL WINAPI |
| +RegisterHWND(hwnd : void *, hins : void *) : VBSPLUGIN_EXPORT void WINAPI        |

#### 5.1.1 OnSimulationStep()

Denne funksjonen blir kalt av VBS2 etter hvert "frame" som blir kjørt. Dette gjør at handlinger som skal skje over tid kan lett manipuleres, men en skal være forsiktig med å legge inn ressurskrevende kode da dette kan føre til lav "frame rate". I funksjonens parameter blir tiden siden siste "frame" tatt i mot.

#### 5.1.2 PluginFunction()

Dette er en funksjon som kan bli kalt direkte fra koden internt i VBS2. Dette gjør det mulig å utføre handlinger når en hendelse oppstår i simuleringen. For eksempel kan en bemanning av et kjøretøy eller avfiring av et våpen settes til å kalle på denne funksjonen sammen med en tekststreng. Ut i fra denne tekststrengen vil det være mulighet til å identifisere hva som har hendt og da kjøre riktig kommando.

#### 5.1.3 ExecuteCommand()

For all kommunikasjon fra VBS2 må funksjonen PluginFunction bli kalt, men når en skal sende kommandoer til VBS2 holder det å sende kommandoen som RegisterCommandFnc har predefinert; ExecuteCommand(), med tilhørende argumenter. Via denne kan vi da sende kommandoer til simulatorsystemet når vi vil noe skal skje.

#### 5.1.4 Funksjoner

| Navn                | Beskrivelse  |
|---------------------|--|
| +OnSimulationStep   | Funksjon som kalles hver gang et frame vises i VBS2. Denne brukes i systemet når noe skal skje kontinuerlig. |
| +RegisterCommandFnc | Aktiverer PluginFunction.  |
| +PluginFunction     | Funksjon som kan kalles fra VBS2. Den blir brukt når en spesielt hendelse skjer som ved en trigger.          |
| +DllMain            | Main-funksjon i dll.   |
| +RegisterHWND       | Henter handler til VBS2.   |



## 5.2 Klasse - VBS2Interface

Denne klassen kobler VBS2 og plugin-systemet sammen og fungerer som et interface. All kommunikasjon som går mellom disse to komponentene vil bli sendt eller mottatt i denne klassen og den blir utelukkende sendt som tekststrenger.

Klassen består av to deler, den ene som hører til RIVET TOLK men også delen GLOBAL (se over) som er selve interfacet til VSB2.

| VBS2 Interface  |
|---|
| +VBS2_Interface(core : Core *)  |
| +Initialize() : void  |
| +GetData() : string   |
| +SetFiringMode(int nMode) : void  |
| +SetNightVision(boolean on) : void  |
| +SetSensitivity(sens : double) : void   |
| +ZoomTo(double zoom) : void   |
| +MoveSight(double x, double y) : void   |
| +Fire() : void  |
| +FireGrenade() : void   |
| +Exit() : void  |
| +Hint(string str) : void  |
| +SetFiring(int a) : void  |
| +GetFiring() : int  |
| +SetOptics() : void   |
| +SetMenuCommand(int command)  |
| +OnSimulationStep(deltaT : float) : VBSPLUGIN_EXPORT void WINAPI                  |
| +RegisterCommandFnc(executeCommandFnc : void *) : VBSPLUGIN_EXPORT void WINAPI    |
| +PluginFunction(input : char *) : VBSPLUGIN_EXPORT char const *WINAPI             |
| +DIIMain(hDll : HINSTANCE, fdwReason : DWORD, lpvReserved : LPVOID) : BOOL WINAPI |
| +RegisterHWND(hwnd : void *, hins : void *) : VBSPLUGIN_EXPORT void WINAPI        |

### 5.2.1 Funksjoner

| Navn            | Beskrivelse   |
|-----------------|---|
| +VBS2Interface  | Konstruktøren til klassen.  |
| +~VBS2Interface | Dekonstruktøren. Sletter alle objekter som er avhengig av denne klassen.                    |
| +Initialize     | Setter startvariabler.  |
| +SetSensitivity | Setter sensitiviteten til joystick.   |
| +ZoomTo         | Setter zoom-nivået til siktet.  |
| +MoveSight      | Endrer synsretningen etter hvert som en beveger på joystick.                                |
| +Fire           | Avfyrrer våpenet.   |
| +FireGrenade    | Avfyrrer granaten.  |
| +Exit           | Kaster brukeren ut av skytterposisjonen   |
| +Hint           | Gir mulighet for å sette en hintmelding innenfor det synlige feltet i FCU. KUN FOR TESTING. |
| +SetFireing     | Setter skuddmodus   |
| +GetFireing     | Henter skuddmodus.  |

|                 |   |
|-----------------|---|
| +SetOptics      | Setter kameramodus.                       |
| +SetMenuCommand | Sender ny menyposisjon til VBS2-scriptet. |

### 5.3 Klasse - Core

Denne klassen er kontrollkjernen i systemet. All kommunikasjon mellom de ulike komponentene går gjennom denne klassen og funksjoner som gjør beregninger eller bestemmer handlinger er lagt her.

| Core                                    |
|---|
| +Core()                                 |
| +OnInit() : bool                        |
| +initialize(int ammo, int fMode) : void |
| +FlipArmed() : void                     |
| +FlipRemote() : void                    |
| +NextFiringMode() : void                |
| +NextVisionMode() : void                |
| +ZoomIn() : void                        |
| +ZoomOut() : void                       |
| +SetFocus() : void                      |
| +FireShot() : void                      |
| +MoveSight(double x, double y) : void   |
| +IncreaseSensitivity() : void           |
| +DecreaseSensitivity() : void           |
| +Exit() : void                          |
| +GetZoom() : double                     |
| +GetVision() : int                      |
| +GetFiringMode() : int                  |
| +SetZoom(zoom : double)                 |
| +SetArmedGrenadeUpper(grenade : bool)   |
| +SetArmedGrenadeLower(grenade : bool)   |
| +SetLocMenu(location : int)             |
| +SetVehicleType(vehicleNumber : int)    |
| +GetInterface() : VBS2_Interface        |
| +GetFacade() : GUI_Facade               |

#### 5.3.1 Funksjoner

| Navn        | Beskrivelse   |
|-------------|---|
| +Core       | Konstruktøren til klassen. Oppretter alle andre klasseobjekter. |
| +~Core      | Dekonstruktøren til klassen.                                    |
| +OnInit     | Returnerer om objektet finnes.                                  |
| +Initialize | Setter startverdier.  |
| +FlipArmed  | Setter armeringsstatus til motsatte av det den er.              |
| +FlipRemote | Setter remote-status til motsatte av det den er.                |

|                       |  |
|-----------------------|--|
|                       | er.  |
| +NextFiringMode       | Setter avfyringsmodus til neste modus.                   |
| +NextVisionMode       | Setter kameramodus til neste modus.                      |
| +ZoomIn               | Zoomer inn siktet.                                       |
| +ZoomOut              | Zoomer ut siktet.  |
| +SetFocus             | Setter riktig FCU ut fra hvilken kjøretøy som er aktivt. |
| +FireShot             | Sender avfyringskommando til interface.                  |
| +FireGrenade          | Sender granat-avfyringskommando til interface.           |
| +MoveSight            | Beveger siktet.  |
| +IncreaseSensitivity  | Setter sensitiviteten til et lavere nivå.                |
| +DecreaseSensitivity  | Setter sensitiviteten til et høyere nivå.                |
| +Exit                 | Sender avslutningskommando til interface                 |
| +GetZoom              | Returnerer gjeldene zoom-nivå.                           |
| +GetVision            | Returnerer gjeldene kameramodus.                         |
| +GetFiringMode        | Returnerer gjeldene avfyringsmodus.                      |
| +SetZoom              | Setter nytt zoom-nivå.                                   |
| +SetArmedGrenadeUpper | Setter øvre granatstatus.                                |
| +SetArmedGrenadeLower | Setter nedre granatstatus.                               |
| +SetLocMenu           | Sender ny menylokasjon til interface                     |
| +SetVehicleType       | Setter gjeldene kjøretøy.                                |

## 5.4 Klasse – VariableStorage

Denne klassen er variabellageret i systemet. Den har som oppgave å ta vare på de forskjellige variablene som vi ønsker å lagre. Disse variablene ar alt fra ammunisjon og avfyringsmodus til nattsikte og zoom.

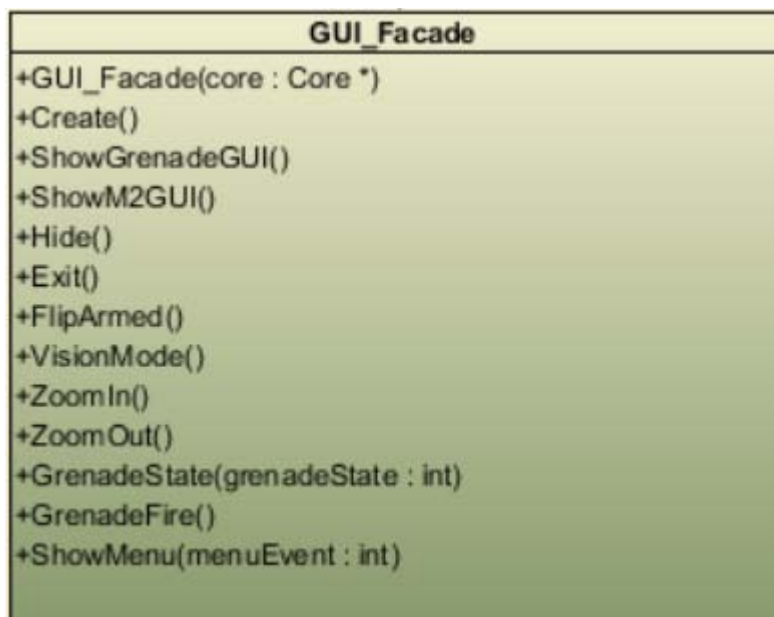
| VariableStorage                              |
|--|
| -ammo : int                                  |
| -firingMode : int                            |
| -visionMode : boolean                        |
| -zoomClicks : int                            |
| -zoom : double                               |
| -sensitivity : double                        |
| -armed : boolean                             |
| -remoteSafe : boolean                        |
| -grenadeUpperArmed : boolean                 |
| -grenadeLowerArmed : boolean                 |
| -grenadeUpper : boolean                      |
| -grenadeLower : boolean                      |
| -menuLocation : int                          |
| -menuArray : int*                            |
| -vehicleType : int                           |
| +VariableStorage()                           |
| +~VariableStorage()                          |
| +SetAmmo(ammo : int) : void                  |
| +GetAmmo() : int                             |
| +SetFiringMode(firingMode : int) : void      |
| +GetFiringMode() : int                       |
| +SetVision(visionMode : int) : void          |
| +IsNighTVision() : int                       |
| +SetZoomClicks(zoomClicks : int) : void      |
| +GetZoomClicks() : int                       |
| +SetZoom(zoom : double) : void               |
| +GetZoom() : double                          |
| +SetSensitivity(sensitivity : int) : void    |
| +GetSensitivity() : double                   |
| +SetArmed(armed : bool) : void               |
| +IsArmed() : bool                            |
| +SetRemoteSafe(remoteSafe : bool) : void     |
| +IsRemoteSafe() : bool                       |
| +SetArmedGrenadeUpper(grenade : bool) : void |
| +GrenadeUpperIsArmed() : bool                |
| +SetArmedGrenadeLower(grenade : bool) : void |
| +GrenadeLowerIsArmed() : bool                |
| +SetGrenadeUpper(grenade : bool) : void      |
| +SetGrenadeLower(grenade : bool) : void      |
| +GrenadeUpperIsReady() : bool                |
| +GrenadeLowerIsReady() : bool                |
| +GetVehicleType() : int                      |
| +SetVehicleType(vehicleType : int) : void    |
| +SetMenuLocation(location : int) : void      |
| +GetMenuArray(index : int) : int             |
| +GetMenuLocation() : int                     |

### 5.4.1 Funksjoner

| Navn                  | Beskrivelse  |
|-----------------------|--|
| +VariableStorage      | Klassens konstruktør. Setter alle startverdier for et nyoppsatt våpen. |
| +~VariableStorage     | Klassens dekonstruktør.  |
| +SetAmmo              | Setter ammotelleren.   |
| +GetAmmo              | Returnerer ammotelleren.   |
| +SetFiringMode        | Setter avfyringsmodus.   |
| +GetFiringMode        | Returnerer avfyringsmodus.   |
| +SetVision            | Setter kameramodus.  |
| +SetZoomClicks        | Setter zoomnivået.   |
| +GetZoomClicks        | Returnerer zoomnivået.   |
| +SetZoom              |  |
| +GetZoom              | Returnerer zoom.   |
| +SetSensitivity       | Setter sensitiviteten.   |
| +GetSensitivity       | Returnerer sensitiviteten.   |
| +SetArmed             | Setter armeringen på våpenet.  |
| +IsArmed              | Returnerer armeringsstatus.  |
| +SetRemoteSafe        | Setter armeringssikring.   |
| +IsRemoteSafe         | Returnerer armeringssikring.   |
| +SetArmedGrenadeUpper | Setter øvre granat armeringssikring.                                   |
| +GrenadeUpperIsArmed  | Returnerer øvre granat armeringssikring.                               |
| +SetArmedGrenadeLower | Setter nedre granat armeringssikring.                                  |
| +GrenadeLowerIsArmed  | Returnerer nedre granat armeringssikring.                              |
| +SetGrenadeUpper      | Setter øvre granat som brukt.  |
| +SetGrenadeLower      | Setter nedre granat som brukt.   |
| +GrenadeUpperIsReady  | Returnerer øvre granatstatus.  |
| +GrenadeLowerIsReady  | Returnerer nedre granatstatus.   |
| +GetVehicleType       | Returnerer gjeldene kjøretøy.  |
| +SetVehicleType       | Setter gjeldene kjøretøy.  |
| +SetMenuLocation      | Setter menylokasjon.   |
| +GetMenuArray         | Returnerer størrelsen på menyen.                                       |
| +GetMenuLocation      | Returnerer menylokasjon.   |

## 5.5 Klasse - GuiFacade

Denne klassen tar seg brukerinteraksjon. Den oppretter alle gui-blokkene og mottar alle kommandoer disse sender, og sender dem videre til Core.



### 5.5.1 Funksjoner

| Navn            | Beskrivelse  |
|-----------------|--|
| +GuiFacade      | Klassens konstruktør.                                |
| +~GuiFacade     | Klassens dekonstruktør. Sletter alle gui-elementene. |
| +Create         | Oppretter alle gui-elementene.                       |
| +ShowGrenadeGUI | Setter alle gui-elementene synlig med granatkaster.  |
| +ShowM2GUI      | Setter alle gui-elementene synlig uten granatkaster. |
| +Hide           | Skjuler alle gui-elementene.                         |
| +Exit           | Sender avslutningskommando til Core.                 |
| +FlipArmed      | Setter armeringsvariabelen.                          |
| +VisionMode     | Setter kameramodus.                                  |
| +FlipRemote     | Setter mekanisk sikring på våpenet.                  |
| +SensDown       | Setter sensitiviteten på joysticken ned.             |
| +SensUp         | Setter sensitiviteten på joysticken opp.             |
| +FireMode       | Setter neste avfyringsmodus.                         |
| +ZoomIn         | Zoomer inn.  |
| +ZoomOut        | Zoomer ut.   |
| +GrenadeState   | Setter granatstatus.                                 |
| +GrenadeFire    | Sender granat avfyringskommando.                     |
| +ShowMenu       | Sender menykommando.                                 |



## 5.6 ArmingSwitch

Denne klassen tar seg av armeringen av våpenet ved et sikkerhetslokk og en bryter. Ved disse tre nivåene sender den kommandoer til GuiFacade. Når den settes til Armed endres en variabel i VariableStorage.

| ArmingSwitch  |
|---|
| -image : wxBitmap<br>-armed : boolean<br>-state : int<br>-mousePos : int<br>-path : wxString  |
| +ArmingSwitch(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+toggleArmed() : void<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void<br>+mouseDown(event : wxMouseEvent &) : void<br>+mouseReleased(event : wxMouseEvent &) : void |

### 5.6.1 Funksjoner

| Navn           | Beskrivelse                                    |
|----------------|--|
| +ArmingSwitch  | Klassens konstruktør. Setter startverdier.     |
| +~ArmingSwitch | Klassens dekonstruktør. Sletter alle objekter. |
| +ToggleArmed   | Sender armeringskommando.                      |
| +PaintEvent    | wxWidgets tegneobjekt.                         |
| +PaintNow      | Tegner objektet.                               |
| +Render        | Rendrer objektet.                              |
| +MouseDown     | Lytter på musen, knappetrykk ned.              |
| +MouseReleased | Lytter på musen, knappetrykk slipp.            |

## 5.7 Background

Denne klassen setter opp områdene knapper og brytere kan leggs på. Det er seks stykker av dem, en for hvert av panelene – bort sett fra det øverste som har ett for og ett uten granatkontroll – og en for den åpne fektet i midtten.

| Background  |
|---|
| -image : wxBitmap   |
| +Background(parent : wxFrame *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void |

### 5.7.1 Funksjoner

| Navn         | Beskrivelse                                    |
|--------------|--|
| +Background  | Klassens konstruktør. Setter startverdier.     |
| +~Background | Klassens dekonstruktør. Sletter alle objekter. |
| +PaintEvent  | wxWidgets tegneobjekt.                         |

|           |                   |
|-----------|-------------------|
| +PaintNow | Tegner objektet.  |
| +Render   | Rendrer objektet. |

## 5.8 FiringMode

Denne klassen bytter mellom de ulike avfyringsmodusene enkeltskudd, femskuddsserie og fullautomatisk. Når den trykkes på endres en variabel i VariableStorage.

| FiringMode  |
|---|
| -image : wxBitmap   |
| +FiringMode(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +nextMode() : void  |
| +paintEvent(evt : wxPaintEvent &) : void  |
| +paintNow() : void  |
| +render(dc : wxDC &) : void   |
| +mouseDown(event : wxMouseEvent &) : void   |

### 5.8.1 Funksjoner

| Navn         | Beskrivelse                                    |
|--------------|--|
| +FiringMode  | Klassens konstruktør. Setter startverdier.     |
| +~FiringMode | Klassens dekonstruktør. Sletter alle objekter. |
| +NextMode    | Sender avfyringsmodus.                         |
| +PaintEvent  | wxWidgets tegneobjekt.                         |
| +PaintNow    | Tegner objektet.                               |
| +Render      | Rendrer objektet.                              |
| +MouseDown   | Lytter på musen, knappetrykk ned.              |

## 5.9 GrenadeArmer

Denne klassen armerer granatdelen av våpenet. Den består av en bryter som har tre posisjoner, to for armering og en for sikring. Når den settes til Armed endres en variabel i VariableStorage.

| GrenadeArmer  |
|---|
| -image : wxBitmap   |
| -path : wxString  |
| -counter : int  |
| -action : int   |
| -armed : boolean  |
| +GrenadeArmer(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +paintEvent(evt : wxPaintEvent &) : void  |
| +paintNow() : void  |
| +render(dc : wxDC &) : void   |
| +mouseDown(event : wxMouseEvent &) : void   |
| +mouseReleased(event : wxMouseEvent &) : void   |



## 5.9.1 Funksjoner

| Navn           | Beskrivelse                                    |
|----------------|--|
| +GrenadeArmer  | Klassens konstruktør. Setter startverdier.     |
| +~GrenadeArmer | Klassens dekonstruktør. Sletter alle objekter. |
| +paintEvent    | wxWidgets tegneobjekt.                         |
| +paintNow      | Tegner objektet.                               |
| +render        | Rendrer objektet.                              |
| +mouseDown     | Lytter på musen, knappetrykk ned.              |
| +mouseReleased | Lytter på musen, knappetrykk slipp.            |

## 5.10 GrenadeFlipper

Denne klassen avfyrer granaten og består av en sikkerhetshette samt en bryter.

| GrenadeFlipper   |
|--|
| -image : wxBitmap<br>-path : wxString<br>-counter : int<br>-action : int<br>-fire : bool   |
| +GrenadeFlipper(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void<br>+mouseDown(event : wxMouseEvent &) : void<br>+mouseReleased(event : wxMouseEvent &) : void |

### 5.10.1 Funksjoner

| Navn             | Beskrivelse                                    |
|------------------|--|
| +GrenadeFlipper  | Klassens konstruktør. Setter startverdier.     |
| +~GrenadeFlipper | Klassens dekonstruktør. Sletter alle objekter. |
| +paintEvent      | wxWidgets tegneobjekt.                         |
| +paintNow        | Tegner objektet.                               |
| +render          | Rendrer objektet.                              |
| +mouseDown       | Lytter på musen, knappetrykk ned.              |
| +mouseReleased   | Lytter på musen, knappetrykk slipp.            |

## 5.11 Menu

Denne klassen lar brukeren navigere i FCU-menyen. Den består av seks knapper. Når denne trykkes på blir det sendt en knappvariabel til core som ut i fra hvilken variabel som er aktiv i VariableStorage, sender en kommando til systemet.

| Menu   |
|--|
| -image : wxBitmap<br>-path : wxString<br>-action : int   |
| +Menu(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void<br>+mouseDown(event : wxMouseEvent &) : void<br>+mouseReleased(event : wxMouseEvent &) : void |

### 5.11.1 Funksjoner

| Navn        | Beskrivelse                                    |
|-------------|--|
| +Menu       | Klassens konstruktør. Setter startverdier.     |
| +~Menu      | Klassens dekonstruktør. Sletter alle objekter. |
| +ToggleMenu | Sender menykommando.                           |
| +PaintEvent | wxWidgets tegneobjekt.                         |
| +PaintNow   | Tegner objektet.                               |
| +Render     | Rendrer objektet.                              |
| +MouseDown  | Lytter på musen, knappetrykk ned.              |

## 5.12 NightVision

Denne klassen skifter mellom de ulike kameraene. Den består av en knapp som skifter mellom vanlig-, lysforsterkende- og infrarødt kamera. Når den trykkes på endres en variabel i VariableStorage.

| NightVision   |
|---|
| -image : wxBitmap   |
| +NightVision(parent : Background *, file : wxString, format : wxBitmapType)<br>+toggle() : void<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void<br>+mouseDown(event : wxMouseEvent &) : void |

### 5.12.1 Funksjoner

| Navn          | Beskrivelse                                    |
|---------------|--|
| +NightVision  | Klassens konstruktør. Setter startverdier.     |
| +~NightVision | Klassens dekonstruktør. Sletter alle objekter. |
| +Toggle       | Sender kameramodus.                            |

|             |                                   |
|-------------|-----------------------------------|
| +PaintEvent | wxWidgets tegneobjekt.            |
| +PaintNow   | Tegner objektet.                  |
| +Render     | Rendrer objektet.                 |
| +MouseDown  | Lytter på musen, knappetrykk ned. |

### 5.13 OnOff

Denne knappen slår av systemet. Den består av en knapp som deaktiverer FCU-en og kaster brukeren ut av posisjonen.

| OnOff  |
|--|
| -image : wxBitmap  |
| +OnOff(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +paintEvent(evt : exPaintEvent &) : void   |
| +paintNow() : void   |
| +render(dc : wxDC &) : void  |
| +mouseDown(event : wxMouseEvent &) : void  |

#### 5.13.1 Funksjoner

| Navn        | Beskrivelse                                    |
|-------------|--|
| +OnOff      | Klassens konstruktør. Setter startverdier.     |
| +~OnOff     | Klassens dekonstruktør. Sletter alle objekter. |
| +PaintEvent | wxWidgets tegneobjekt.                         |
| +PaintNow   | Tegner objektet.                               |
| +Render     | Rendrer objektet.                              |
| +MouseDown  | Lytter på musen, knappetrykk ned.              |

## 5.14 RemoteSafe

Denne klassen er en overstyrer for en sikring som avgrensner siktet for områder som ikke skal avfyres i. Den består av et sikkerhetslokk og en bryter. Når den settes til Charge endres en variabel i VariableStorage.

| RemoteSafe   |
|--|
| -image : wxBitmap<br>-path : wxString<br>-counter : int<br>-action : int<br>-armed : boolean   |
| +RemoteSafe(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void<br>+mouseDown(event : wxMouseEvent &) : void<br>+mouseReleased(event : wxMouseEvent &) : void |

### 5.14.1 Funksjoner

| Navn           | Beskrivelse                                    |
|----------------|--|
| +RemoteSafe    | Klassens konstruktør. Setter startverdier.     |
| +~RemoteSage   | Klassens dekonstruktør. Sletter alle objekter. |
| +paintEvent    | wxWidgets tegneobjekt.                         |
| +paintNow      | Tegner objektet.                               |
| +render        | Rendrer objektet.                              |
| +mouseDown     | Lytter på musen, knappetrykk ned.              |
| +mouseReleased | Lytter på musen, knappetrykk slipp.            |

## 5.15 Sensitivity

Denne klassen styrer sensitiviteten på stikken. Den består av to knapper. Når den trykkes på endres en variabel i VariableStorage.

| Sensitivity   |
|---|
| -image : wxBitmap   |
| +Sensitivity(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade)<br>+sensUp() : void<br>+sensDown() : void<br>+paintEvent(evt : wxPaintEvent &) : void<br>+paintNow() : void<br>+render(dc : wxDC &) : void |

### 5.15.1 Funksjoner

| Navn          | Beskrivelse                                    |
|---------------|--|
| +Sensitivity  | Klassens konstruktør. Setter startverdier.     |
| +~Sensitivity | Klassens dekonstruktør. Sletter alle objekter. |
| +SensUp       | Setter sensitiviteten opp.                     |
| +SensDown     | Setter sensitiviteten ned.                     |

|             |                                   |
|-------------|-----------------------------------|
| +PaintEvent | wxWidgets tegneobjekt.            |
| +PaintNow   | Tegner objektet.                  |
| +Render     | Rendrer objektet.                 |
| +MouseDown  | Lytter på musen, knappetrykk ned. |

## 5.16 Zoom

Denne klassen styrer zoom-nivået til siktet. Den består av to knapper. Når den trykkes på endres en variabel i VariableStorage.

| Zoom  |
|---|
| -image : wxBitmap   |
| +Zoom(parent : Background *, file : wxString, format : wxBitmapType, gUI_Facade : GUI_Facade) |
| +zoomIn() : void  |
| +zoomOut() : void   |
| +paintEvent(evt : wxPaintEvent &) : void  |
| +mouseDown(event : wxMouseEvent &) : void   |
| +paintNow() : void  |
| +render(dc : wxDC &) : void   |

### 5.16.1 Funksjoner

| Navn        | Beskrivelse                                    |
|-------------|--|
| +Zoom       | Klassens konstruktør. Setter startverdier.     |
| +~Zoom      | Klassens dekonstruktør. Sletter alle objekter. |
| +ZoomIn     | Setter sensitiviteten opp.                     |
| +ZoomOut    | Setter sensitiviteten ned.                     |
| +PaintEvent | wxWidgets tegneobjekt.                         |
| +MouseDown  | Lytter på musen, knappetrykk ned.              |
| +PaintNow   | Tegner objektet.                               |
| +Render     | Rendrer objektet.                              |

## 6 Løsninger og kodeeksempler

Dette kapitlet viser en beskrivelse av løsningene som ble brukt for å fullføre oppgavene, der vi ønsker å spesifisere hva som har blitt gjort. For å gi en bedre struktur blir disse vist via use case.

### 6.1 UC1 – Bemanne systemet

Når en bemanner systemet vil det bli sendt en kommando med et tall. Dette beskriver hvilket kjøretøy som bemannes.

```
if(input[2] == 49)
{
    core->SetVehicleType(1);
    vehicle = "bilenbrom";
}
else if(input[2] == 50)
{
    core->SetVehicleType(2);
    vehicle = "pellemilitarbil";
}

string run = vehicle + "disableGunnerInput[[0], true]";
MoveWindow(stuffy, 360, 200, 840, 700, true);
```

Dette lagres så i VariableStorage samtidig som gui-et blir startet og programvinduet justert. Muligheten for å bruke VBS2 kommandoer via joystick blir også slått av.

### 6.2 UC2 – Sikte

For å få siktet til å bevege seg i det man beveger joysticken er dette gjort i interface-funksjonen OnSimulationStep. Denne tar i tillegg hensyn til en dødssone for å unngå at joysticken vandrer av gårde på egenhånd.

```
if((abs(((joystick->GetPosition().x) - 32768)) > 500) ||
    abs(((joystick->GetPosition().y) - 32768)) > 500)
{
    vbsInterface->MoveSight((((joystick->GetPosition().x) -
    32768) * deltaT * ((int)(pow(10.0, desimal))) / (32768 /
    90)),(((joystick->GetPosition().y) - 32768) * deltaT *
    ((int)(pow(10.0, desimal))) / (32768 / 90)));
}
```

### 6.3 UC3 – Armering på/av

Armeringen av våpenet består av en variabel som setter til true eller false i henhold til om den skal være armert eller ikke. Denne settes ved armeringsknapp, lagres i VariableStorage og hentes av avfyringsfunksjonen hver gang denne kalles.

```
if(storage->IsArmed() && storage->IsRemoteSafe())
{
    vbsInterface->Fire();
}
```

## 6.4 UC4 – Skyte

Når en skal avfyre våpenet blir det først sjekket om det er armert. Om dette er sant blir våpenet avfyrt.

```
string fire = vehicle + "fire \"VBS2_M2_HMG\"";
const char* result= fire.c_str ();
ExecuteCommand(result, NULL, 0);
```

## 6.5 UC5 – Velge skuddhastighet

Når en velger avfyringsmodus vil denne hentes fra VariableStorage og sjekke hvilken måte en skal skyte på.

```
switch(core->GetFiringMode())
{
    case 1:
        if(fire && bille)
        {
            //fire code
        }
        break;
    case 2:
        if(vbsInterface->GetFireing() > 0)
        {
            //fire code
        }
        break;
    case 3:
        if(skyt)
        {
            //fire code
        }
        break;
}
```

Ved enkeltskudd vil en boolean hindre videre kall på avfyringskoden.

Ved fem-takt vil det telles ned fem skudd.

Ved fullautomatisk vil metoden kontinuerlig kalles.

## 6.6 Simulere servoer

Vi startet med å lage en simulasjon av servoene til våpenet via en matematisk modell og fullførte denne. Det viste seg senere at for å kunne få siktet til å virke slik vi ønsket det måtte vi bruke VBS2 sin implementerte simulering.

```
string fullString = "setOpticsState [true, 2, 0, 0,
\"VBS2_M2_HMG\", 0]"; //true for servo simulation
```

## 6.7 UC7 – Zoom

Når en trykker på zoom-knappene vil den tilhørende variabelen settes opp eller ned etter hvilken knapp man trykket.

Zoom fungerer på samme måte som UC8 og tilhører samme funksjon. Man kan her se hvordan en bygger tekststrengen som skal sendes til VBS2

```

stringstream streamZoom;
streamZoom << core->GetZoom();
string strZoom = "";
strZoom.append(streamZoom.str());

string fullString = "setOpticsState [true," + strZoom + "," +
    opticStr + ", 0, \"VBS2_M2_HMG\", 0]";

const char* result = fullString.c_str();
ExecuteCommand(result, NULL,0);

```

## 6.8 UC8 – Slå av/på nattmodus

Når en trykker på kameramodusknappen vil den tilhørende variabelen settes i neste modus. Denne hentes opp og det riktige kameraet aktiveres.

Kameramodus fungerer på samme måte som UC7 og tilhører samme funksjon. En kan her se hvordan en bygger tekststrengen som skal sendes til VBS2.

```

stringstream optic;
string opticStr = "";
optic << core->GetVision();
opticStr.append(optic.str());

string fullString = "setOpticsState [true," + strZoom + "," +
    opticStr + ", 0, \"VBS2_M2_HMG\", 0]";

const char* result = fullString.c_str();
ExecuteCommand(result, NULL,0);

```

## 6.9 UC9 – Sikte: sensitivitet på joystick

Når en trykker på sensitivitetsknappene vil den tilhørende variabelen settes opp eller ned etter hvilken knapp man trykket.

```

dirx += xIn * sensitivity;
diry += yIn * sensitivity;

```

## 6.10 Ekstrafunksjoner

Følgende vil det bli vist løsningene som ble brukt for å løse de ønskede ekstrafunksjonene som ble definert i slutten av prosjektet.

### 6.10.1 Menysystem

For å få til dette måtte vi bruke VBS2-script. Dette er et egendefinert script som bygger på programmeringsspråkene C/C++.

I brukergrensesnittdelen blir hver knapp tildelt en int-verdi som sendes til Core. Her blir menyvariabel hentet og ut i fra den blir det riktige valget sendt via kommando til VBS2-scriptet. 10 for første menyoverskrift, 20, for neste osv. 11 for første menyvalg, 12 for neste osv.



```

switch (location)
{
    case 0: //onOff
        if (menuLocation == 0)
        {
            menuLocation = 10;
        }
        else
        {
            menuLocation = 0;
        }
        break;
    case 1: //selZero
        if (menuLocation % 10 != 0)
        {
            //runFunction(menuLocation); //not implemented
        }
        else
        {
            menuLocation++;
        }
        break;
    case 2: //Up
        if (menuLocation % 10 != 0)
        {
            menuLocation--;
        }
        break;
    case 3: //Down
        if ((menuLocation % 10) <
            storage->GetMenuArray((menuLocation / 10) - 1))
        {
            menuLocation++;
        }
        break;
    case 4: //Left
        menuLocation = 10;
        break;
    case 5: //Right
        if (menuLocation % 10 == 0 && menuLocation != 30)
        {
            menuLocation += 10;
        }
        break;
}

```

## 6.11 Mulighet for å velge FCU

Når en setter seg inn i et kjøretøy blir det sendt en variabel via PluginFunction. Ut i fra dette vil det bli valgt riktig brukergrensesnitt som skal kjøres.

```
if(storage->GetVehicleType() == 1)
{
    gui->ShowM2GUI();
}

else if(storage->GetVehicleType() == 2)
{
    gui->ShowGrenadeGUI();
}
```

## 6.12 Referanser

[1] Bjarne Stroustrup, The C++ Programming Language, Special Edition, Addison-Westly, 2008, side 27.

[2] Erich Gamma, Richard Helm, Ralph Johnson og John M. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. ISBN-13: 978-0201633610

### **Andre dokumenter som er lest under utformingen av dette dokumentet:**

- 08/09 E-Pos – Implementasjonsdokument
- 08/09 KDA Raven – Implementasjonsdokument – SW

# KDA RIVET



## Iterasjonsplan 1

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research .....                               | 4 |
| 2.3 | Krav/Analyse.....                            | 4 |
| 2.4 | Design .....                                 | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test .....                                   | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 1.....                     | 7 |
| 4.1 | Milepæler .....                              | 7 |
| 4.2 | Primære mål .....                            | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 5   | Gantt-diagram.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimat på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem  
**LK:** Lone Knutsen  
**OMG:** Ole-Martin Grøtterud  
**TT:** Thomas Transeth  
**ZV:** Zoran Vukobratovic  
**VBS:** Virtual Battle Space

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                            | Redaktør             |
|---------------|----------|--------------------------------------|----------------------|
| 1.99          | 31.05.10 | Korrekturlest og ny dokumentstandard | Ole-Martin Grøtterud |
| 2.0           | 31.05.10 | Klargjort for innlevering            | Thomas Transeth      |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Iterasjonsplan

Første iterasjon går fra vi satte sammen gruppa, frem til første presentasjon. Under er en liste over planlagte aktiviteter, estimert tidsbrukt og ansvarsområder for iterasjonen. På ansvarsområder der det står "Alle" går ansvaret på rundgang i gruppa.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 100         |
| 101 | Prosjektstyring  | TT        | 20          |
| 102 | Korrekturlesing  | Alle      | 30          |
| 103 | Prosjektplan     | TT        | 10          |
| 104 | Presentasjoner   | Alle      | 80          |
| 105 | Dokumentstandard | LK        | 5           |
| 106 | Økonomi          | ZV        | 3           |
| 107 | Diskusjon        | Alle      |             |
| 108 | Dokumentmaler    | LK        | 10          |
| 109 | Møtereferat      | Alle      | 7,5         |
| 110 | Møteinnkalling   | Alle      | 2,5         |
| 111 | Bedriftsmøte     | Alle      | 30          |
| 112 | Webutvikling     | AKÅ       | 30          |
| 113 | Planlegging      | OMG, ZV   | 15          |
| 114 | Backup           | Alle      |             |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 20          |
| 201 | Trainer-funksjoner                   | TT        | 3           |
| 202 | Interface mellom joystick og trainer |           |             |
| 203 | Add-ons til VBS2                     | AKÅ       | 15          |
| 204 | VTK vs Fusion                        | ZV        | 15          |
| 205 | WxWidgets                            | LK        | 15          |

### 2.3 Krav/Analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 10          |
| 301 | Use case          | LK        | 10          |
| 302 | Kravspesifikasjon | LK        | 5           |
| 304 | Analyse           | OMG       | 5           |
| 305 | Forstudie         | TT        | 20          |
| 306 | Risikoanalyse     | OMG       | 5           |
| 307 | Brukermanual      | ZV        |             |

## 2.4 Design

| Nr. | Aktivitet      | Ansvarlig | Timeestimat |
|-----|----------------|-----------|-------------|
| 400 | Klasser        | AKÅ       |             |
| 401 | Designdokument | AKÅ       |             |
| 402 | Designgrammer  | AKÅ       |             |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        |             |
| 501 | Prototype                    | TT        |             |
| 502 | Implementasjon               | TT        |             |
| 503 | Implementasjonsdokumentasjon | TT        |             |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 20          |
| 601 | Testspesifikasjon | ZV        | 25          |
| 602 | Testrapport       | ZV        |             |
| 603 | Test              | ZV        |             |



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen.

- Microsoft Office
- Microsoft Project
- VBS2
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop

## 4 Mål for iterasjon 1

---

I løpet av første iterasjon skal vi ha kartlagt mesteparten av teknologien vi har tenkt til å bruke, så vi vet hva vi skal gå videre på i neste iterasjon. Vi skal også produsere førsteutkast av de grunnleggende dokumentene.

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres:

- Forstudierapport
- Iterasjonsplan 1
- Iterasjonsrapport 1
- Iterasjonsplan 2
- Kravspesifikasjon
- Teststrategi
- Testspesifikasjon
- Risikodokument
- Prosjektplan

Følgende teknologidokumenter skal også produseres:

- VBS 2
- Application Scripting Interface
- Add-ons
- WxWidgets

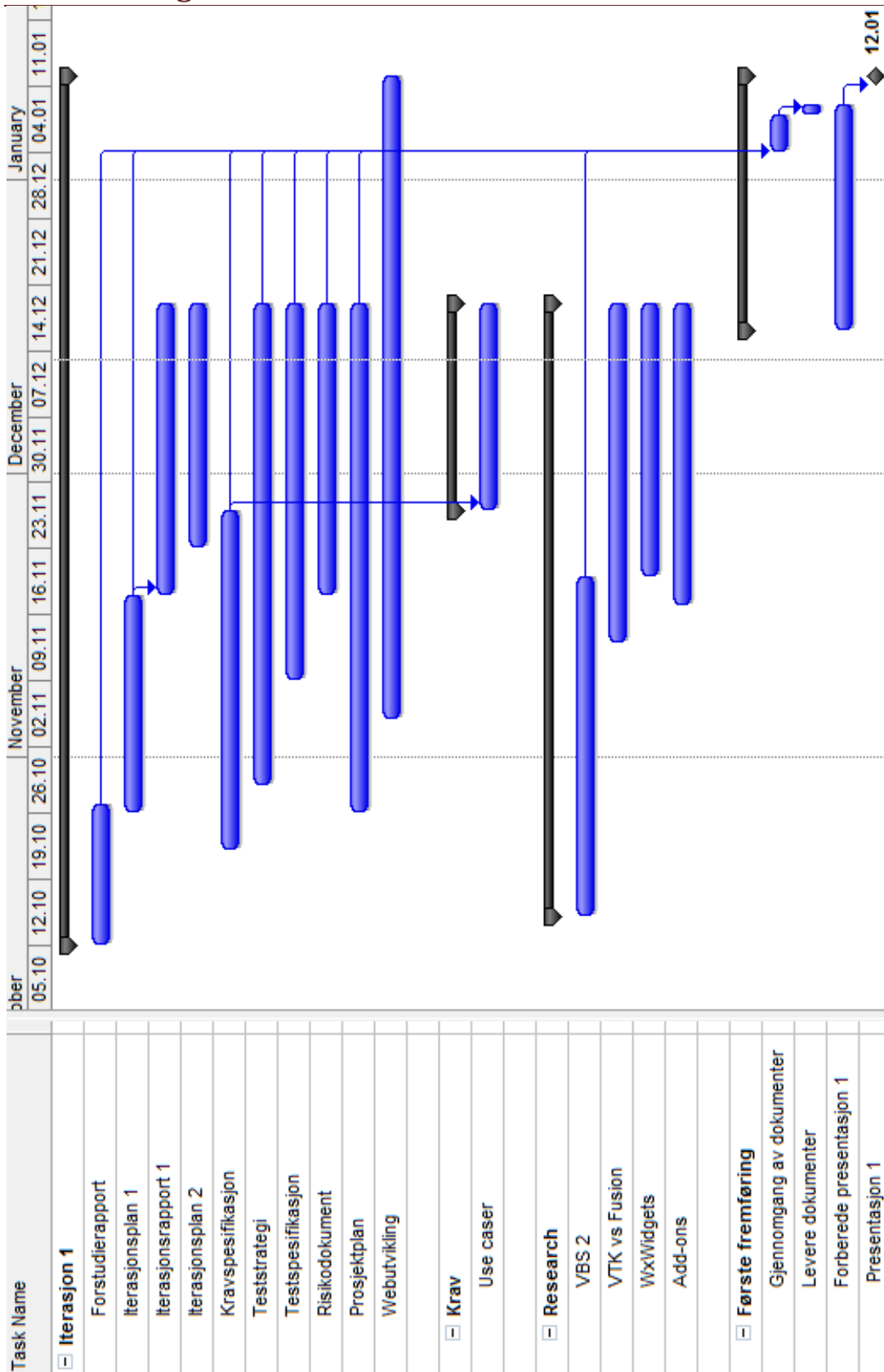
### 4.2 Primære mål

- Gjennomføre nok research til å kunne velge beste verktøy å jobbe videre med
- Skrive førsteutkast av alle dokumentene under punkt 4.1 Milepæler og sørge for at alle følger samme dokumentstandard
- Utarbeide flest mulig use caser og forsikre oss om at de er slik prosjektgiver vil ha de

### 4.3 Sekundære mål

- Få opp ett websystem for lagring av dokumenter på websiden
- Lage et system på websiden for å lagre timelister og automatgenerere rapporter

## 5 Gantt-diagram



# KDA RIVET



## Iterasjonsplan 2

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research .....                               | 4 |
| 2.3 | Krav/analyse .....                           | 4 |
| 2.4 | Design .....                                 | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test .....                                   | 5 |
| 2.7 | Tidsberegning .....                          | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 2.....                     | 7 |
| 4.1 | Milepæler .....                              | 7 |
| 4.2 | Primære mål .....                            | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 5   | Gantt-diagram .....                          | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimat på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem

**LK:** Lone Knutsen

**OMG:** Ole-Martin Grøtterud

**TT:** Thomas Transeth

**ZV:** Zoran Vukobratovic

**VBS:** Virtual Battle Space

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                        | Redaktør             |
|---------------|----------|----------------------------------|----------------------|
| 1.1           | 20.01.10 | Oppdatert                        | Ole-Martin Grøtterud |
| 1.2           | 11.03.10 | Oppdatert etter dokumentstandard | Lone Knutsen         |
| 1.99          | 23.03.10 | Korrekturlest                    | Lone Knutsen         |
| 2.0           | 31.05.10 | Klargjort for innlevering        | Thomas Transeth      |

## 1.4 Ansvarlig

Ole-Martin Grøtterud.

## 2 Iterasjonsplan

Andre iterasjon går fra dagen etter første presentasjon(13. januar), frem til 1.februar. Under er en liste over planlagte aktiviteter, estimert tidsbrukt og ansvarsområder for iterasjonen. På ansvarsområder der det står "Alle" er ansvaret delt på alle i gruppa.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 25          |
| 101 | Prosjektstyring  | TT        | 10          |
| 102 | Korrekturlesing  | Alle      | 0           |
| 103 | Prosjektplan     | TT        | 5           |
| 104 | Presentasjoner   | Alle      | 0           |
| 105 | Dokumentstandard | LK        | 0           |
| 106 | Økonomi          | ZV        | 2           |
| 107 | Diskusjon        | Alle      |             |
| 108 | Dokumentmaler    | LK        | 3           |
| 109 | Møtereferat      | Alle      | 0           |
| 110 | Møteinnkalling   | Alle      | 0           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 40          |
| 113 | Planlegging      | OMG       | 5           |
| 114 | Backup           | TT        | 0           |
| 115 | Kvalitetssikring | TT        | 0           |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 0           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer |           |             |
| 203 | Add-ons til VBS2                     | AKÅ       | 1           |
| 204 | VTK vs Fusion                        | ZV        | 1           |
| 205 | WxWidgets                            | LK        | 40          |

### 2.3 Krav/analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 0           |
| 301 | Use caser         | LK        | 0           |
| 302 | Kravspesifikasjon | LK        | 0           |
| 304 | Analyse           | OMG       | 30          |
| 305 | Forstudie         | TT        | 0           |
| 306 | Risikoanalyse     | OMG       | 2           |
| 307 | Brukermanual      | ZV        |             |

## 2.4 Design

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 400 | Klasser          | AKÅ       |             |
| 401 | Designdokument   | AKÅ       |             |
| 402 | Designdiagrammer | AKÅ       |             |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 5           |
| 501 | Prototype                    | TT        | 20          |
| 502 | Implementasjon               | TT        |             |
| 503 | Implementasjonsdokumentasjon | TT        |             |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 15          |
| 601 | Testspesifikasjon | ZV        | 15          |
| 602 | Testrapport       | ZV        |             |
| 603 | Test              | ZV        |             |
|     |                   |           |             |

**Totalt timeestimat: 227 timer**

## 2.7 Tidsberegning

Fra forrige iterasjon har vi tatt mye lærdom på tidsberegning. Vi bruker blant annet langt mindre tid på møter enn vi hadde trodd, og vi har nå bestemt en fast arbeidsmengde per uke. Iterasjon 2 består av 3 arbeidsuker hvor vi beregner 15 timer per person per uke. Det er da lagt inn omtrentlig 40 timer på hver enkeltes ansvarsområder, hvor de siste 5 timene som er beregnet går med til møter. Selvfølgelig arbeider ikke hver person kun med sitt eget ansvarsområde, men beregningen her er gjort med at alle hjelper alle like mye.



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen.

- Microsoft Office
- Microsoft Project
- VBS2
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 2

---

Andre iterasjon er første tredjedel i Elaboration fasen. I løpet av denne perioden skal vi ha analysert systemet og ha kommet i gang med en prototype av systemet. I tillegg skal vi forbedre websystemet betraktelig, og gå grundig igjennom wxWidgets.

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres:

- Iterasjonsplan 3
- Iterasjonsrapport 2
- Analysedokument
- Kodestandard

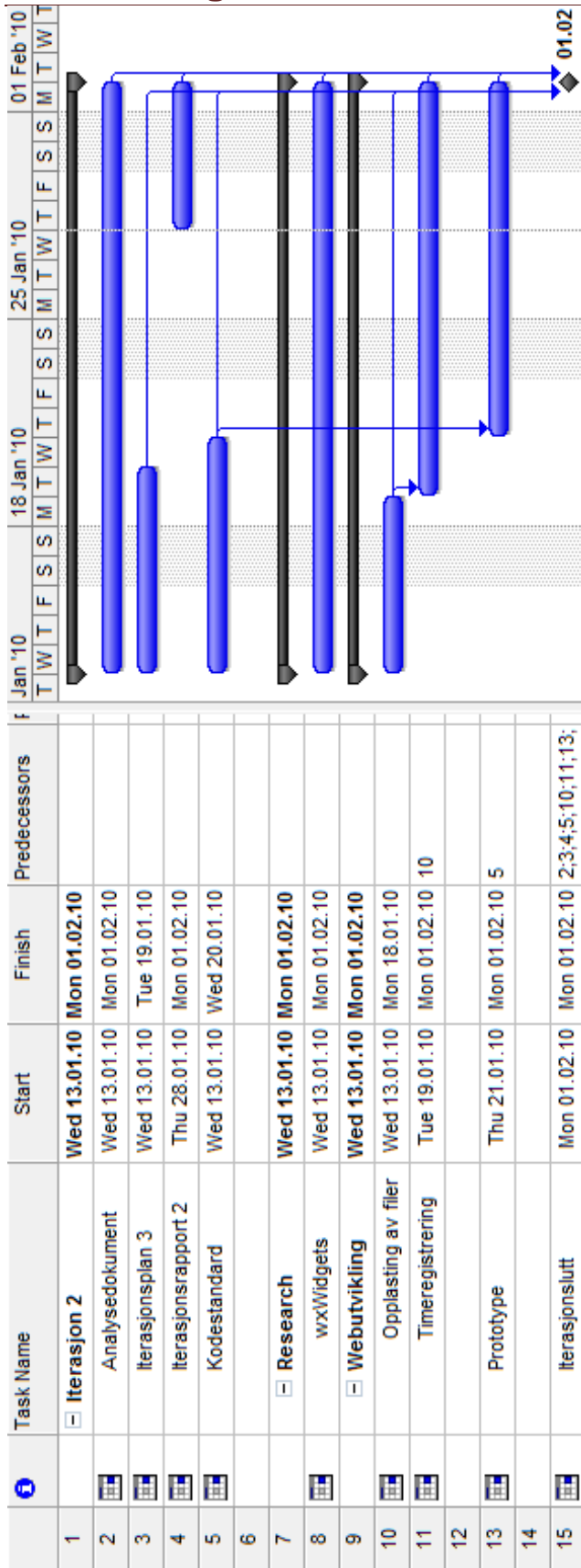
### 4.2 Primære mål

- Timeregistreringsfunksjonalitet på websiden.
- Skrive førsteutkast av alle dokumentene under punkt 4.1
- Utarbeide en testlog mal.
- Gjennomføre en grundig analyse av systemet og produsere kommunikasjon, klasse og sekvensdiagrammer for alle use casene, samt flere diagrammer hvis nødvendig.
- Få en prototype til å kommunisere enveis med VBS2 ved å sende en kommando.
- Sette opp Visual Studio med wxWidgets slik at det inneholder alle biblioteker og linker for å kompilere.

### 4.3 Sekundære mål

- Forbedre referanser i tidligere produserte dokumenter
- Gjøre lesning rundt designdokumentet
- Testing av prototypen

## 5 Gantt-diagram



# KDA RIVET



## Iterasjonsplan 3

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig.....                               | 3 |
| 2   | Iterasjonsplan 3.....                        | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research.....                                | 4 |
| 2.3 | Krav/Analyse.....                            | 4 |
| 2.4 | Design.....                                  | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test.....                                    | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 3.....                     | 7 |
| 4.1 | Milepæler.....                               | 7 |
| 4.2 | Primære mål.....                             | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 5   | Gantt-diagram.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimat på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem  
**LK:** Lone Knutsen  
**OMG:** Ole-Martin Grøtterud  
**TT:** Thomas Transeth  
**ZV:** Zoran Vukobratovic  
**VBS:** Virtual Battle Space

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør             |
|---------------|----------|---------------------------|----------------------|
| 0.1           | 17.01.10 | Førsteutkast              | Lone Knutsen         |
| 0.2           | 10.03.10 | Ny dokumentstandard       | Ole-Martin Grøtterud |
| 0.99          | 23.03.10 | Korrekturlest             | Ole-Martin Grøtterud |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth      |

## 1.4 Ansvarlig

Lone Knutsen

## 2 Iterasjonsplan 3

. Under er en liste over planlagte aktiviteter, estimert tidsbrukt og ansvarsområder for iterasjonen. På ansvarsområder der det står "Alle" går ansvaret på rundgang i gruppa. Denne iterasjonen går fra 2.feb til 22.feb.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 20          |
| 101 | Prosjektstyring  | TT        | 10          |
| 102 | Korrekturlesing  | Alle      | 5           |
| 103 | Prosjektplan     | TT        | 5           |
| 104 | Presentasjoner   | Alle      | 0           |
| 105 | Dokumentstandard | LK        | 2           |
| 106 | Økonomi          | ZV        | 1           |
| 107 | Diskusjon        | Alle      |             |
| 108 | Dokumentmaler    | LK        | 2           |
| 109 | Møtereferat      | Alle      | 2           |
| 110 | Møteinnkalling   | Alle      | 2           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 5           |
| 113 | Planlegging      | ZV,LK     | 10          |
| 114 | Backup           | Alle      | 3           |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 5           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer |           |             |
| 203 | Add-ons til VBS2                     | AKÅ       | 0           |
| 204 | VTK vs Fusion                        | ZV        | 0           |
| 205 | WxWidgets                            | LK,OMG    | 50          |

### 2.3 Krav/Analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 0           |
| 301 | Use case          | LK        | 0           |
| 302 | Kravspesifikasjon | LK        | 0           |
| 304 | Analyse           | OMG       | 5           |
| 305 | Forstudie         | TT        | 0           |
| 306 | Risikoanalyse     | OMG       | 5           |
| 307 | Brukermanual      | ZV        |             |
|     |                   |           |             |

## 2.4 Design

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 400 | Klasser          | AKÅ       | 0           |
| 401 | Designdokument   | AKÅ       | 35          |
| 402 | Designdiagrammer | AKÅ       | 0           |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 2           |
| 501 | Prototype                    | TT,ZV     | 45          |
| 502 | Implementasjon               | TT        | 0           |
| 503 | Implementasjonsdokumentasjon | TT        | 0           |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 0           |
| 601 | Testspesifikasjon | ZV        | 2           |
| 602 | Testrapport       | ZV        | 0           |
| 603 | Test              | ZV        | 9           |

**Totalt antall timer: 225 timer.**



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen:

- wxWidgets
- Microsoft Office
- Microsoft Project
- VBS2
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 3

---

I løpet av tredje iterasjon skal vi ha kommet godt i gang med tre prototyper og vi skal ha kommet i gang med GUI-utviklingen. I tillegg så skal web-siden bli ferdig og vi skal ha påbegynt testingen av prototypen.

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres:

- Testrapport av prototypene
- Prototypedokument for prototypene
- Iterasjonsplan 4
- Iterasjonsrapport 3
- Designdokument
- GUI-dokument

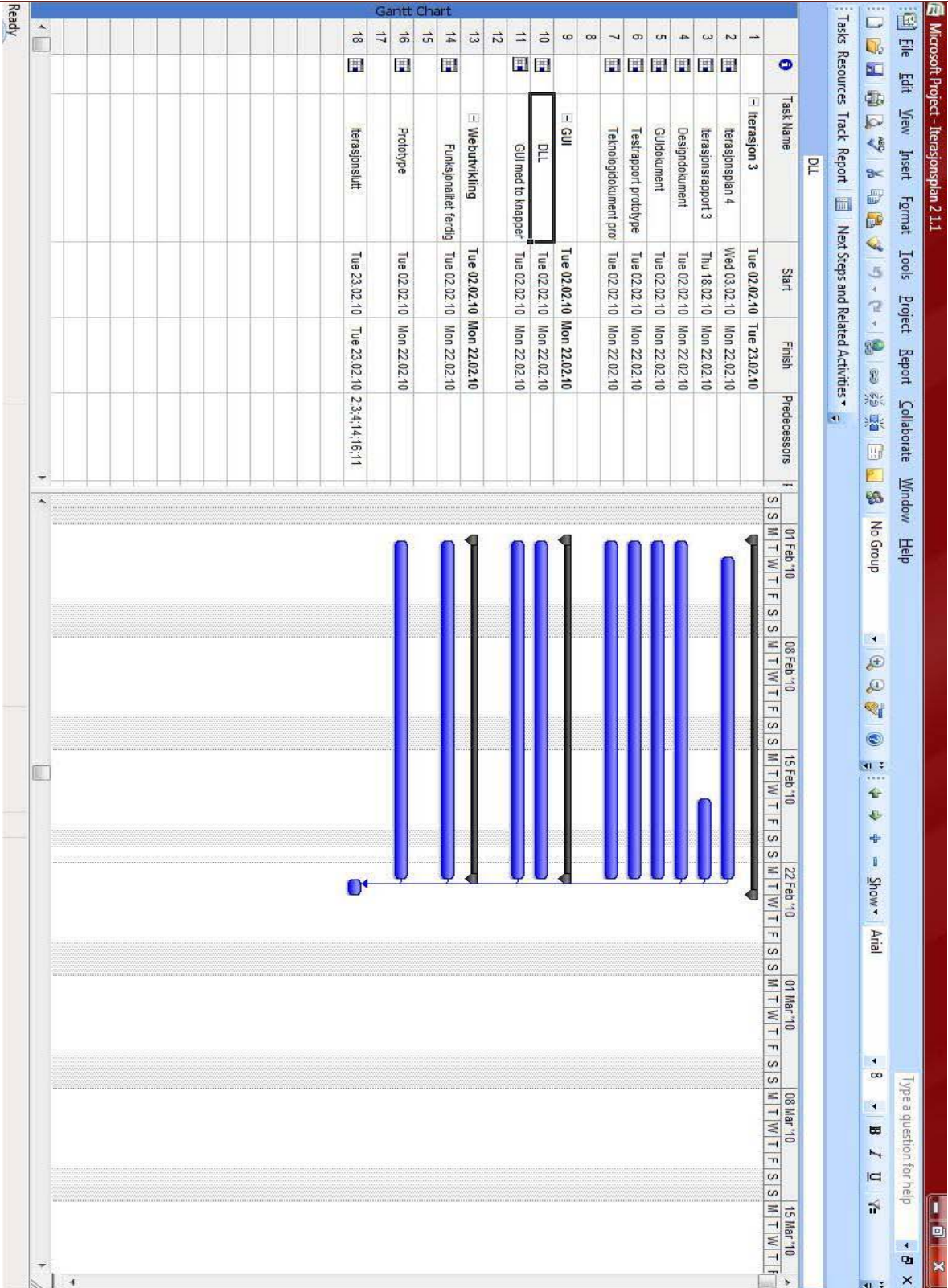
### 4.2 Primære mål

- Vi skal ha en GUI-prototype med to knapper, som igjen skal ha en funksjon hver.
- Dll-prototypen skal bruke flerveiskommunikasjon.
- Skrive førsteutkast av alle dokumentene under punkt 4.1 Milepæler og sørge for at alle følger samme dokumentstandard.

### 4.3 Sekundære mål

- Koble sammen GUI-prototype og Dll-prototype.

# 5 Gantt-diagram



# KDA RIVET



## Iterasjonsplan 4

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research .....                               | 4 |
| 2.3 | Krav/Analyse.....                            | 4 |
| 2.4 | Design .....                                 | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test .....                                   | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 4.....                     | 7 |
| 4.1 | Milepæler .....                              | 7 |
| 4.2 | Primære mål .....                            | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 5   | Gantt-diagram.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimat på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem

**LK:** Lone Knutsen

**OMG:** Ole-Martin Grøtterud

**TT :** Thomas Transeth

**ZV :** Zoran Vukobratovic

**VBS2 :** Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør             |
|---------------|----------|---------------------------|----------------------|
| 0.1           | 18.02.10 | Førsteutkast              | Zoran Vukobratovic   |
| 0.2           | 03.03.10 | Ny dokumentstandard       | Ole-Martin Grøtterud |
| 0.99          | 17.03.10 | korrekturlest             | Arne Kr              |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth      |

## 1.4 Ansvarlig

Zoran Vukobratovic

## 2 Iterasjonsplan

Under er en liste over planlagte aktiviteter, estimert tidsbrukt og ansvarsområder for iterasjonen. På ansvarsområder der det står "Alle" går ansvaret på rundgang i gruppa. Denne iterasjonen går fra 23.feb til 09.mar.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 15          |
| 101 | Prosjektstyring  | TT        | 15          |
| 102 | Korrekturlesing  | Alle      | 5           |
| 103 | Prosjektplan     | TT        | 5           |
| 104 | Presentasjoner   | Alle      | 10          |
| 105 | Dokumentstandard | LK        | 0           |
| 106 | Økonomi          | ZV        | 0           |
| 107 | Diskusjon        | Alle      | 0           |
| 108 | Dokumentmaler    | LK        | 0           |
| 109 | Møtereferat      | Alle      | 2           |
| 110 | Møteinnkalling   | Alle      | 2           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 15          |
| 113 | Planlegging      | ZV,LK     | 10          |
| 114 | Backup           | Alle      | 3           |
| 115 | Kvalitetssikring | TT        | 10          |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 0           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer | TT        | 0           |
| 203 | Add-ons til VBS2                     | AKÅ       | 0           |
| 204 | VTK vs Fusion                        | ZV        | 0           |
| 205 | WxWidgets                            | LK,OMG    | 0           |

### 2.3 Krav/Analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 0           |
| 301 | Use case          | LK        | 0           |
| 302 | Kravspesifikasjon | LK        | 1           |
| 304 | Analyse           | OMG       | 0           |
| 305 | Forstudie         | TT        | 0           |
| 306 | Risikoanalyse     | OMG       | 0           |
| 307 | Brukermanual      | ZV        | 0           |

## 2.4 Design

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 400 | Klasser          | AKÅ       | 0           |
| 401 | Designdokument   | AKÅ       | 5           |
| 402 | Designdiagrammer | AKÅ       | 5           |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 0           |
| 501 | Prototype - DLL              | TT,ZV     | 0           |
| 502 | Implementasjon               | TT        | 62          |
| 503 | Implementasjonsdokumentasjon | TT        | 15          |
| 505 | Prototype – wxWidgets        | LK,OMG    | 0           |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 0           |
| 601 | Testspesifikasjon | ZV        | 0           |
| 602 | Testrapport       | ZV        | 5           |
| 603 | Test              | ZV        | 15          |

**Totalt antall timer: 200 timer.**



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen:

- wxWidgets
- Microsoft Office
- Microsoft Project
- VBS2 VTK
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 4

---

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres:

- Iterasjonsrapport 4.
- Iterasjonsplan 5.
- Implementasjonsdokument

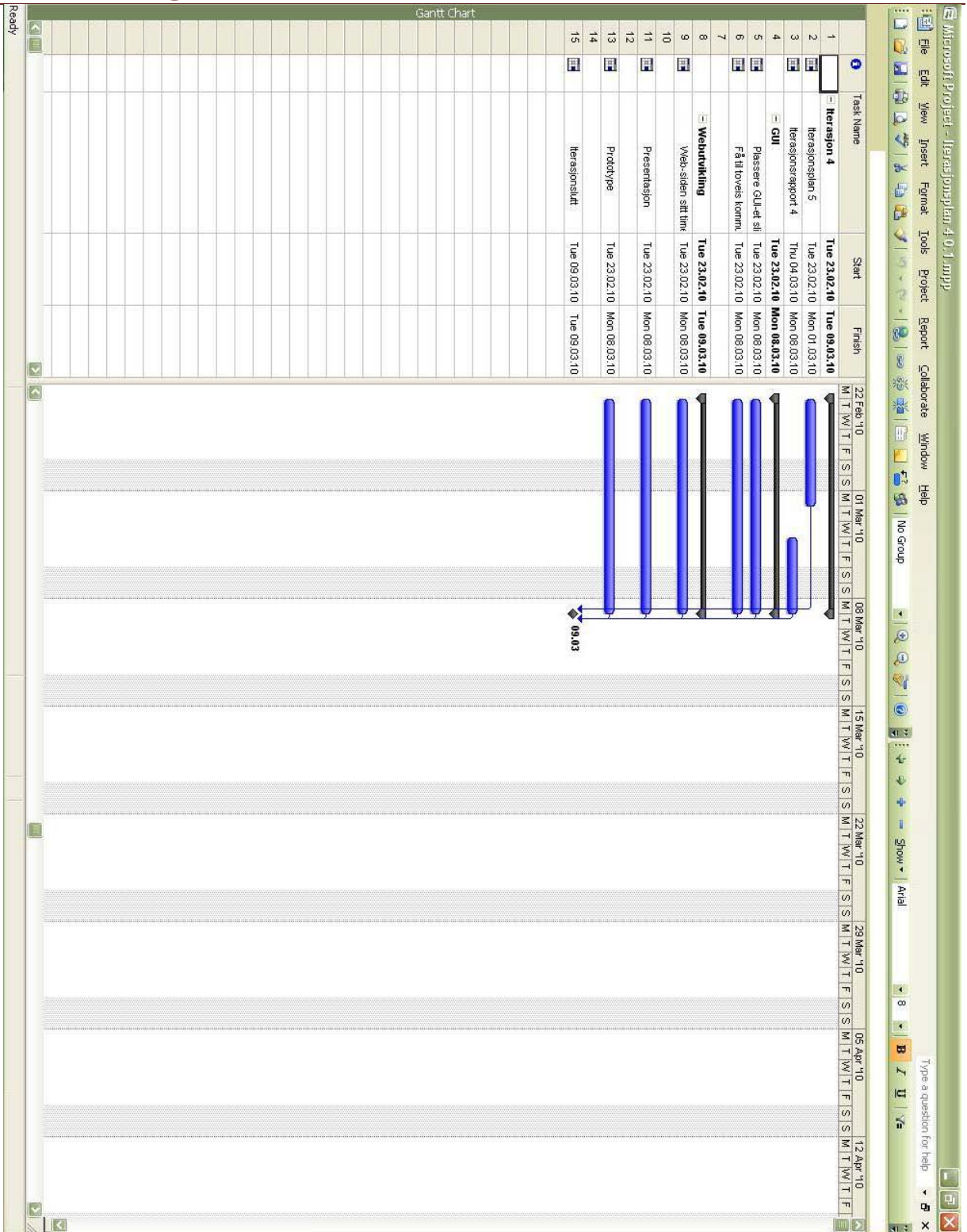
### 4.2 Primære mål

- Generere og gjøre klar framework.
- Lage minst 2 slider per emne til presentasjon 2.
- Web-siden sitt timelistesystem skal gjøres helt ferdig.
- Produsere Implementasjonsdokument med dokumentasjon om GUI-et og kommunikasjonen.

### 4.3 Sekundære mål

- Endre størrelsesforhold på synsfeltet til VBS2.
- Endre plassering på programvinduet til VBS2.
- Få til toveis kommunikasjon mellom GUI og VBS2.

# 5 Gantt-diagram



# KDA RIVET



## Iterasjonsplan 5

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research .....                               | 4 |
| 2.3 | Krav/Analyse.....                            | 4 |
| 2.4 | Design .....                                 | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test .....                                   | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 5.....                     | 7 |
| 4.1 | Milepæler .....                              | 7 |
| 4.2 | Primære mål .....                            | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 4.4 | Dokumenter.....                              | 7 |
| 5   | Gantt-diagram.....                           | 8 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimat på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem

**LK:** Lone Knutsen

**OMG:** Ole-Martin Grøtterud

**TT :** Thomas Transeth

**ZV :** Zoran Vukobratovic

**VBS2 :** Virtual Battle Space 2

**DLL:** Dynamic Link Library

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør             |
|---------------|----------|---------------------------|----------------------|
| 0.1           | 08.03.10 | Utforming                 | Thomas Transeth      |
| 0.2           | 10.03.10 | Ny dokumentstandard       | Ole-Martin Grøtterud |
| 0.99          | 17.03.10 | Korrekturlest             | Arne Kristian        |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth      |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Iterasjonsplan

Femte iterasjon er noe annerledes enn resten av construction-iterasjonene. Vi skal nå igjen presentere prosjektet vårt, men denne gangen med en mer teknisk vri og for å kunne vise dette må den stabile arkitekturen virkelig være på plass. Tiden framover vil derfor være fylt av mye dokumentskriving og korrekturlesing, men også en del teknisk arbeid slik vi kan vise dette fram både i presentasjonen og i dokumentene.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 20          |
| 101 | Prosjektstyring  | TT        | 15          |
| 102 | Korrekturlesing  | Alle      | 25          |
| 103 | Prosjektplan     | TT        | 5           |
| 104 | Presentasjoner   | Alle      | 80          |
| 105 | Dokumentstandard | LK        | 0           |
| 106 | Økonomi          | ZV        | 2           |
| 107 | Diskusjon        | Alle      | 0           |
| 108 | Dokumentmaler    | Alle      | 15          |
| 109 | Møtereferat      | Alle      | 5           |
| 110 | Møteinnkalling   | Alle      | 5           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 5           |
| 113 | Planlegging      | OMG       | 5           |
| 114 | Backup           | TT        | 3           |
| 115 | Kvalitetssikring | TT        | 10          |
| 116 | PR               | Alle      | 5           |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 0           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer | TT        | 0           |
| 203 | Add-ons til VBS2                     | AKÅ       | 0           |
| 204 | VTK vs Fusion                        | ZV        | 0           |
| 205 | WxWidgets                            | LK,OMG    | 0           |

### 2.3

#### Krav/Analyse

| Nr. | Aktivitet | Ansvarlig | Timeestimat |
|-----|-----------|-----------|-------------|
| 300 | Krav      | LK        | 0           |
| 301 | Use caser | LK        | 0           |

|     |                   |     |   |
|-----|-------------------|-----|---|
| 302 | Kravspesifikasjon | LK  | 0 |
| 304 | Analyse           | OMG | 0 |
| 305 | Forstudie         | TT  | 0 |
| 306 | Risikoanalyse     | OMG | 0 |
| 307 | Brukermanual      | ZV  | 0 |

## 2.4 Design

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 400 | Klasser          | AKÅ       | 0           |
| 401 | Designdokument   | AKÅ       | 0           |
| 402 | Designdiagrammer | AKÅ       | 0           |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 5           |
| 501 | Prototype - DLL              | TT,ZV     | 0           |
| 502 | Implementasjon               | TT        | 40          |
| 503 | Implementasjonsdokumentasjon | TT        | 30          |
| 505 | Prototype – wxWidgets        | LK,OMG    | 0           |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 0           |
| 601 | Testspesifikasjon | ZV        | 0           |
| 602 | Testrapport       | ZV        | 5           |
| 603 | Test              | ZV        | 20          |
| 604 | Kodeinspeksjon    | ZV        | 5           |

**Totalt antall timer: 300 timer.**



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen:

- wxWidgets
- Microsoft Office
- Microsoft Project
- VBS2 VTK
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 5

---

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres/klargjøres:

- Iterasjonsrapport 5.
- Iterasjonsplan 6.
- Presentasjon 2

### 4.2 Primære mål

- Klargjøre og øve til presentasjon 2.
- Lage ny dokumentstandard og oppdatere alle dokumenter til denne.
- Klargjøre alle dokumenter til innlevering.
- Implementere joystick-funksjon for å kunne frigjøre mus til overlay.
- Lage et stabilt rammeverk til implementasjonen.
- Sende kommandoer mellom overlay og VBS2 via samme kommunikasjonskanal, men med ulike resultater.

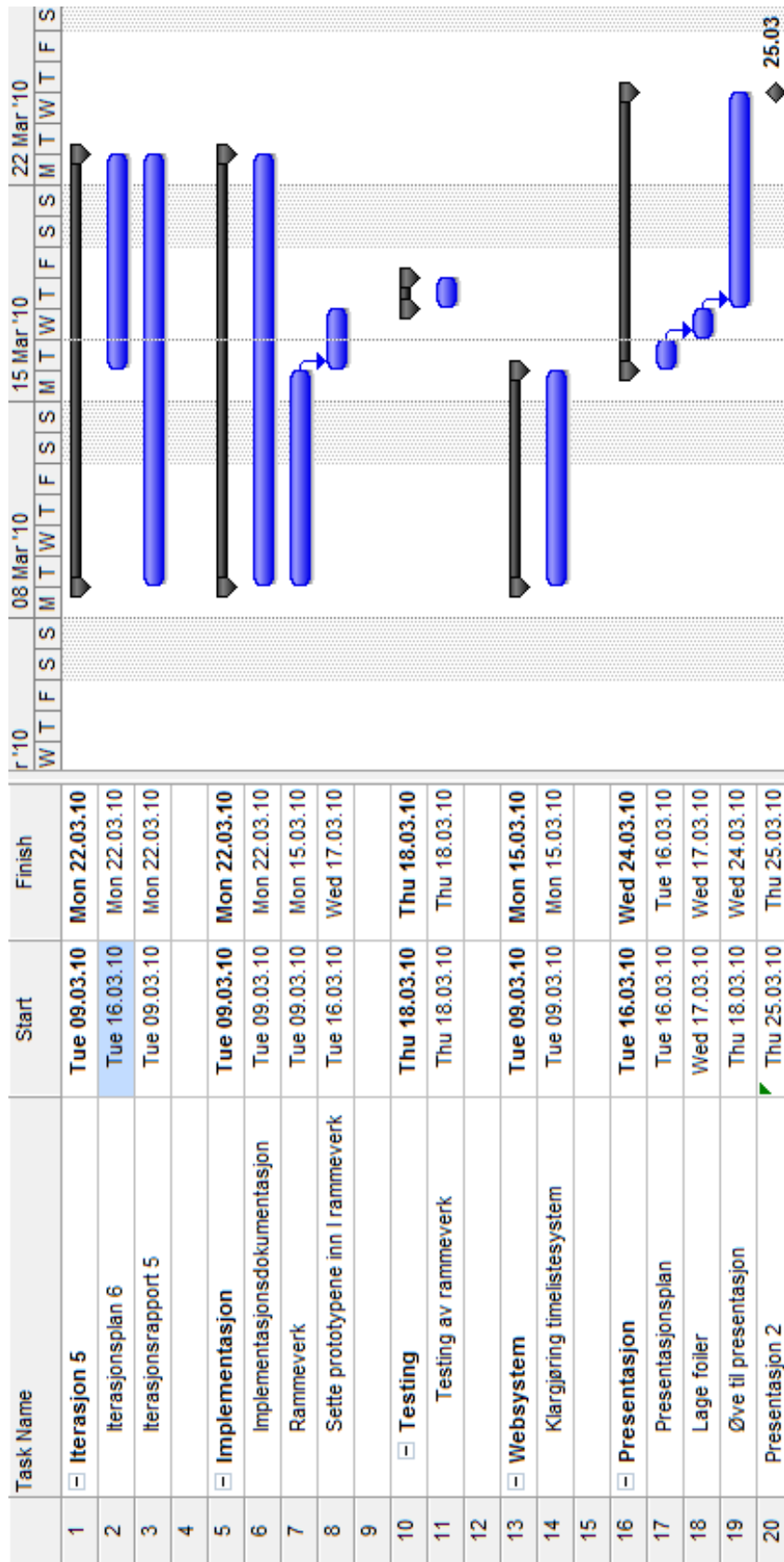
### 4.3 Sekundære mål

- Endre størrelsesforhold på synsfeltet til VBS2.
- Sette opp kode som gjør uavhengige inkluderingsfiler til .dll for å lettere gjøre systemet skalerbart.

### 4.4 Dokumenter

- Prosjektplan.
- Analysedokument.
- Designdokument.
- Implementasjonsdokumentasjon.
- Kodestandard.
- Teknologidokument – dll.
- Teknologidokument – wxWidgets.
- Testrapporter.

## 5 Gantt-diagram



# KDA RIVET



## Iterasjonsplan 6

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Åmellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig.....                               | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research.....                                | 4 |
| 2.3 | Krav/Analyse.....                            | 5 |
| 2.4 | Design.....                                  | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test.....                                    | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 6.....                     | 7 |
| 4.1 | Milepæler.....                               | 7 |
| 4.2 | Primære mål.....                             | 7 |
|     | Sekundære mål.....                           | 7 |
| 4.3 | Dokumenter.....                              | 7 |
| 5   | Gantt-diagram.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimater på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem

**LK:** Lone Knutsen

**OMG:** Ole-Martin Grøtterud

**TT:** Thomas Transeth

**ZV:** Zoran Vukobratovic

**VBS2:** Virtual Battle Space 2

**DLL:** Dynamic Link Library

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 12.04.10 | Utforming                 | Arne Kr. Åmellem   |
| 0.2           | 23.04.10 | Dokumentstandard          | Lone Knutsen       |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Iterasjonsplan

Dette er den sjettede iterasjon og er en av de fem iterasjonene i løpet av iterasjonen med en planlagt ukentlig arbeidsbelastning per person på 35 timer, summert til 175 timer pr uke på hele gruppen. Siden store deler av denne iterasjonen gikk bort til påskeferie og eksamener, så har vi beregnet en total på ca 175, dvs 1 uke. Under har vi satt opp en tabell som viser de ulike aktivitetene knyttet til iterasjonen. Hvor mange timer som skal brukes for å fullføre aktiviteten er estimert i en egen kolonne. Hvis ansvarsperson er angitt er denne personen hovedansvarlig for å etterse at aktiviteten er fullført i henhold til de angitte rammene. Hvis det ikke er angitt noen ansvarsperson betyr det at ansvaret for denne aktiviteten roterer i gruppa. Resurser angir hvilke personer som skal jobbe med aktiviteten.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 10          |
| 101 | Prosjektstyring  | TT        | 2           |
| 102 | Korrekturlesing  | Alle      | 0           |
| 103 | Prosjektplan     | TT        | 5           |
| 104 | Presentasjoner   | Alle      | 0           |
| 105 | Dokumentstandard | LK        | 0           |
| 106 | Økonomi          | ZV        | 0           |
| 107 | Diskusjon        | Alle      | 0           |
| 108 | Dokumentmaler    | Alle      | 0           |
| 109 | Møtereferat      | Alle      | 1           |
| 110 | Møteinnkalling   | Alle      | 1           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 0           |
| 113 | Planlegging      | OMG       | 5           |
| 114 | Backup           | TT        | 2           |
| 115 | Kvalitetssikring | TT        | 5           |
| 116 | PR               | Alle      | 0           |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 0           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer | TT        | 0           |
| 203 | Add-ons til VBS2                     | AKÅ       | 0           |
| 204 | VTK vs Fusion                        | ZV        | 0           |
| 205 | WxWidgets                            | LK,OMG    | 0           |

## 2.3 Krav/Analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 0           |
| 301 | Use case          | LK        | 0           |
| 302 | Kravspesifikasjon | LK        | 0           |
| 304 | Analyse           | OMG       | 0           |
| 305 | Forstudie         | TT        | 0           |
| 306 | Risikoanalyse     | OMG       | 0           |
| 307 | Brukermanual      | ZV        | 3           |

## 2.4 Design

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 400 | Klasser           | AKÅ       | 1           |
| 401 | Design dokument   | AKÅ       | 5           |
| 402 | Design diagrammer | AKÅ       | 5           |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 5           |
| 501 | Prototype - DLL              | TT,ZV     | 0           |
| 502 | Implementasjon               | TT        | 80          |
| 503 | Implementasjonsdokumentasjon | TT        | 10          |
| 504 | Testing av teknologi         | Alle      | 15          |
| 505 | Prototype – wxWidgets        | LK,OMG    | 0           |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 0           |
| 601 | Testspesifikasjon | ZV        | 0           |
| 602 | Testrapport       | ZV        | 5           |
| 603 | Test              | ZV        | 10          |
| 604 | Kodeinspeksjon    | ZV        | 5           |

**Totalt antall timer: 175 timer.**



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen:

- wxWidgets
- Microsoft Office
- Microsoft Project
- VBS2 VTK
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 6

---

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres/klargjøres:

- Iterasjonsrapport 6.
- Iterasjonsplan 7.

### 4.2 Primære mål

- Oppdatere design med hensiktsmessige patterns.
- Implementere og dokumentere patterns som ble funnet i punktet over.

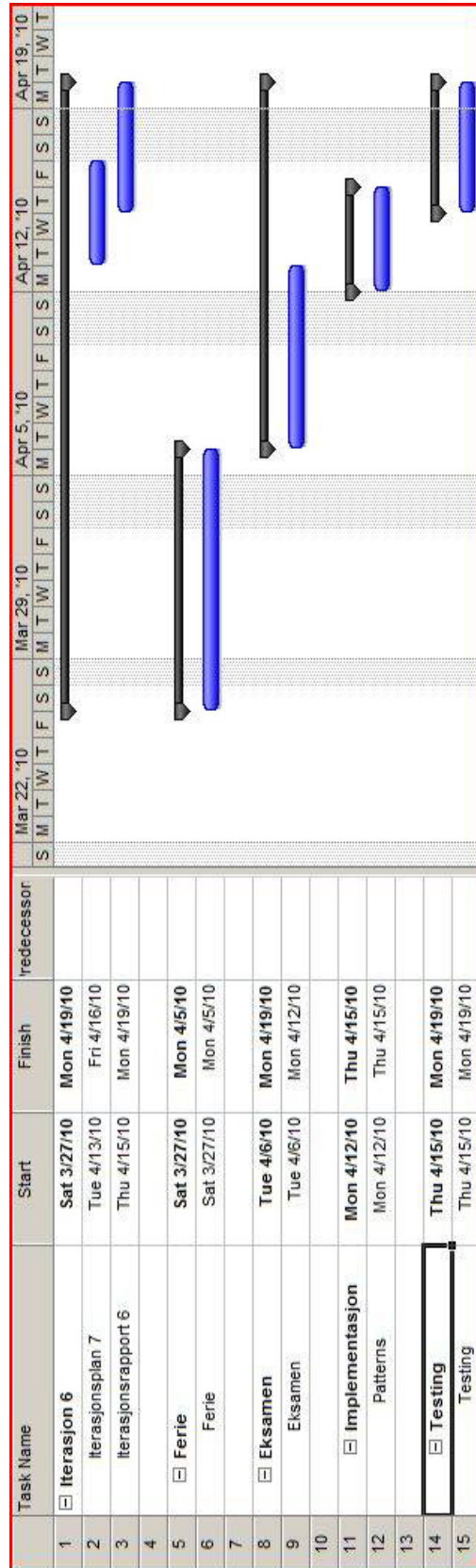
#### Sekundære mål

- Få Joysticken til å skyte og styre kamera i VBS2.
- Armeringskapp funksjonalitet

### 4.3 Dokumenter

- Prosjektplan.
- Designdokument.
- Implementasjonsdokumentasjon.
- Testrapporter.

## 5 Gantt-diagram



# KDA RIVET



## Iterasjonsplan 7

---

Lone Knutsen

---

Arne Kristian Åmellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig.....                               | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research.....                                | 4 |
| 2.3 | Krav/Analyse.....                            | 5 |
| 2.4 | Design.....                                  | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test.....                                    | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 7.....                     | 7 |
| 4.1 | Milepæler.....                               | 7 |
| 4.2 | Primære mål.....                             | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 4.4 | Prosjektdokumenter som oppdateres.....       | 7 |
| 5   | Gantt-diagram.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimat på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem

**LK:** Lone Knutsen

**OMG:** Ole-Martin Grøtterud

**TT :** Thomas Transeth

**ZV :** Zoran Vukobratovic

**VBS2 :** Virtual Battle Space 2

**DLL:** Dynamic Link Library

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør             |
|---------------|----------|---------------------------|----------------------|
| 0.1           | 12.04.10 | Utforming                 | Ole-Martin Grøtterud |
| 0.2           | 23.04.10 | Dokumentstandard          | Lone Knutsen         |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic   |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth      |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Iterasjonsplan

Dette er den sjuende iterasjonen og er en av de fem iterasjonene i construction. Den har en varighet på to uker og går fra 20.April til 3.Mai . I løpet av iterasjonen med en planlagt ukentlig arbeidsbelastning per person på 35 timer, summert til 175 timer pr uke på hele gruppen og 350 timer til sammen. Under har vi satt opp en tabell som viser de ulike aktivitetene knyttet til iterasjonen. Hvor mange timer som skal brukes for å fullføre aktiviteten er estimert i en egen kolonne. Hvis ansvarsperson er angitt er denne personen hovedansvarlig for å etterse at aktiviteten er fullført i henhold til de angitte rammene. Hvis det ikke er angitt noen ansvarsperson betyr det at ansvaret for denne aktiviteten roterer i gruppa. Resurser angir hvilke personer som skal jobbe med aktiviteten.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 10          |
| 101 | Prosjektstyring  | TT        | 15          |
| 102 | Korrekturlesing  | Alle      | 0           |
| 103 | Prosjektplan     | TT        | 10          |
| 104 | Presentasjoner   | Alle      | 0           |
| 105 | Dokumentstandard | LK        | 0           |
| 106 | Økonomi          | ZV        | 0           |
| 107 | Diskusjon        | Alle      | 0           |
| 108 | Dokumentmaler    | Alle      | 0           |
| 109 | Møtereferat      | Alle      | 2           |
| 110 | Møteinnkalling   | Alle      | 0           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 10          |
| 113 | Planlegging      | OMG       | 7           |
| 114 | Backup           | TT        | 1           |
| 115 | Kvalitetssikring | TT        | 5           |
| 116 | PR               | Alle      | 0           |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 0           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer | TT        | 0           |
| 203 | Add-ons til VBS2                     | AKÅ       | 0           |
| 204 | VTK vs Fusion                        | ZV        | 0           |
| 205 | WxWidgets                            | LK,OMG    | 0           |

## 2.3 Krav/Analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 5           |
| 301 | Use case          | LK        | 5           |
| 302 | Kravspesifikasjon | LK        | 3           |
| 304 | Analyse           | OMG       | 5           |
| 305 | Forstudie         | TT        | 0           |
| 306 | Risikoanalyse     | OMG       | 2           |
| 307 | Brukermanual      | ZV        | 10          |

## 2.4 Design

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 400 | Klasser           | AKÅ       | 3           |
| 401 | Design dokument   | AKÅ       | 3           |
| 402 | Design diagrammer | AKÅ       | 4           |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 10          |
| 501 | Prototype - DLL              | TT,ZV     | 0           |
| 502 | Implementasjon               | TT        | 90          |
| 503 | Implementasjonsdokumentasjon | TT        | 30          |
| 504 | Testing av teknologi         | TT        | 50          |
| 505 | Prototype – wxWidgets        | LK,OMG    | 0           |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 0           |
| 601 | Testspesifikasjon | ZV        | 0           |
| 602 | Testrapport       | ZV        | 20          |
| 603 | Test              | ZV        | 40          |
| 604 | Kodeinspeksjon    | ZV        | 10          |

**Totalt antall timer: 350 timer.**



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen:

- wxWidgets
- Microsoft Office
- Microsoft Project
- VBS2 VTK
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 7

---

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres/klargjøres:

- Iterasjonsrapport 7.
- Iterasjonsplan 8.

### 4.2 Primære mål

- Følgende funksjonalitet i implementasjonen skal være til stede:
  - Bemanne systemet.
  - Sikte: elevasjon og rotasjon.
  - Armering på/av.
  - Skyte.
- Funksjonaliteten over skal være testet
- Alle patterns skal være implementert og dokumentert

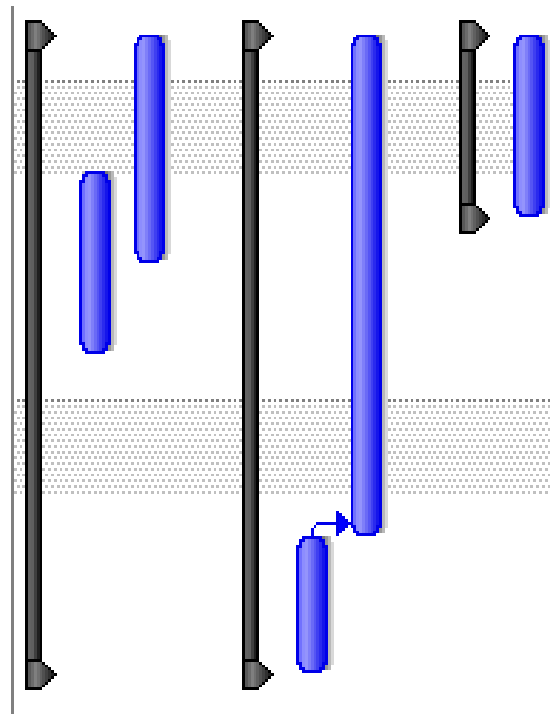
### 4.3 Sekundære mål

- Følgende funksjonalitet i implementasjonen skal være til stede:
  - Velge skuddhastighet.
  - Sikte: simulere servoene.
  - Zoome.
- Funksjonaliteten over skal være testet

### 4.4 Prosjektdokumenter som oppdateres

- Prosjektplan.
- Designdokument.
- Implementasjonsdokumentasjon.
- Kodestandard.
- Testrapporter.

## 5 Gantt-diagram



|    |                     |              |              |
|----|---------------------|--------------|--------------|
| 1  | [-] Iterasjon 7     | Tue 20.04.10 | Mon 03.05.10 |
| 2  | Iterasjonsplan 8    | Tue 27.04.10 | Fri 30.04.10 |
| 3  | Iterasjonsrapport 7 | Thu 29.04.10 | Mon 03.05.10 |
| 4  |                     |              |              |
| 5  | [-] Implementasjon  | Tue 20.04.10 | Mon 03.05.10 |
| 6  | Patterns            | Tue 20.04.10 | Thu 22.04.10 |
| 7  | A-Krav              | Fri 23.04.10 | Mon 03.05.10 |
| 8  |                     |              |              |
| 9  | [-] Testing         | Fri 30.04.10 | Mon 03.05.10 |
| 10 | Testing             | Fri 30.04.10 | Mon 03.05.10 |

# KDA RIVET



## Iterasjonsplan 8

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig.....                               | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| 2.2 | Research.....                                | 4 |
| 2.3 | Krav/Analyse.....                            | 5 |
| 2.4 | Design.....                                  | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test.....                                    | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 7.....                     | 7 |
| 4.1 | Milepæler.....                               | 7 |
| 4.2 | Primære mål.....                             | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 4.4 | Prosjektdokumenter som oppdateres.....       | 7 |
| 5   | Gantt-diagram.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimater på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem

**LK:** Lone Knutsen

**OMG:** Ole-Martin Grøtterud

**TT:** Thomas Transeth

**ZV:** Zoran Vukobratovic

**VBS2:** Virtual Battle Space 2

**DLL:** Dynamic Link Library

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 03.05.10 | Utforming                 | Thomas Transeth    |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Iterasjonsplan

Dette er den åttende iterasjonen i prosjektet og den siste i construction-fasen. Det er satt av to uker (04.05 til 17.05) til denne perioden og er ment til klargjøring av systemet og siste bug-sjekker. Siden vi har gode resultater så langt vil det også bli introdusert nye krav.

I løpet av iterasjonen med en planlagt ukentlig arbeidsbelastning per person på 35 timer, summert til 175 timer pr uke på hele gruppen og 350 timer til sammen. Under har vi satt opp en tabell som viser de ulike aktivitetene knyttet til iterasjonen. Hvor mange timer som skal brukes for å fullføre aktiviteten er estimert i en egen kolonne. Hvis ansvarsperson er angitt er denne personen hovedansvarlig for å etterse at aktiviteten er fullført i henhold til de angitte rammene. Hvis det ikke er angitt noen ansvarsperson betyr det at ansvaret for denne aktiviteten roterer i gruppa. Resurser angir hvilke personer som skal jobbe med aktiviteten.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 10          |
| 101 | Prosjektstyring  | TT        | 15          |
| 102 | Korrekturlesing  | Alle      | 0           |
| 103 | Prosjektplan     | TT        | 10          |
| 104 | Presentasjoner   | Alle      | 10          |
| 105 | Dokumentstandard | LK        | 0           |
| 106 | Økonomi          | ZV        | 0           |
| 107 | Diskusjon        | Alle      | 0           |
| 108 | Dokumentmaler    | Alle      | 0           |
| 109 | Møtereferat      | Alle      | 2           |
| 110 | Møteinnkalling   | Alle      | 0           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 0           |
| 113 | Planlegging      | OMG       | 10          |
| 114 | Backup           | TT        | 1           |
| 115 | Kvalitetssikring | TT        | 5           |
| 116 | PR               | Alle      | 0           |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 0           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer | TT        | 0           |
| 203 | Add-ons til VBS2                     | AKÅ       | 0           |
| 204 | VTK vs Fusion                        | ZV        | 0           |
| 205 | WxWidgets                            | LK,OMG    | 5           |

## 2.3 Krav/Analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 5           |
| 301 | Use case          | LK        | 5           |
| 302 | Kravspesifikasjon | LK        | 3           |
| 304 | Analyse           | OMG       | 5           |
| 305 | Forstudie         | TT        | 0           |
| 306 | Risikoanalyse     | OMG       | 2           |
| 307 | Brukermanual      | ZV        | 18          |

## 2.4 Design

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 400 | Klasser           | AKÅ       | 3           |
| 401 | Design dokument   | AKÅ       | 3           |
| 402 | Design diagrammer | AKÅ       | 3           |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 0           |
| 501 | Prototype - DLL              | TT,ZV     | 0           |
| 502 | Implementasjon               | TT        | 110         |
| 503 | Implementasjonsdokumentasjon | TT        | 30          |
| 504 | Testing av teknologi         | TT        | 20          |
| 505 | Prototype – wxWidgets        | LK,OMG    | 0           |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 0           |
| 601 | Testspesifikasjon | ZV        | 0           |
| 602 | Testrapport       | ZV        | 20          |
| 603 | Test              | ZV        | 40          |
| 604 | Kodeinspeksjon    | ZV        | 15          |

**Totalt antall timer: 350 timer.**



### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen:

- wxWidgets
- Microsoft Office
- Microsoft Project
- VBS2 VTK
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 7

---

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres/klargjøres:

- Iterasjonsrapport 8.
- Iterasjonsplan 9.

### 4.2 Primære mål

- Alle A, B og C krav skal tilfredsstille de satte kriteriene før de nye kravene innføres.
- Alle funksjoner skal testes.
- All dokumentasjon skal klargjøres.

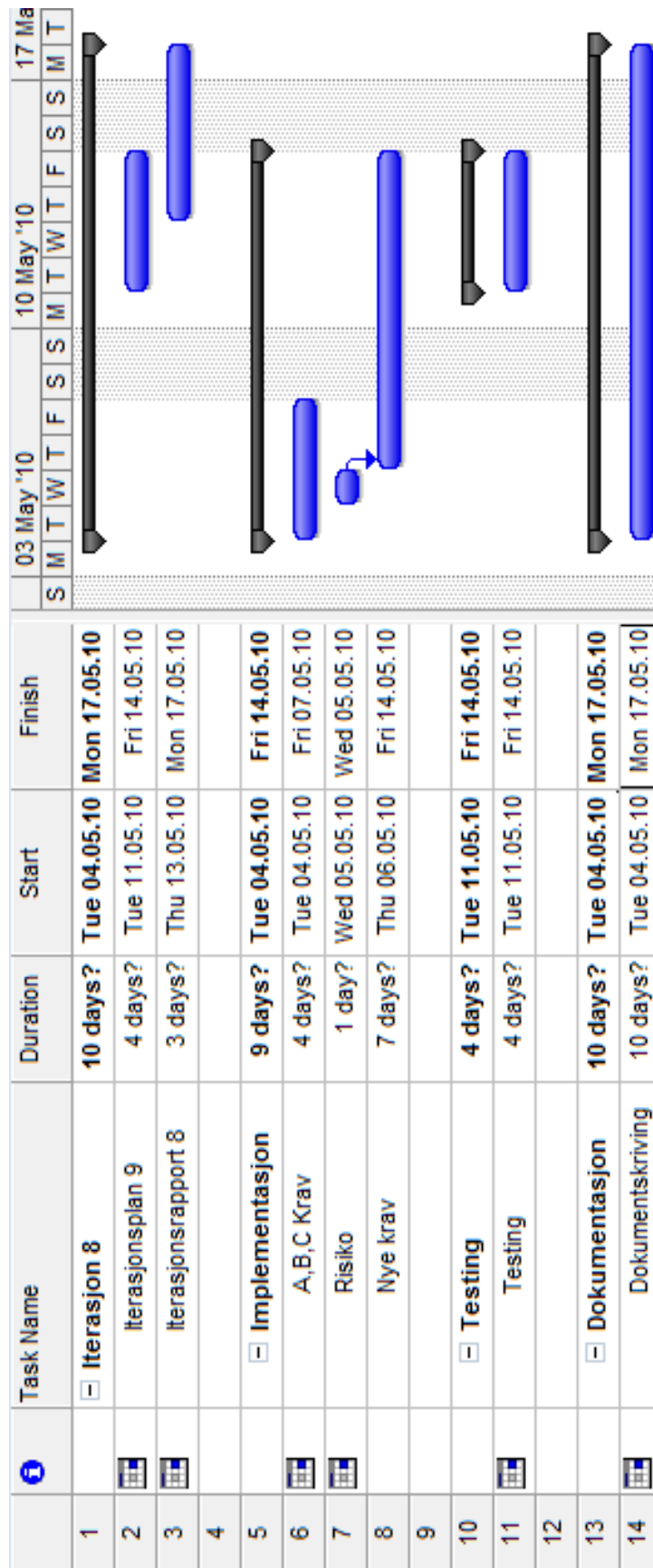
### 4.3 Sekundære mål

- Legge til følgende funksjonalitet med tilhørende prioritet.
  - Brukeren skal kunne skifte "cross hair".
  - Implementere "face of direction".
  - Lage en navigerbar meny med riktige menyfelter.

### 4.4 Prosjektdokumenter som oppdateres

- Prosjektplan.
- Designdokument.
- Implementasjonsdokumentasjon.
- Kodestandard.
- Testrapporter.
- Kravdokument.
- Risikodokument.

## 5 Gantt-diagram





# KDA RIVET



## Iterasjonsplan 9

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig.....                               | 3 |
| 2   | Iterasjonsplan.....                          | 4 |
| 2.1 | Administrasjon og prosjektstyring.....       | 4 |
| +   | <b>Error! Bookmark not defined.</b>          |   |
| 2.2 | Research.....                                | 4 |
| 2.3 | Krav/Analyse.....                            | 5 |
| 2.4 | Design.....                                  | 5 |
| 2.5 | Implementasjon.....                          | 5 |
| 2.6 | Test.....                                    | 5 |
| 3   | Verktøy.....                                 | 6 |
| 4   | Mål for iterasjon 9.....                     | 7 |
| 4.1 | Milepæler.....                               | 7 |
| 4.2 | Primære mål.....                             | 7 |
| 4.3 | Sekundære mål.....                           | 7 |
| 4.4 | Prosjektdokumenter som oppdateres.....       | 7 |
| 5   | Gantt-diagram.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hva som skal gjøres i løpet av iterasjonen, samt gi en kort forklaring av alt som skal gjøres. Den skal også ha med tidsestimat på alle aktivitetene og ansvarspersoner for disse, noe som vil hjelpe oss framover mot neste milepæl.

## 1.2 Definisjoner, akronymer og forkortelser

**AKÅ:** Arne Kristian Åmellem

**LK:** Lone Knutsen

**OMG:** Ole-Martin Grøtterud

**TT :** Thomas Transeth

**ZV :** Zoran Vukobratovic

**VBS2 :** Virtual Battle Space 2

**DLL:** Dynamic Link Library

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 18.05.10 | Utforming                 | Arne Kristian      |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Iterasjonsplan

Det er den niende iterasjonen i prosjektet og den eneste i Transition-fasen. Det er satt av 2 uker (18.05 til 31.05) til denne perioden og er ment til og ferdig stille prosjektet for levering til kunden. Siden vi har gjennomført alle A, B og C kravene gjenstår bare bug-hunting og dokument skriving som gjenstår. I løpet av iterasjonen med en planlagt ukentlig arbeidsbelastning per person på 35 timer, summert til 175 timer pr uke på hele gruppen og 350 timer til sammen. Under har vi satt opp en tabell som viser de ulike aktivitetene knyttet til iterasjonen. Hvor mange timer som skal brukes for å fullføre aktiviteten er estimert i en egen kolonne. Hvis ansvarsperson er angitt er denne personen hovedansvarlig for å etterse at aktiviteten er fullført i henhold til de angitte rammene. Hvis det ikke er angitt noen ansvarsperson betyr det at ansvaret for denne aktiviteten roterer i gruppa. Ressurser angir hvilke personer som skal jobbe med aktiviteten.

### 2.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Ansvarlig | Timeestimat |
|-----|------------------|-----------|-------------|
| 100 | Møter            | Alle      | 10          |
| 101 | Prosjektstyring  | TT        | 15          |
| 102 | Korrekturlesing  | Alle      | 56          |
| 103 | Prosjektplan     | TT        | 10          |
| 104 | Presentasjoner   | Alle      | 35          |
| 105 | Dokumentstandard | LK        | 0           |
| 106 | Økonomi          | ZV        | 0           |
| 107 | Diskusjon        | Alle      | 0           |
| 108 | Dokumentmaler    | Alle      | 0           |
| 109 | Møtereferat      | Alle      | 2           |
| 110 | Møteinnkalling   | Alle      | 2           |
| 111 | Bedriftsmøte     | Alle      | 0           |
| 112 | Webutvikling     | AKÅ       | 10          |
| 113 | Planlegging      | OMG       | 10          |
| 114 | Backup           | TT        | 1           |
| 115 | Kvalitetssikring | TT        | 5           |
| 116 | PR               | Alle      | 5           |

### 2.2 Research

| Nr. | Aktivitet                            | Ansvarlig | Timeestimat |
|-----|--------------------------------------|-----------|-------------|
| 200 | VBS2 funksjoner og bruk              | OMG       | 0           |
| 201 | Trainer-funksjoner                   | TT        | 0           |
| 202 | Interface mellom joystick og trainer | TT        | 0           |
| 203 | Add-ons til VBS2                     | AKÅ       | 0           |
| 204 | VTK vs Fusion                        | ZV        | 0           |
| 205 | WxWidgets                            | LK,OMG    | 0           |



## 2.3 Krav/Analyse

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 300 | Krav              | LK        | 0           |
| 301 | Use case          | LK        | 0           |
| 302 | Kravspesifikasjon | LK        | 0           |
| 304 | Analyse           | OMG       | 0           |
| 305 | Forstudie         | TT        | 0           |
| 306 | Risikoanalyse     | OMG       | 0           |
| 307 | Brukermanual      | ZV        | 5           |

## 2.4 Design

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 400 | Klasser           | AKÅ       | 0           |
| 401 | Design dokument   | AKÅ       | 6           |
| 402 | Design diagrammer | AKÅ       | 3           |

## 2.5 Implementasjon

| Nr. | Aktivitet                    | Ansvarlig | Timeestimat |
|-----|------------------------------|-----------|-------------|
| 500 | Kodestandard                 | TT        | 0           |
| 501 | Prototype - DLL              | TT,ZV     | 0           |
| 502 | Implementasjon               | TT        | 80          |
| 503 | Implementasjonsdokumentasjon | TT        | 30          |
| 504 | Testing av teknologi         | TT        | 10          |
| 505 | Prototype – wxWidgets        | LK,OMG    | 0           |

## 2.6 Test

| Nr. | Aktivitet         | Ansvarlig | Timeestimat |
|-----|-------------------|-----------|-------------|
| 600 | Teststrategier    | ZV        | 0           |
| 601 | Testspesifikasjon | ZV        | 0           |
| 602 | Testrapport       | ZV        | 20          |
| 603 | Test              | ZV        | 30          |
| 604 | Kodeinspeksjon    | ZV        | 5           |

**Totalt antall timer: 350 timer.**

### 3 Verktøy

---

Her er en oversikt over de verktøyene vi planlegger å bruke i løpet av iterasjonen:

- wxWidgets
- Microsoft Office
- Microsoft Project
- VBS2 VTK
- Microsoft Visual Studio 2008 SP1
- Microsoft Paint
- Dreamweaver
- Adobe Photoshop
- Dropbox

## 4 Mål for iterasjon 9

---

### 4.1 Milepæler

I løpet av iterasjonen skal følgende dokumenter produseres/klargjøres:

- Iterasjonsrapport 9.

### 4.2 Primære mål

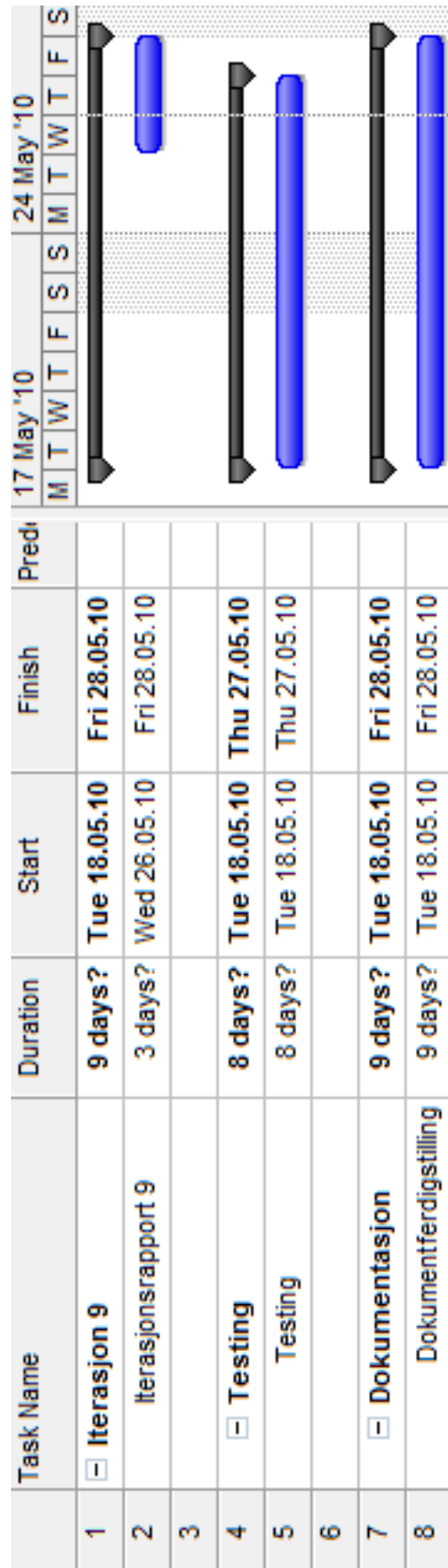
- Ferdigstille alle dokumenter
- Gjennomføre alle tester
- Ferdig stille prosjekt til levering
- Presentasjon 3

### 4.3 Sekundære mål

### 4.4 Prosjektdokumenter som oppdateres

- Alle

## 5 Gantt-diagram



# KDA RIVET



## Iterasjonsrapport 1

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....                            | 3 |
| 1.1   | Hensikt med dokumentet.....                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3   | Dokumenthistorie.....                        | 3 |
| 1.4   | Forfatter(e).....                            | 3 |
| 1.5   | Ansvarlig.....                               | 3 |
| 2     | Mål og iterasjonsprodukter.....              | 4 |
| 2.1   | Mål for iterasjonen.....                     | 4 |
| 2.2   | Primære mål.....                             | 4 |
| 2.3   | Sekundære mål.....                           | 4 |
| 3     | Timeforbruk.....                             | 5 |
| 3.1   | Timeoversikt.....                            | 5 |
| 3.1.1 | Administrasjon og prosjektstyring.....       | 5 |
| 3.1.2 | Research.....                                | 5 |
| 3.1.3 | Krav/Analyse.....                            | 5 |
| 3.1.4 | Design.....                                  | 6 |
| 3.1.5 | Implementasjon.....                          | 6 |
| 3.1.6 | Test.....                                    | 6 |
| 3.2   | Kommentarer til timeforbruk.....             | 6 |
| 4     | Konklusjon.....                              | 8 |
| 4.1   | Milepæler.....                               | 8 |
| 4.2   | Primære mål.....                             | 8 |
| 4.3   | Sekundære mål.....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

Ingen

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Ansvar          |
|---------------|----------|---------------------------|-----------------|
| 1.99          | 31.05.10 | Korrekturlest             | Lone Knutsen    |
| 2.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth |

## 1.4 Forfatter(e)

Ole-Martin Grøtterud

## 1.5 Ansvarlig

Ole-Martin Grøtterud

## 2 Mål og iterasjonsprodukter

---

I løpet av første iterasjon har vi kartlagt mesteparten av teknologien vi har tenkt til å bruke, slik at vi nå vet hva vi skal gå videre på i neste iterasjon. Vi har også produsert førsteutkast av de grunnleggende dokumentene, og dermed laget ett grunnlag for prosjektet.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Forstudierapport
- Iterasjonsplan 1
- Iterasjonsrapport 1
- Iterasjonsplan 2
- Kravspesifikasjon
- Teststrategi
- Testspesifikasjon
- Risikodokument
- Prosjektplan

Og i tillegg følgende teknologidokumenter:

- VBS 2
- Application Scripting Interface
- Add-ons
- WxWidgets

### 2.2 Primære mål

- Gjennomføre nok research til å kunne velge beste verktøy å jobbe videre med.
- Skrive førsteutkast av alle dokumentene under punkt 4.1 Milepæler og sørge for at alle følger samme dokumentstandard.
- Utarbeide flest mulig use caser og forsikre oss om at de er slik prosjektgiver vil ha de

### 2.3 Sekundære mål

- Få opp ett websystem for lagring av dokumenter på websiden
- Lage et system på websiden for å lagre timelister og automatgenerere rapporter



## 3 Timeforbruk

### 3.1 Timeoversikt

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 100               | 75,5             | -24,5 |
| 101 | Prosjektstyring  | 20                | 22,5             | 2,5   |
| 102 | Korrekturlesing  | 30                | 9,5              | -20,5 |
| 103 | Prosjektplan     | 10                | 11               | 1     |
| 104 | Presentasjoner   | 80                | 60,5             | -19,5 |
| 105 | Dokumentstandard | 5                 | 4,5              | -0,5  |
| 106 | Økonomi          | 3                 | 2                | -1    |
| 107 | Diskusjon        |                   |                  | 0     |
| 109 | Møtereferat      | 7,5               | 1                | -6,5  |
| 110 | Møteinnkalling   | 2,5               | 1,5              | -1    |
| 111 | Bedriftsmøte     | 30                | 28               | -2    |
| 112 | Webutvikling     | 30                | 12               | -18   |
| 113 | Planlegging      | 15                | 23,5             | 8,5   |
| 114 | Backup           |                   | 1                | 1     |
| 115 | Kvalitetssikring | 5                 | 4                | -1    |

#### 3.1.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 20                | 20,5             | 0,5   |
| 201 | Trainer-funksjoner                   | 3                 | 3                | 0     |
| 202 | Interface mellom joystick og trainer |                   |                  | 0     |
| 203 | Add-ons til VBS2                     | 15                | 5                | -10   |
| 204 | VTK vs Fusion                        | 15                | 26,5             | 11,5  |
| 205 | WxWidgets                            | 15                | 17               | 2     |

#### 3.1.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 10                | 14               | 4     |
| 301 | Use case          | 10                | 2                | -8    |
| 302 | Kravspesifikasjon | 5                 |                  | -5    |
| 304 | Analyse           | 5                 | 2                | -3    |
| 305 | Forstudie         | 20                | 11               | -9    |
| 306 | Risikoanalyse     | 5                 | 7                | 2     |
| 307 | Brukermanual      |                   |                  |       |

### 3.1.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          |                   |                  |       |
| 401 | Designdokument   |                   | 1                | 1     |
| 402 | Designdiagrammer |                   |                  |       |

### 3.1.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 |                   |                  |       |
| 501 | Prototype                    |                   |                  |       |
| 502 | Implementasjon               |                   |                  |       |
| 503 | Implementasjonsdokumentasjon |                   |                  |       |

### 3.1.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 20                | 13               | -7    |
| 601 | Testspesifikasjon | 25                | 10               | -15   |
| 602 | Testrapport       |                   |                  |       |
| 603 | Test              |                   |                  |       |

**Totalt planlagt timebruk: 506 timer**

**Totalt timer brukt: 388,5 timer**

**Totalt avvik: -117,5**

## 3.2 Kommentarer til timeforbruk

Timeforbruket totalt sett er en del lavere enn vi hadde planlagt. Dette kan skyldes at vi har jobbet mer effektivt enn vi hadde trodd og derfor ble ferdig med dokumentene raskere enn ventet. Vi har redusert møtetiden ved å ta i bruk daglige scrum-møter og dermed også brukt tiden mer effektivt og kuttet nesten ut tid på møtereferater. På vanlige møter skrev vi referatet mens møtet pågikk og førte det derfor ikke på som ekstra tidsbruk.

Siden dokumentasjonen skal leveres to arbeidsdager før presentasjonen vil det komme en del flere timer under rubrikken "Presentasjoner" enn faktisk timebruk viser på nåværende tidspunkt.

Websiden er som dokumentet vil tilsi, noe etter skjema, men den vil bli prioritert i neste iterasjon.

På teknologidokumentene viste det seg at add-ons teknologidokumentet var betydelig mindre jobb enn vi trodde da det fantes ferdig manualer for det på nettet. VTK vs Fusion på den andre siden ble jobbet mer med da dette er tungt å sette seg inn i.

Arbeidet med krav, use-caser og kravspesifikasjon har gått hånd-i-hånd som det burde, men timene har blitt skrevet for det meste på krav ettersom det har vært grunnsteinen for de to andre.

På testingen har vi tydeligvis estimert for mye da det ikke var grunnlag for mer arbeid på testing i denne iterasjonen.

## 4 Konklusjon

---

Alle har jobbet godt med prosjektet og visst god arbeidsinnsats hele veien og resultatet har blitt bra. Vi har hatt noen forekomster av sykdom og skader, men siden det var omtalt i risikoanalysen ble det raskt tatt hånd om. Ellers har det vært en del tekniske utfordringer med teknologien vi bruker, men ingenting vi ikke har klart å løse enda. Arbeidet vi har gjort i denne fasen har gitt oss mye kunnskap og erfaringer som vi vil få mye bruk for videre fremover.

### 4.1 Milepæler

- Vi har produsert første versjon av alle dokumentene som var milepæler for denne iterasjonen. Alle milepælene er nådd.

### 4.2 Primære mål

- Alle de primære målene som var satt opp for iterasjonen er nådd.

### 4.3 Sekundære mål

- Websiden er oppe, men har ikke nådd de sekundære målene for iterasjonen, dette vil bli prioritert i neste iterasjon

# KDA RIVET



## Iterasjonsrapport 2

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Mål og iterasjonsprodukter .....             | 4 |
| 2.1 | Mål for iterasjonen .....                    | 4 |
| 2.2 | Primære mål .....                            | 4 |
| 2.3 | Sekundære mål.....                           | 4 |
| 3   | Timeforbruk .....                            | 5 |
| 3.1 | Administrasjon og prosjektstyring.....       | 5 |
| 3.2 | Research .....                               | 5 |
| 3.3 | Krav/Analyse.....                            | 5 |
| 3.4 | Design .....                                 | 6 |
| 3.5 | Implementasjon.....                          | 6 |
| 3.6 | Test .....                                   | 6 |
| 3.7 | Kommentarer til timeforbruk.....             | 6 |
| 4   | Konklusjon.....                              | 7 |
| 4.1 | Milepæler .....                              | 7 |
| 4.2 | Primære mål .....                            | 7 |
| 4.3 | Sekundære mål.....                           | 7 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

Ingen

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør        |
|---------------|----------|---------------------------|-----------------|
| 1.99          | 31.05.10 | Korrekturlest             | Lone Knutsen    |
| 2.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Mål og iterasjonsprodukter

---

I løpet av andre iterasjon har vi analysert hele systemet og lagt grunnlag for design, slik at vi nå vet hva vi skal gå videre på i neste iterasjon. Vi har også produsert førsteutkast av de grunnleggende dokumentene, og dermed laget et grunnlag for prosjektet.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Iterasjonsplan 3
- Iterasjonsrapport 2
- Analysedokument
- Kodestandard

### 2.2 Primære mål

- Timeregistreringsfunksjonalitet på websiden.
- Skrive førsteutkast av alle dokumentene under punkt 4.1
- Utarbeide en testlog mal.
- Gjennomføre en grundig analyse av systemet og produsere kommunikasjon, klasse og sekvensdiagrammer for alle use casene, samt flere diagrammer hvis nødvendig.
- Få en prototype til å kommunisere enveis med VBS2 ved å sende en kommando.
- Sette opp Visual Studio med wxWidgets slik at det inneholder alle biblioteker og linker for å kompilere.

### 2.3 Sekundære mål

- Forbedre referanser i tidligere produserte dokumenter
- Gjøre lesning rundt designdokumentet
- Testing av prototypen



### 3 Timeforbruk

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 25                | 20,5             | -4,5  |
| 101 | Prosjektstyring  | 10                | 7,5              | -2,5  |
| 102 | Korrekturlesing  | 0                 | 0                | 0     |
| 103 | Prosjektplan     | 5                 | 0                | -5    |
| 104 | Presentasjoner   | 0                 | 0                | 0     |
| 105 | Dokumentstandard | 0                 | 5                | 5     |
| 106 | Økonomi          | 2                 | 1,5              | -0,5  |
| 107 | Diskusjon        | 0                 | 0                | 0     |
| 108 | Dokumentmaler    | 3                 | 0                | -3    |
| 109 | Møtereferat      | 0                 | 2,5              | 2,5   |
| 110 | Møteinnkalling   | 0                 | 0                | 0     |
| 111 | Bedriftsmøte     | 0                 | 0                | 0     |
| 112 | Webutvikling     | 40                | 38               | -2    |
| 113 | Planlegging      | 5                 | 6,5              | 1,5   |
| 114 | Backup           | 0                 | 0                | 0     |
| 115 | Kvalitetssikring | 0                 | 0,5              | 0,5   |

#### 3.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 0                 | 0                | 0     |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 1                 | 0                | -1    |
| 204 | VTK vs Fusion                        | 1                 | 0                | -1    |
| 205 | WxWidgets                            | 40                | 27               | -13   |

#### 3.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 0                 | 0                | 0     |
| 301 | Use case          | 0                 | 4                | 4     |
| 302 | Kravspesifikasjon | 0                 | 0                | 0     |
| 304 | Analyse           | 30                | 23,5             | -6,5  |
| 305 | Forstudie         | 0                 | 0                | 0     |
| 306 | Risikoanalyse     | 2                 | 0                | -2    |

|     |              |   |   |   |
|-----|--------------|---|---|---|
| 307 | Brukermanual | 0 | 0 | 0 |
|-----|--------------|---|---|---|

### 3.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 0                 | 0                | 0     |
| 401 | Designdokument   | 0                 | 1                | 1     |
| 402 | Designdiagrammer | 0                 | 0                | 0     |

### 3.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 5                 | 11,5             | 6,5   |
| 501 | Prototype                    | 20                | 15               | -5    |
| 502 | Implementasjon               | 0                 | 0                | 0     |
| 503 | Implementasjonsdokumentasjon | 0                 | 0                | 0     |

### 3.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 15                | 0                | -15   |
| 601 | Testspesifikasjon | 15                | 4,5              | -10,5 |
| 602 | Testrapport       | 0                 | 0                |       |
| 603 | Test              | 0                 | 0                | 0     |

**Totalt planlagt timebruk: 227 timer**

**Totalt timer brukt: 168,5 timer**

**Totalt avvik: 58,5 timer**

### 3.7 Kommentarer til timeforbruk

Grunnen til avviket på 58,5 timer kan forklares ved at vi tok fri resten av presentasjonsdagen og dagen etter, noe som gjorde at vi mistet to ekstra dager som vi ikke hadde regnet med. Etter neste presentasjon vil vi derfor passe godt på å få med ett par hviledager i tidsberegningen. Testansvarlig var også syk en uke, og de andre fikk ikke gjort oppgavene hans i denne perioden, derfor er timeavviket på test også ganske stort. Avviket på wxWidgets skyldes at den ansvarlige måtte brukes i andre sammenhenger. I tillegg har det blitt brukt mer tid på planlegging denne iterasjonen, ettersom vi har delt opp prosjektet fra 6 til 9 iterasjoner for å øke tempoet og få en bedre oppdeling.

## 4 Konklusjon

---

Selv om vi brukte betydelig færre timer enn vi hadde planlagt denne iterasjonen, har vi jobbet effektivt nok til å få i havn alle de primære målene for iterasjonen. Dette tyder på at vi allerede har en bedre innsikt i systemet enn vi hadde regnet med og vil øke tempoet deretter. Dette har vi gjort ved å dele prosjektet fra 6 til 9 iterasjoner.

### 4.1 Milepæler

- Vi har produsert første versjon av alle dokumentene som var milepæler for denne iterasjonen. Alle milepælene er nådd.

### 4.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd.

### 4.3 Sekundære mål

- Det har blitt foretatt noe lesning om design dokumentene. Ingen av de øvrige sekundære målene har blitt nådd, men vil bli prioritert i neste iterasjon.

# KDA RIVET



## Iterasjonsrapport 3

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....                            | 3 |
| 1.1   | Hensikt med dokumentet.....                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3   | Dokumenthistorie.....                        | 3 |
| 1.4   | Ansvarlig .....                              | 3 |
| 2     | Mål og iterasjonsprodukter .....             | 4 |
| 2.1   | Mål for iterasjonen .....                    | 4 |
| 2.2   | Primære mål .....                            | 4 |
| 2.3   | Sekundære mål.....                           | 4 |
| 3     | Timeforbruk .....                            | 5 |
| 3.1   | Timeoversikt .....                           | 5 |
| 3.1.1 | Administrasjon og prosjektstyring .....      | 5 |
| 3.1.2 | Research .....                               | 5 |
| 3.1.3 | Krav/Analyse.....                            | 5 |
| 3.1.4 | Design .....                                 | 6 |
| 3.1.5 | Implementasjon .....                         | 6 |
| 3.1.6 | Test .....                                   | 6 |
| 3.2   | Kommentarer til timeforbruk.....             | 6 |
| 4     | Konklusjon.....                              | 7 |
| 4.1   | Milepæler .....                              | 7 |
| 4.2   | Primære mål .....                            | 7 |
| 4.3   | Sekundære mål.....                           | 7 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

**GUI:** Graphical User Interface.

**DII:** Dynamic linked library

**VBS2:** Virtual Battle Space 2.

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør             |
|---------------|----------|---------------------------|----------------------|
| 0.1           | 20.02.10 | Førsteutkast              | Lone Knutsen         |
| 0.2           | 10.03.10 | Ny dokumentstandard       | Ole-Martin Grøtterud |
| 0.99          | 23.03.10 | Korrekturlest             | Ole-Martin Grøtterud |
| 1.0           | 31.05.10 | Klargjort til innlevering | Thomas Transeth      |

## 1.4 Ansvarlig

Lone Knutsen

## 2 Mål og iterasjonsprodukter

---

I løpet av iterasjonen har vi kartlagt og fått på plass tre prototyper og en stabil arkitektur på disse. Vi har fått opp en prototype på GUI'et med et identisk GUI slik det skal være og to fungerende knapper. Og en for Dll-kommunikasjon med VBS2, hvor vi nå har en flerveiskommunikasjon. Den siste prototypen er en påbygning av DLL prototypen og håndterer joystick posisjon, denne kom i tillegg til de to planlagte grunnet utsettelse av iterasjonen.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Testrapport av prototypene
- Prototypedokumenter for prototypene
- Iterasjonsplan 4
- Iterasjonsrapport 3
- Designdokument
- GUI-dokument

### 2.2 Primære mål

- Vi skal ha en GUI-prototype med to knapper, som igjen skal ha en funksjon hver.
- Dll-prototypen skal bruke flerveiskommunikasjon.
- Skrive førsteutkast av alle dokumentene under punkt 4.1 Milepæler og sørge for at alle følger samme dokumentstandard.

### 2.3 Sekundære mål

- Forbedre referanser i tidligere produserte dokumenter
- Koble sammen GUI-prototype og Dll-prototype.

## 3 Timeforbruk

### 3.1 Timeoversikt

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 20                | 26,5             | +6,5  |
| 101 | Prosjektstyring  | 10                | 7                | -3    |
| 102 | Korrekturlesing  | 5                 | 0                | -5    |
| 103 | Prosjektplan     | 5                 | 12               | +7    |
| 104 | Presentasjoner   | 0                 | 0                | 0     |
| 105 | Dokumentstandard | 2                 | 0                | -2    |
| 106 | Økonomi          | 1                 | 0                | -1    |
| 107 | Diskusjon        | 0                 | 0                | 0     |
| 108 | Dokumentmaler    | 2                 | 0                | -2    |
| 109 | Møtereferat      | 2                 | 0                | -2    |
| 110 | Møteinnkalling   | 2                 | 0                | -2    |
| 111 | Bedriftsmøte     | 0                 | 0                | 0     |
| 112 | Webutvikling     | 5                 | 8,5              | +3,5  |
| 113 | Planlegging      | 10                | 7                | -3    |
| 114 | Backup           | 3                 | 0                | -3    |
| 115 | Kvalitetssikring | 0                 | 4                | +4    |
| 116 | PR               | 0                 | 3,5              | +3,5  |

#### 3.1.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 5                 | 0,5              | -4,5  |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 0                 | 0                | 0     |
| 204 | VTK vs Fusion                        | 0                 | 0                | 0     |
| 205 | WxWidgets                            | 50                | 24               | -26   |

#### 3.1.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 0                 | 0                | 0     |
| 301 | Use caser         | 0                 | 2,5              | +2,5  |
| 302 | Kravspesifikasjon | 0                 | 0                | 0     |
| 304 | Analyse           | 5                 | 3                | -2    |
| 305 | Forstudie         | 0                 | 0                | 0     |



|     |               |   |   |    |
|-----|---------------|---|---|----|
| 306 | Risikoanalyse | 5 | 0 | -5 |
| 307 | Brukermanual  | 0 | 0 | 0  |

### 3.1.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 0                 | 0                | 0     |
| 401 | Designdokument   | 35                | 22               | -13   |
| 402 | Designdiagrammer | 0                 | 21,5             | +21,5 |

### 3.1.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 2                 | 0                | -2    |
| 501 | Prototype                    | 45                | 15               | -30   |
| 502 | Implementasjon               | 0                 | 0                | 0     |
| 503 | Implementasjonsdokumentasjon | 0                 | 0                | 0     |
| 504 | Testing av teknologi         | 0                 | 8                | +8    |
| 505 | Prototype wxWidgets          | 0                 | 30               | +30   |

### 3.1.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 0                 | 0                | 0     |
| 601 | Testspesifikasjon | 2                 | 18,5             | +16,5 |
| 602 | Testrapport       | 0                 | 4                | +4    |
| 603 | Test              | 9                 | 1,5              | -7,5  |

**Totalt planlagt timebruk:** 225 timer

**Totalt timer brukt:** 219 timer

**Totalt avvik:** - 6 timer

## 3.2 Kommentarer til timeforbruk

Vi fikk noen flere timer på møter enn vi hadde regnet med. Vi måtte ta et evalueringsmøte på samarbeidet vårt, og dette trakk opp møtetimene litt. Et stykke ut i iterasjonen vår, så bestemte vi oss for å opprette en ny aktivitet (nr 505) som vi kalte "Prototype wxWidgtes". Vi fikk derfor et stort avvik (-26 timer) i de planlagte timene på pkt 205 *wxWidgtes*, og et påfølgende avvik på pkt 505 *wxWidgtes prototype* på + 30 timer. I alt så blir avviket der på +4 timer.

Når det gjelder pkt 501 Prototype, så viste det seg at å få opp en Dll-flerveiskommunikasjon med VBS2 ikke ble så vrient som fryktet. Vi klarte altså å spare inn 30 timer på denne. Testingen (pkt 601, 602 og 603) tok også en del mer tid enn beregnet (+13 timer), og dette kommer av at noen av testspesifikasjonene måtte oppdateres.

## 4 Konklusjon

---

I denne iterasjonen så har vi oppnådd alle de primære målene våre, og litt til. Vi har fått på plass en stabil arkitektur på begge prototypene våre, og vi har testet at disse fungerer slik de skal. I tillegg har vi fått opp et korrekt GUI på wxWidgets prototypen.

### 4.1 Milepæler

- Vi har produsert første versjon av alle dokumentene som var milepæler for denne iterasjonen. Alle milepælene er nådd.

### 4.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd.

### 4.3 Sekundære mål

- Vi har ikke knyttet sammen wxWidgets prototypen og Dll-prototypen.
- Har forbedret referansene i noen dokumenter.

# KDA RIVET



## Iterasjonsrapport 4

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....                            | 3 |
| 1.1   | Hensikt med dokumentet.....                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3   | Dokumenthistorie.....                        | 3 |
| 1.4   | Ansvarlig .....                              | 3 |
| 2     | Mål og iterasjonsprodukter .....             | 4 |
| 2.1   | Mål for iterasjonen .....                    | 4 |
| 2.2   | Primære mål .....                            | 4 |
| 2.3   | Sekundære mål.....                           | 4 |
| 3     | Timeforbruk .....                            | 5 |
| 3.1   | Timeoversikt .....                           | 5 |
| 3.1.1 | Administrasjon og prosjektstyring .....      | 5 |
| 3.1.2 | Research .....                               | 5 |
| 3.1.3 | Krav/Analyse.....                            | 5 |
| 3.1.4 | Design .....                                 | 6 |
| 3.1.5 | Implementasjon .....                         | 6 |
| 3.1.6 | Test .....                                   | 6 |
| 3.2   | Kommentarer til timeforbruk.....             | 6 |
| 4     | Konklusjon.....                              | 7 |
| 4.1   | Milepæler .....                              | 7 |
| 4.2   | Primære mål .....                            | 7 |
| 4.3   | Sekundære mål.....                           | 7 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

**GUI:** Graphical User Interface.

**DLL:** Dynamic linked library

**VBS2:** Virtual Battle Space 2.

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 07.03.10 | Førsteutkast              | Zoran Vukobratovic |
| 0.1           | 08.03.10 | Legge til timeforbruk     | Zoran Vukobratovic |
| 0.2           | 11.03.10 | Ny dokumentstandard       | Zoran Vukobratovic |
| 0.99          | 17.03.10 | Korrekturlest             | Arne Kristian      |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Zoran Vukobratovic

## 2 Mål og iterasjonsprodukter

---

I løpet av fjerde iterasjon har vi begynt på implementasjonen og implementasjonsdokumentasjonen. Vi har opprettet frameworket fra design diagrammene og arbeidet den til slik at den er klar til å brukes. I tillegg har vi produsert førsteutkast av de grunnleggende dokumentene.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen skal følgende dokumenter produseres:

- Iterasjonsrapport 4.
- Iterasjonsplan 5.
- Implementasjonsdokument

### 2.2 Primære mål

- Generere og gjøre klar framework.
- Lage minst 2 slider per emne til presentasjon 2.
- Web-siden sitt timelistesystem skal gjøres helt ferdig.
- Produsere Implementasjonsdokument med dokumentasjon om GUI-et og kommunikasjonen.

### 2.3 Sekundære mål

- Endre størrelsesforhold på synsfeltet til VBS2.
- Endre plassering på programvinduet til VBS2.
- Få til toveis kommunikasjon mellom GUI og VBS2.

## 3 Timeforbruk

### 3.1 Timeoversikt

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 15                | 16               | +1    |
| 101 | Prosjektstyring  | 15                | 5                | -10   |
| 102 | Korrekturlesing  | 5                 | 0                | -5    |
| 103 | Prosjektplan     | 5                 | 2                | -3    |
| 104 | Presentasjoner   | 10                | 5                | -5    |
| 105 | Dokumentstandard | 0                 | 0                | 0     |
| 106 | Økonomi          | 0                 | 0                | 0     |
| 107 | Diskusjon        | 0                 | 10               | +10   |
| 108 | Dokumentmaler    | 0                 | 0                | 0     |
| 109 | Møtereferat      | 2                 | 1                | -1    |
| 110 | Møteinnkalling   | 2                 | 1                | -1    |
| 111 | Bedriftsmøte     | 0                 | 0                | 0     |
| 112 | Webutvikling     | 15                | 2                | -13   |
| 113 | Planlegging      | 10                | 4                | -6    |
| 114 | Backup           | 3                 | 3                | 0     |
| 115 | Kvalitetssikring | 10                | 0                | -10   |
| 116 | PR               | 15                | 0                | -15   |
| 122 | Web              | 0                 | 2                | +2    |

#### 3.1.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 0                 | 0                | 0     |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 0                 | 0                | 0     |
| 204 | VTK vs Fusion                        | 0                 | 0                | 0     |
| 205 | WxWidgets                            | 0                 | 0                | 0     |

#### 3.1.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 0                 | 0                | 0     |
| 301 | Use case          | 0                 | 0                | 0     |
| 302 | Kravspesifikasjon | 1                 | 0                | -1    |
| 304 | Analyse           | 0                 | 0                | 0     |

|     |               |   |   |   |
|-----|---------------|---|---|---|
| 305 | Forstudie     | 0 | 0 | 0 |
| 306 | Risikoanalyse | 0 | 0 | 0 |
| 307 | Brukermanual  | 0 | 0 | 0 |

### 3.1.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 0                 | 0                | 0     |
| 401 | Designdokument   | 5                 | 0                | 0     |
| 402 | Designdiagrammer | 5                 | 0                | 0     |

### 3.1.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 0                 | 0                | 0     |
| 501 | Prototype                    | 0                 | 0                | 0     |
| 502 | Implementasjon               | 32                | 28               | -4    |
| 503 | Implementasjonsdokumentasjon | 15                | 2                | -13   |
| 504 | Testing av teknologi         | 0                 | 10               | +10   |
| 505 | Prototype wxWidgets          | 0                 | 0                | 0     |

### 3.1.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 0                 | 0                | 0     |
| 601 | Testspesifikasjon | 0                 | 0                | 0     |
| 602 | Testrapport       | 5                 | 0                | 0     |
| 603 | Test              | 15                | 0                | -15   |
| 604 | Kodeinspeksjon    | 0                 | 11               | +11   |

**Totalt planlagt timebruk:** 170 timer

**Totalt timer brukt:** 102 timer

**Totalt avvik:** -68 timer

## 3.2 Kommentarer til timeforbruk

Denne iterasjonen måtte endres og tilpasses pga. iterasjonsutsettelsen forrige uke. Det gjorde at vi mistet en uke av denne iterasjonen, som tilsvarte rundt 30 % av tiden.

Timene som var satt opp under punkt 603 (test) ble i stedet flyttet over til et nytt punkt, 604 (Kodeinspeksjon), da dette dokumentet var viktig for elaboration-fasen.

Timeforbruket denne uken var også en del mindre fordi de forskjellige oppgavene som falt under implementasjon og testing av teknologi, punkt 502 og 504, ikke kunne gjøres samtidig på den ene PC-en vi har fått fra KDA. Vi har nå planlagt å lage et system for en rotasjon slik at alle oppgaver kan bli gjort i tide i stedet for at gruppedlemmer må sitte og vente på tur.



## 4 Konklusjon

---

I denne iterasjonen så har vi oppnådd alle de primære målene våre, og alle sekundære bortsett fra ett. Frameworket er oppe og går og det fungerer som det skal med all kommunikasjon på plass.

### 4.1 Milepæler

- Vi har produsert første versjon av alle dokumentene som var milepæler for denne iterasjonen. Alle milepælene er nådd.

### 4.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd.

### 4.3 Sekundære mål

- Etter en diskusjon kom vi frem til at toveiskommunikasjon mellom GUI og VBS2 ikke var nødvendig på dette stadiet. Alle resterende sekundære mål er oppfylt.

# KDA RIVET



## Iterasjonsrapport 5

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Mål og iterasjonsprodukter .....             | 4 |
| 2.1 | Mål for iterasjonen .....                    | 4 |
| 2.2 | Primære mål .....                            | 4 |
| 2.3 | Sekundære mål.....                           | 4 |
| 3   | Timeforbruk .....                            | 5 |
| 3.1 | Administrasjon og prosjektstyring.....       | 5 |
| 3.2 | Research .....                               | 5 |
| 3.3 | Krav/Analyse.....                            | 5 |
| 3.4 | Design .....                                 | 6 |
| 3.5 | Implementasjon.....                          | 6 |
| 3.6 | Test .....                                   | 6 |
| 3.7 | Kommentarer til timeforbruk.....             | 6 |
| 4   | Konklusjon.....                              | 7 |
| 4.1 | Milepæler .....                              | 7 |
| 4.2 | Primære mål .....                            | 7 |
| 4.3 | Sekundære mål.....                           | 7 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS2** – Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 12.04.10 | Førsteutkast              | Thomas Transeth    |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Mål og iterasjonsprodukter

---

Vi har i femte iterasjon i hovedsak klargjort prosjektet til framføringen, samt få en grundig kontroll på hvordan systemet skal bygges oss og implementasjonen av dette. Resultatet er et godt rammeverk og kontroll på kommunikasjonen.

Vi har også fått en ny dokumentstandard som alle dokumenter nå følger samt begynt arbeidet med styring mus – joystick.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Iterasjonsrapport 5.
- Iterasjonsplan 6.
- Klargjøring av alle dokumenter til innlevering.

### 2.2 Primære mål

- Klargjøre og øve til presentasjon 2.
- Lage ny dokumentstandard og oppdatere alle dokumenter til denne.
- Klargjøre alle dokumenter til innlevering.
- Implementere joystick-funksjon for å kunne frigjøre mus til overlay.
- Lage et stabilt rammeverk til implementasjonen.
- Sende kommandoer mellom overlay og VBS2 via samme kommunikasjonskanal, men med ulike resultater.

### 2.3 Sekundære mål

- Endre størrelsesforhold på synsfeltet til VBS2.
- Sette opp kode som gjør uavhengige inkluderingsfiler til .dll for å lettere gjøre systemet skalerbart.

### 3 Timeforbruk

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 20                | 9                | -11   |
| 101 | Prosjektstyring  | 15                | 6                | -9    |
| 102 | Korrekturlesing  | 25                | 10               | -15   |
| 103 | Prosjektplan     | 5                 | 0                | -5    |
| 104 | Presentasjoner   | 80                | 211              | 131   |
| 105 | Dokumentstandard | 0                 | 6                | 6     |
| 106 | Økonomi          | 2                 | 0                | -2    |
| 107 | Diskusjon        | 0                 | 0                | 0     |
| 108 | Dokumentmaler    | 15                | 0                | -15   |
| 109 | Møtereferat      | 5                 | 0                | -5    |
| 110 | Møteinnkalling   | 5                 | 0                | -5    |
| 111 | Bedriftsmøte     | 0                 | 0                | 0     |
| 112 | Webutvikling     | 5                 | 9                | 4     |
| 113 | Planlegging      | 5                 | 0                | -5    |
| 114 | Backup           | 3                 | 1                | -2    |
| 115 | Kvalitetssikring | 10                | 9                | -1    |
| 116 | PR               | 5                 | 0                | -5    |

#### 3.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 0                 | 0                | 0     |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 0                 | 0                | 0     |
| 204 | VTK vs Fusion                        | 0                 | 0                | 0     |
| 205 | WxWidgets                            | 0                 | 0                | 0     |

#### 3.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 0                 | 0                | 0     |
| 301 | Use case          | 0                 | 0                | 0     |
| 302 | Kravspesifikasjon | 0                 | 0                | 0     |
| 304 | Analyse           | 0                 | 0                | 0     |
| 305 | Forstudie         | 0                 | 0                | 0     |

|     |               |   |   |   |
|-----|---------------|---|---|---|
| 306 | Risikoanalyse | 0 | 0 | 0 |
| 307 | Brukermanual  | 0 | 0 | 0 |

### 3.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 0                 | 0                | 0     |
| 401 | Designdokument   | 0                 | 0                | 0     |
| 402 | Designdiagrammer | 0                 | 0                | 0     |

### 3.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 5                 | 0                | -5    |
| 501 | Prototype - DLL              | 0                 | 0                | 0     |
| 502 | Implementasjon               | 40                | 27               | -13   |
| 503 | Implementasjonsdokumentasjon | 30                | 7                | -23   |
| 505 | Prototype – wxWidgets        | 0                 | 0                | 0     |

### 3.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 0                 | 0                | 0     |
| 601 | Testspesifikasjon | 0                 | 0                | 0     |
| 602 | Testrapport       | 5                 | 0                | -5    |
| 603 | Test              | 20                | 0                | -20   |
| 604 | Kodeinspeksjon    | 5                 | 4                | -1    |

**Totalt planlagt timebruk: 300 timer**

**Totalt timer brukt: 305 timer**

**Totalt avvik: 5 timer**

### 3.7 Kommentarer til timeforbruk

Når en ser timebruket totalt sett målt opp mot det gruppen planla før iterasjonen, kommer det tydelig fram at vi godt har klart å beregne faktisk arbeidsmengde.

Når en ser på inndelingen av timene ser en klart at størsteparten av iterasjonen har blitt brukt til presentasjonsforberedelser. Feilestimeringen skyldes at vi ble raskt ferdig med primærmålene og valgte derfor å bruke mer tid til å forberede presentasjonen, og da bøte på noen av kommentarene som vi fikk som tilbakemeldinger fra forrige presentasjon.

Mye av klargjøringen av dokumentene ble også raskt unnagjort da disse var av høyere kvalitet enn sist levering og derfor ikke trengte like mye etterarbeid. Dette er en av godene vi fikk fra den nye dokumentstandarden.

## 4 Konklusjon

---

Selv om presentasjonen krevde mesteparten av tiden vår i denne iterasjonen klarte vi å bli ferdig med alle primære mål på en tilfredsstillende måte. Vi har nå et godt rammeverk for koden og prototypene er satt sammen og kommuniserer.

### 4.1 Milepæler

- Alle milepælene er nådd da alle dokumentene ble klargjort til innleveringen og presentasjon 2 ble holdt.
- Iterasjonsplan 5 og iterasjonsrapport 6 er produsert.

### 4.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd.

### 4.3 Sekundære mål

- For å endre størrelsesforholdet til VBS2 kreves det mer jobbing og dette ble ikke prioritert i denne iterasjonen.
- Kode for uavhengige inkluderingsfiler til .dll er avlyst da systemet ikke har behov for dette og implementasjonen skal gjøres på en annen måte.



# KDA RIVET



## Iterasjonsrapport 6

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |   |   |
|-----|---|---|
| 1   | Introduksjon .....                            | 3 |
| 1.1 | Hensikt med dokumentet .....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser ..... | 3 |
| 1.3 | Dokumenthistorie .....                        | 3 |
| 1.4 | Ansvarlig .....                               | 3 |
| 2   | Mål og iterasjonsprodukter .....              | 4 |
| 2.1 | Mål for iterasjonen .....                     | 4 |
| 2.2 | Primære mål .....                             | 4 |
| 2.3 | Sekundære mål .....                           | 4 |
| 3   | Timeforbruk .....                             | 5 |
| 3.1 | Administrasjon og prosjektstyring .....       | 5 |
| 3.2 | Research .....                                | 5 |
| 3.3 | Krav/Analyse .....                            | 5 |
| 3.4 | Design .....                                  | 6 |
| 3.5 | Implementasjon .....                          | 6 |
| 3.6 | Test .....                                    | 6 |
| 4   | Kommentarer til timeforbruk .....             | 7 |
| 5   | Konklusjon .....                              | 8 |
| 5.1 | Milepeler .....                               | 8 |
| 5.2 | Primære mål .....                             | 8 |
| 5.3 | Sekundære mål .....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS2** – Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 19.04.10 | Førsteutkast              | Arne Kr.           |
| 0.2           | 23.04.10 | Dokumentstandard          | Lone Knutsen       |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Arne Kristian Åmellem

## 2 Mål og iterasjonsprodukter

---

Store deler av sjettede iterasjon har gått med til påskeferie og eksamensforberedelser. Vi har også fått implementert et designpattern i frameworket våres, samt det å kunne bevege våpenet med joysticken. Resultatet er at vi nå kan styre vindusstørrelsen på VBS2 via vårt program, roterer våpen ved hjelp av joystick og har et kjent designpattern implementert.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Iterasjonsrapport 6.
- Iterasjonsplan 7.

### 2.2 Primære mål

- Identifisere designpatterns som passer systemet vårt.
- Implementere designpattern fra designet.

### 2.3 Sekundære mål

- Styre våpen ved hjelp av joystick
- Implementere armeringsbryter funksjonalitet



### 3 Timeforbruk

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 10                | 10               | 0     |
| 101 | Prosjektstyring  | 2                 | 5                | 3     |
| 102 | Korrekturlesing  | 0                 | 0                | 0     |
| 103 | Prosjektplan     | 5                 | 0                | -5    |
| 104 | Presentasjoner   | 0                 | 0                | 0     |
| 105 | Dokumentstandard | 0                 | 0                | 0     |
| 106 | Økonomi          | 0                 | 1                | 1     |
| 107 | Diskusjon        | 0                 | 0                | 0     |
| 108 | Dokumentmaler    | 0                 | 0                | 0     |
| 109 | Møtereferat      | 1                 | 1                | 0     |
| 110 | Møteinnkalling   | 1                 | 0                | -1    |
| 111 | Bedriftsmøte     | 0                 | 0                | 0     |
| 112 | Webutvikling     | 0                 | 1                | 1     |
| 113 | Planlegging      | 5                 | 10               | -5    |
| 114 | Backup           | 2                 | 0                | -2    |
| 115 | Kvalitetssikring | 5                 | 3                | -2    |
| 116 | PR               | 0                 | 0                | 0     |

#### 3.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 0                 | 0                | 0     |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 0                 | 0                | 0     |
| 204 | VTK vs Fusion                        | 0                 | 0                | 0     |
| 205 | WxWidgets                            | 0                 | 0                | 0     |

#### 3.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 0                 | 0                | 0     |
| 301 | Use case          | 0                 | 0                | 0     |
| 302 | Kravspesifikasjon | 0                 | 0                | 0     |
| 304 | Analyse           | 0                 | 0                | 0     |
| 305 | Forstudie         | 0                 | 0                | 0     |

|     |               |   |   |    |
|-----|---------------|---|---|----|
| 306 | Risikoanalyse | 0 | 0 | 0  |
| 307 | Brukermanual  | 3 | 0 | -3 |

### 3.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 1                 | 0                | 0     |
| 401 | Designdokument   | 5                 | 5                | 0     |
| 402 | Designdiagrammer | 5                 | 4                | 0     |

### 3.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 5                 | 0                | -5    |
| 501 | Prototype - DLL              | 0                 | 0                | 0     |
| 502 | Implementasjon               | 80                | 66               | -14   |
| 503 | Implementasjonsdokumentasjon | 10                | 9                | -1    |
| 504 | Testing av Teknologi         | 15                | 21               | -6    |
| 505 | Prototype – wxWidgets        | 0                 | 0                | 0     |

### 3.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 0                 | 0                | 0     |
| 601 | Testspesifikasjon | 0                 | 0                | 0     |
| 602 | Testrapport       | 5                 | 0                | -5    |
| 603 | Test              | 10                | 11               | +1    |
| 604 | Kodeinspeksjon    | 5                 | 0                | -5    |

**Totalt planlagt timebruk: 175 timer**

**Totalt timer brukt: 147 timer**

**Totalt avvik: 28 timer**

#### **4 Kommentarer til timeforbruk**

---

Denne itarasjonen bommet vi litt på timeantallet. Dette skyldes ferie og eksamen og det da var litt vanlig og estimere hvor mye vi fikk jobbet innimellom. Men som beregnet har det gått mest tid til implementasjon.

Det ble dobbelt så mye planleggingsarbeid som vi beregnet, men dette skyldes endringer vi valgte å ta etter feedback fra eksternveileder.

## 5 Konklusjon

---

Selv om vi jobbet noe mindre enn beregnet har vi klart og bli ferdig med primærkravene våre, samt ett av sekundærmålene.

Selv om presentasjonen krevde mesteparten av tiden vår i denne iterasjonen klarte vi å bli ferdig med alle primære mål på en tilfredsstillende måte. Vi har nå et godt rammeverk for koden og prototypene er satt sammen og kommuniserer.

### 5.1 Milepeler

- Alle milepælene er nådd da alle dokumentene ble klargjort til innleveringen og presentasjon 2 ble holdt.
- Iterasjonsplan 7 og iterasjonsrapport 6 er produsert.

### 5.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd.

### 5.3 Sekundære mål

- Styre våpen ved hjelp av joystick



# KDA RIVET



## Iterasjonsrapport 7

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |   |                                     |
|-----|---|-------------------------------------|
| 1   | Introduksjon .....                            | 3                                   |
| 1.1 | Hensikt med dokumentet .....                  | 3                                   |
| 1.2 | Definisjoner, akronymer og forkortelser ..... | 3                                   |
| 1.3 | Dokumenthistorie .....                        | 3                                   |
| 1.4 | Ansvarlig .....                               | 3                                   |
| 2   | Mål og iterasjonsprodukter .....              | 4                                   |
| 2.1 | Mål for iterasjonen .....                     | 4                                   |
| 2.2 | Primære mål .....                             | 4                                   |
| 2.3 | Sekundære mål .....                           | 4                                   |
| 3   | Timeforbruk .....                             | 5                                   |
| 3.1 | Administrasjon og prosjektstyring .....       | 5                                   |
| 3.2 | Research .....                                | 5                                   |
| 3.3 | Krav/Analyse .....                            | 5                                   |
| 3.4 | Design .....                                  | 6                                   |
| 3.5 | Implementasjon .....                          | 6                                   |
| 3.6 | Test .....                                    | 6                                   |
| 3.7 | Kommentarer til timeforbruk .....             | 7                                   |
| 4   | Konklusjon .....                              | 8                                   |
| 4.1 | Milepæler .....                               | 8                                   |
| 4.2 | Primære mål .....                             | 8                                   |
| 4.3 | Sekundære mål .....                           | 8                                   |
| 5   | Tracking Gantt-diagram .....                  | <b>Error! Bookmark not defined.</b> |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS2** – Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør             |
|---------------|----------|---------------------------|----------------------|
| 0.1           | 04.05.10 | Førsteutkast              | Ole-Martin Grøtterud |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic   |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth      |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Mål og iterasjonsprodukter

---

I denne iterasjonen har vi klart å implementere mye av de viktige funksjonene, og ser lyst på fremtiden. Arbeidet har vært krevende teknisk, men vi mener vi har funnet gode løsninger og fått det til på en godt fungerende måte.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Iterasjonsrapport 7.
- Iterasjonsplan 8.

### 2.2 Primære mål

- Følgende funksjonalitet i implementasjonen skal være til stede:
  - Bemanne systemet.
  - Sikte: elevasjon og rotasjon.
  - Armering på/av.
  - Skyte.
- Funksjonaliteten over skal være testet
- Alle patterns skal være implementert og dokumentert

### 2.3 Sekundære mål

- Følgende funksjonalitet i implementasjonen skal være til stede:
  - Velge skuddhastighet.
  - Sikte: simulere servoene.
  - Zoome.
- Funksjonaliteten over skal være testet

### 2.4 Prosjektdokumenter som oppdateres

- Prosjektplan.
- Designdokument.
- Implementasjonsdokumentasjon.
- Kodestandard.
- Testrapporter.

### 3 Timeforbruk

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 10                | 8                | -2    |
| 101 | Prosjektstyring  | 15                | 12               | -3    |
| 102 | Korrekturlesing  | 0                 | 0                | 0     |
| 103 | Prosjektplan     | 10                | 2                | -8    |
| 104 | Presentasjoner   | 0                 | 0                | 0     |
| 105 | Dokumentstandard | 0                 | 5                | 5     |
| 106 | Økonomi          | 0                 | 0                | 0     |
| 107 | Diskusjon        | 0                 | 0                | 0     |
| 108 | Dokumentmaler    | 0                 | 0                | 0     |
| 109 | Møtereferat      | 2                 | 1                | -1    |
| 110 | Møteinnkalling   | 0                 | 0                | 0     |
| 111 | Bedriftsmøte     | 0                 | 0                | 0     |
| 112 | Webutvikling     | 10                | 0                | -10   |
| 113 | Planlegging      | 7                 | 5                | -2    |
| 114 | Backup           | 1                 | 0                | -1    |
| 115 | Kvalitetssikring | 5                 | 3                | -2    |
| 116 | PR               | 0                 | 0                | 0     |

#### 3.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 0                 | 0                | 0     |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 0                 | 0                | 0     |
| 204 | VTK vs Fusion                        | 0                 | 0                | 0     |
| 205 | WxWidgets                            | 0                 | 0                | 0     |

#### 3.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 5                 | 0                | -5    |
| 301 | Use caser         | 5                 | 0                | -5    |
| 302 | Kravspesifikasjon | 3                 | 0                | -3    |
| 304 | Analyse           | 5                 | 0                | -5    |
| 305 | Forstudie         | 0                 | 0                | 0     |

|     |               |    |   |    |
|-----|---------------|----|---|----|
| 306 | Risikoanalyse | 2  | 0 | -2 |
| 307 | Brukermanual  | 10 | 9 | -1 |

### 3.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 3                 | 0                | -3    |
| 401 | Designdokument   | 3                 | 3                | 0     |
| 402 | Designdiagrammer | 4                 | 2                | -2    |

### 3.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 10                | 1                | -9    |
| 501 | Prototype - DLL              | 0                 | 0                | 0     |
| 502 | Implementasjon               | 90                | 134              | 44    |
| 503 | Implementasjonsdokumentasjon | 30                | 44               | 14    |
| 504 | Testing av Teknologi         | 50                | 41               | -9    |
| 505 | Prototype – wxWidgets        | 0                 | 0                | 0     |

### 3.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 0                 | 4                | 4     |
| 601 | Testspesifikasjon | 0                 | 0                | 0     |
| 602 | Testrapport       | 20                | 16               | -4    |
| 603 | Test              | 40                | 32               | -8    |
| 604 | Kodeinspeksjon    | 10                | 12               | 2     |

**Totalt planlagt timebruk: 350 timer**

**Totalt timer brukt: 323 timer**

**Totalt avvik: -27 timer**

## 4 Kommentarer til timeforbruk

---

Vi hadde planlagt å legge til nye krav i denne iterasjonen, men etter møte med bedriften kom vi frem til at vi beholder kravene vi har og istedet optimaliserer implementasjonen. Dette førte til at vi ikke brukte tid på analyse og krav, men kun noen få timer på å endre designet for å tilpasse implementasjonen. Vi brukte betydelig mer timer på implementasjonen for å perfeksjonere systemet vårt, og resultatet har vært veldig bra. Alle A og B -kravene har også blitt testet og korreksjoner har blitt gjort som følge av dem.



## 5 Konklusjon

---

Alle primære og sekundære mål ble nådd denne iterasjonen, bortsett fra zoom som skyltes en bug i VBS2 vi ikke får gjort noe med. I tillegg til de oppsatte målene har vi også fått implementert nesten alle C-kravene og regner med at vi får nye krav i neste iterasjon.

### 5.1 Milepeler

- Iterasjonsplan 8 og iterasjonsrapport 7 er produsert.

### 5.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd.

### 5.3 Sekundære mål

- Alle sekundære mål bortsett fra zoom ble nådd.



# KDA RIVET



## Iterasjonsrapport 8

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |   |                                     |
|-----|---|-------------------------------------|
| 1   | Introduksjon .....                            | 3                                   |
| 1.1 | Hensikt med dokumentet .....                  | 3                                   |
| 1.2 | Definisjoner, akronymer og forkortelser ..... | 3                                   |
| 1.3 | Dokumenthistorie .....                        | 3                                   |
| 1.4 | Ansvarlig .....                               | 3                                   |
| 2   | Mål og iterasjonsprodukter .....              | 4                                   |
| 2.1 | Mål for iterasjonen .....                     | 4                                   |
| 2.2 | Primære mål .....                             | 4                                   |
| 2.3 | Sekundære mål .....                           | 4                                   |
| 3   | Timeforbruk .....                             | 5                                   |
| 3.1 | Administrasjon og prosjektstyring .....       | 5                                   |
| 3.2 | Research .....                                | 5                                   |
| 3.3 | Krav/Analyse .....                            | 5                                   |
| 3.4 | Design .....                                  | 6                                   |
| 3.5 | Implementasjon .....                          | 6                                   |
| 3.6 | Test .....                                    | 6                                   |
| 3.7 | Kommentarer til timeforbruk .....             | 7                                   |
| 4   | Konklusjon .....                              | 8                                   |
| 4.1 | Milepæler .....                               | 8                                   |
| 4.2 | Primære mål .....                             | 8                                   |
| 4.3 | Sekundære mål .....                           | 8                                   |
| 5   | Tracking Gantt-diagram .....                  | <b>Error! Bookmark not defined.</b> |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS2** – Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 13.05.10 | Førsteutkast              | Thomas Transeth    |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Mål og iterasjonsprodukter

---

Denne iterasjonen startet 04.05 og holdt på fram til 17.05. I dette tidsrommet, som er siste iterasjon i Contruction, har vi brukt mye tid på å fullføre og forbedre systemet, og vi har lagt til en del nye oppgaver som vi har jobbet med. Spesielt har vi brukt tid på å undersøke muligheten for å bruke VBS2-skript som løsning på de nye utfordringene.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Iterasjonsrapport 8.
- Iterasjonsplan 9.

### 2.2 Primære mål

- Alle A, B og C krav skal tilfredsstillte de satte kriteriene før de nye kravene innføres.
- Alle funksjoner skal testes.
- All dokumentasjon skal klargjøres.

### 2.3 Sekundære mål

- Legge til følgende funksjonalitet med tilhørende prioritet.
  - Brukeren skal kunne skifte "crosshair".
  - Implementere "face of direction".
  - Lage en navigerbar meny med riktige menyfelter.

### 2.4 Prosjektdokumenter som oppdateres

- Prosjektplan.
- Designdokument.
- Implementasjonsdokumentasjon.
- Kodestandard.
- Testrapporter.
- Kravdokument.
- Risikodokument.

### 3 Timeforbruk

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 10                | 13               | 2     |
| 101 | Prosjektstyring  | 15                | 10               | 5     |
| 102 | Korrekturlesing  | 0                 | 5                | -5    |
| 103 | Prosjektplan     | 10                | 10               | 0     |
| 104 | Presentasjoner   | 10                | 10               | 0     |
| 105 | Dokumentstandard | 0                 | 10               | -10   |
| 106 | Økonomi          | 0                 | 0                | 0     |
| 107 | Diskusjon        | 0                 | 0                | 0     |
| 108 | Dokumentmaler    | 0                 | 0                | 0     |
| 109 | Møtereferat      | 2                 | 2                | 0     |
| 110 | Møteinnkalling   | 0                 | 0                | 0     |
| 111 | Bedriftsmøte     | 0                 | 0                | 0     |
| 112 | Webutvikling     | 0                 | 0                | 0     |
| 113 | Planlegging      | 10                | 7                | 3     |
| 114 | Backup           | 1                 | 1                | 0     |
| 115 | Kvalitetssikring | 5                 | 4                | 1     |
| 116 | PR               | 0                 | 0                | 0     |

#### 3.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 0                 | 0                | 0     |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 0                 | 0                | 0     |
| 204 | VTK vs Fusion                        | 0                 | 0                | 0     |
| 205 | WxWidgets                            | 5                 | 0                | 5     |

#### 3.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 5                 | 0                | 5     |
| 301 | Use caser         | 5                 | 0                | 5     |
| 302 | Kravspesifikasjon | 3                 | 0                | 3     |
| 304 | Analyse           | 5                 | 1                | 4     |
| 305 | Forstudie         | 0                 | 0                | 0     |

|     |               |    |   |    |
|-----|---------------|----|---|----|
| 306 | Risikoanalyse | 2  | 0 | 2  |
| 307 | Brukermanual  | 18 | 0 | 18 |

### 3.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 3                 | 0                | 3     |
| 401 | Designdokument   | 3                 | 0                | 3     |
| 402 | Designdiagrammer | 3                 | 5                | -2    |

### 3.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 0                 | 0                | 0     |
| 501 | Prototype - DLL              | 0                 | 0                | 0     |
| 502 | Implementasjon               | 110               | 105              | 5     |
| 503 | Implementasjonsdokumentasjon | 30                | 20               | 10    |
| 504 | Testing av Teknologi         | 20                | 56               | -36   |
| 505 | Prototype – wxWidgets        | 0                 | 0                |       |

### 3.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 0                 | 0                | 0     |
| 601 | Testspesifikasjon | 0                 | 1                | -0    |
| 602 | Testrapport       | 20                | 4                | 16    |
| 603 | Test              | 40                | 25               | 15    |
| 604 | Kodeinspeksjon    | 15                | 10               | 5     |

**Totalt planlagt timebruk: 350 timer.**

**Totalt timer brukt: 300 timer**

**Totalt avvik: 50 timer**

## 4 Kommentarer til timeforbruk

---

I denne iterasjonen planlagte vi å fullføre systemet vårt fir å gjøre det mulig å avslutte prosjektet i tide. Det ble brukt noe liten tid på testing, da vi kjapt startet på utvidelsen av oppgaven som ble utarbeidet i samarbeid med oppdragsgiver. Vi startet tidlig på dette for å ha mulighet for å utforske teknologien før vi la den til implementasjonen, og derfor vises det en del avvik på timene.



## 5 Konklusjon

---

De primære målene er nådd for iterasjonen, men noen av testene måtte kjøres flere ganger da det var funksjonalitet som ikke ble godkjent de første rundene.

Gruppen valgte og ikke å implementere to av de ekstra oppgavene, men menysystemet er fullt implementert og fungerende.

### 5.1 Milepeler

- Iterasjonsplan 9 og iterasjonsrapport 8 er produsert.

### 5.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd, men noen tester måtte kjøres flere ganger.

### 5.3 Sekundære mål

- Det viste seg at det ikke lot seg implementere verken crosshair eller face of direction da dette viste seg en større jobb enn det vi har igjen av prosjekttid.
- Menysystemet er satt opp.



# KDA RIVET



## Iterasjonsrapport 9

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |   |   |
|-----|---|---|
| 1   | Introduksjon .....                            | 3 |
| 1.1 | Hensikt med dokumentet .....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser ..... | 3 |
| 1.3 | Dokumenthistorie .....                        | 3 |
| 1.4 | Ansvarlig .....                               | 3 |
| 2   | Mål og iterasjonsprodukter .....              | 4 |
| 2.1 | Mål for iterasjonen .....                     | 4 |
| 2.2 | Primære mål .....                             | 4 |
| 2.3 | Sekundære mål .....                           | 4 |
| 2.4 | Prosjektdokumenter som oppdateres .....       | 4 |
| 3   | Timeforbruk .....                             | 5 |
| 3.1 | Administrasjon og prosjektstyring .....       | 5 |
| 3.2 | Research .....                                | 5 |
| 3.3 | Krav/Analyse .....                            | 5 |
| 3.4 | Design .....                                  | 6 |
| 3.5 | Implementasjon .....                          | 6 |
| 3.6 | Test .....                                    | 6 |
| 4   | Kommentarer til timeforbruk .....             | 7 |
|     | Konklusjon .....                              | 8 |
| 4.1 | Milepeler .....                               | 8 |
| 4.2 | Primære mål .....                             | 8 |
| 4.3 | Sekundære mål .....                           | 8 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å oppsummere iterasjonen og evaluere arbeidet som er gjort i forhold til målene som er satt.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS2** – Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør              |
|---------------|----------|---------------------------|-----------------------|
| 0.1           | 31.05.10 | Førsteutkast              | Arne Kristian Åmellem |
| 0.99          | 31.05.10 | Korrekturlesing           | Zoran Vukobratovic    |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth       |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Mål og iterasjonsprodukter

---

Denne iterasjonen startet 18.05 og holdt på fram til 31.05. I dette tidsrommet som siste iterasjon, nemlig Transition, har vi brukt mye tid på å ferdigstille systemet, samt klargjøre alle dokumenter for innlevering. Dette har vært den store oppgaven denne iterasjonen. Alle dokumenter måtte gjennomgås og klargjøres. Noe dokumentasjon måtte også produseres.

### 2.1 Mål for iterasjonen

I løpet av iterasjonen har vi hatt som mål å produsere følgende dokumenter:

- Iterasjonsrapport 9.

### 2.2 Primære mål

- Ferdigstille alle dokumenter.
- Gjennomføre alle tester.
- Ferdig stille prosjekt til levering.
- Presentasjon 3.

### 2.3 Sekundære mål

### 2.4 Prosjektdokumenter som oppdateres

- Alle

### 3 Timeforbruk

Nedenfor er en oversikt over planlagt timeforbruk fra iterasjonsplanen og faktisk timebruk samt avviket på disse.

#### 3.1 Administrasjon og prosjektstyring

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 100 | Møter            | 10                | 9                | 1     |
| 101 | Prosjektstyring  | 15                | 10               | 5     |
| 102 | Korrekturlesing  | 56                | 67               | -11   |
| 103 | Prosjektplan     | 10                | 0                | 10    |
| 104 | Presentasjoner   | 35                | 0                | 35    |
| 105 | Dokumentstandard | 0                 | 15               | -15   |
| 106 | Økonomi          | 0                 | 0                | 0     |
| 109 | Møtereferat      | 2                 | 1                | 1     |
| 110 | Møteinnkalling   | 2                 | 0                | 0     |
| 111 | Bedriftsmøte     | 0                 | 8                | -8    |
| 112 | Webutvikling     | 10                | 4                | 6     |
| 113 | Planlegging      | 10                | 7                | 3     |
| 114 | Backup           | 1                 | 0                | 1     |
| 115 | Kvalitetssikring | 5                 | 39               | -34   |
| 116 | PR               | 5                 | 9                | -4    |

#### 3.2 Research

| Nr. | Aktivitet                            | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|--------------------------------------|-------------------|------------------|-------|
| 200 | VBS2 funksjoner og bruk              | 0                 | 0                | 0     |
| 201 | Trainer-funksjoner                   | 0                 | 0                | 0     |
| 202 | Interface mellom joystick og trainer | 0                 | 0                | 0     |
| 203 | Add-ons til VBS2                     | 0                 | 0                | 0     |
| 204 | VTK vs Fusion                        | 0                 | 0                | 0     |
| 205 | WxWidgets                            | 0                 | 0                | 0     |

#### 3.3 Krav/Analyse

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 300 | Krav              | 0                 | 2                | -2    |
| 301 | Use caser         | 0                 | 0                | 0     |
| 302 | Kravspesifikasjon | 0                 | 0                | 0     |
| 304 | Analyse           | 0                 | 1                | -1    |
| 305 | Forstudie         | 0                 | 0                | 0     |
| 306 | Risikoanalyse     | 0                 | 0                | 0     |
| 307 | Brukermanual      | 5                 | 4                | 1     |

### 3.4 Design

| Nr. | Aktivitet        | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------|-------------------|------------------|-------|
| 400 | Klasser          | 0                 | 0                | 0     |
| 401 | Designdokument   | 6                 | 6                | 0     |
| 402 | Designdiagrammer | 3                 | 2                | 1     |

### 3.5 Implementasjon

| Nr. | Aktivitet                    | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|------------------------------|-------------------|------------------|-------|
| 500 | Kodestandard                 | 0                 | 0                | 0     |
| 501 | Prototype - DLL              | 0                 | 0                | 0     |
| 502 | Implementasjon               | 80                | 107              | -27   |
| 503 | Implementasjonsdokumentasjon | 30                | 27               | 3     |
| 504 | Testing av Teknologi         | 10                | 7                | 3     |
| 505 | Prototype – wxWidgets        | 0                 | 0                | 0     |

### 3.6 Test

| Nr. | Aktivitet         | Planlagt timebruk | Faktisk timebruk | Avvik |
|-----|-------------------|-------------------|------------------|-------|
| 600 | Teststrategier    | 0                 | 0                | 0     |
| 601 | Testspesifikasjon | 0                 | 0                | 0     |
| 602 | Testrapport       | 20                | 11               | 9     |
| 603 | Test              | 30                | 14,5             | -15,5 |
| 604 | Kodeinspeksjon    | 5                 | 0                | 5     |

**Totalt planlagt timebruk: 350 timer.**

**Totalt timer brukt: 350,5 timer**

**Totalt avvik: 0,5 timer**

## 4 Kommentarer til timeforbruk

---

I denne iterasjonen planla vi å gjøre ferdig stille prosjektet vårt i tide for innlevering. Siden vi utvidet systemet vårt men noen ekstra funksjoner så ble det lagt litt mer i Implementasjon enn beregnet. Det ble også testet litt mindre enn planlagt. Grunnen for dette er at vi brukte en del tid på å utforske teknologien på forhånd, så vi var godt kjent med det vi skulle implementere.

## Konklusjon

---

De primære målene er nådd for iterasjonen. Alle dokumenter er klare og skal leveres mandag 31. Mai.

### 4.1 Milepeler

- Iterasjonsrapport 9 er produsert og samtlige dokumenter klargjort.

### 4.2 Primære mål

- Alle de primære målene for iterasjonen ble nådd

### 4.3 Sekundære mål

- Denne Iterasjonen hadde ingen sekundærmål



# KDA RIVET



## Kodestandard

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                            | 3  |
| 1.1   | Hensikt med dokumentet.....                  | 3  |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3  |
| 1.3   | Dokumenthistorie.....                        | 3  |
| 1.4   | Ansvarlig.....                               | 3  |
| 2     | Navngiving.....                              | 4  |
| 2.1   | Variabelnavn.....                            | 4  |
| 2.2   | Funksjonsnavn.....                           | 4  |
| 2.3   | Klassenavn og struct.....                    | 5  |
| 2.4   | Parameternavn.....                           | 5  |
| 2.5   | Namespace-navn.....                          | 5  |
| 3     | Kodestruktur.....                            | 6  |
| 3.1   | Filnavn.....                                 | 6  |
| 3.2   | Inkluderingsbeskyttelse.....                 | 6  |
| 3.3   | Inkludering av filer.....                    | 6  |
| 3.4   | Namespace.....                               | 7  |
| 4     | Formatering.....                             | 8  |
| 4.1   | Grensesnitt.....                             | 8  |
| 4.2   | If-else-setninger.....                       | 8  |
| 4.3   | Løkker.....                                  | 9  |
| 4.4   | Unntakshåndtering.....                       | 9  |
| 5     | Kommentering av kode.....                    | 10 |
| 5.1   | Deklarasjonsfil (.h).....                    | 10 |
| 5.1.1 | Globalt for fil.....                         | 10 |
| 5.1.2 | Metoder/funksjoner i deklarasjonsfil.....    | 11 |
| 5.2   | Implementasjonsfil.....                      | 11 |
| 5.2.1 | Globalt for fil.....                         | 11 |
| 5.2.2 | Metoder/funksjoner i implementasjonsfil..... | 11 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Dette dokumentet skal sikre at all kode som blir produsert blir skrevet på samme måte, uavhengig av hvem som programmerer den. Det vil her bli vist med eksempler på hvordan alle standardløkker og setninger skal bli kodet, samt generelt oppsett og dokumentasjon. Dokumentet er skrevet for kodeutvikling i programmeringsspråket C++.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet  | Redaktør                           |
|---------------|----------|--|------------------------------------|
| 1.1           | 20.04.10 | Oppdatering etter gjennomgang med ekstern veileder | Thomas Transeth                    |
| 1.2           | 23.04.10 | Oppdatert iht dokumentstandard                     | Lone Knutsen                       |
| 1.99          | 24.05.10 | Korrekturlesing                                    | Zoran Vukobratovic og Lone Knutsen |
| 2.0           | 28.05.10 | Klargjort for innlevering                          | Thomas Transeth                    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Navngiving

---

For at all kode lettere skal bli forstått, er det viktig at alle programmererne følger samme standard og jobber etter denne. Når en kode er blitt skrevet skal den også kommenteres og dokumenteres etter følgende standarder.

All kode og dokumentasjon skal skrives på engelsk.

### 2.1 Variabelnavn

Alle variabelnavn skal være forklarende og kunne utrykke hva den representerer uten at en trenger å kjenne til resten av koden og det anbefales å bruke et substantiv.

De skal starte med liten bokstav og om det skal være mer enn ett ord langt skal vært nye ord starte med stor bokstav. Det skal ikke forekomme tall for ord og det brukes kun engelske bokstaver.

Eksempel:

```
int distance;
string file;
string weaponType;
```

### 2.2 Funksjonsnavn

Alle funksjoner skal ha forklarende navn som beskriver hva de gjør og det anbefales å bruke et verb. De skal starte med stor bokstav og alle etterfølgende ord skal starte med stor bokstav som i følge med PascalCasing. Krøllparentesene starter linjen under metodenavnet og slutter linjen under siste kodelinje.

Eksempel:

```
void SendDataToClient()
{
    //code
}
```

## 2.3 Klassenavn og struct

Klassene og struct skal ha et så forklarende navn som det lar seg gjøre. De skal starte med stor bokstav og alle etterfølgende ord skal starte med stor bokstav som i følge med PascalCasing. Krøllparantesene starter linjen under klassenavnet og slutter linjen under siste kodelinje.

Eksempel:

```
Struct types
{
    //code
};

class WeaponControl()
{
    //code
};
```

## 2.4 Parameternavn

Alle parametere skal være forklarende og kunne utrykke hva den representerer uten at en trenger å kjenne til resten av koden og det anbefales å bruke et substantiv.

De skal starte med liten bokstav og om det skal være mer enn ett ord langt skal vært nye ord starte med stor bokstav. Det skal ikke forekomme tall for ord og det brukes kun engelske bokstaver.

Om parameteret er til for å oppdatere en global verdi skal den hete det samme som denne globale verdien.

Eksempel:

```
string weaponType;

void setWeaponType(string weaponType)
{
    this->weaponType = weaponType;
}
```

## 2.5 Namespace-navn

Namespace-ene skal ha et navn som forklarer klassene den har ansvaret for. Alle skal ha stor bokstav og alle etterfølgende ord skal ha stor bokstav som i følge PascalCasing.

Krøllparantesene starter linjen under namespace-navnet og slutter linjen under siste kodelinje.

Eksempel:

```
namespace GuiButtons
{
    //code
}
```

## 3 Kodestruktur

---

For å gjøre systemet strukturert, lettlest og skalerbart er det valgt å bruke følgende løsninger:

### 3.1 Filnavn

I dette prosjektet er det tre typer filer som spesielt blir brukt:

- .h for deklarasjonsfiler
- .cpp for implementasjonsfiler
- .dll for dynamiske biblioteker

For filtypene .h og .cpp skal disse hete det samme når de hører sammen og navnet skal være selvforklarende. For filtypen .dll blir det brukt en mer beskrivende tittel over hva den representerer.

Alle filtypene skal ha stor bokstav og alle etterfølgende ord skal ha stor bokstav som i følge PascalCasing og skal hete det samme som klassen den inneholder.

Eksempel:

```
WeaponControl.h
WeaponControl.cpp
GuiControl.dll
```

### 3.2 Inkluderingsbeskyttelse

For å sikre at koden ikke blir unødig inkludert skal det være inkluderingsbeskyttelse på alle filer. Dette skal gjøres ved å bruke "pragma once".

Eksempel:

```
#pragma once
#define WEAPON_CONTROL_H
//code
```

### 3.3 Inkludering av filer

Det er to typer filer som skal inkluderes. Den ene er deklarasjonsfiler som er en del av utviklingsplattformen og skal inkluderes med <>, og skal deklarerer først. Den andre er lokale filer som blir utviklet spesielt til dette prosjektet og disse skal inkluderes med " ", og skal deklarerer sist.

Om det skal inkluderes spesifikke namespaces skal disse deklarerer etter dette igjen.

Eksempel:

```
#include <string>
#include <iostream>
#include "WeaponTypes.h"
#include "VehicleTypes.h";

using namespace std;
```

### 3.4 Namespace

Systemet som skal utvikles består av mange små klasser som ofte kun har et lite ansvarsområde. For ikke å miste kontrollen på disse har vi valgt å bruke namespace som grupperingsmetode.

Alle klasser som direkte har noe med hverandre å gjøre skal stå innenfor samme namespace. Eksempler på dette er grafiske klasse som for eksempel knapp, bryter og bakgrunn i namespace-et SystemComponents.

Eksempel:

```
namespace SystemComponents
{
    //code
}
```

## 4 Formatering

---

For at koden i store trekk skal se lik ut og at forskjellige deler av systemet skal være lettere å identifisere, har vi valgt å bruke strukturene som er forklart under.

Semikolon ";" skal brukes direkte etter en setning uten mellomrom og det skal ikke forekomme flere setninger på samme linje.

### 4.1 Grensesnitt

Alle klasser skal ha spesifisert tilgangsrettighet på alle variabler og funksjoner/metoder. Først skal rettigheten skrives etterfulgt av et kolon ":", deretter et linjeskift og så koden. De skal skrives på samme innrykk som i klassen de står og all kode som tilhører tilgangsrettigheten skal ha innrykk.

Det skal brukes følgende rekkefølge:

Eksempel:

```
class Weapon
{
    private:
        //code
    protected:
        //code
    public:
        //code
}
```

### 4.2 If-else-setninger

If-else-setningene skal formateres som følgende.

Eksempel:

```
if(condition)
{
    //code
}
else if(anotherCondition)
{
    //code
}
else
{
    //code
}
```



### 4.3 Løkker

For-løkker skal starte telleren på i, og videre nøstede løkker skal ha etterfølgende bokstav som teller:

Eksempel:

```
for(int i = startValue; condition; i++)
{
    for(int j = startValue; booleanCheck; j++)
    {
        //code
    }
}
```

While-løkker skal formateres på følgende måte:

Eksempel:

```
while(condition)
{
    //code
}
```

### 4.4 Unntakshåndtering

Alle uttrykk som skal/kan generere unntak skal settes i en try-catch blokk:

Eksempel:

```
try
{
    //code that throws an exception
}
catch(Exception e)
{
    //code for exception handling
}
```

## 5 Kommentering av kode

---

Kommentering skal skje ved Javadoc-standarden. Vi bruker denne metoden fordi gruppen er grundig kjent med denne type kommentering, noe som vil skape en bedre helhet og forståelse.

Alle kommentarer skal skrives på engelsk.

### 5.1 Deklarasjonsfil (.h)

Header-filer skal kommenteres på følgende måte:  
(se liste under 5.1.1)

#### 5.1.1 Globalt for fil

1. @file filnavn
2. Beskrivelse av filen
3. Hvilke krav den stammer fra
4. @author Forfattere
5. @version Versjonsnummer
6. Metode/funksjonsbeskrivelser

Eksempel:

```
/**
 * @file core.h
 * This file is the core header and has responsibility for
 * the main control of the system.
 * From requirement: A(controllability)
 * @author Thomas Transeth, Arne Kristian Åmellem
 * @version 1.4 (12.03.10)
 *
 * bool fire(): Fires the active gun.
 * int getAmmoCount: Returns the counter for remaining ammo
 */

#ifndef CORE_H
#define CORE_H

//class

#endif
```

### 5.1.2 Metoder/funksjoner i deklarasjonsfil

Det skal ikke kommenteres for metoden/funksjonen der den predefineres. Dette skal skje i kommentarfeltet øverst i filen.

## 5.2 Implementasjonsfil

Implementasjonsfiler skal kommenteres som vist under.

### 5.2.1 Globalt for fil

Det skal ikke kommenteres globalt for denne filen, da dette skjer i deklarasjonsfilen.

### 5.2.2 Metoder/funksjoner i implementasjonsfil

1. Beskrivelse av metoden/funksjonen
2. @param Parameterbeskrivelse
3. @return Returbeskrivelse

Eksempel:

```
/**
 * This function fires the active gun.
 * @param The degrees of the turret
 * @return True if the gun successfully fired, false otherwise
 */
bool Fire(double deg)
{
    //code
    return isFired;
}
```

# KDA RIVET



## Kostnadsdokument

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|  |   |
|--|---|
| 0T1 Introduksjon.....                            | 3 |
| 1.1 Hensikt med dokumentet.....                  | 3 |
| 1.2 Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 Dokumenthistorie.....                        | 3 |
| 1.4 Ansvarlig .....                              | 3 |
| 2 Kostnadsdokument.....                          | 4 |
| 2.1 Kostnader.....                               | 4 |
| 2.2 Budsjett.....                                | 4 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å få en oversikt over kostnadene prosjektgruppen trenger å få dekket i løpet av prosjektarbeidet.

## 1.2 Definisjoner, akronymer og forkortelser

**FFI:** Forsvarets Forskningsinstitutt

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Ansvar                                      |
|---------------|----------|---------------------------|---|
| 2.99          | 24.05.10 | Korrekturlest             | Arne Kristian Åmellem<br>Zoran Vukobratovic |
| 3.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                             |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Kostnadsdokument

---

### 2.1 Kostnader

Siden KDA dekker utgiftene for maskin- og programvare, trenger gruppen kun de utgiftene de vil ha i forbindelse med fremføringer, kopiering på skolen, dokumentasjon og reise.

Det vil bli fremføringsutgifter i løpet av prosjektet, ettersom det skal holdes tre fremføringer. På hver av disse forventes det av prosjektgruppene å stille med oppvarming til sensorer og veiledere. Vi har valgt å ha et bredt, men kostnadseffektivt, utvalg. Dette får vi til ved å gjøre en ekstra innsats på egenhånd, som å lage eget fruktfat og bake kaker fra bunnen av. Fremføringsutgiftene inkluderer kaffe, te, kaker, penner og annet materiale som blir brukt i sammenheng med fremføringene.

Vi skal ha to møter med FFI på Kjeller, så det vil bli noen reisekostnader på dette. Satsen som er brukt er 3kr/km + 0,50kr/passasjer. Det er 100km hver vei mellom Kongsberg og FFI.

I tillegg kommer det kopieringskostnader for å kopiere på skolen, noe vi blir nødt til å gjøre i mindre sammenhenger. Vi trenger også permer og skilleark til å sette inn dokumentasjonen, samt en poster som skal henges opp på skolen.

### 2.2 Budsjett

| Type:                   | Totalt:            |
|-------------------------|--------------------|
| Fremføringsutgifter     | kr 1 500,00        |
| Reiseutgifter           | kr 2 000,00        |
| Administrative utgifter | kr 600,00          |
| <b>Sum</b>              | <b>kr 4 100,00</b> |

Estimatene dekker alle 3 fremføringer og resten av prosjektet for øvrig.

# KDA RIVET



## Kravdokument

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth



## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                            | 4  |
| 1.1   | Hensikt med dokumentet.....                  | 4  |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 4  |
| 1.3   | Dokumenthistorie.....                        | 4  |
| 1.4   | Ansvarlig.....                               | 4  |
| 2     | Beskrivelse.....                             | 5  |
| 3     | Krav.....                                    | 6  |
| 3.1   | Grunnleggende om krav:.....                  | 6  |
| 3.2   | A-krav:.....                                 | 6  |
| 3.2.1 | Funksjonelle krav:.....                      | 6  |
| 3.2.2 | Ikke-funksjonelle krav:.....                 | 6  |
| 3.3   | B-krav:.....                                 | 6  |
| 3.3.1 | Funksjonelle krav:.....                      | 6  |
| 3.4   | C-krav:.....                                 | 6  |
| 3.4.1 | Funksjonelle krav:.....                      | 6  |
| 4     | Detaljering av krav.....                     | 7  |
| 4.1   | A-krav.....                                  | 8  |
| 4.1.1 | 1F – Bemanne systemet.....                   | 8  |
| 4.1.2 | 2F – Sikte: elevasjon og rotasjon.....       | 9  |
| 4.1.3 | 3F - Armere.....                             | 10 |
| 4.1.4 | 4F - Skyte.....                              | 11 |
| 4.1.5 | 1IF – GUI i henhold til spesifikasjon.....   | 12 |
| 4.2   | B-krav.....                                  | 13 |
| 4.2.1 | 5F - Velge skuddhastighet.....               | 13 |
| 4.2.2 | 6F – Sikte: simulere servoene.....           | 14 |
| 4.2.3 | 7F - Zoom.....                               | 15 |
| 4.3   | C-Krav.....                                  | 16 |
| 4.3.1 | 8F - Nattmodus.....                          | 16 |
| 4.3.2 | 9F – Sikte: sensitivitet på joystick.....    | 17 |
| 5     | Analyse.....                                 | 18 |



|         |  |    |
|---------|--|----|
| 5.1     | Use case-diagram .....                     | 18 |
| 5.2     | Beskrivelse av aktører.....                | 18 |
| 5.2.1   | User: .....                                | 18 |
| 5.2.2   | System: .....                              | 18 |
| 5.3     | Deltaljing av use caser .....              | 19 |
| 5.3.1   | Aktør: User .....                          | 19 |
| 5.3.1.1 | UC1 – Bemanne systemet .....               | 19 |
| 5.3.1.2 | UC2 – Sikte: Elevasjon og rotasjon .....   | 20 |
| 5.3.1.3 | UC3 – Armering på/av .....                 | 20 |
| 5.3.1.4 | UC4 – Skyte .....                          | 21 |
| 5.3.1.5 | UC5 – Velge skuddhastighet.....            | 21 |
| 5.3.1.6 | UC6 – Slå på/av nattmodus.....             | 22 |
| 5.3.1.7 | UC7 – Zoome .....                          | 22 |
| 5.3.1.8 | UC9 – Sikte: sensitivitet på joystick..... | 23 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å avdekke kravene til systemet, samt de tilhørende use-casene.

## 1.2 Definisjoner, akronymer og forkortelser

**GUI:** Graphical user interface

**RWS:** Remote weapon station

**VBS2:** Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                                     | Redaktør                                |
|---------------|----------|---|---|
| 2.1           | 04.05.10 | Oppdatert kravtabellene med test og kommentar | Lone Knutsen                            |
| 2.99          | 25.05.10 | Korrekturlest                                 | Ole-Martin Grøtterud og Thomas Transeth |
| 3.0           | 28.05.10 | Klargjort for innlevering                     | Thomas Transeth                         |

## 1.4 Ansvarlig

Lone Knutsen

## 2 Beskrivelse

---

Utviklingen av en kravspesifikasjon er en dynamisk prosess som holdes gående gjennom hele prosjektet. Kravene i dette dokumentet er utarbeidet fra oppgaveteksten og samtaler med oppdragsgiver. Disse kravene er med på å gi oss en oversikt over hva vi skal gjøre i prosjektet vårt, og hjelper oss med å unngå misforståelser i forhold til prosjektet og hva det går ut på.

Dokumentet skal inneholde alle de kravene vi utvikler i løpet av prosjektet, samt tilhørende use-caser. Det er kun de funksjonelle kravene som er mulig å oppfylle i use-caser, ettersom de ikke-funksjonelle kravene ikke er testbare på samme måte og kun omhandler systemet og ikke bruker.

## 3 Krav

---

### 3.1 Grunnleggende om krav:

Det finnes i hovedsak to hovedtyper krav: *funksjonelle krav* og *ikke-funksjonelle krav*. De funksjonelle kravene handler om hva systemer skal kunne gjøre, og de ikke-funksjonelle kravene handler om hvordan systemet er ment å være.

Videre finnes det forskjellige grupperinger på hvor viktige kravene er for systemet. Disse deler vi inn i tre kategorier: A-, B-, og C-krav. *A-kravene* kan defineres som "må ha" krav, og er nødvendige for at systemet skal fungere som det skal. *B-kravene* er definert som "kan ha" krav, og er krav som man kan legge inn for at systemet skal fungere bedre. Til slutt så har man *C-kravene* som er "kjekt å ha" krav. Dette vil si at de er fine å ha med, men absolutt ingen nødvendighet for systemet i seg selv.

Funksjonelle krav vil få navn som heter xF, ikke-funksjonelle krav xIF.

### 3.2 A-krav:

#### 3.2.1 Funksjonelle krav:

- 1F - bemanne systemet.
- 2F - sikte: elevasjon og rotasjon.
- 3F - armering på/av.
- 4F - skyte.

#### 3.2.2 Ikke-funksjonelle krav:

- 1IF - GUI i henhold til spesifikasjon.

### 3.3 B-krav:

#### 3.3.1 Funksjonelle krav:

- 5F - velge skuddhastighet.
- 6F - sikte: simulere servoene.
- 7F - zoom.

### 3.4 C-krav:

#### 3.4.1 Funksjonelle krav:

- 8F - nattmodus.
- 9F - sikte: sensitivitet på joystick.

## 4 Detaljering av krav

For å holde en god oversikt, vil alle kravene settes opp på tabellform. Videre vil det lages kryssreferanser til testdokumenter og risikodokumenter.

Det generelle utseendet på disse tabellene er som følger:

### Identifisering:

|                                     |                                |                           |      |                  |         |
|-------------------------------------|--------------------------------|---------------------------|------|------------------|---------|
| <b>Opphav til krav</b>              | [bakgrunn for kravet]          | <b>ID</b>                 | [ID] | <b>Prioritet</b> | [A/B/C] |
| <b>Beskrivelse</b>                  | [beskrivelse av kravet]        |                           |      |                  |         |
| <b>Dato for registrering</b>        | [dd.mm.åå]                     | <b>Oppfylt i use case</b> |      | [UcID]           |         |
| <b>Funksjonelt/ikke-funksjonelt</b> | [funksjonelt/ikke-funksjonelt] |                           |      |                  |         |

### Kryssreferanser:

|               |            |                           |  |                      |  |
|---------------|------------|---------------------------|--|----------------------|--|
| <b>Risiko</b> | [RisikoID] |                           |  |                      |  |
| <b>Test</b>   | [TestID]   | <b>Testet/Ikke testet</b> |  | [Testet/ikke testet] |  |

### Annet:

|                  |                             |
|------------------|-----------------------------|
| <b>Status</b>    | [Foreslått/avslått/godtatt] |
| <b>Kommentar</b> |                             |

Tabell 1: Tabellforklaring

## 4.1 A-krav

### 4.1.1 1F - Bemanne systemet

#### Identifisering:

|                                     |                                   |                           |     |                  |   |
|-------------------------------------|-----------------------------------|---------------------------|-----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver           | <b>ID</b>                 | 1F  | <b>Prioritet</b> | A |
| <b>Beskrivelse</b>                  | Beskrives ved å bemanne systemet. |                           |     |                  |   |
| <b>Dato for registrering</b>        | 09.11.09                          | <b>Oppfylt i use case</b> | UC1 |                  |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt                       |                           |     |                  |   |

#### Kryssreferanser:

|               |                  |                           |        |  |
|---------------|------------------|---------------------------|--------|--|
| <b>Risiko</b> | R1, R4, R11      |                           |        |  |
| <b>Test</b>   | TF01, TF13, TF14 | <b>Testet/ikke testet</b> | Testet |  |

#### Annet:

|                  |  |
|------------------|--|
| <b>Status</b>    | Godtatt                                    |
| <b>Kommentar</b> | Godtatt og testet. Fungerer slik det skal. |

Tabell 2: Krav 1F

#### 4.1.2 2F - Sikte: elevasjon og rotasjon

##### Identifisering:

|                                     |   |                           |    |                  |   |
|-------------------------------------|---|---------------------------|----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver   | <b>ID</b>                 | 2F | <b>Prioritet</b> | A |
| <b>Beskrivelse</b>                  | En skal kunne snu skyttertårnet 360 grader uten stoppepunkt. En runde skal ta 4 sek. Elevasjonen skal gå fra -30 til 60 grader. |                           |    |                  |   |
| <b>Dato for registrering</b>        | 09.11.09  | <b>Oppfylt i use case</b> |    | UC2              |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt   |                           |    |                  |   |

##### Kryssreferanser:

|               |             |                           |        |  |
|---------------|-------------|---------------------------|--------|--|
| <b>Risiko</b> | R1, R4, R11 |                           |        |  |
| <b>Test</b>   | TF02, TBL07 | <b>Testet/Ikke testet</b> | Testet |  |

##### Annet:

|                  |                                 |
|------------------|---------------------------------|
| <b>Status</b>    | Godtatt                         |
| <b>Kommentar</b> | Testet. Fungerer slik den skal. |

Tabell 3: Krav 2F



### 4.1.3 3F - Armere

#### Identifisering:

|                                     |   |                           |    |                  |   |
|-------------------------------------|---|---------------------------|----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver   | <b>ID</b>                 | 3F | <b>Prioritet</b> | A |
| <b>Beskrivelse</b>                  | Det skal være mulig å armere og desarmere våpenet ved å flippe "Arm"- og "Remote safe"-bryteren i GUI'et. |                           |    |                  |   |
| <b>Dato for registrering</b>        | 09.11.09  | <b>Oppfylt i use case</b> |    | UC3              |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt   |                           |    |                  |   |

#### Kryssreferanser:

|               |  |                           |        |  |
|---------------|--|---------------------------|--------|--|
| <b>Risiko</b> | R1, R4, R11                            |                           |        |  |
| <b>Test</b>   | TF03, TF04, TBL01, TBL02, TBL10, TBL11 | <b>Testet/Ikke testet</b> | Testet |  |

#### Annet:

|                  |                                    |
|------------------|------------------------------------|
| <b>Status</b>    | Godtatt                            |
| <b>Kommentar</b> | Testet, og fungerer slik den skal. |

Tabell 4: Krav 3F

#### 4.1.4 4F - Skyte

##### Identifisering:

|                                     |   |                           |    |                  |   |
|-------------------------------------|---|---------------------------|----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver   | <b>ID</b>                 | 4F | <b>Prioritet</b> | A |
| <b>Beskrivelse</b>                  | Det skal være mulig å skyte med våpenet ved å bruke skyteknappen på joysticken, etter at våpenet er armert. Dersom dette ikke er oppfylt, skal det ikke fyres av skudd. |                           |    |                  |   |
| <b>Dato for registrering</b>        | 09.11.09  | <b>Oppfylt i use case</b> |    | UC4              |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt   |                           |    |                  |   |

##### Kryssreferanser:

|               |                   |                           |        |  |
|---------------|-------------------|---------------------------|--------|--|
| <b>Risiko</b> | R1, R4, R11       |                           |        |  |
| <b>Test</b>   | TF05, TBL06, TF17 | <b>Testet/Ikke testet</b> | Testet |  |

##### Annet:

|                  |                      |
|------------------|----------------------|
| <b>Status</b>    | Godtatt              |
| <b>Kommentar</b> | Testet, og fungerer. |

Tabell 5: Krav 4F

#### 4.1.5 1IF – GUI i henhold til spesifikasjon

##### Identifisering:

|                                     |   |                           |     |                  |   |
|-------------------------------------|---|---------------------------|-----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver   | <b>ID</b>                 | 1IF | <b>Prioritet</b> | A |
| <b>Beskrivelse</b>                  | GUI'et skal være i henhold til spesifikasjon fra oppdragsgiver. |                           |     |                  |   |
| <b>Dato for registrering</b>        | 09.11.09  | <b>Oppfylt i use case</b> |     | n/a              |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Ikke-funksjonelt  |                           |     |                  |   |

##### Kryssreferanser:

|               |                  |                           |        |  |
|---------------|------------------|---------------------------|--------|--|
| <b>Risiko</b> | R1, R4, R11      |                           |        |  |
| <b>Test</b>   | TB01, TB02, TB03 | <b>Testet/Ikke testet</b> | Testet |  |

##### Annet:

|                  |                                   |
|------------------|-----------------------------------|
| <b>Status</b>    | Godtatt                           |
| <b>Kommentar</b> | Testet, og er slik det skal være. |

Tabell 6: Krav 1IF

## 4.2 B-krav

### 4.2.1 5F - Velge skuddhastighet

#### Identifisering:

|                                     |  |                           |    |                  |   |
|-------------------------------------|--|---------------------------|----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver  | <b>ID</b>                 | 5F | <b>Prioritet</b> | B |
| <b>Beskrivelse</b>                  | Det skal være mulig å velge skuddhastighet/type (serie, singelskudd, automat) ved bruk av "Rate"-knappen i GUI'et. |                           |    |                  |   |
| <b>Dato for registrering</b>        | 09.11.09   | <b>Oppfylt i use case</b> |    | UC5              |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt  |                           |    |                  |   |

#### Kryssreferanser:

|               |                   |                           |        |  |  |
|---------------|-------------------|---------------------------|--------|--|--|
| <b>Risiko</b> | R2, R4, R11       |                           |        |  |  |
| <b>Test</b>   | TF06, TBL05, TF18 | <b>Testet/Ikke testet</b> | Testet |  |  |

#### Annet:

|                  |                                    |
|------------------|------------------------------------|
| <b>Status</b>    | Godtatt                            |
| <b>Kommentar</b> | Testet, og fungerer slik den skal. |

Tabell 7: Krav 5F

#### 4.2.2 6F – Sikte: simulere servoene

##### Identifisering:

|                                     |  |                           |     |                  |   |
|-------------------------------------|--|---------------------------|-----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver  | <b>ID</b>                 | 6F  | <b>Prioritet</b> | B |
| <b>Beskrivelse</b>                  | Simulere servoene slik en de bruker tid på å akselerere opp i riktig hastighet. Det skal også gå en gitt tid for å snu 360 grader. |                           |     |                  |   |
| <b>Dato for registrering</b>        | 09.11.09   | <b>Oppfylt i use case</b> | UC8 |                  |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt  |                           |     |                  |   |

##### Kryssreferanser:

|               |             |                           |        |  |
|---------------|-------------|---------------------------|--------|--|
| <b>Risiko</b> | R2, R4, R11 |                           |        |  |
| <b>Test</b>   | TF07        | <b>Testet/ikke testet</b> | Testet |  |

##### Annet:

|                  |                         |
|------------------|-------------------------|
| <b>Status</b>    | Godtatt.                |
| <b>Kommentar</b> | Fungerer slik det skal. |

Tabell 8: Krav 6F

### 4.2.3 7F - Zoom

#### Identifisering:

|                                     |   |                           |    |                  |   |
|-------------------------------------|---|---------------------------|----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver   | <b>ID</b>                 | 7F | <b>Prioritet</b> | B |
| <b>Beskrivelse</b>                  | Det skal være mulig å zoome inn og ut ved bruk av "Zoom"-knappene på GUI'et.<br><br>Zoom fungerer ved at kameraet beveger seg kjapt på faste steg og har 1x – 52x zoom. Nattkamera og dagkamera har ulike steg. |                           |    |                  |   |
| <b>Dato for registrering</b>        | 09.11.09  | <b>Oppfylt i use case</b> |    | UC7              |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt   |                           |    |                  |   |

#### Kryssreferanser:

|               |                         |                           |        |  |
|---------------|-------------------------|---------------------------|--------|--|
| <b>Risiko</b> | R2, R4, R11             |                           |        |  |
| <b>Test</b>   | TF08, TF09, TBL04, TF19 | <b>Testet/Ikke testet</b> | Testet |  |

#### Annet:

|                  |  |
|------------------|--|
| <b>Status</b>    | Godtatt  |
| <b>Kommentar</b> | Testet, og fungerer. Zoomen i VBS2 har maks limit på 34x zoom, så det er ikke mulig å oppnå høyere zoom enn dette. Stegvis forflytning er ikke tatt med da oppdragsgiver ikke satt dette som en prioritet. |

Tabell 8: Krav 9F

## 4.3 C-Krav

### 4.3.1 8F - Nattmodus

#### Identifisering:

|                                     |  |                           |    |                  |   |
|-------------------------------------|--|---------------------------|----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver  | <b>ID</b>                 | 8F | <b>Prioritet</b> | C |
| <b>Beskrivelse</b>                  | Det skal være mulig å slå på og av nattmodus ved å trykke på "Day/Night"-knappen i GUI'et. |                           |    |                  |   |
| <b>Dato for registrering</b>        | 09.11.09   | <b>Oppfylt i use case</b> |    | UC6              |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt  |                           |    |                  |   |

#### Kryssreferanser:

|               |                        |                           |        |  |
|---------------|------------------------|---------------------------|--------|--|
| <b>Risiko</b> | R3, R4, R11            |                           |        |  |
| <b>Test</b>   | TF10,TF11, TBL03, TF20 | <b>Testet/Ikke testet</b> | Testet |  |

#### Annet:

|                  |  |
|------------------|--|
| <b>Status</b>    | Foreslått  |
| <b>Kommentar</b> | Testet, og fungerende. Kan veksle mellom dag og natt, samt thermal vision. |

Tabell 7: Krav 10F

### 4.3.2 9F – Sikte: sensitivitet på joystick

#### Identifisering:

|                                     |  |                           |     |                  |   |
|-------------------------------------|--|---------------------------|-----|------------------|---|
| <b>Opphav til krav</b>              | Tekst fra oppdragsgiver  | <b>ID</b>                 | 9F  | <b>Prioritet</b> | C |
| <b>Beskrivelse</b>                  | Sette sensitivitet på joysticken til de samme verdiene som i Protector treneren. |                           |     |                  |   |
| <b>Dato for registrering</b>        | 09.11.09   | <b>Oppfylt i use case</b> | UC9 |                  |   |
| <b>Funksjonelt/ikke funksjonelt</b> | Funksjonelt  |                           |     |                  |   |

#### Kryssreferanser:

|               |             |                           |        |  |
|---------------|-------------|---------------------------|--------|--|
| <b>Risiko</b> | R3, R4, R11 |                           |        |  |
| <b>Test</b>   | TF12, TBL08 | <b>Testet/Ikke testet</b> | Testet |  |

#### Annet:

|                  |                                    |
|------------------|------------------------------------|
| <b>Status</b>    | Godtatt                            |
| <b>Kommentar</b> | Testet, og fungerer slik den skal. |

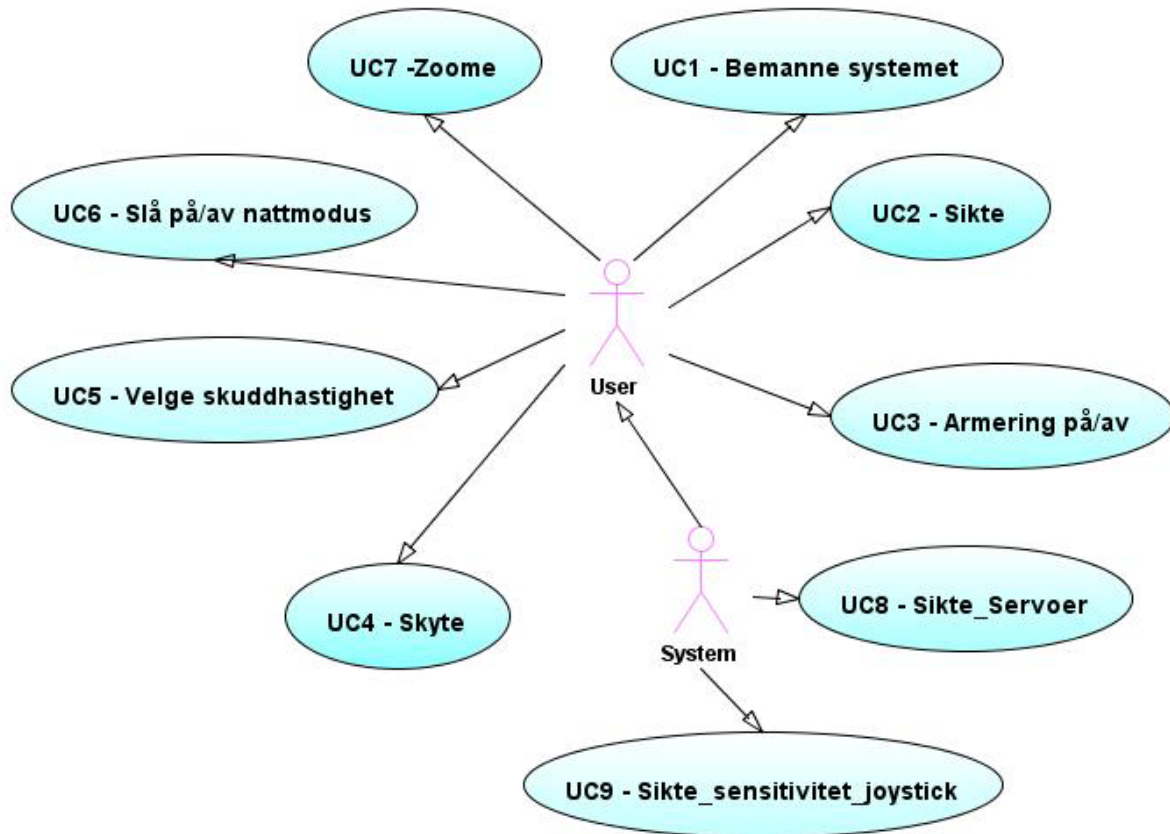
Tabell 7: Krav 11F



## 5 Analyse

---

### 5.1 Use case-diagram



Figur1: Use case-diagram

### 5.2 Beskrivelse av aktører

#### 5.2.1 User:

Brukeren av systemet.

#### 5.2.2 System:

Systemet brukeren bruker til å "spille" på (VBS2).

## 5.3 Deltaljing av use caser

### 5.3.1 Aktør: User

#### 5.3.1.1 UC1 – Bemanne systemet

**Beskrivelse:** Brukeren skal kunne bemanne en RWS.

**Oppfyller krav:** 1F

**Forhåndsbetingelser:**

- Brukeren må være pålogget som soldat i VBS, og det må være en RWS i nærheten som kan bemannes.

**Hendelsesforløp:**

- 1. Brukeren bemanner RWS'en.

**Sluttbetingelser:**

- Systemet er nå bemannet, og operativt.

**Tabell 8: UC1**

### 5.3.1.2 UC2 – Sikte: Elevasjon og rotasjon

**Beskrivelse:** Brukeren skal ha muligheten til å sikte med joysticken.

**Oppfyller krav:** 2F

**Forhåndsbedingungen:**

- Systemet er slått på og operativt.

**Hendelsesforløp:**

- 1. Brukeren sikter seg inn på ønsket mål.
- 2a. brukeren skal kunne vri siktet 360 grader.
- 2b. brukeren skal kunne heve og senke siktet mellom -30 og 60 grader.

**Sluttbedingungen:**

- Brukeren har nå siktet seg inn på ønsket mål med ønsket rotasjon og elevasjon.

**Tabell 9: UC2**

### 5.3.1.3 UC3 – Armering på/av

**Beskrivelse:** Brukeren skal ha muligheten til å armere våpenet.

**Oppfyller krav:** 3F

**Forhåndsbedingungen:**

- Systemet er slått på og operativt.

**Hendelsesforløp:**

- 1. Brukeren flipper bryteren for armering på/av.

**Sluttbedingungen:**

- Våpenet er nå armert, og klart til å fyres av.

**Tabell 10: UC3**

#### 5.3.1.4 UC4 – Skyte

**Beskrivelse:** Brukeren skal ha muligheten til fyre av våpenet.

**Oppfyller krav:** 4F

**Forhåndsbetingelser:**

- Systemet er slått på, operativt og armert.

**Hendelsesforløp:**

1. Brukeren trykker på avtrekker knappen på joysticken.

**Sluttbetingelser:**

- Våpenet fyrer av et skudd.

**Tabell 11: UC4**

#### 5.3.1.5 UC5 – Velge skuddhastighet

**Beskrivelse:** Brukeren skal ha muligheten til å velge skuddhastigheten på våpenet ved å trykke på "Rate"-knappen i displayet.

**Oppfyller krav:** 5F

**Forhåndsbetingelser:**

- Systemet er slått på, operativt og armert.

**Hendelsesforløp:**

1. Brukeren trykker på "Rate"-knappen inntil han har fått opp ønsket skuddtype.

**Sluttbetingelser:**

- Brukeren har nå valgt ønsket skuddhastighet.

**Tabell 12: UC5**

### **5.3.1.6 UC6 – Slå på/av nattmodus**

**Beskrivelse:** Brukeren skal ha muligheten til å slå på eller av nattmodus dersom han/hun ønsker det.

**Oppfyller krav:** 6F

**Forhåndsbetingelser:**

- Systemet er slått på og operativt.

**Hendelsesforløp:**

1. Brukeren trykker på knappen "Night/Day" i displayet.

**Sluttbetingelser:**

- Nattmodusen er nå slått på eller av avhengig av hva som var ønsket.

**Tabell 13: UC6**

### **5.3.1.7 UC7 – Zoome**

**Beskrivelse:** Brukeren skal ha muligheten til å zoome seg inn på ønskede mål.

**Oppfyller krav:** 7F

**Forhåndsbetingelser:**

- Systemet er slått på og operativt.

**Hendelsesforløp:**

1. Brukeren trykker på "zoom"-knappene i displayet eller på joysticken.

**Sluttbetingelser:**

- Brukeren har nå zoomet seg inn på ønsket mål.

**Tabell 14: UC7**

### 5.3.1.8 UC9 – Sikte: sensitivitet på joystick

**Beskrivelse:** Sensitiviteten på joysticken skal kunne endres om ønskelig.

**Oppfyller krav:** 9F

**Forhåndsbetingelser:**

- Systemet er slått på og operativt. Brukeren skal kunne sikte seg inn på ønsket mål med 360 graders rotasjon og -30 til 60 graders elevasjon.

**Hendelsesforløp:**

- 1. Sensitiviteten på joysticken skal kunne endres for å bli mer eller mindre sensitiv ved innsikting på mål.

**Sluttbetingelser:**

- Rotasjon- og elevasjonshastigheten øker eller minker ettersom sensitiviteten på joysticken endres.

**Tabell 16: UC9**

# KDA RIVET



## Presentasjonsrapport 1

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....  | 3 |
| 1.1   | Hensikt med dokumentet.....                                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser.....                 | 3 |
| 1.3   | Dokumenthistorie.....  | 3 |
| 1.4   | Ansvarlig .....  | 3 |
| 2     | Presentasjon 1.....  | 4 |
| 2.1   | Del 1.....   | 4 |
| 2.2   | Resultat del 1 .....   | 4 |
| 2.3   | Del 2.....   | 4 |
| 2.3.1 | Er det noe du vil trekke fram som bra i presentasjonen?..... | 5 |
| 2.3.2 | Er det noe du synes vi burde forbedre?.....                  | 5 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Dette dokumentet er en samling av tilbakemeldingene som ble gitt under prosjektets første presentasjon, og gir svar på hvordan publikum opplevde den.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 19.01.10 | Førsteutkast              | Thomas Transeth    |
| 0.99          | 30.05.10 | Korrekturlest             | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Presentasjon 1

---

Som en del av kvalitetssikringen for prosjektet ble det delt ut et tilbakemeldingsskjema til alle deltakerne under første presentasjon. Ved å få publikum til å skrive positive og negative sider ved presentasjonen vil dette gi et grunnlag for å gjøre presentasjonene mer interessante og informative.

Prosjektgruppen vil bruke dette for å sikre en høyere kvalitet på de to neste presentasjonene.

Tilbakemeldingsskjemaet var todelt og første del bestod av fire spørsmål der en kunne krysse av for god, middels og dårlig. Siste del bestod av to spørsmål der en kunne svare mer utdypende.

### 2.1 Del 1

Følgende spørsmål var gitt i denne delen:

1. Forstod du hva KDA RIVET skal utvikle?
2. Snakket gruppen høyt og tydelig?
3. Var det bra flyt i presentasjonen?
4. Synes du powerpoint-slidene var et bra virkemiddel?

### 2.2 Resultat del 1

| Nr:    | Tilbakemelding |         |        | Totalt | Prosentandel |          |         |
|--------|----------------|---------|--------|--------|--------------|----------|---------|
|        | God            | Middels | Dårlig |        | %God         | %Middels | %Dårlig |
| 1      | 20             | 3       | 0      | 23     | 86,9         | 13,1     | 0       |
| 2      | 21             | 2       | 0      | 23     | 91,3         | 8,7      | 0       |
| 3      | 19             | 4       | 0      | 23     | 82,6         | 17,4     | 0       |
| 4      | 18             | 5       | 0      | 23     | 78,3         | 21,7     | 0       |
| Totalt | 78             | 14      | 0      | 92     | 84,78        | 15,22    | 0       |

### 2.3 Del 2

Følgende spørsmål var gitt i denne delen:

1. Er det noe du vil trekke fram som bra i presentasjonen?
2. Er det noe du synes vi burde forbedre?

### 2.3.1 Er det noe du vil trekke fram som bra i presentasjonen?

Følgende tilbakemeldinger ble gitt:

- Gode illustrasjoner, gode forklaringer.
- Gode til å svare på spørsmål. Godt forberedt med foiler.
- Meget proft.
- Virket som dere hadde peiling.
- God forberedelse, kan svar på spørsmål. Forberedt seg grunnet dokumentrisiko. God flyt.
- Trodde først at kaken var litt tørr, men som jeg til eplebitene. Da oppdaget jeg at kaken var himmelsk god. God struktur.
- Kake! Var generelt bra.
- God mat. Flink til å snakke.
- God presentasjon. Flott med vittigheter.
- Fint presentert.
- Veldig god kake!
- Bra struktur.
- Flyten, litt og velplassert humor, bekledning, gode svar på spørsmål.
- Dere fikk fram det dere skal lage bra.

### 2.3.2 Er det noe du synes vi burde forbedre?

Følgende tilbakemeldinger ble gitt:

- Mer kake.
- Litt for mye "meta for bilder".
- Bedre plass.
- Kake, plass.
- Te, og mer kake!
- Litt vel fargerike slides tok bort fokus.
- Var det avtalte spørsmål?!
- Neida, alt var kjempebra.
- Nei.
- Nei.
- [Veldig god kake!] Følte ut som litt smisking, men det liker vi!
- Rent presentasjonsmessig: Hvis dere skal bevege dere for eksempel for å finne "slider" for å besvare spørsmål, prøv å se selvsikker ut! Ikke smyg dere bort til PC. Ellers; PUST!
- Hvis du ber publikum stille spørsmål, så virker det veldig arrangert å ha slides for spørsmålene.

# KDA RIVET



## Presentasjonsrapport 2

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....  | 3 |
| 1.1   | Hensikt med dokumentet.....                                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser.....                 | 3 |
| 1.3   | Dokumenthistorie.....  | 3 |
| 1.4   | Ansvarlig .....  | 3 |
| 2     | Presentasjon 2.....  | 4 |
| 2.1   | Del 1.....   | 4 |
| 2.2   | Resultat del 1 .....   | 4 |
| 2.3   | Del 2.....   | 5 |
| 2.3.1 | Er det noe du vil trekke fram som bra i presentasjonen?..... | 5 |
| 2.3.2 | Er det noe du synes vi burde forbedre?.....                  | 5 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Dette dokumentet er en samling av tilbakemeldingene som ble gitt under prosjektets andre presentasjon, og gir svar på hvordan publikum opplevde den.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør           |
|---------------|----------|---------------------------|--------------------|
| 0.1           | 16.04.10 | Førsteutkast              | Thomas Transeth    |
| 0.99          | 30.05.10 | Korrekturlesing           | Zoran Vukobratovic |
| 1.0           | 31.05.10 | Klargjort for innlevering | Thomas Transeth    |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Presentasjon 2

---

Som en del av kvalitetssikringen for prosjektet ble det delt ut et tilbakemeldingsskjema til alle deltakerne under andre presentasjon. Ved å få publikum til å skrive positive og negative sider ved presentasjonen vil dette gi et grunnlag for å gjøre presentasjonene mer interessante og informative. Prosjektgruppen vil så bruke dette for å sikre en høyere kvalitet på den siste og avgjørende presentasjonen.

Tilbakemeldingsskjemaet var todelt og første del bestod av fem spørsmål der en kunne krysse av for god, middels og dårlig. Siste del bestod av to spørsmål der en kunne svare mer utdypende.

Det var óg lagt ett spørsmål til i forhold til den tidligere tilbakemeldingen.

### 2.1 Del 1

Følgende spørsmål var gitt i denne delen:

1. Hva synes du om presentasjonen generelt?
2. Forstod du hva KDA RIVET skal utvikle?
3. Snakket gruppen høyt og tydelig?
4. Var det bra flyt i presentasjonen?
5. Synes du powerpoint-slidene var et bra virkemiddel?

### 2.2 Resultat del 1

| Nr:    | Tilbakemelding |         |        |        | Prosentandel |          |         |
|--------|----------------|---------|--------|--------|--------------|----------|---------|
|        | God            | Middels | Dårlig | Totalt | %God         | %Middels | %Dårlig |
| 1      | 25             | 2       | 0      | 27     | 92,6         | 7,4      | 0       |
| 2      | 22             | 5       | 0      | 27     | 81,5         | 18,5     | 0       |
| 3      | 25             | 2       | 0      | 27     | 92,6         | 7,4      | 0       |
| 4      | 25             | 2       | 0      | 27     | 92,6         | 7,4      | 0       |
| 5      | 24             | 3       | 0      | 27     | 88,8         | 11,0     | 0       |
|        |                |         |        |        |              |          |         |
| Totalt | 120            | 15      | 0      | 135    | 89,6         | 10,4     | 0       |

## 2.3 Del 2

Følgende spørsmål var gitt i denne delen:

6. Er det noe du vil trekke fram som bra i presentasjonen?
7. Er det noe du synes vi burde forbedre?

### 2.3.1 Er det noe du vil trekke fram som bra i presentasjonen?

Følgende tilbakemeldinger ble gitt:

- Eplekake.
- Godt humør. Selvsikre talere.
- It seemed smooth and well presented. I did not understand a word sadly, but received a good feeling about it.
- Kaker er bra.
- Godt sammensatt, alle kunne hva de snakket om, og det var tydelig at presentasjonen var godt planlagt!
- Strukturert og ryddig.
- Virket som alt hang bra sammen.
- Litt humor så det ikke ble for tørt/seriøst.
- Lett og forstå for alle som hører på! Gode illustrasjoner.
- Høyt og tydelig uttale og artikulasjon.
- Powerpoint og humor!
- Gir inntrykk at dere vet hva dere snakker om. Kake = bra. Har egen nettside = bra.
- Det var mer kake.
- Godt innøvd, ikke noe "eeh". Thumbs up!

### 2.3.2 Er det noe du synes vi burde forbedre?

Følgende tilbakemeldinger ble gitt:

- Kaldere kaffe.
- Trenger sjokoladekake.
- Større forelesningsplass.
- Present in english next time?? (joke)
- Synes alle må kneppe opp flere sjorteknapper. Vise litt mer hud!
- Thomas må stress litt mindre. Jenta trenger å snakke roligere og tydeligere.
- Kjør SVN istedenfor Dropbox.
- Nah.
- Nei!
- Litt mer mimikk!
- Powerpoint-slidene har for små skalering.
- Ikke egentlig.
- Pekestokk kan være et fint virkemiddel.



- Jeg vil ha cola neste gang.
- Noe kaldt å drikke.
- Mer plass.
- For mye kake denne gang.
- Litt statisk og stivt til tider. Kan til tider høres ut som opplesing fra manus. En litt mer naturlig framføring er ønsket.



# KDA RIVET



## Prototypedokument - DLL

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....                            | 3 |
| 1.1   | Hensikt med dokumentet.....                  | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3   | Dokumenthistorie.....                        | 3 |
| 1.4   | Ansvarlig.....                               | 3 |
| 2     | Dll-prototypen.....                          | 4 |
| 2.1   | Oppsett og oppbygging.....                   | 4 |
| 2.1.1 | RegisterCommandFnc.....                      | 4 |
| 2.1.2 | OnSimulationStep.....                        | 4 |
| 2.1.3 | PluginFunction.....                          | 4 |
| 2.1.4 | Kommunikasjon.....                           | 4 |
| 2.2   | Importerte ekstrarfunksjoner.....            | 5 |
| 3     | Eksempel på funksjonalitet.....              | 6 |
| 3.1   | Mål for prototypen.....                      | 6 |
| 3.2   | Metode for handlingen.....                   | 6 |
| 3.3   | Detaljert beskrivelse av systemet.....       | 6 |
| 4     | Eksempler fra prototypen.....                | 7 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en forklaring på hvordan prototypen fungerer og å gi en beskrivelse på hvordan man setter opp et enkelt, fungerende dll-prosjekt som kommuniserer med VBS2.

## 1.2 Definisjoner, akronymer og forkortelser

**Dll:** Dynamic Link Library

**VBS2:** Virtual Battle Space 2

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                             |
|---------------|----------|--------------------------------|--------------------------------------|
| 1.98          | 19.05.10 | Gjennomgang av egne dokumenter | Thomas Transeth                      |
| 1.99          | 23.05.10 | Korrekturlest                  | Lone Knutsen og Ole-Martin Grøtterud |
| 2.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                      |

## 1.4 Ansvarlig

Thomas Transeth

## 2 Dll-prototypen

---

Før en kan arbeide med en god implementasjon av oppgaven, må både kunnskapen innenfor dll-programmering og forståelsen av hvordan VBS2 kommuniserer med tilleggsprogrammer testes. Dette blir gjort stegvis og dette dokumentet skal gi en forståelse på oppbygging, koding og kommunikasjon mellom dll-filen og VBS2.

**De viktigste punktene er som følger:**

- Oppsett, oppbygging og kommunikasjon
- Importerte ekstrarfunksjoner

### 2.1 Oppsett og oppbygging

VBS2 krever et noe spesielt oppsett av dll-filen og det er derfor mest hensiktsmessig å bruke de predefinerte testbibliotekene som mal for videre arbeid. Når en tar bort all funksjonell kode står en igjen med fire punkter:

#### 2.1.1 RegisterCommandFnc

Denne funksjonen tar for seg initialiseringen av kommandofunksjonen beskrevet under. VBS2 trenger denne funksjonen for å kunne ta i mot eksterne script-kommandoer. Dette er en funksjon som ikke skal endres.

#### 2.1.2 OnSimulationStep

Denne funksjonen blir kalt av VBS2 etter hvert "frame" som blir kjørt. Dette gjør at handlinger som skal skje over tid kan lett manipuleres, men en skal være forsiktig med å legge inn ressurskrevende kode da dette kan føre til lav "frame rate". I funksjonens parameter blir tiden siden siste "frame" tatt i mot.

#### 2.1.3 PluginFunction

Dette er en funksjon som kan bli kalt direkte fra koden internt i VBS2. Dette gjør det mulig å utføre handlinger når en hendelse oppstår i simuleringen. For eksempel kan en bemanning av et kjøretøy eller avfiring av et våpen settes til å kalle på denne funksjonen sammen med en tekststreng. Ut i fra denne tekststrengen vil det være mulighet til å identifisere hva som har hendt og da kjøre riktig kommando.

#### 2.1.4 Kommunikasjon

For all kommunikasjon *fra* VBS2 må funksjonen PluginFunction bli kalt, men når en skal sende kommandoer *til* VBS2 holder det å sende kommandoen som RegisterCommandFnc har predefinert; ExecuteCommand(), med tilhørende argumenter.

## 2.2 Importerte ekstrarfunksjoner

For at det skal være mulig å legge til funksjoner på en fornuftig og ren måte vil størsteparten av all kode bli importert inn til dll-filen. Dette gjør at kommunikasjonen blir like enkel som det er ment den skal være, i tillegg til at det kan lages kode som lett kan videreutvikles senere.

Siden dll-filen direkte bygger på C++ og språket for wxWidgets er det samme, er det mulig å bruke importerte klasser og funksjoner slik en vanligvis gjør i denne typen programmering. Dette er en metode prosjektgruppen kjenner godt til og gir gode muligheter for å lage et program som blir verdig tiden vi bruker.

### 3 Eksempel på funksjonalitet

---

Det er laget flere filer for å teste funksjonaliteten til systemet og følgende er en forklaring på den som tester systemet best hittil.

#### 3.1 Mål for prototypen

Målet for dette systemet er å teste en flerveiskommunikasjon der dll-filen står i sentrum og manipulerer VBS2. Den skal også kunne få et kall tilbake, som igjen skal endre en variabel i systemet og videreføre den.

#### 3.2 Metode for handlingen

For å få til dette er følgende handlinger bestemt:

1. Sett opp en tekst som et hint i VBS2.
2. La simuleringen via en spesifikk handling gi beskjed tilbake om noe har skjedd.
3. Endre teksten i hintet.
4. Skriv teksten som ble sendt i punkt 2, ti ganger i en tekstfil.

#### 3.3 Detaljert beskrivelse av systemet.

Følgende er satt opp i dll-fil og VBS2 scenario:

1. I metoden OnSimulationStep er det satt opp en kommando som gjør at VBS2 setter opp et hint. Denne kommandoen er som følger:

```
ExecuteCommand("hint 'Dette er stadiet før en hendelse'", NULL, 0);
```

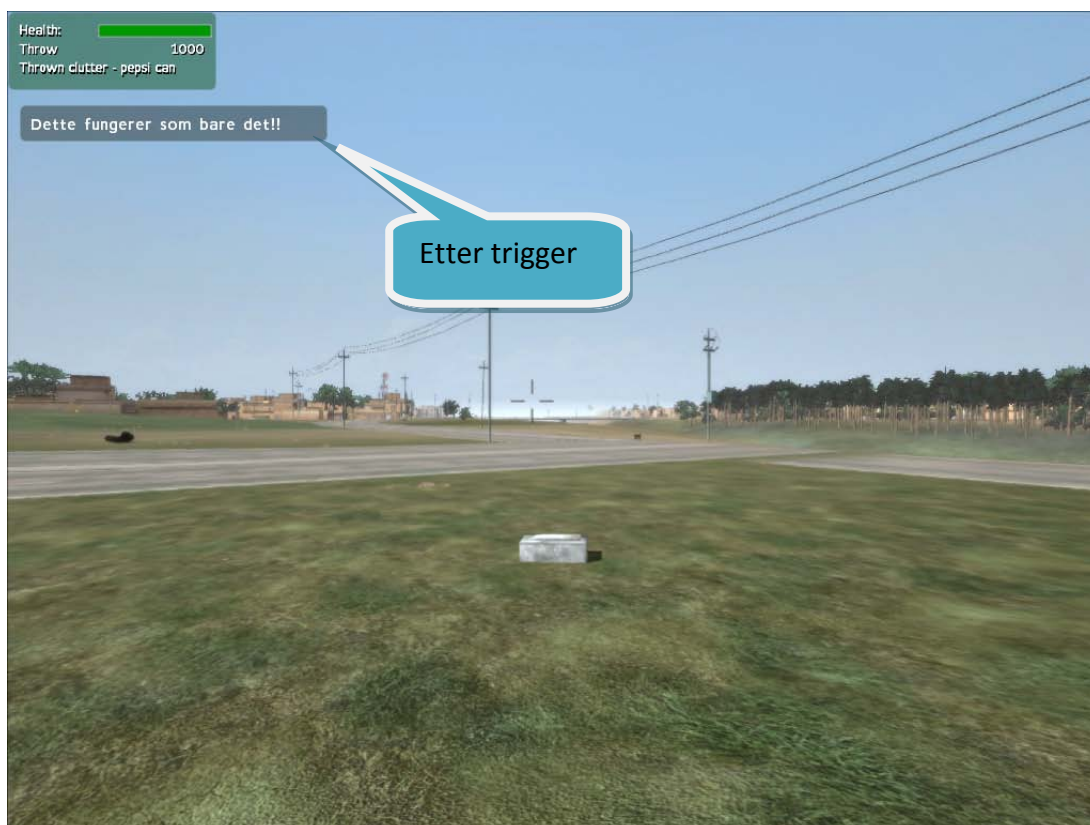
2. For å sikre seg at det er lett å fange opp hendelsen blir det laget et enkelt scenario i VBS2 der en tønne er satt opp i et åpent område. En blir satt til å styre en person som skal gå bort til denne tønningen og da samtidig aktivere en sjekk for om en er kommet innenfor en viss avstand. Det blir da aktivert en kommando som sender teksten "Bazinga!" tilbake til dll-filen.

3. I det PluginFunction blir kalt endres det en boolsk variabel som gjør at kommandoen i OnSimulationStep endres til:

```
ExecuteCommand("hint 'Dette fungerer som bare det!!!'", NULL, 0);
```

4. Rett etter punkt 3 er utført blir en funksjon i en annen klasse kjørt. Denne inneholder kode for å skrive til fil og via en for-løkke skriver den "Bazinga!" ti ganger.

## 4 Eksempler fra prototypen





# KDA RIVET



## Prototypedokument - Joystick

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig.....                               | 3 |
| 2   | Hensikt med prototypen.....                  | 4 |
| 3   | Joystick Prototypen.....                     | 5 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å kartlegge prototypen for joysticken og beskrive det som er gjort. Prototypen skal bruke utprøvd teknologi for å sjekke en enkelt del av systemet. Alt som er gjort i prototypen skal være laget på en slik måte at hoveddelen av det kan tas med videre til implementasjonen.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                                       |
|---------------|----------|--------------------------------|--|
| 1.98          | 19.05.10 | Gjennomgang av egne dokumenter | Ole-Martin Grøtterud                           |
| 1.99          | 24.05.10 | Korrekturlesing                | Arne Kristian Åmellem<br>og Zoran Vukobratovic |
| 2.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                                |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Hensikt med prototypen

---

Hensikten med prototypen er å bruke teknologien vi har så langt til å få input fra joysticken angående posisjon, og printe det ut på skjermen i VBS2. Prototypen tar da for seg input kontroll på joysticken, kommunikasjon til VBS2 og å bygge opp et char array av variabel lengde for å sende det til.



### 3 Joystick Prototypen

---

Prototypen ble satt opp i miljøet til DLL prototypen slik at alle de nødvendige koblinger mellom koden og VBS2 allerede var der.

wxWidgets har en ferdiglaget klasse for å ta imot joystick som vi vil bruke. For å få tilgang til den, må vi legge til wxmsw28d\_adv.lib biblioteket.

Vi må også legge til .h filen for joysticken:

```
#include <wx/joystick.h>
```

#### Joystick peker:

```
wxJoystick *stick;  
stick = new wxJoystick();
```

#### Finder joystick posisjonen og oppdaterer den på skjermen hver frame:

```
VBSPLUGIN_EXPORT void WINAPI OnSimulationStep(float deltaT)  
{  
    //Henter x og y posisjon på joystick  
    int x = stick->GetPosition().x;  
    int y = stick->GetPosition().y;  
  
    //Oppretter ett char array  
    char hi[30];  
  
    //Skriver inn hint  
    hi[0] = 'h';  
    hi[1] = 'i';  
    hi[2] = 'n';  
    hi[3] = 't';  
    hi[4] = ' ';  
    hi[5] = '\\';  
  
    //h forteller hvor neste skal plasseres, settes til 6 fordi 6 tegn  
    allerede er satt inn  
    int h = 6;  
  
    //Tellere for å finne siffer  
    int j = 0;  
    int k = x;  
    //Teller antall siffer i x-koordinaten  
    while(k>10 || k<-10)  
    {  
        k = k/10;  
        j++;  
    }  
  
    //Setter inn x-koordinaten  
    for(int i = 0 ; x>10 || x<-10 ; i++)  
    {  
        hi[h+j-i] = (x%10)+48;  
        x=x/10;  
    }  
}
```

```

hi[h] = x+48;

//Oppdaterer plasseringstilleren
h = h+j;

h++;
//Bindestrek før y-koordinat
hi[h] = '-';
h++;

//Tellere for å finne siffer
j = 0;
k = y;

//Teller antall siffer i y-koordinaten
while(k>10 || k<-10)
{
    k = k/10;
    j++;
}
//Setter inn y-koordinaten
for(int i = 0 ; y>10 || y<-10 ; i++)
{
    hi[h+j-i] = (y%10)+48;
    y=y/10;
}

hi[h] = y+48;

//Avslutter char pekeren ved å sette inn \0
hi[h+j+1] = '\0';
hi[h+j+2] = '\0';

//Kopierer til en const char peker
const char* he = hi;

//Sender kommandoen til VBS2
ExecuteCommand(he, NULL, 0);
}

```

# KDA RIVET



## Prototypedokument - wxWidgets

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Hensikt med prototypen.....                  | 4 |
| 3   | wxWidgets Prototypen .....                   | 5 |
| 3.1 | Design .....                                 | 5 |
| 3.2 | Forklaring av designet.....                  | 6 |





# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å kartlegge prototypen for wxWidgets og beskrive det som er gjort. Prototypen skal bruke utprøvd teknologi for å sjekke en enkelt del av systemet. Alt som er gjort i prototypen skal være laget på en slik måte at hoveddelen av det kan tas med videre til implementasjonen.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                       | Redaktør                                       |
|---------------|----------|---------------------------------|--|
| 1.1           | 04.05.10 | Dokumentgjennomgang og standard | Lone Knutsen                                   |
| 1.98          | 19.05.10 | Gjennomgang av egne dokumenter  | Lone Knutsen                                   |
| 1.99          | 24.05.10 | Korrekturlest                   | Arne Kristian Åmellem<br>og Zoran Vukobratovic |
| 2.0           | 28.05.10 | Klargjort for innlevering       | Thomas Transeth                                |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Hensikt med prototypen

---

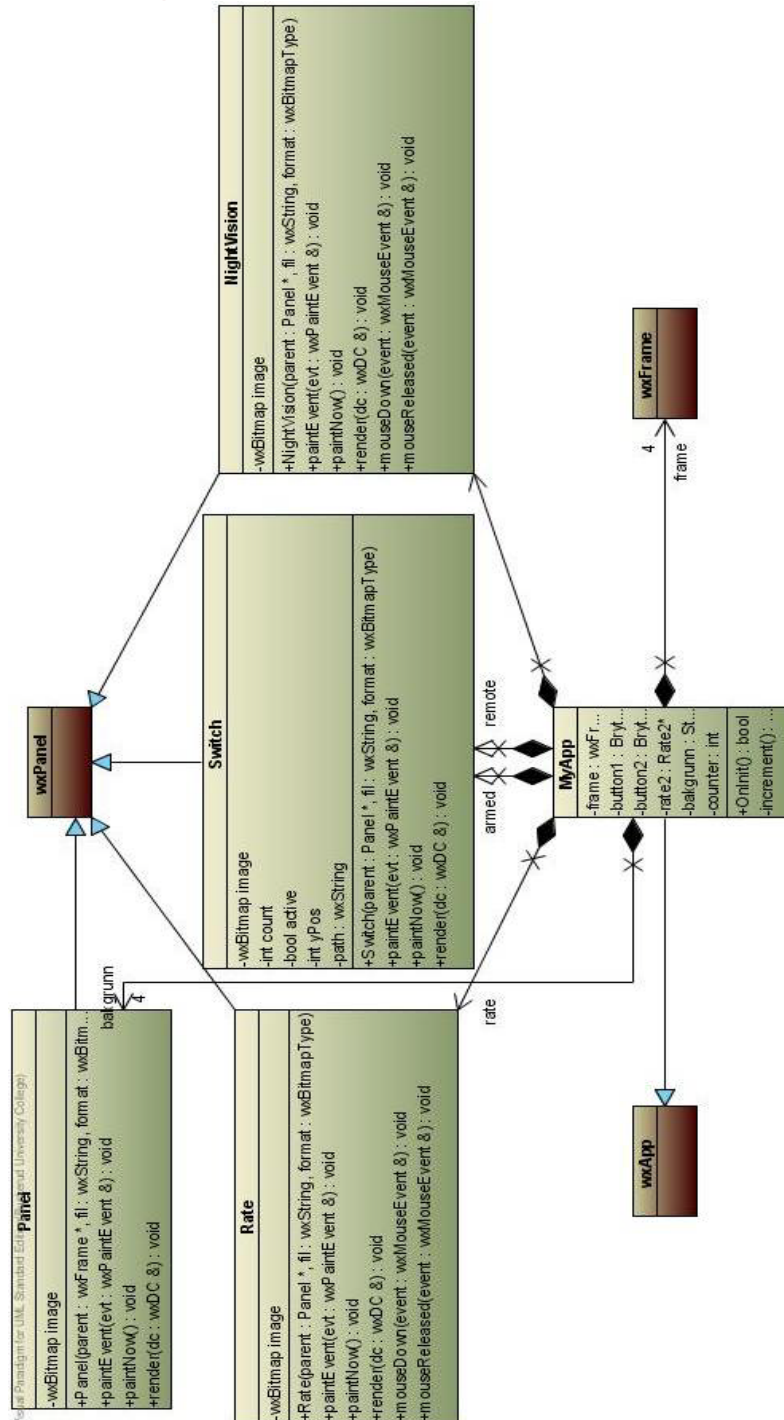
Hensikten med prototypen i wxWidgets er å finne ut om vi får laget det korrekte grensesnittet og om vi får det til å kommunisere mellom bruker og kode ved hjelp av dette biblioteket. For å få til dette er prosjektet satt opp i henhold til Teknologidokument – wxWidgets.



### 3 wxWidgets Prototypen

Prototypen skal settes opp med to brytere og en knapp, som alle skal reagere når de blir trykket på. Bryterne er av typen med en ekstra sikkerhetsflapp, og den må først åpnes før selve bryteren kan trykkes. Bryteren skal skrus av dersom sikkerhetsflappen tas ned mens den er aktivert. Knappene skal utføre samme operasjon hver gang de trykkes ned.

#### 3.1 Design

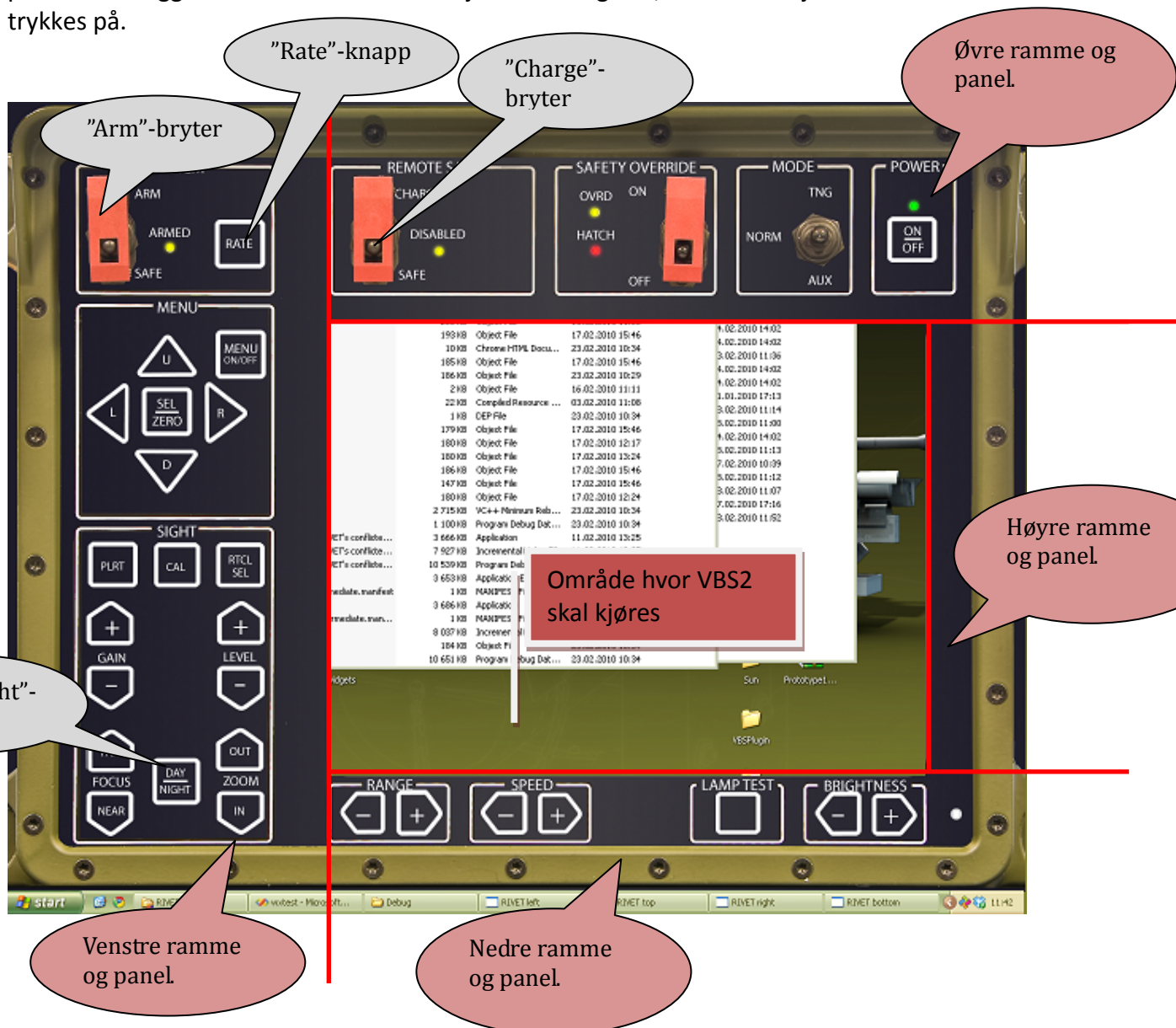


Over er ett designklassediagram av prototypen i et tidlig stadie. Knappene som er lagt til senere, er bygget på samme måte som Rate-klassen.

### 3.2 Forklaring av designet

For å få opp GUI'et slik det skal være så har vi måttet splitte det opp i flere rammer og paneler. Det ble totalt fire rammer, som hver har et tilhørende panel hvor det blir kjørt opp et bilde for å få opp den eksakte bakgrunnen på GUI'et. Dette til sammen danner rammen, og gjør at vi får et "tomt" område på midten som vi kan kjøre VBS2 i. Eller for å si det på en annen måte, vi kan legge GUI'et som en ramme rundt VBS2.

I tillegg så har vi laget fire fungerende knapper, som igjen har forskjellige funksjoner. Vi har en "Day/Night"-knapp, en "Arm"-bryter, en "Charge"-bryter og en "Rate"-knapp. Hver av disse tar inn en parameter som sier hvilket panel de tilhører, og kjøres dermed opp i dette panelet. I tillegg har hver av dem en funksjon som fungerer, slik at det skjer noe når de trykkes på.



# KDA RIVET



## Risikodokument

\_\_\_\_\_  
Lone Knutsen

\_\_\_\_\_  
Arne Kristian Amellem

\_\_\_\_\_  
Zoran Vukobratovic

\_\_\_\_\_  
Ole-Martin Grøttenud

\_\_\_\_\_  
Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                            | 3  |
| 1.1   | Hensikt med dokumentet.....                  | 3  |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3  |
| 1.3   | Dokumenthistorie.....                        | 3  |
| 1.4   | Ansvarlig .....                              | 3  |
| 2     | Innledning .....                             | 4  |
| 3     | Risikoanalyse.....                           | 5  |
| 3.1   | Risikovurderingsmal .....                    | 5  |
| 3.1.1 | Fargekoder for risiko .....                  | 5  |
| 3.2   | Risikoer .....                               | 5  |
| 3.2.1 | Kravrelaterte risikoer .....                 | 5  |
| 3.2.2 | Tekniske risikoer.....                       | 6  |
| 3.2.3 | Menneskelige risikoer .....                  | 6  |
| 3.2.4 | Utenforliggende risikoer .....               | 9  |
| 4     | Oppståtte risikoer .....                     | 10 |
| 5     | Referanser.....                              | 11 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å få en oversikt over de risikoer som kan oppstå i løpet av prosjektet, slik at de skal kunne forebygges. Det skal også stå hvordan man løser risikoer dersom de oppstår og konsekvensene de vil ha for prosjektet.

## 1.2 Definisjoner, akronymer og forkortelser

**KDA:** Kongsberg Defence and Aerospace

**AK:** Arne Kristian Åmellem

**GUI:** Graphical User Inteface

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Ansvar                                   |
|---------------|----------|--------------------------------|--|
| 2.1           | 14.05.10 | Oppdatert risikoer             | Ole-Martin Grøtterud                     |
| 2.98          | 21.05.10 | Gjennomgang av egne dokumenter | Ole-Martin Grøtterud                     |
| 2.99          | 24.05.10 | Korrekturlesing                | Zoran Vukobratovic<br>og Thomas Transeth |
| 3.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                          |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Innledning

---

Risikodokumentet inneholder risikovurderinger av forskjellige risikoer gruppen kan komme over. Risikodokumentet er ikke et endelig dokument før prosjektet er over, og vil bli oppdatert fortløpende. I tillegg til tilføyning av oppståtte risikoer underveis, kan det være mange grunner til at risikovurderinger må endres eller nye føyes til i løpet av prosjektet. En av de vanligste årsakene til dette er at risikoen har blitt laget på feil grunnlag. Det er derfor viktig med god kommunikasjon mellom alle gruppemedlemmene og relevante kontakter, slik at man får tilgang til all relevant informasjon. Vurderingene som gjøres i dette dokumentet er ment for å bistå prosjektgruppen ved gjennomføring av aktiviteter, prioriteringer, strategiske valg av løsninger og prosjektstyring. Ved å kartlegge risikoene på forhånd vil det ikke bare være enklere å unngå de, men også enklere å komme seg videre dersom de oppstår.[1]

Alle risikoene som er vurdert i dette dokumentet blir tallfestet med et risikonummer som kan refereres til av andre dokumenter i prosjektet, dersom dette er nødvendig. Hver risikovurdering skal også inneholde navnet på risikoen og en beskrivelse av den. Risikoene blir så analysert, og får estimert en sannsynlighet samt beskrevet konsekvensen hvis den oppstår med ord[1]. Produktet av sannsynligheten og konsekvensen, blir så tildelt en risikograd, som skal gi helheten i hvor alvorlig risikoen er. I tillegg til dette skal det beskrives hva som kan gjøres for å forebygge risikoen for å minske sannsynligheten for at den oppstår. På alle risikoer som ikke er kravrelaterte, skal det også beskrives en løsning som forklarer hvordan gruppen skal gå frem dersom en risiko inntreffer.



## 3 Risikoanalyse

### 3.1 Risikovurderingsmal

Under er malen på hvordan risikoene er skrevet med forklaringer:


|                      |  |             |
|----------------------|--|-------------|
| <b>[RisikoNr]</b>    | [Risikonavn]   | [Fargekode] |
| <b>Beskrivelse</b>   | [Beskrivelse av risikoen]  |             |
| <b>Sannsynlighet</b> | [Veldig usannsynlig/Usannsynlig/Mulig/Sannsynlig/Veldig Sannsynlig]                  |             |
| <b>Forebygging</b>   | [Tiltak som bør gjøres for å unngå risikoen]   |             |
| <b>Konsekvens</b>    | [Konsekvensene risikoen har for prosjektet hvis den oppstår]                         |             |
| <b>Løsning</b>       | [Tiltak som bør gjøres dersom risikoen oppstår] (Ikke med på kravrelaterte risikoer) |             |


#### 3.1.1 Fargekoder for risiko

|   |                   |
|---|-------------------|
|    | Veldig lav risiko |
|    | Lav risiko        |
|   | Middels risiko    |
|  | Høy risiko        |
|  | Kritisk risiko    |

## 3.2 Risikoer

### 3.2.1 Kravrelaterte risikoer

|                      |  |   |
|----------------------|--|---|
| <b>R1</b>            | Ikke oppfylte A-krav   |  |
| <b>Beskrivelse</b>   | Prosjektgruppen har ikke klart å oppfylle alle A-kravene   |   |
| <b>Sannsynlighet</b> | Usannsynlig  |   |
| <b>Forebygging</b>   | Jobbe målrettet for å få oppfylt de viktigste kravene, overholde frister og velge løsninger med minst mulig problemer              |   |
| <b>Konsekvens</b>    | Konsekvens av ikke oppfylte A-krav vil være ett system som ikke er tilfredsstillende for KDA, og vil mangle viktig funksjonalitet. |   |

|                      |   |   |
|----------------------|---|---|
| <b>R2</b>            | Ikke oppfylte B-krav  |  |
| <b>Beskrivelse</b>   | Prosjektgruppen har ikke klart å oppfylle alle B-kravene  |   |
| <b>Sannsynlighet</b> | Mulig   |   |
| <b>Forebygging</b>   | Jobbe målrettet for å få oppfylt de viktigste kravene, overholde frister og velge løsninger med minst mulig problemer |   |
| <b>Konsekvens</b>    | Konsekvens av ikke oppfylte B-krav vil være ett system som vil mangle ønsket funksjonalitet.                          |   |

|                      |   |  |
|----------------------|---|--|
| <b>R3</b>            | Ikke oppfylte C-krav  |  |
| <b>Beskrivelse</b>   | Prosjektgruppen har ikke klart å oppfylle alle C-kravene  |  |
| <b>Sannsynlighet</b> | Mulig   |  |
| <b>Forebygging</b>   | Jobbe målrettet for å få oppfylt de viktigste kravene, overholde frister og velge løsninger med minst mulig problemer |  |
| <b>Konsekvens</b>    | Konsekvens av ikke oppfylte C-krav vil være ett system som vil mangle noe ekstra funksjonalitet.                      |  |

### 3.2.2 Tekniske risikoer

|                      |  |  |
|----------------------|--|--|
| <b>R4</b>            | Feil GUI løsning   |  |
| <b>Beskrivelse</b>   | Gruppen har valgt en løsning for å lage GUI som ikke fungerer  |  |
| <b>Sannsynlighet</b> | Usannsynlig  |  |
| <b>Forebygging</b>   | Gjennomføre grundig research av mulighetene til å lage GUI for å velge best mulig løsning å jobbe videre på  |  |
| <b>Konsekvens</b>    | Gruppen vil ikke kunne gjennomføre oppgaven på en tilfredsstillende måte   |  |
| <b>Løsning</b>       | Avhengig av hvor langt vi er i prosjektet kan vi prøve oss på en ny og bedre løsning hvis vi har tid til det, hvis ikke må vi gjøre det beste ut av løsningen vi har valgt |  |

|                      |   |  |
|----------------------|---|--|
| <b>R5</b>            | Datatap som følge av PC krasj   |  |
| <b>Beskrivelse</b>   | En pc krasjer av en eller annen grunn slik at dataen på den blir utilgjengelig  |  |
| <b>Sannsynlighet</b> | Mulig   |  |
| <b>Forebygging</b>   | Ta backup ofte  |  |
| <b>Konsekvens</b>    | Alt som er gjort siden siste backup vil bli borte   |  |
| <b>Løsning</b>       | Låne en annen PC til man får orden på pc'en igjen. Hente ned backup og komme seg dit man var før krasjen så fort som mulig. |  |

### 3.2.3 Menneskelige risikoer

|                      |  |  |
|----------------------|--|--|
| <b>R6</b>            | Kortvarig sykdom eller skade   |  |
| <b>Beskrivelse</b>   | Ett eller flere gruppemedlemmer blir borte eller mister arbeidsevne i opptil 3 dager på grunn av sykdom eller skade. |  |
| <b>Sannsynlighet</b> | Sannsynlig   |  |
| <b>Forebygging</b>   | Alle gruppemedlemmene oppfordres til å spise vitaminbjørner  |  |
| <b>Konsekvens</b>    | Oppgavene gruppemedlemmet holder på med blir utsatt  |  |
| <b>Løsning</b>       | Sykt gruppemedlem må ta igjen arbeidet han/hun ikke fikk gjort i sykdomsperioden                                     |  |

|                      |  |  |
|----------------------|--|--|
| <b>R7</b>            | Frafall grunnet fengselsopphold  |  |
| <b>Beskrivelse</b>   | Ett eller flere gruppedlemmer blir satt inne i fengsel og får ikke gjort oppgavene sine            |  |
| <b>Sannsynlighet</b> | Veldig usannsynlig   |  |
| <b>Forebygging</b>   | Alle gruppedlemmene oppfordres til å følge lover og regler, eller være flinke til ikke å bli tatt. |  |
| <b>Konsekvens</b>    | Mer arbeid på de andre så lenge grupped medlemmet er i fengsel                                     |  |
| <b>Løsning</b>       | Fordele personens arbeidsoppgaver til han/hun er ute av fengselet                                  |  |

|                      |  |  |
|----------------------|--|--|
| <b>R8</b>            | Langvarig sykdom, skade eller dødsfall   |  |
| <b>Beskrivelse</b>   | Ett eller flere gruppedlemmer får en langvarig sykdom, skade eller dør, slik at de ikke kan arbeide over lengre tid.   |  |
| <b>Sannsynlighet</b> | Veldig usannsynlig   |  |
| <b>Forebygging</b>   | Alle gruppedlemmene oppfordres til å spise vitaminbjørner, vaske hendene ofte og bruke brodder når de skal gå på isen. |  |
| <b>Konsekvens</b>    | Mye mer arbeid på de andre gruppedlemmene, det kan være nødvendig å revidere kravene med arbeidsgiver                  |  |
| <b>Løsning</b>       | Gå igjennom resterende arbeid og fordele flest mulig oppgaver.   |  |

|                      |  |  |
|----------------------|--|--|
| <b>R9</b>            | Dårlig arbeidsmoral  |  |
| <b>Beskrivelse</b>   | Ett eller flere gruppedlemmer får dårlig arbeidsmoral i løpet av prosjektet.   |  |
| <b>Sannsynlighet</b> | Sannsynlig   |  |
| <b>Forebygging</b>   | "Team building" med "hode skulder kne og tå" eller "hjulene på bussen" før gruppemøtene og deling av morsomme YouTube videoer. |  |
| <b>Konsekvens</b>    | Arbeidet vil gå tregere/stagnere   |  |
| <b>Løsning</b>       | Ha "team building" med "hode skulder kne og tå" eller "hjulene på bussen" og se morsomme YouTube videoer eller Terkel i Knipe. |  |

|                      |  |  |
|----------------------|--|--|
| <b>R10</b>           | Noen blir sinte under "team building"  |  |
| <b>Beskrivelse</b>   | Ett eller flere gruppedlemmer blir sinte når det gjennomføres "team building" tiltak.  |  |
| <b>Sannsynlighet</b> | Mulig  |  |
| <b>Forebygging</b>   | Gruppedlemmene må instrueres i at med "hode skulder kne og tå" og "hjulene på bussen" faktisk er morsomt og forebyggende for R6. |  |
| <b>Konsekvens</b>    | Ole-Martin får seg en på trynet  |  |
| <b>Løsning</b>       | Sende Ole-Martin til legevakten og revurdere "team building" aktivitetene.   |  |

|                      |  |  |
|----------------------|--|--|
| <b>R11</b>           | Mangel på kunnskap   |  |
| <b>Beskrivelse</b>   | Gruppemedlemmene mangler den kunnskapen de trenger for å kunne gjennomføre en eller flere oppgaver i prosjektet. |  |
| <b>Sannsynlighet</b> | Sannsynlig   |  |
| <b>Forebygging</b>   | Produksjon av teknologidokumenter for teknologien vi regner med å bruke  |  |
| <b>Konsekvens</b>    | Oppgaven kan bli utsatt eller i verste fall utført feil  |  |
| <b>Løsning</b>       | Lage teknologidokumenter på teknologien  |  |

|                      |   |  |
|----------------------|---|--|
| <b>R12</b>           | Miste veileder  |  |
| <b>Beskrivelse</b>   | Intern eller ekstern veileder forsvinner eller blir utilgjengelig for gruppa  |  |
| <b>Sannsynlighet</b> | Usannsynlig   |  |
| <b>Forebygging</b>   | Jobbe selvstendig, få tak i mest mulig informasjon tidlig i prosjektet og overbevise ekstern veileder om at piggdekk på sykkel ikke er for pyser. |  |
| <b>Konsekvens</b>    | Det blir vanskeligere å få veiledning om skolens eller KDA sine krav.   |  |
| <b>Løsning</b>       | Finne ny veileder   |  |

|                      |   |  |
|----------------------|---|--|
| <b>R13</b>           | Samlivsbrudd  |  |
| <b>Beskrivelse</b>   | Ett eller flere gruppemedlemmer opplever samlivsbrudd   |  |
| <b>Sannsynlighet</b> | Usannsynlig   |  |
| <b>Forebygging</b>   | Alle må være flinke til å pleie sin bedre halvdel i perioder med lite jobbing og skille mellom privatliv og skole |  |
| <b>Konsekvens</b>    | Gruppemedlemmet kan få dårlig moral hvis han/hun ikke har vært flink nok til å skille mellom privatliv og skole   |  |
| <b>Løsning</b>       | Få en god klem av AK og komme seg videre.   |  |

|                      |   |  |
|----------------------|---|--|
| <b>R16</b>           | Uenighet om teknisk løsning   |  |
| <b>Beskrivelse</b>   | To eller flere gruppemedlemmer er uenige om valg av en teknisk løsning  |  |
| <b>Sannsynlighet</b> | Sannsynlig  |  |
| <b>Forebygging</b>   | Gruppemedlemmene må hjernevaskes til å ha samme mening om alle tekniske løsninger   |  |
| <b>Konsekvens</b>    | Implementasjonen av den tekniske løsningen vil bli forsinket eller må endres  |  |
| <b>Løsning</b>       | De forskjellige løsningene skal legges kort frem på neste gruppemøte med fordeler og ulemper. Flertallet velger beste løsning og konklusjonen er endelig. |  |

### 3.2.4 Utenforliggende risikoer

|                      |   |  |
|----------------------|---|--|
| <b>R14</b>           | KDA går konkurs   |  |
| <b>Beskrivelse</b>   | KDA eller underavdelingen vi har oppdrag for går konkurs eller forsvinner fra Kongsberg |  |
| <b>Sannsynlighet</b> | Veldig usannsynlig  |  |
| <b>Forebygging</b>   | Passe på at pappen til Lone gjør en god jobb(han jobber på KDA)                         |  |
| <b>Konsekvens</b>    | Vi kan selge vår egen trener!   |  |
| <b>Løsning</b>       | Vi må bytte prosjekt eller få skolen til å ta seg av den ekstern sensuren               |  |

|                      |  |  |
|----------------------|--|--|
| <b>R15</b>           | Skolen legges ned  |  |
| <b>Beskrivelse</b>   | Skolen legges ned  |  |
| <b>Sannsynlighet</b> | Veldig usannsynlig   |  |
| <b>Forebygging</b>   | Passe på at intern veileder gjør en god jobb                                     |  |
| <b>Konsekvens</b>    | Vi står uten intern veileder og sensor   |  |
| <b>Løsning</b>       | Vi må finne en annen høgskole eller universitet hvor vi kan fortsette prosjektet |  |

## 4 Oppståtte risikoer

---

Hittil så har R6 oppstått tre ganger. Det ene var et tilfelle av svineinfluensa (H1N1) i høst, det andre var en knekt finger i vinter og det tredje var en runde med omgangssyken. Ellers har vi klart å unngå alle risikoer, enten på grunn av flaks eller fordi vi har vært flinke til å forebygge de kartlagte risikoene på forhånd.

## 5 Referanser

---

[1] Standard Norge, NS 5814:2008 - Krav til risikovurderinger, Standard Online AS, 2008, side 3-13

# KDA RIVET



## Teknologidokument - ASI vs API

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth



## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig.....                               | 3 |
| 2   | Innledning.....                              | 4 |
| 3   | Detaljert beskrivelse av VTK og Fusion.....  | 5 |
| 3.1 | VTK.....                                     | 5 |
| 3.2 | Fusion.....                                  | 5 |
| 4   | Mer om ASI.....                              | 6 |
| 4.1 | ASI.....                                     | 6 |
| 5   | Teknologien i bruk.....                      | 7 |
| 6   | Referanser.....                              | 8 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en innføring i forskjellene mellom ASI som den vanlige VBS2 VTK versjonen bruker, og API som tilleggspakken FUSION bruker.

## 1.2 Definisjoner, akronymer og forkortelser

**GUI:** Graphical user interface

**VBS2:** Virtual Battle Space 2

**VTK:** Virtual Training Kit

**ASI:** Application Scripting Interface

**API:** Application Programming Interface

**AI:** Artificial Intelligence

**NPC:** Non Player Character (en simulert figur som blir styrt av datamaskinen)

**C:** Programmeringsspråket C

**C++:** Programmeringsspråket C++

**DLL:** Dynamic Link Library

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                             |
|---------------|----------|---------------------------|--------------------------------------|
| 0.1           | 27.05.10 | Førsteutkast              | Zoran Vukobratovic                   |
| 0.99          | 27.05.10 | Korrekturlest             | Lone Knutsen & Arne Kristian Åmellem |
| 1.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                      |

## 1.4 Ansvarlig

Zoran Vukobratovic

## 2 Innledning

---

VBS2 bruker sitt eget skriptspråk for å utvide mulighetene for funksjonalitet og logikk. Hele listen over alle skriptkommandoer kan finnes på Bohemia sine sider. [1]

Disse kommandoene kan enten legges inn som "statements" på objekter inne i VBS2 mission editoren, eller de kan settes inn i bl.a. DLL filer som add-on's.



## 3 Detaljert beskrivelse av VTK og Fusion

---

### 3.1 VTK

Virtual Training Kit (VTK) er den grunnleggende pakken av VBS2, og inneholder hele simuleringsmiljøet, samt en rekke predefinerte simuleringsscenarioer. Her brukes scripting for å sette sammen, kontrollere og endre alt fra værforhold til å kommandere mennesker og deres oppførsel. VTK er ikke gratis software, og må kjøpes av Bohemia Interactive til en pris på 15.000kr.

### 3.2 Fusion

Fusion er en add-on modul som selges separat for VBS2, og er utviklet av SimCentric Technologies. Fusion ble utviklet på grunn av ønsket om å kunne simulere ting ved å bruke programmering i stedet for scripting. Det valgte programmeringsspråket er C++, på grunn av kompatibiliteten den har til C biblioteker. Fusion kommer i to versjoner, versjon 1.0 abstraherer scriptingen til VBS2, mens versjon 2.0 gir direkte tilgang til simuleringskjernen i VBS2, og fjerner behovet for å bruke scripting i det hele tatt. Denne softwaren er heller ikke gratis, og kommer med en prislapp på 150.000kr.

## 4 Mer om ASI

---

Alle skriptkommandoer som er relaterte til vårt prosjekt er forklart på wiki siden til Bohemia Interactive under:

[http://community.bistudio.com/wiki/Category:Scripting\\_Commands\\_VBS2](http://community.bistudio.com/wiki/Category:Scripting_Commands_VBS2)

### 4.1 ASI

VBS2 sitt skriptspråk er veldig omfattende[2], og kan brukes til nesten hva som helst. Alle kommandoene har som nevnt flere måter å kunne brukes som. Hvert objekt som legges til i VBS2 Mission Editor kan ha mange "initialization statements" som blir kjørt med en gang en mission starter. Sammen med dette kan det også legges inn triggere som eksekverer script kommandoer eller kjører diverse andre filer som .dll-, .ext- og .sqf-filer.

I alle disse filene kan brukeren spesifisere hendelser, parametere for mission, begrensninger til realitet, kjøring av addons osv.

## 5 Teknologien i bruk

---



I vårt scenario vil VBS2 først kjøre alle "initialization statements" til hvert objekt som har dem, deretter vil den kjøre opp .ext filen som også blir startet sammen med et scenario. Initialization statements har som regel skriptkommandoer som bare påvirker det spesifikke objektet og ikke hele scenarioet, her kan vi f.eks. angi BMI-en (body mass index) til hver figur som er et menneske.

.dll og .ext filene inneholder skripting kommandoer som påvirker scenarioet fortløpende. Her har vi f.eks. skytefunksjonen "fire"[3] som ligger i .dll og som blir sendt via en tekststreng til VBS2 for å avfyre et gitt våpen hver gang noe skjer.

I .ext filen ligger all kode for menyen som kommer opp når vi trykker på "MENU ON/OFF" knappen på GUI'et vårt.

## 6 Referanser

---

[1] Bohemia Interactive Community, *Scripting*,

[http://community.bistudio.com/wiki/Category:Scripting\\_Commands\\_VBS2](http://community.bistudio.com/wiki/Category:Scripting_Commands_VBS2)

(sist besøkt 27.05.10)

[2] Bohemia Interactive, VBS2

[http://virtualbattlespace.vbs2.com/index.php?option=com\\_content&task=view&id=95&Itemid=88](http://virtualbattlespace.vbs2.com/index.php?option=com_content&task=view&id=95&Itemid=88)

(sist besøkt 27.05.10)

[3] Bohemia Interactive Community, *Scripting*,

[http://community.bistudio.com/wiki/fire\\_%28VBS2%29](http://community.bistudio.com/wiki/fire_%28VBS2%29)

(sist besøkt 27.05.10)



# KDA RIVET



## Teknologidokument - Brukergrensesnitt

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth



## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Prototypen .....                             | 4 |
| 3   | Videreutvikling av brukergrensesnittet ..... | 5 |
| 3.1 | Hotkeys og fjerning av "alt + F4" .....      | 5 |
| 4   | Konklusjon.....                              | 6 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å forklare hvordan brukergrensesnittet har blitt utvidet etter at prototypedokumentet ble skrevet, og hvilke nye funksjoner som er lagt til.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                                   |
|---------------|----------|---------------------------|--|
| 0.1           | 20.05.10 | Førsteutkast              | Lone Knutsen                               |
| 0.99          | 27.05.10 | Korrekturlesing           | Zoran Vukobratovic & Arne Kristian Åmellem |
| 1.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                            |

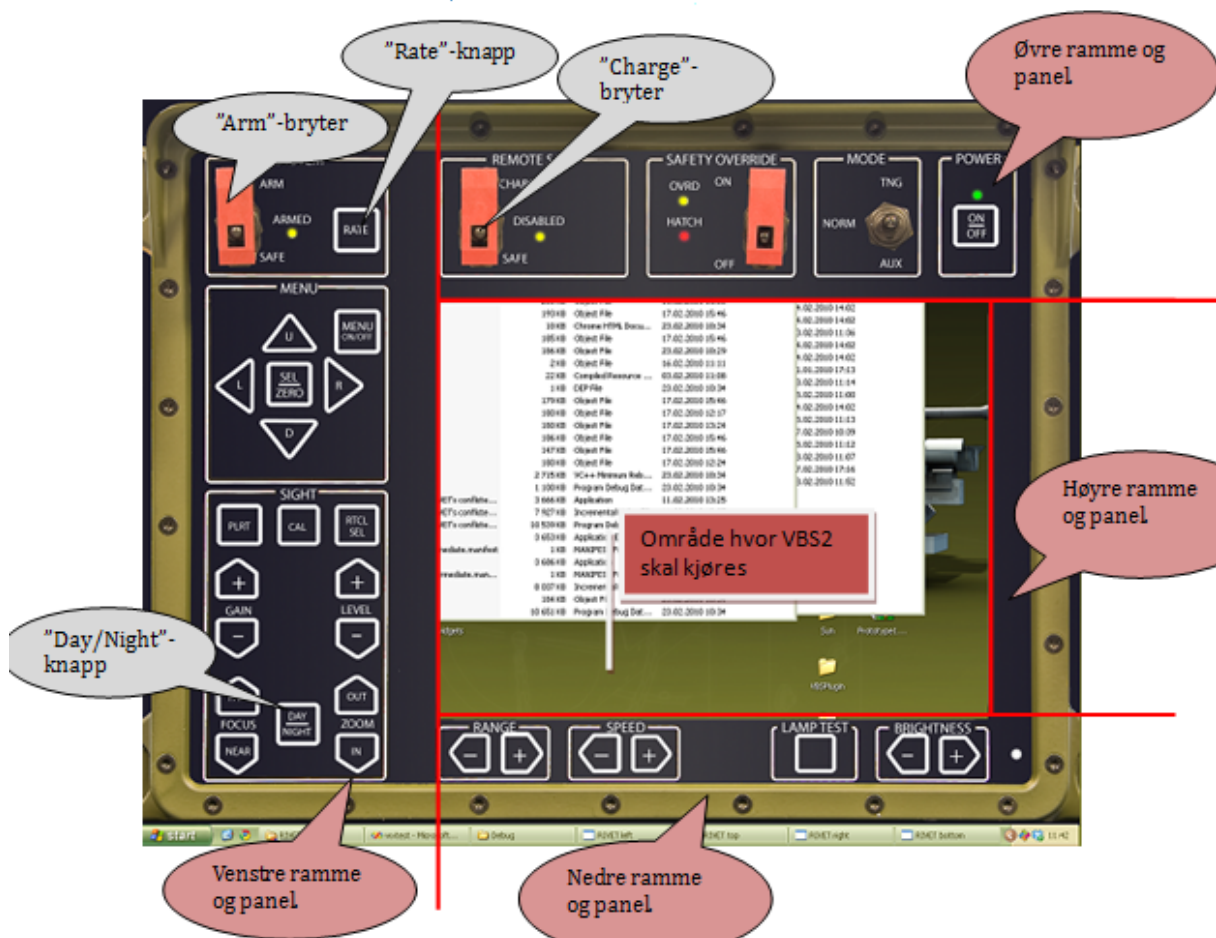
## 1.4 Ansvarlig

Lone Knutsen

## 2 Prototypen

Prototypen vår var delt inn i fire forskjellige rammer, som igjen hadde hvert sitt bildepanel plassert på seg. Rammene var uavhengige av hverandre, og var i utgangspunktet ikke knyttet sammen på noen måte. I tillegg var området i midten der hvor VBS2 plassert uten noen form for "beskyttelse", noe som gjorde det mulig å bevege VBS2-vinduet rundt som man ville.

I prinsippet så var prototypen slik:



### 3 Videreutvikling av brukergrensesnittet

---

I det ferdige produktet, så har det blitt gjort noen endringer på prototypen. Designet ble endret slik at alt nå er avhengig av én hovedramme, med fem underrammer. Fire av underrammene er de samme, og den femte rammen er en gjennomsiktig ramme over VBS2 vinduet, som gjør at dette vinduet ikke lenger kan flyttes på.

Grunnen til at det ble løst på denne måten er at dersom en ramme settes gjennomsiktig, så blir alt annet man legger på denne rammen like gjennomsiktig. Dersom man derimot legger på en ny ramme, og denne ikke er gjennomsiktig, så blir resten av den nye rammen synlig og man har da et synlig bilde og synlige knapper og brytere.

**Kodemessig så er dette utført slik:**

```
wxFrameBack->Show(true);
wxFrameBack->SetTransparent(1);

wxFrameMiddle->Show(true);
wxFrameMiddle->SetTransparent(1);

wxFrameLeft->Show(true);
wxFrameTop->Show(true);
wxFrameRight->Show(true);
wxFrameBottom->Show(true);
```

wxFrameBackSetTransparent(1) og wxFrameMiddle->SetTransparent(1) betyr at disse nå er blitt usynlige.

#### 3.1 Hotkeys og fjerning av "alt + F4"

Slik designet var opprinnelig, så var det slik at dersom man trykket på en av bitene og trykket "alt + F4" så forsvant den delen av brukergrensesnittet man hadde markert. Dette betydde at man fortsatt hadde tre av fire rammer oppe og at man i prinsippet kunne fjerne brukergrensesnittet helt og fortsatt ha VBS2-vinduet åpent. Dette var ikke noen ønskelig situasjon, og for å løse dette fjernet vi "alt + F4" kommandoen fra brukergrensesnittet helt.

**Dette ble gjort ved å legge inn to kodelinjer:**

```
HWND rivetHandle = (HWND)wxFrameLeft->GetHWND();
RegisterHotKey(rivetHandle, 100, MOD_ALT, 0x73);
```

I praksis så vil dette si at vi henter Windows handleren til en av rammene våre og legger inn en hotkey på denne hvor vi setter at den skal ignorere kombinasjonen "alt + F4".

## 4 Konklusjon

---

Vi har gjort noen tekniske endringer i selve designet til brukergrensesnittet og dette fungerer slik som ønsket. Nå er alt avhengig av en hoveddramme og vi har løst utfordringen med at deler av grensesnittet forsvant ved bruk av "alt + F4".

# KDA RIVET



## Teknologidokument - Servo-simulering

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøtterud

---

Thomas Transeth

## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Innledning .....                             | 4 |
| 3   | Beskrivelse .....                            | 4 |
| 3.1 | Statisk fysikk .....                         | 4 |
| 3.2 | Dynamisk modell .....                        | 5 |
| 4   | Konklusjon.....                              | 6 |
| 5   | Referanser.....                              | 7 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å få en oversikt over hvordan vi ser for oss simulering av servoer.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS:** Virtual Battle Space

**RWS:** Remote Weapons System

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                                   |
|---------------|----------|---------------------------|--|
| 0.1           | 27.04.10 | Servo Sim.                | Arne Kristian Åmellem                      |
| 0.99          | 24.05.10 | Korrekturlesing           | Zoran Vukobratovic og Ole-Martin Grøtterud |
| 1.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                            |

## 1.4 Ansvarlig

Arne Kristian Åmellem



## 2 Innledning

---

Som en ingeniør er det viktig å se på helheten og kunne bestemme hva som er relevant for en simuleringsmodell. Velger man å lage en for detaljert modell, er det ofte den blir for komplisert og tar for lang tid å kalkulere i forhold til relevansen. På en annen side lager man en modell som er for enkel er den heller ikke brukbar. Vi har prøvd å finne en balanse mellom disse to for å lage en relevant matematisk modell av hvordan servoene vil oppføre seg fra en operatørs ståsted.

Simulatorer er noe man ønsker at skal være så nøyaktig som overhodet mulig. Det er en mengde variabler som man må ta hensyn til for at en simulering skal bli så nøyaktig som vi ønsker. En måte å få en simulering til å bli så nøyke er å benytte seg av matematisk modellering. Dette er en matematisk modell på et system eller en situasjon som vi ønsker å se på.

I dette dokumentet skal vi skrive om vår matematiske modell.

## 3 Beskrivelse

---

### 3.1 Statisk fysikk

I et servo system for våpen er det viktig at man har et pålitelig og nøyaktig system. En elektrisk servo er en overføring av elektrisk energi til mekanisk. Forsinkelsen fra man beveger på kontrollen, til våpenet beveger seg, er i hovedsak avhengig av servo styrke, friksjon og våpenets vekt. Desto større friksjon og/eller vekt systemet har i forhold til servokraften, desto større vil forsinkelsen være. På en annen side har vi med så kraftige servoer å gjøre at friksjonen og tregheten uansett vil være så lav at vi kan se bort i fra denne.

Vi har også den utfordringen med at ammunisjonskassen har variabel vekt ut i fra mengden ammunisjon den inneholder. Etter å ha sett på systemet så ser vi at vektballansens forskyvning er nominell i forhold til våpenets tyngde og kraft, så vi ser bort ifra dette også.

Fra fysikken er vi kjent formeler for vinkel hastighet og vinkel akselerasjon

Hastighet:

$$\omega = \frac{|\mathbf{v}| \sin(\theta)}{|\mathbf{r}|}$$

Akselerasjon:

$$\alpha = \frac{d\omega}{dt} = \frac{d^2\theta}{dt^2}$$

Ved å bruke disse formlene kan man lage seg en veldig enkel matematisk modell. Men siden vi snakker om rotasjon av en litt tung gjenstand må ta med oss treghetsmomentet. Vi har valgt å se på RWSen som en sylinder for å gjøre den litt enklere.

$$I = \frac{1}{2}mr^2$$

Her ser vi formelen for å beregne treghets moment av en sylinder

Dette vil gi oss en formel som vi kan beregne hvordan en sylinder vil spinne opp fra 0. Den lar oss også beregne tiden det tar for å stoppe redskapen ved en eventuell endring i ønsket rotasjonsretning.

Her ser vi formelen for vinkel hastigheten hvor treghetsmomentet er tatt med.

$$\omega = \omega_f - \omega_I = \left( \frac{|v| \sin(\theta)}{|r|} \right) - \left( \frac{1}{2} m r^2 \right)$$

### 3.2 Dynamisk modell

Først så vi for oss at vi måtte ha en komplisert formel som måtte være ikke lineær. Siden våpenet står på et kulelager, vil friksjonen endre seg ut i fra hastigheten våpenet roterer med.

$$v + i + \sigma \left( 1 - \frac{F}{F_c} \operatorname{sgn} v \right)^\alpha v$$

Dette er i følge Dahl dynamiske modell for kulelager.

Mens i følge Bristle modellen ser den slik ut

$$v + i + \sum_{i=1}^N \sigma_0 (x_i - b_i)$$

Dette viste seg at var noe overdrevent i forhold til det vi trengte til vår simulering. Vårt system er avhengig av at operasjonene ikke er for tunge, da alle beregninger kjøres hver eneste "frame". Dette ville redusert "frame raten" til systemet og økt sjansen for desync.

Etter nærmere studie kom vi frem til at friksjonen er så liten i forhold til servoene sin kraft at det er ubetydelig for vår modell.

## 4 Konklusjon

---

Først ble systemet sett på som alt for detaljert og den matematiske modellen ble alt for tung og komplisert. Vi valgte derfor å se bort i fra elementer som vil påvirke systemet minimalt, dermed forenkle formelen ned til basis fysikk. Rotasjon med treghetsmoment.

## 5 Referanser

---

Hibbeler, Russell C. (2009). [\*Engineering Mechanics\*](#). Upper Saddle River, New Jersey: Pearson Prentice Hall. pp. 314, 153. [ISBN 9780136077916](#).

[Landau, L.D.](#); [Lifshitz, E.M.](#) (1997). *Mechanics*. Butterworth-Heinemann. [ISBN 0-750-62896-0](#).



# KDA RIVET



## Teknologidokument VBS2 - AddOn

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                            | 3  |
| 1.1   | Hensikt med dokumentet.....                  | 3  |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3  |
| 1.3   | Dokumenthistorie.....                        | 3  |
| 1.4   | Ansvarlig .....                              | 3  |
| 2     | Innledning .....                             | 4  |
| 3     | Detaljert beskrivelse .....                  | 5  |
| 3.1   | Utvikingsverktøy .....                       | 5  |
| 3.2   | Modellering .....                            | 5  |
| 3.3   | Eksport og import.....                       | 5  |
| 3.3.1 | FBX.....                                     | 5  |
| 3.3.2 | 3DS: .....                                   | 6  |
| 3.3.3 | P3D Tidlig versjon.....                      | 6  |
| 3.4   | Konfigurasjonsfiler.....                     | 7  |
| 3.5   | PBO-pakking.....                             | 9  |
| 4     | Konklusjon.....                              | 10 |
| 5     | Referanser.....                              | 11 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å få en oversikt over hvordan man lager en Addon til VBS2.

## 1.2 Definisjoner, akronymer og forkortelser

**Add on:** Tilleggsmodul, eller utvidelse av den originale varen.

**3DMAX:** 3D Studio Max

**VBS:** Virtual Battle Space

**PBO:** Pixel Buffer Object

**RWS:** Remote Weapons System

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                                 |
|---------------|----------|---------------------------|--|
| 1.98          | 21.05.10 | Gjennomgang av egne dok.  | Arne Kristian Åmellem                    |
| 1.99          | 24.05.10 | Korrekturlesing           | Zoran Vukobratovic og<br>Thomas Transeth |
| 2.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                          |

## 1.4 Ansvarlig

Arne Kristian Åmellem

## 2 Innledning

---

Utvikling av 3d-modeller er noe man trenger å kunne gjøre om man ønsker å utvikle sine egne kjøretøy, bygninger eller annet innhold til et scenario. VBS2 har i tillegg en rekke ferdige modeller man kan bruke for å befolke den verdenen man lager.

I dette dokumentet skal vi skrive litt om hvordan man skal gå frem for å implementere slike modeller, siden det er en del man må passe på at blir gjort riktig.



## 3 Detaljert beskrivelse

---

Teknologien som beskrives her er utvikling av 3d-modeller til VBS2 og hvordan lage en addon ut av det, som kan brukes i VBS.

### 3.1 Utvilkingsverktøy

For å kunne lage en modell til VBS2 bruker man et 3D-modelleringsverktøy som 3D Studio Max(3DMAX), Oxygen, Maya eller Blender. Vi skal holde oss til 3DMAX siden det er dette - verktøyet som er relevant for gruppen. Man er også nødt til å legge inn en plug-in i 3DMAX, som gjør at man kan eksportere modellen man lager ut til det korrekte filformatet.

3D Studio Max er ett av de mest brukte modelleringsprogrammene. Det er et stort og tungt program som krever litt kunnskap om modellering, men det er også noe av det bedre som er på markedet. Du kan modellere, legge til teksturer og animere i dette verktøyet.

Det må nevnes at Oxygen - også kjent som O2 - er et modelleringsverktøy som er utviklet av Bohemia Interactive med tanke på VBS2. På grunn av noen svakheter i dette verktøyet foretrekker de fleste fremdeles å bruke programmer som 3DMAX eller Maya for å lage selve modellen, men importerer modellene inn i O2 for å optimalisere dem for VBS2.

### 3.2 Modellering

Selve modelleringen av et objekt er ikke noe annerledes enn vanlig. Man lager en modell slik man ønsker at den skal se ut, deretter UV-mapper man modellen og lager teksturkartet. Det er alltid lurt å lagre modellen i standard formatet til programmet man jobber i, før man begynner å forsøke å eksportere modellene.

Når modellen er eksportert til VBS2 kan man bruke Oxygen for å forandre på modellen.

### 3.3 Eksport og import

VBS2 benytter filformatet P3D. Det er det samme filformatet som ble brukt i VBS1, men denne benytter en eldre versjon.

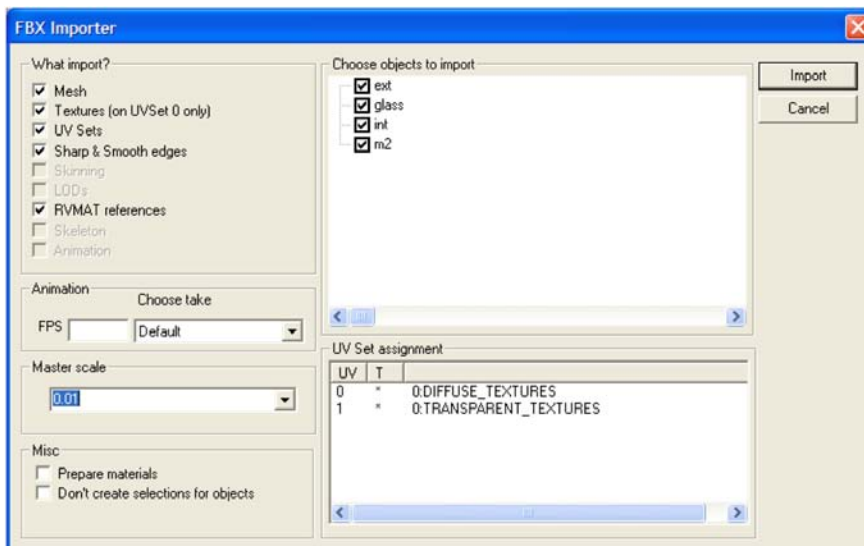
Siden mange ønsker å bruke 3DMAX som utviklingsverktøy, er det laget plugg-ins slik at man kan importere disse modellene inn i Oxygen. Det samme gjelder for FBX formatet som også er mye brukt.

Alle disse filformatene som er beskrevet under kan importeres inn i Oxygen for videre bruk. Når man har fått importert modellen inn i Oxygen kan man lage en pakke/addon (se PBO pakking)

#### 3.3.1 FBX

FBX er en open source plattform, uavhengig 3D filformat som har bedre støtte for viktigere funksjoner enn 3DMAX har. Som nevnt tidligere er det tilgjengelige plug-ins til en rekke programmer.

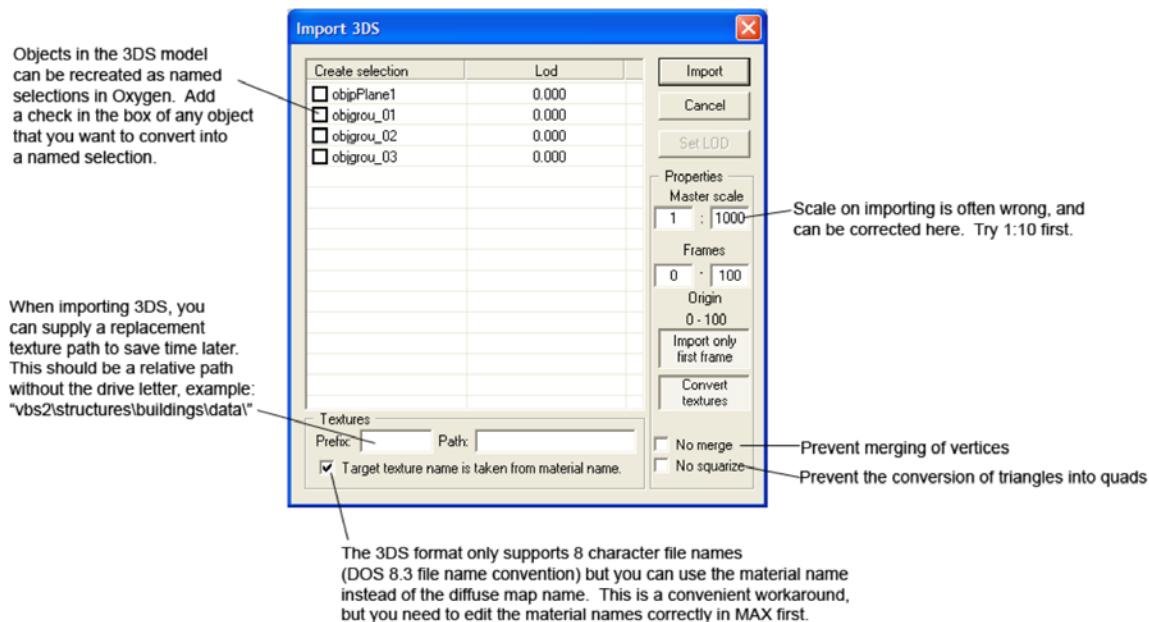
Her ser du vinduet du vil se om du skal importere en FBX modell.



Det å importere er ganske rett frem. Du bare velger skalering og hvilket objekt fra FBX scenen du vil importere. Merk deg at enhver relativ teksturbeliggenhet blir absolutt. Så det lønner seg å legge filene på samme sted som P3D filene lagres. Da slipper man mye unødvendig trøbbel.

### 3.3.2 3DS:

Den største begrensningen ved 3DS-formatet er at teksturene kan kun ha 8 tegn. Vi kan komme rundt dette ved og be Oxygen om å endre Teksturens "map name" isteden for teksturens navn. Dette gjør man eksempelvis ved å navngi materialet med samme navn som teksturen som brukes. For eks. "Bricks\_red" om man har en bricks\_red.tga. Her er et bilde av hvordan import av 3DS formatet ser ut.



### 3.3.3 P3D Tidlig versjon

Denne filstandarden er den som opprinnelig ble brukt for VBS1. Den er en del enklere enn 3DS og FBX. Det er kun eksport funksjon for dette i Oxygen, da det ikke trengs noen import. Filene kan åpnes direkte.

### 3.4 Konfigurasjonsfiler

Konfigurasjonsfiler trenger man for å knytte modellene opp mot VBS2. I disse filene spesifiserer du om et kjøretøy er fiendtlig eller ikke. Du spesifiserer hvilken type mannskap den skal inneholde, om noen, og lyder også alle egenskapene VBS2 trenger for å fungere. Når man lager en konfigurasjonsfil, tar man en som allerede er laget og endrer på denne. Det minimerer sjansen for feil. Minimumskravet på hva man må endre er som vist på figuren under:

Linje 7, prosjektmappe lokasjonen.

Linje 10, navnet på prosjektmappen.

Linje 17, en ny kjøretøysklasse kategori.

Linje 19, et "display name" på den nye kjøretøyklassen.

Linje 30, det nye navnet på kjøretøyklassen, i alfanumeriske tegn, uten mellomrom og lav bindestrektegn skal brukes.

Linje 33, "display name" for dette kjøretøyet.

Linje 34, filnavnet på din P3D modell, uten .p3d filendingen

Linje 35, skriv inn kjøretøyklassen du lagde på linje 17.

Linje 40, om det ikke er noen passasjerer så er transportSoldier=0;

Linje 44, ingen endringer gjort i dette eksempelet, men pass på at driverAction stringen har samme filnavn som navnet på sjåførposisjonen på modellen.

Linje 47 og 48, Her kontrollerer vi ikoner. Setter man adressen til "" så deaktiverer man ikoner, og forhindrer at vi får feilmeldinger in game.

```

1 //Always include the define section in all configs at the top of the file
2 #include "vbs2\customer\defines.hpp"
3
4
5
6 //Always change this to be the path (or full path) to the directory you are working in
7 #define __CurrentDir__ "vbs2\customer\vehicles\bia_basicvehicles"
8
9 //Put the end of the folder you are working in here (or path, just for names)
10 FOLDER_NAME(bia_basicvehicles);
11
12
13
14 //All entities we are creating categories for the listing of objects in the editor
15 class CfgVehicleClasses
16 {
17     class bia_basic_vehicles_category //This class is used later
18     {
19         displayname = "BIA Basic Vehicles"; //Always name for this category then press up to edit
20     };
21 };
22
23
24
25 class VBS2_Fusses_king; //This object from the base game
26
27 class CfgVehicles
28 {
29     class vbs2_Car; //This class for all created vehicles
30     class bia_baseCar : VBS2_Car //Always use vbs2_baseCar with our own classes
31     {
32         scope = public; //Always the class is public in the editor
33         displayName = "Basic Car"; //Name of vehicle as shown in the editor
34         model = __CurrentDir__\bia_basecar; //Name of file used
35         vehicleClass = bia_basic_vehicles_category; //This class was created in CfgVehicleClasses above
36
37         //Color
38         side = 000_0000; //Color you this vehicle belongs to (Always 000_0000, 000_0000, 000_0000, 000_0000)
39         crew = vbs2_tq_sneurg_01_ahm; //Response of the type of crew that pilots the vehicle (used in editor)
40         transportSlotIDler = 0; //How many slots can be carried in cargo (Just look the number of provide in the model as well)
41         typicalCargo[] = {vbs2_tq_sneurg_01_ahm}; //Categories of provide that can start out in each cargo slot (Leave empty for none)
42
43         //PROPRIETARY
44         driveAction = vbs2_Default_Drives; //Response of the parameter used by the driver. A listing of all items will be available in a later update
45         cargoAction[] = {vbs2_Default_Cargo[]}; //Categories of priorities used by cargo
46         wheelCircumference = 1.0; //Circumference of the wheels, in meters. Used to properly load what vehicles while driving
47         picture = ""; //Name is ignored but for vehicle
48         preview = ""; //Ignore preview picture
49         icon = vbs2_icon_nato_opfor_Transport; //Always use Path listing of picture will be available in a later update
50
51         //Used for the vehicle engine Parameter
52         //Get the sound file, the file is not used in sound (taken from F:\vbs2\obj\audio\obj\prop\mod\fan\Default\fan_of_normal\fanmod01.p3d)
53         soundEngine[] = {__CurrentDir__\data\sound\400_100p1, CR_ENGINE_CAR, 1};
54
55         //armor
56         armor = 0.1; //This number, along with the size of the vehicle, determines how much damage it can take before destruction
57         threat[] = {0, 0, 0}; //How dangerous (0-1) the AI considers this vehicle versus enemy types (0=AI, 1=Human, 2=)
58         unloadInCombat = true; //If true, this is cargo (not driver/gunner) will unload during engagements
59         crewVulnerable = true; //If true, the AI will shoot it up into the grass. If false, AI will only target the vehicle, not the crew.
60
61         //MOBILITY
62         maxSpeed = 30; //How fast vehicle speed (in m/s), m/s
63         terrainCoef = 0.0; //How much rough terrain of terrain vehicle speed (higher or vehicle is slower in rough terrain)
64         turnCoef = 0.0; //How fast can slowly the vehicle can turn
65         brakeDistance = 1.7; //Response of vehicle brake
66         canFly = false; //If true, vehicle can fly (using)
67         fuelCapacity = 0.0; //Fuel capacity, in m/s (1000 liter units)
68     };
69 };
70

```



### 3.5 PBO-pakking

For å få brukt modellen og konfigurasjonsfilen i VBS2 så må man lage en addon-pakke som skal legges et bestemt sted slikt at VBS finner den.

Det du må gjøre er å åpne PBO-pakkeren i Oxygen, og peke på mappen som filene dine ligger i (de må ligge alene). Så kjører du pakkeprogrammet. Det lønner seg å sjekke loggen etter du har laget pakken, den vil vise deg om det er noen problemer og vil spare deg for en masse bortkastet tid. Du finner loggen i for eksempel: `export\logs\tag_projectname.log`

Når pakken er laget uten feil, kan du legge den i VBS2 sin addons-undermappe. Deretter er det bare å fyre opp VBS2, gå inn i editoren og finne modellen. Plassere den ut og se om den fungerer som den skal.

## 4 Konklusjon

---

Det å implementere en egen modell i VBS2 skal ikke bli veldig vanskelig. Vi kommer nok til å pirke litt med konfigurasjonsfilene for å få det helt riktig, men det å importere en egenprodusert eller kjøpt modell skal gå helt uproblematisk.



## 5 Referanser

---

Bohemia Interactive Australia, *VBS2*, <http://virtualbattlespace.vbs2.com/>, (sist besøkt 30.12.09)

<http://developer.vbs2.com/manuals/>

Bohemia Interactive Australia, *Adding content*, <http://developer.vbs2.com/onlinehelp/>, (sist besøkt 30.12.09)

# KDA RIVET



## Teknologidokument - Script

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth



## Innhold

---

|     |  |   |
|-----|--|---|
| 1   | Introduksjon.....                            | 3 |
| 1.1 | Hensikt med dokumentet.....                  | 3 |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3 |
| 1.3 | Dokumenthistorie.....                        | 3 |
| 1.4 | Ansvarlig .....                              | 3 |
| 2   | Innledning .....                             | 4 |
| 3   | Detaljert beskrivelse .....                  | 5 |
| 3.1 | Description.ext.....                         | 5 |
| 3.2 | .sqf-filer.....                              | 5 |
| 3.3 | Via ASI.....                                 | 5 |
| 4   | Teknologien i bruk.....                      | 6 |
| 5   | Referanser.....                              | 7 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en innføring i VBS2 sitt eget skripspråk. Hva det er, hvordan det brukes og hvilke muligheter man har.

## 1.2 Definisjoner, akronymer og forkortelser

**HUD:** Heads-Up Display

**VBS:** Virtual Battle Space

**GUI:** Graphical User Interface

**UI:** User Interface

**RWS:** Remote Weapons System

**AI:** Artificial Intelligence

**ASI:** Application Scripting Interface

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                           |
|---------------|----------|---------------------------|------------------------------------|
| 0.1           | 04.05.10 | Førsteutkast              | Ole-Martin Grøtterud               |
| 0.99          | 23.05.10 | Korrekturlest             | Lone Knutsen og<br>Thomas Transeth |
| 1.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                    |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Innledning

---

VBS2 bruker et eget skriptspråk for å utvide mulighetene for funksjonalitet og logikk. Disse kan enten legges inn som "statements" forskjellige steder inne i VBS2 editoren, eller legges ut som egne skriptfiler.

Skriptfilene for et scenario ligger under Mine dokumenter/VBS2 Profiles(Evt VBS2 Other profiles)/"Scenario navn". Det er filene description.ext og alle .sqf filer som er de mest interessante i skriptsammenheng. [1]



## 3 Detaljert beskrivelse

---

Alle skriptkommandoer som kan brukes er forklart på wiki siden til Bohemia Interactive under: [http://community.bistudio.com/wiki/Category:Scripting\\_Commands](http://community.bistudio.com/wiki/Category:Scripting_Commands)

### 3.1 Description.ext

Description.ext ligger under hvert enkelt scenario, som forklart i innledningen, og blir kjørt når det lastes. I denne kan man spesifisere mange hendelser for scenarioet ved hjelp av skript, og definere selvlagde skriptklasser. Eksempelvis kan man spesifisere om spilleren skal ha muligheten til å "respawne", hvordan han/hun skal gjøre dette og hvor lang tid det tar før han/hun gjør dette. I tillegg kan man sette opp egenlagde GUI-komponenter, lydavspilling eller logiske funksjoner som skal brukes i løpet av sesjonen.[2]

### 3.2 .sqf-filer

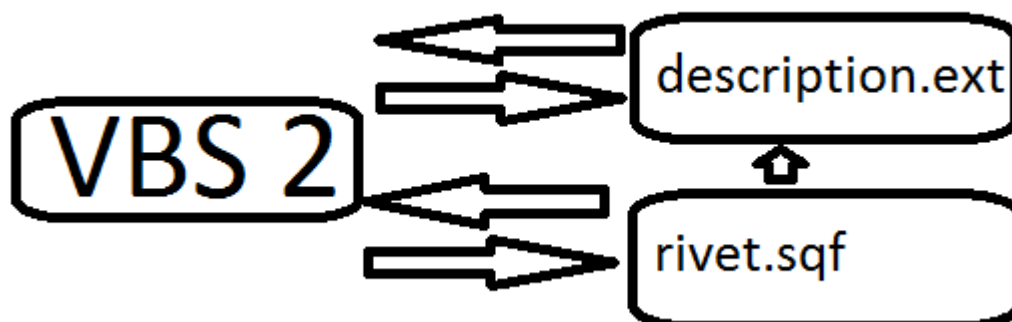
Sqf filer er skriptfiler som enten kan inkluderes i Description.ext for å dele den opp i mindre deler, eller kjøres separat. De kan inneholde logisk kode og deler variabler med de som er deklartert inne i VBS2 scenarioet. For å kjøre en skriptfil må man kalle de opp på formen: nul = [argument] execVM "myscript.sqf" [1]. Hvor argument kan være tomt for ikke å sende med noe. Skriptfilen kan ikke inneholde klassesdefinisjoner, men kan opprette klasser som ble definert i description.ext eller filer som ble inkludert i den.

### 3.3 Via ASI

Ved å bruke ExecuteCommand-funksjonen som ligger i ASI delen via C++, kan vi eksekvere skript inne i VBS2. Denne sender med en const char\* som skal bestå av skriptlinjen som skal kjøres.

## 4 Teknologien i bruk

---



I vårt scenario, vil VBS2 først laste Description.ext. I denne deklarerer vi alle menyklassene som brukes av menyen, i tillegg til scenariospesifikke variabler.

Rivet.sqf vil bli kjørt via VBS2 skript fra koden idet vi trykker på menyknappen. I denne ligger kode for å opprette riktig menyklasse fra description og lukke de igjen, utifra en global variabel i VBS2.

## 5 Referanser

---

- [1]Bohemia Interactive Community, *Scripting*,  
[http://community.bistudio.com/wiki/Category:Scripting\\_Topics/](http://community.bistudio.com/wiki/Category:Scripting_Topics/), (sist besøkt 20.05.10)
- [2]Bohemia Interactive Community, *description.ext*,  
<http://community.bistudio.com/wiki/description.ext/>, (sist besøkt 20.05.10)

# KDA RIVET



## Teknologidokument - VBS2

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|     |  |    |
|-----|--|----|
| 1   | Introduksjon.....                            | 3  |
| 1.1 | Hensikt med dokumentet.....                  | 3  |
| 1.2 | Definisjoner, akronymer og forkortelser..... | 3  |
| 1.3 | Dokumenthistorie.....                        | 3  |
| 1.4 | Ansvarlig.....                               | 3  |
| 2   | Innledning.....                              | 4  |
| 3   | Detaljert beskrivelse.....                   | 5  |
| 3.1 | Scenarier.....                               | 5  |
| 3.2 | Tilgang.....                                 | 5  |
| 3.3 | ASI.....                                     | 5  |
| 3.4 | Tilføyning av egne modeller.....             | 6  |
| 4   | Teknologien i bruk.....                      | 7  |
| 4.1 | Plassering av objekter.....                  | 7  |
| 4.2 | Egenskaper for objekter.....                 | 8  |
| 4.3 | Bevege objekter:.....                        | 9  |
| 5   | Konklusjon.....                              | 10 |
| 6   | Referanser.....                              | 11 |





# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en forklaring av VBS2, hvordan det fungerer og hvordan man oppretter nye scenarioer.

## 1.2 Definisjoner, akronymer og forkortelser

**HUD:** Heads-Up Display

**VBS:** Virtual Battle Space

**GUI:** Graphical User Interface

**UI:** User Interface

**RWS:** Remote Weapons System

**AI:** Artificial Intelligence

**ASI:** Application Scripting Interface

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                              |
|---------------|----------|--------------------------------|---------------------------------------|
| 1.1           | 04.05.10 | Ny dokumentstandard            | Ole-Martin Grøtterud                  |
| 1.98          | 19.05.10 | Gjennomgang av egne dokumenter | Ole-Martin Grøtterud                  |
| 1.99          | 23.05.10 | Korrekturlest                  | Lone Knutsen og Arne Kristian Åmellem |
| 2.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                       |

## 1.4 Ansvarlig

Ole-Martin Grøtterud



## 2 Innledning

---

VBS2 er et fullstendig 3-dimensjonelt treningsystem med bred støtte for trening og utprøving av militære situasjoner eller lignende. Det er laget av Bohemia Interactive Australia og gir virtuell og konstruktiv innsikt i ett realistisk miljø. VBS2 blir brukt av blant annet US Army, USMC, ADF and UK MoD.[1]



### 3 Detaljert beskrivelse

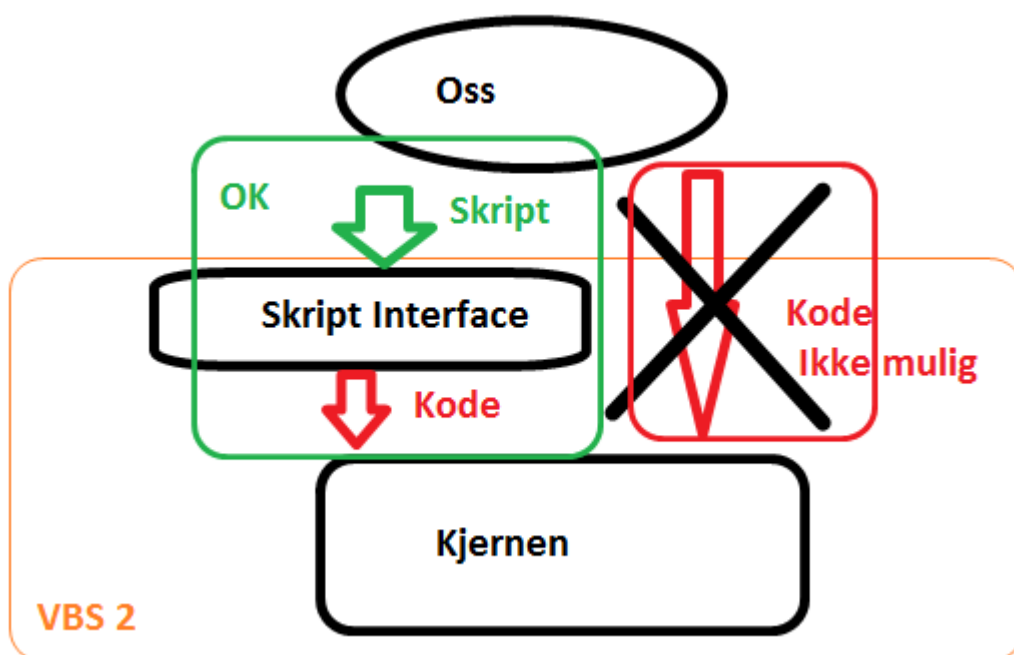
Teknologien som beskrives her er for VBS2 VTK som er den versjonen prosjektgruppen har fått utdelt.

#### 3.1 Scenarier

For å kunne lage scenarier i VBS2 brukes en editor som er inne i administrator delen av programmet. Dette gir deg tilgang til å sette ut alt fra esler til mennesker og helikoptre, og å bestemme oppførselen deres. Man kan derimot ikke sette ut bygninger eller forandre på terrenget i denne.

#### 3.2 Tilgang

I VBS2 er selve kjernen av programmet låst, det er derimot ett lag med skripting over dette som gjør at vi får mulighet til å gjøre modifikasjoner inntil en viss grad. Dette gjør at vi må gjøre hacks med skripting, for å klare å oppnå målene våre. Siden programmet ikke er designet for å kunne lage ett egendefinert grensesnitt. Det kan også bety at det ikke ligger støtte for det vi trenger og at oppgaven viser seg å være ugjennomførbar på en tilfredsstillende måte.



Tegningen over viser hvordan vi kan bruke VBS og hvordan vi ikke kan bruke det. Vi har ikke mulighet til å gjøre noe direkte med kjernen, noe som vil si at alt må gå gjennom "script interfacet".

#### 3.3 ASI

VBS 2 har noe som heter Application Scripting Interface(ASI), som lar brukere kommunisere med VBS2 gjennom andre programmer ved å sende script kommandoer. Dette er nærmere beskrevet i Teknologidokument – VTK vs Fusion.

### 3.4 Tilføyning av egne modeller

Det er mulig å legge til egenlagde modeller i VBS2, alt som trengs en 3d modell og to konfigurasjonsfiler. [3]

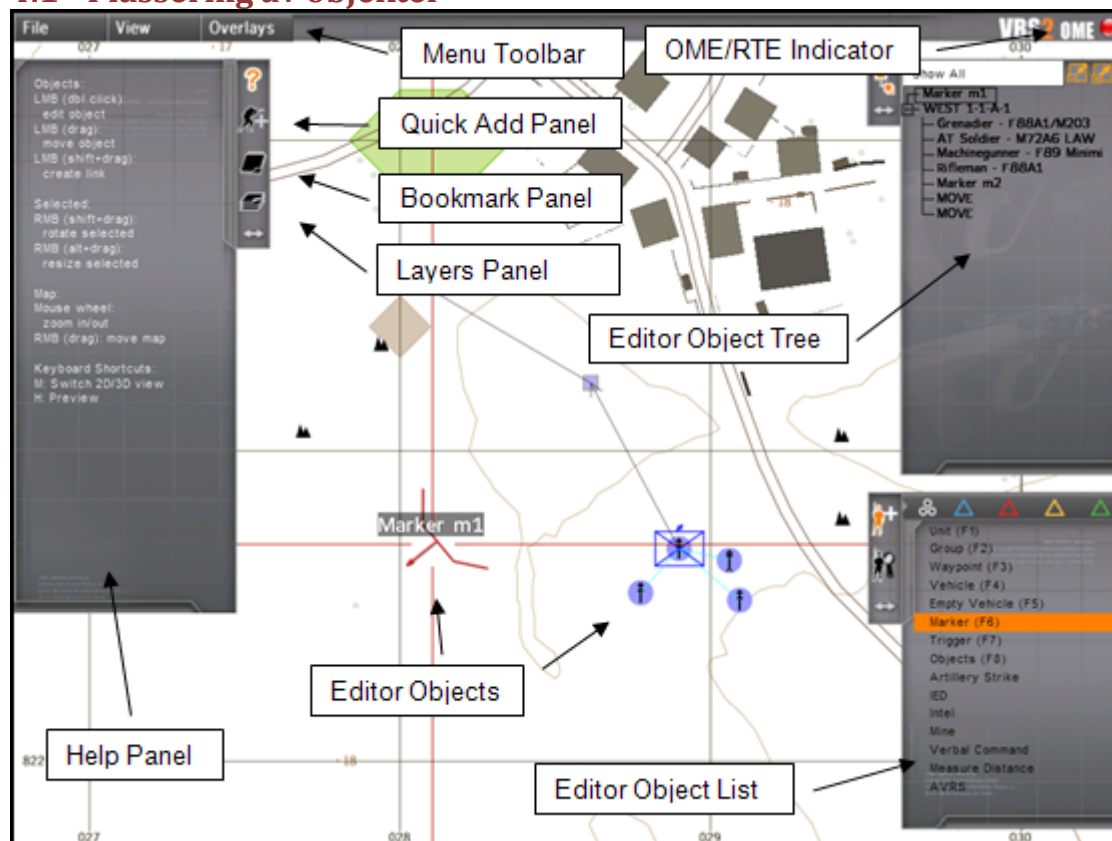
Dette er nærmere beskrevet i Teknologidokument – VBS Add-on



## 4 Teknologien i bruk

Følgende viser hvordan man kan bruke Scenario Editoren i administratordelen av VBS2 til å lage ett scenario.

### 4.1 Plassering av objekter

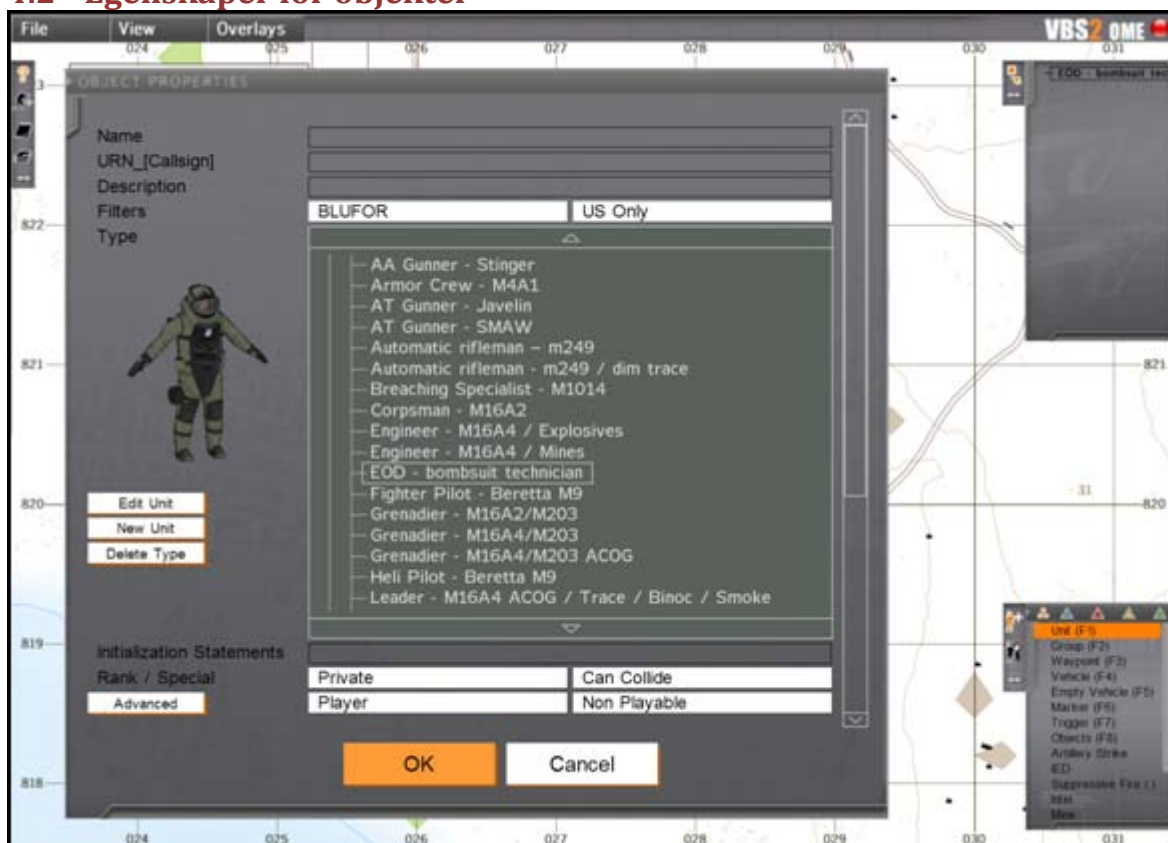


Panelet nede til høyre kalt "Editor Object List", er det vi kommer til å bruke mest. Det er der vi kan spesifisere nye modeller, og legge inn parametere slik vi vil ha de. For å legge til modeller må vi først velge type vi trenger, se f.eks vehicle, empty vehicle, eller trigger, og sørge for at vi er i Add modus (mannen i oransje med + tegn på siden av "Editor Object List" vinduet). Deretter dobbelt klikker vi der vi vil ha objektet på kartet. Det vil da åpnes et properties-vindu for det vi legger til, trykk OK for å sette ut modellen med standard innstillinger. [2]

**For å rotere objekter:** Hold nede skift og høyre museknapp og dra musa til sidene. Dobbeltklikk på ett objekt for å få opp properties vinduet på dem igjen.

**Quick Add Panel:** Velg det du vil legge inn, hold høyre Ctrl inne og trykk inn der du vil ha objektet.

## 4.2 Egenskaper for objekter



**Navn:** Navnet vi refererer til i scripts. For eksempel burde spillerens avatar hete Player eller lignende.

URN\_ kan vi se bort ifra.

**Description:** Hvis du har veldig lyst til å informere verden om at personen liker fargen lyseblå og har brune øyne selvom det ikke syntes på modellen. Evt viktig informasjon om hva hensikten med objektet er. [2]

**Filters:** Filtrering av listen under for å velge modell, merk at du ikke kan velge en BLUFOR(venner) modell og sette den inn som OPFOR(fienden), samme for vica versa og civ etc. Det må da opprettes en ny modell med samme skin med andre configs. [2]

**Initialization Statements:** Skriptlinjer som skal kjøres når den settes ut. F.eks vil vi kanskje bruke Player moveInGunner bilMedRWS for å flytte spilleren inn som gunner i en bilMedRWS bilen. Man kan også si at player skal starte som gunner, men man kan da ikke spesifisere hva slags modell player er. [2]

**Rank:** Høyere rank gjør at du kan gi ordre til botter, botter med høyere rank vil gi ordre til deg.

**Special:** "In Cargo" hvis objektet skal starte som last i en bil. "Flying" hvis det er flyvende objekter. "In formation" er nyttig hvis du vil at objektet skal starte i formasjon rundt gruppelederen(mer om det senere) i stedet for nøyaktig hvor den er plassert. "Can collide(default)" hvis du vil at den skal stå der den står(den vil unngå å kolliderer med noe). [2]

**Player – Non Playable:** Om den skal være spillbar eller AI styrt som default. Spillbare modeller vil kun være der om det er en spiller for den.

**Player – Playable:** Samme effekt som over.

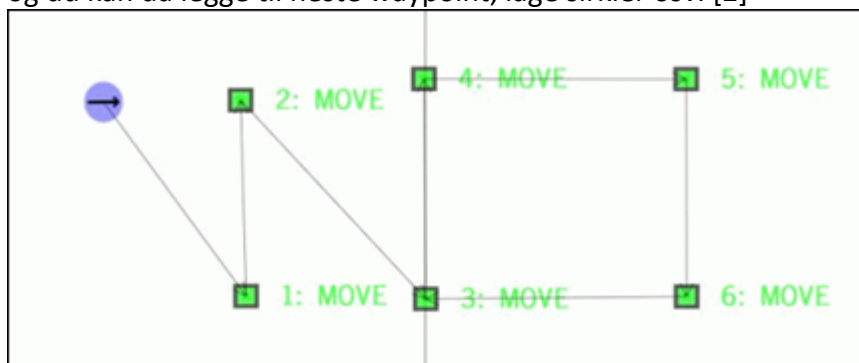
**AI – Non Playable:** En ikke spillbar AI.

**AI – Playable:** AI som utgangspunkt, men kan bli brukt av spiller. (Driver i vår sammenheng.) [2]

### 4.3 Bevege objekter:

For å bevege ting brukes waypoints. Kun ett objekt kan bruke ett waypoint, men du kan knytte flere modeller sammen til en gruppe og be gruppelederen følge waypoints for å få hele gruppen til å gjøre det. [2]

For å legge til waypoint kan du enten høyreklikke en modell og velge "Assign new waypoint", eller velge det fra "Editor Object List" og dobbeltklikke det inn. Waypoints kan høyreklikkes og du kan da legge til neste waypoint, lage sirkler osv. [2]



På hvert waypoint kan oppførselen til objektet/gruppen på vei dit endres, f.eks fart, formasjon ventetid etc. Waypoints kan også synkroniseres for å få hendelser til å samkjøre. [2]

Du kan "branche" waypoints ved å høyreklikke ett waypoint og velge "create branch". Du kan så sette "branch condition" for at den skal følge ett annet mønster ved visse hendelser. For eksempel kan du si at en returnerer til basen hvis den har mistet noen eller de får ett helikopter til å plukke de opp og dra. Det er mange muligheter for å lage bra scenarier med branching. [2]

## 5 Konklusjon

---

VBS2 er helt klart et kjempegodt program for simulering, men det er vanskelig å endre funksjonaliteten når man bare har tilgang på ett sett av ferdiglagde skript. Med de innebygde funksjonene kan man skreddersy scenarioer, og lage de dynamiske slik at de forandrer seg fra gang til gang. De største problemene vi vil ha, vil bli knyttet til å lage GUI'et.





## 6 Referanser

---

[1] Bohemia Interactive Australia, *VBS2*, <http://virtualbattlespace.vbs2.com/>, (sist besøkt 25.11.09)

[2] Bohemia Interactive Australia, *VBS2 Offline Mission Editor OME*,  
<http://developer.vbs2.com/manuals/>, (sist besøkt 25.11.09)

[3] Bohemia Interactive Australia, *Adding content*, <http://developer.vbs2.com/onlinehelp/>,  
(sist besøkt 25.11.09)



# KDA RIVET



## Teknologidokument - wxJoystick

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |   |   |
|-------|---|---|
| 1     | Introduksjon.....                                 | 3 |
| 1.1   | Hensikt med dokumentet.....                       | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser.....      | 3 |
| 1.3   | Dokumenthistorie.....                             | 3 |
| 1.4   | Ansvarlig .....                                   | 3 |
| 2     | Detaljert beskrivelse .....                       | 4 |
| 2.1   | Metoder vi får bruk for.....                      | 4 |
| 2.1.1 | Konstruktør – wxJoystick() .....                  | 4 |
| 2.1.2 | GetPosition() .....                               | 4 |
| 3     | Teknologien i bruk.....                           | 5 |
| 3.1   | Konvertere x og y verdier til grader/sekund ..... | 5 |
| 3.2   | Sensitivitet .....                                | 5 |
| 3.3   | Bruke verdiene til å regne ut posisjon.....       | 5 |
| 4     | Konklusjon.....                                   | 6 |
| 5     | Referanser.....                                   | 7 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en innføring i joystick håndtering ved hjelp av wxJoystick klassen, og hvilke muligheter som finnes.

## 1.2 Definisjoner, akronymer og forkortelser

**VBS:** Virtual Battle Space

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                      | Redaktør                                 |
|---------------|----------|--------------------------------|--|
| 0.1           | 25.04.10 | Førsteutkast                   | Ole-Martin Grøtterud                     |
| 0.98          | 19.05.10 | Gjennomgang av egne dokumenter | Ole-Martin Grøtterud                     |
| 0.99          | 24.05.10 | Korrekturlest                  | Arne Kristian Åmellem<br>og Lone Knutsen |
| 1.0           | 28.05.10 | Klargjort for innlevering      | Thomas Transeth                          |

## 1.4 Ansvarlig

Ole-Martin Grøtterud

## 2 Detaljert beskrivelse

---

Klassen wxJoystick tillater applikasjonen å styre én eller flere joysticker. Det er del av "wxAdvanced" biblioteket, og man må da legge til "wxmsw28d\_adv.lib" i linkerene. [1]

### 2.1 Metoder vi får bruk for

#### 2.1.1 Konstruktør - wxJoystick()

Har default parameter som wxJOYSTICK1, trengs kun å sette inn dersom man har 2 joysticker tilkoblet og skal bruke nr2, da putter man inn wxJOYSTICK2.

#### 2.1.2 GetPosition()

Returnerer i utgangspunktet ett wxPoint, men ved å kalle GetPosition().x eller GetPosition().y får man x eller y verdi som int. Verdien går fra 0 til 65536 for begge punkter.

## 3 Teknologien i bruk

---

Konstruktøren brukes med default parametere.

### 3.1 Konvertere x og y verdier til grader/sekund

I tillegg til x og y verdiene vil vi ha tilgjengelig en deltaT verdi, som gir oss tidsrommet siden sist metoden ble kjørt, i sekunder. For å få det på grader/sekund må vi først skille mellom positive og negative verdier, dette gjøres ved å trekke fra halve maks summen, slik at vi får en skala fra -32768 til 32768. Siden maksimale hastighet tårnet kan rotere med er (1rotasjon/4sek = 90grader/sekund) deler vi skalaen på  $(32768/90) \approx 364,09$  for å få grader/sekund < 90.

De endelige formlene for x og y verdiene blir da:

X-verdi:  $Joystick \rightarrow GetPosition().x / (32768/90) * deltaT$

Y-verdi:  $Joystick \rightarrow GetPosition().y / (32768/90) * deltaT$

### 3.2 Sensitivitet

For å legge til sensitivitet multipliseres bare verdiene over med en sensitivitetsvariabel før de legges til det totale antall grader

### 3.3 Bruke verdiene til å regne ut posisjon

Når vi skal bruke det vi har funnet over til å regne ut hvor vi er, har vi en metode som mottar x og y verdien ferdig utregnet som over. Klassen den ligger i inneholder en double verdi for x og y som er det totale antallet grader for våpenet. For hver gang metoden kjøres, multipliseres x og y verdiene som ble mottatt med sensitiviteten, før de legges til klassens totale x og y verdier. Siden grader går fra 0 til 359 grader modulodividerer vi de totale verdiene med 360.

$xTotal += xParam * Sensitivitet$

$xTotal = xTotal \% 360$

## 4 Konklusjon

---

Joysticken kan brukes til å styre siktet i VBS2 ved å håndtere bevegelsene i vår plug in. Dette vil være svært gunstig i motsetning til å måtte sende individuelle signaler til VBS2 med joystickposisjoner.



## 5 Referanser

---

[1] wxWidgets, wxJoystick-dokumentasjon,  
[http://docs.wxwidgets.org/2.8.5/wx\\_wxjoystick.html](http://docs.wxwidgets.org/2.8.5/wx_wxjoystick.html), 25.04.10.





# KDA RIVET



## Teknologidokument - wxWidgets

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |    |
|-------|--|----|
| 1     | Introduksjon.....                            | 3  |
| 1.1   | Hensikt med dokumentet.....                  | 3  |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 3  |
| 1.3   | Dokumenthistorie.....                        | 3  |
| 1.4   | Ansvarlig.....                               | 3  |
| 2     | Beskrivelse av wxWidgets.....                | 4  |
| 3     | Installering og bruk.....                    | 5  |
| 3.1   | Installering.....                            | 5  |
| 3.2   | Bygging.....                                 | 5  |
| 3.2.1 | wxWidgets-løsning.....                       | 5  |
| 3.2.2 | wxWidgets eksempler.....                     | 5  |
| 4     | Teknologien i bruk.....                      | 7  |
| 4.1   | Opprette nytt prosjekt:.....                 | 7  |
| 4.2   | Sette opp riktige konfigurasjoner:.....      | 7  |
| 4.3   | Tilpasse prosjektet:.....                    | 7  |
| 5     | Konklusjon.....                              | 9  |
| 6     | Referanser.....                              | 10 |
| 6.1   | Webreferanser:.....                          | 10 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en innføring i hva wxWidgets er og hvordan det skal brukes.

## 1.2 Definisjoner, akronymer og forkortelser

**GUI:** Graphical User Interface

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                  | Redaktør   |
|---------------|----------|----------------------------|--|
| 2.1           | 04.05.10 | Gjennomgang og dokstandard | Lone Knutsen                                     |
| 2.99          | 21.05.10 | Korrekturlest              | Ole-Martin Grøtterud og<br>Arne Kristian Åmellem |
| 3.0           | 28.05.10 | Klargjort for innlevering  | Thomas Transeth                                  |

## 1.4 Ansvarlig

Lone Knutsen

## 2 Beskrivelse av wxWidgets

---

wxWidgets er et bibliotek som inneholder forskjellige deler som til sammen gjør det mulig å sette sammen et grafisk bruker grensesnitt. Det leveres til forskjellige plattformer, og kan brukes i forskjellige kompilatorer. Vi vil bruke Windows som plattform og Visual Studio 2008 som kompilator.



## 3 Installering og bruk

---

### 3.1 Installering

For å kunne bruke wxWidgets, så må man først laste ned, lagre og installere wxWidgets. Vi skal i prosjektet vårt bruke Visual Studio 2008, og da er det wxWidgets versjon 2.8.10 som er kompatibel med denne. Man får blant annet lastet ned wxWidgets fra wxWidgets sine hjemmesider[1].

Ettersom vi alle bruker Windows som operativsystem, så bør man velge Windows-versjonen. Når denne er lastet ned (det er en .exe-fil), må den installeres. Man må da følge installasjonsguiden slik at applikasjonen blir riktig satt opp.

### 3.2 Bygging

Etter at man har installert wxWidgets, så må man åpne Visual Studio for å bygge prosjektene og eksemplene som følger med.

#### 3.2.1 wxWidgets-løsning

Det første man gjør er å velge **"Fil" → "åpne eksisterende prosjekt/løsning"** og deretter å klikke seg frem til mappen hvor wxWidgets er lagret. Her åpner du så **mappen "Build" → "Msw"**. Her ligger det to forskjellige løsninger som kan brukes. Vi skal bruke den som heter **wx\_dll.dsw**. Nå vil du få spørsmål om du vil konvertere filene, og dette svarer du **"Yes – to all"** på. Nå vil løsningen lastes inn i Visual Studio.

Når denne har blitt lastet inn i Visual Studio, så må løsningen bygges på følgende måte. Høyreklikk på løsningen, og velg **"Batch build"**. Merk av for de som heter **"Debug"** og **"Release"**. Etter at dette er gjort så trykker du **"Build"**.

Ettersom en del av filene er inkrementelt linket, så er en del av disse knyttet sammen. Det kan det hende at det dukker opp feilmeldinger under byggingen av løsningen. Dersom dette skjer så er det bare å høyreklikke på løsningen og å velge **"Build"**. Da vil de som ikke ble bygget i første runde bygges nå.

#### 3.2.2 wxWidgets eksempler

Du skal nå ha gjort ferdig byggingen av selve wxWidgets applikasjonen. For å teste om dette ble gjort riktig, så må du bygge **eksemplene** som følger med. Dette gjøres på følgende måte.

**Fil" → "åpne eksisterende prosjekt/løsning"**. Deretter klikker du deg frem til mappen **"Samples"** som ligger inne i wxWidgets-mappen. Her åpner du fila som heter **"samples.dsw"**. Her velger du også **"Yes – to all"**, og løsningen vil så lastes inn. Etter dette, høyreklikker du på løsningen, velger **"Batch build"** og merker av **alle "Debug"** og **"Release"**. Trykk så **"Build"**.

Når dette er gjort så kan du lukke løsningen, og åpne et av eksemplene (f.eks "Image").  
Dette må så bygges, og etter dette kan du kompilere.



## 4 Teknologien i bruk

---

For å kunne opprette et prosjekt, så er det to forskjellige måter å gjøre dette på. Den ene måten er å sette opp prosjektet fra bunn av, og den andre måten er å ta utgangspunkt i et tidligere prosjekt.

Skal man sette opp sitt eget prosjekt så må man sette en del variabler, "path" og innstillinger for prosjektet. Hvilke "path" og variable som skal inn, avhenger av hvilket operativsystem, Visual Studio-versjon og hvilken versjon av wxWidgtes man har.

Setter man opp et prosjekt fra et annet prosjekt, så slipper man en del kringling med path, variabler, preprocessor og linker. Skal man sette opp et prosjekt fra et annet prosjekt, så gjøres dette på følgende måte *med Visual Studio 2008, Windows 7 og wxWidgtes 2-8-10*:

### 4.1 Opprette nytt prosjekt

1. File → new project.
2. Velg Visual studio c++ prosjekt
3. Gi prosjektet et navn og velg ok.

### 4.2 Sette opp riktige konfigurasjoner:

1. Høyreklikk på prosjektet og velg åpne i et windows explorer vindu.
2. Åpne enda et Windows Explorer vindu.
3. I det andre vinduet åpner du mappen hvor du har lagret wxWidgtes og åpner mappen **samples** → **minimal**.
4. Kopier **minimal.cpp** og **minimal.rc** til mappen hvor du har lagret prosjektet ditt.
5. Lukk vinduet som har wxWidgtes-filene oppe.
6. I mappen hvor du har prosjektet ditt, merk **minimal.cpp** og **minimal.rc** å dra de inn i prosjektet i Visual studio.
7. **Minimal.cpp** skal nå havne under source-files. **Minimal.rc** skal havne under resources.

### 4.3 Tilpasse prosjektet:

1. Høyreklikk på prosjektet og velg **properties**.
2. I venstre hjørne på properties-vinduet, så er det en drop-down meny. Her velger man "**all configurations**".
3. Utvid menyen til **Confugration properties**.
4. Utvid menyen til **c++**
5. Klikk på **General**
6. I "**Additional Include Directories**" så må følgende legges inn (wxWidgtes liker ikke miljøvariabler, så vi må jobbe med hele path-er):  
"path'en til mappen hvor wxWidgtes ligger\include";  
" path-en til mappen hvor wxWidgtes ligger \lib\vc\_lib\mswd"

7. Klikk på **preprocessor**, og legg inn følgende i **preprocessor definitions**:

WIN32;\_\_WXMSW\_\_;\_WINDOWS;\_DEBUG;\_\_WXDEBUG\_\_

8. Utvid **Linker**.

9. Klikk på **General**

10. Legg til følgende i **Additional Library Directories**:

"path'en til mappen hvor wxWidgtes ligger \lib\vc\_lib"

11. Klikk på **Input**.

12. Legg inn følgende i **Additional Dependencies**:

wxmsw28d\_core.lib wxmsw28d\_adv.lib(for joysticken) wxbase28d.lib wxtiffd.lib wxjpegd.lib  
wxpngd.lib wxzlibd.lib wxregexd.lib wxexpatd.lib winmm.lib comctl32.lib rpcrt4.lib  
wsock32.lib odbc32.lib

13. Utvid **resources**

14. Klikk på **general**

15. Legg inn følgende i **preprocessor definitions**:

\_DEBUG;\_\_WXMSW\_\_;\_\_WXDEBUG\_\_;\_WINDOWS;NOPCH

16. Legg inn følgende i **Additional Include Directories**:

"path'en til mappen hvor wxWidgtes ligger\include";" path'en til mappen hvor wxWidgtes  
ligger \lib\vc\_lib\mswd"

17. **Velg ok.** [2]

Du kan nå bygge prosjektet ditt. Dette gjøres ved å velge **Debug** fra dropdown-menyen på toppen og å klikke "**Build**". Når denne er ferdig, så gjentas det samme for "**Release**". (Dersom byggingen krasjer, så gjøres denne på nytt. Det er en del filer som er avhengig av hverandre).



## 5 Konklusjon

---

Hittil ser det ut som om wxWidgets skal inneholde den funksjonaliteten som trengs for å kunne lage et GUI til treneren. Vi har hittil fått opp et GUI som likner på det vi skal ha, og vi har lagt til slik at vi kan "flippe" brytere. I tillegg eksisterer det en "plug-in" som skal gjøre det mulig å kommunisere med VBS2.

## 6 Referanser

---

### 6.1 Webreferanser:

[1] wxWidgets, hjemmeside,  
<http://www.wxwidgets.org> (sist besøkt 04.02.10)

[2] Ukjent blogger, blogg,  
<http://rhyous.com/2009/12/16/how-to-compile-a-wxwidgets-application-in-visual-studio-2008/> (sist besøkt 16.02.10)

# KDA RIVET



## Testspesifikasjon

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|        |  |    |
|--------|--|----|
| 1      | Introduksjon.....                            | 5  |
| 1.1    | Hensikt med dokumentet.....                  | 5  |
| 1.2    | Definisjoner, akronymer og forkortelser..... | 5  |
| 1.3    | Dokumenthistorie.....                        | 5  |
| 1.4    | Ansvarlig .....                              | 5  |
| 2      | Testutføring.....                            | 6  |
| 3      | Overordnet testbeskrivelse.....              | 7  |
| 3.1    | Tester.....                                  | 8  |
| 3.1.1  | Funksjonstest.....                           | 8  |
| 3.1.2  | Brukergrensesnitttest.....                   | 9  |
| 3.1.3  | Ytelsesprofilering.....                      | 9  |
| 3.1.4  | Belastningstest .....                        | 9  |
| 3.1.5  | Konfigurasjonstest.....                      | 9  |
| 3.1.6  | Stresstest .....                             | 9  |
| 3.1.7  | Volumtest .....                              | 9  |
| 3.1.8  | Sikkerhetstest .....                         | 10 |
| 3.1.9  | Gjenopprettingstest .....                    | 10 |
| 3.1.10 | Installasjonstest.....                       | 10 |
| 3.1.11 | Kodeinspeksjon .....                         | 10 |
| 3.1.12 | Referansetest .....                          | 10 |
| 3.1.13 | Data og databaseintegritetstest.....         | 10 |
| 3.1.14 | Regresjonstest.....                          | 10 |
| 3.2    | Testgruppering.....                          | 10 |
| 3.2.1  | Testing av krav.....                         | 11 |
| 4      | Detaljert testbeskrivelse .....              | 12 |
| 4.2    | Funksjonstester .....                        | 13 |
| 4.2.1  | TF01 – Bemannet systemet .....               | 13 |
| 4.2.2  | TF02 – Sikte: elevasjon og rotasjon.....     | 14 |
| 4.2.3  | TF03 – Armering på .....                     | 15 |
| 4.2.4  | TF04 – Armering av .....                     | 16 |



|        |  |    |
|--------|--|----|
| 4.2.5  | TF05 - Skyte .....                           | 17 |
| 4.2.6  | TF06 – Velge skuddhastighet .....            | 18 |
| 4.2.7  | TF07 – Simulering av servoer .....           | 19 |
| 4.2.8  | TF08 – Zoome inn .....                       | 20 |
| 4.2.9  | TF09 – Zoome ut.....                         | 21 |
| 4.2.10 | TF10 – Nattmodus på .....                    | 22 |
| 4.2.11 | TF11 – Nattmodus av .....                    | 23 |
| 4.2.12 | TF12 – Sensitivitet .....                    | 24 |
| 4.2.13 | TF20 Nattmodul & Thermal vision .....        | 25 |
| 4.2.14 | TF22 Granatkaster .....                      | 26 |
| 4.3    | Brukergrensesnitttest .....                  | 27 |
| 4.3.1  | TB01 – GUI i henhold til spesifikasjon ..... | 27 |
| 4.3.2  | TB02 - Meny .....                            | 28 |
| 4.4    | Ytelsesprofilering .....                     | 30 |
| 4.5    | Belastningstest.....                         | 30 |
| 4.5.1  | TBL01 - Armering.....                        | 30 |
| 4.5.2  | TBL02 – Armering: Safe Remote .....          | 31 |
| 4.5.3  | TBL03 - Nightvision.....                     | 32 |
| 4.5.4  | TBL04 - Zoom.....                            | 33 |
| 4.5.5  | TBL05 – Rate of fire .....                   | 34 |
| 4.5.6  | TBL06 - Skyte .....                          | 35 |
| 4.5.7  | TBL07 - Sikte .....                          | 36 |
| 4.5.8  | TBL08 - Sensitivitet .....                   | 37 |
|        | Konfigurasjonstest .....                     | 38 |
| 4.6    | Stresstest .....                             | 38 |
| 4.7    | Volumtest .....                              | 38 |
| 4.8    | Sikkerhetstest .....                         | 38 |
| 4.9    | Gjenopprettingstest.....                     | 38 |
| 4.10   | Installasjonstest .....                      | 38 |
| 4.11   | Kodeinspeksjon.....                          | 39 |
| 4.11.1 | TKI01 - Kodestandard .....                   | 39 |
| 4.12   | Referansetest.....                           | 40 |



|      |                                       |    |
|------|---------------------------------------|----|
| 4.13 | Data- og databaseintegritetstest..... | 40 |
| 4.14 | Regresjonstest .....                  | 40 |



# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi en innføring i hva slags tester som er knyttet opp mot hvilke krav, samt gi en oversikt over alle testene som vi har planlagt å gjøre.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                   | Redaktør             |
|---------------|----------|-----------------------------|----------------------|
| 1.1           | 29.04.10 | Ny dokumentstandard og TF12 | Ole-Martin Grøtterud |
| 1.2           | 04.05.10 | Oppdatering, TBLXX, TKI01   | Zoran Vukobratovic   |
| 1.3           | 24.05.10 | Oppdatering                 | Zoran Vukobratovic   |
| 2.0           | 28.05.10 | Klargjort for innlevering   | Thomas Transeth      |

## 1.4 Ansvarlig

Zoran Vukobratovic

## 2 Testutføring

---

**For å utføre en test må følgende prosedyrer utføres:**

1. Identifiser hvilken modul som skal testes. I kapittel 3.2 omtaler hvilke tester som omfatter hver modul.
2. Identifiser alle tester som skal utføres.
3. Generer en Testrapport fra testrapport malen med navn, modul, dato osv.

**Utførelsen av testen kan nå begynne, og følgende prosedyrer må utføres:**

1. Testene gjennomføres etter rekkefølgen satt for modulen.
2. Sjekk nøye at riktige kriterier er oppnådd før du gjør en vurdering.
3. Når riktige kriterier er oppnådd, gjennomgå fremgangsmåten.
4. Noter det faktiske resultatet og gjør en rask sammenligning med det forventede resultatet.
5. Skriv inn navn og dato for utførelse av tester, samt legg til status "utført"
6. Skriv feillogg og snakk med testansvarlig. Retting av feil blir delegert til de rette personene.



### 3 Overordnet testbeskrivelse

---

De ulike testene som skal utføres deles basert på testtyper. Følgende testtyper inngår:

- Funksjonstest
- Brukergrensesnitt test
- Ytelsesprofilering
- Belastningstest
- Konfigurasjonstest
- Stresstest
- Volumtest
- Sikkerhetstest
- Gjenopprettingstest
- Installasjonstest
- Kodeinspeksjon
- Referansetest
- Data- og databaseintegritetstest
- Regresjonstest

Hver test består av to kolonner: *Testidentifikasjon* og *Testbeskrivelse*.

#### Testidentifikasjonen består av tre komponenter:

- Nummerering på heltallsform som inkrementeres
- Testtype:
  - Bokstaven "F" for Funksjonstest
  - Bokstaven "B" for Brukergrensesnitttest
  - Bokstaven "Y" for Ytelsesprofilering
  - Bokstavene "BL" for Belastningstest
  - Bokstaven "K" for Konfigurasjonstest
  - Bokstaven "S" for Stresstest
  - Bokstaven "V" for Volumtest
  - Bokstavene "SI" for Sikkerhetstest
  - Bokstaven "G" for Gjenopprettingstest
  - Bokstaven "I" for Installasjonstest
  - Bokstavene "KI" for Kodeinspeksjon
  - Bokstaven "R" for Referansetest
  - Bokstaven "D" for Data- og databaseintegritetstest
  - Bokstavene "RE" for Regresjonstest
- Bokstaven "T" som står for Test

En testidentifikasjon kan da være gitt av "TF01", som innebærer at dette er en test, nærmere bestemt en funksjonstest, og kommer som nummer 1 i rekka.

## 3.1 Tester

### 3.1.1 Funksjonstest

| Testidentifikasjon | Testbeskrivelse                                  |
|--------------------|--|
| TF01               | Bemanne systemet                                 |
| TF02               | Går det an å sikte med joysticken?               |
| TF03               | Går det an å armere RWS-en?                      |
| TF04               | Går det an å desarmere RWS-en?                   |
| TF05               | Går det an å skyte med våpenet?                  |
| TF06               | Bytter systemet mellom skuddhastighet riktig?    |
| TF07               | Er simuleringen av servoene riktig?              |
| TF08               | Zoomer systemet inn mellom stegene som det skal? |
| TF09               | Zoomer systemet ut mellom stegene som det skal?  |
| TF10               | Byttes modusen til nattmodus?                    |
| TF11               | Byttes modusen til vanlig modus?                 |
| TF12               | Sensitivitet                                     |
| TF13               | Bemanne systemet – utvidet                       |
| TF14               | Bemanne systemet                                 |
| TF15               | Sikte  |
| TF16               | Armering   |
| TF17               | Skyte  |
| TF18               | Rate of fire                                     |
| TF19               | Zoom   |
| TF20               | Nattmodus & Thermal vision                       |
| TF21               | Sensitivitet                                     |
| TF22               | Granatkaster                                     |
| TF23               | Bemanne systemet                                 |
| TF24               | Sikte  |
| TF25               | Rate of fire                                     |
| TF26               | Zoom   |
| TF27               | Nattmodus & Thermal vision                       |
| TF28               | Sensitivitet                                     |
| TF29               | Granatkaster                                     |

### 3.1.2 Brukergrensesnittstest

| Testidentifikasjon | Testbeskrivelse  |
|--------------------|--|
| TB01               | Stemmer GUI-en overens med den spesifikke fra oppdragsgiver? |
| TB02               | Meny   |
| TB03               | Meny   |

### 3.1.3 Ytelsesprofilering

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.4 Belastningstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
| TBL01              | Armering        |
| TBL02              | Remote Safe     |
| TBL03              | Nightvision     |
| TBL04              | Zoom            |
| TBL05              | Rate of fire    |
| TBL06              | Skyte           |
| TBL07              | Sikte           |
| TBL08              | Sensitivitet    |
| TBL09              | Skyte           |
| TBL10              | Armering        |
| TBL11              | Remote Safe     |

### 3.1.5 Konfigurasjonstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.6 Stresstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.7 Volumtest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.8 Sikkerhetstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.9 Gjenopprettingstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.10 Installasjonstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.11 Kodeinspeksjon

| Testidentifikasjon | Testbeskrivelse           |
|--------------------|---------------------------|
| TKI01              | Er kodestandarden riktig? |

### 3.1.12 Referansetest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.13 Data og databaseintegritetstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

### 3.1.14 Regresjonstest

| Testidentifikasjon | Testbeskrivelse |
|--------------------|-----------------|
|--------------------|-----------------|

## 3.2 Testgruppering

I forbindelse med testingen kan det være slik at mange ulike tester grupperes inn i en modul. Når det skal gjøres modultesting gjelder det derfor å teste alle testene som ligger under hver modul. For hver modultest skal det skrives en testrapport.

### **3.2.1 Testing av krav**

Kapitlene under beskriver hvilke tester som assosieres med hvilke krav. Hvert krav kan inneholde flere tester.

#### **3.2.1.1 Krav 1F**

- TF01

#### **3.2.1.2 Krav 2F**

- TF02

#### **3.2.1.3 Krav 3F**

- TF03
- TF04

#### **3.2.1.4 Krav 4F**

- TF05

#### **3.2.1.5 Krav 5F**

- TF06

#### **3.2.1.6 Krav 6F**

- TF07

#### **3.2.1.7 Krav 7F**

- TF08
- TF09

#### **3.2.1.8 Krav 8F**

- TF10
- TF11

#### **3.2.1.9 Krav 9F**

- TF12

#### **3.2.1.10 Krav 1IF**

- TB01

## 4 Detaljert testbeskrivelse

Testene beskrevet i kapittel "Overordnet testbeskrivelse" må i tillegg detaljeres for at testerene skal kunne vite hvilken fremgangsmåte som skal benyttes ved testingen. De forskjellige testene detaljeres derfor som i "Figur 1" nedenfor:

### Identifisering:

|                            |  |
|----------------------------|--|
| <b>Testidentifikasjon:</b> | [IDnr]   |
| <b>Beskrivelse:</b>        | [her skal det stå en beskrivelse av hvordan testen skal utføres] |
| <b>Type:</b>               | Funksjonell/ikke-funksjonell                                     |

### Kryssreferanse:

|                  |            |
|------------------|------------|
| <b>Krav:</b>     | [KraVID]   |
| <b>Use Case:</b> | [UCid]     |
| <b>Risiko:</b>   | [Risikoid] |

### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | [hva som skal til for at testen er gjennomførbar] |
| <b>Fremgangsmåte:</b>      | [hvordan testen skal utføres]                     |
| <b>Forventet resultat:</b> | [hva forventet resultat er]                       |
| <b>Faktisk resultat:</b>   | [hva det faktiske resultatet ble]                 |

### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | [navn]             | <b>Dato:</b> | [dd.mm.åå] |
| <b>Utført av:</b>   | [navn]             | <b>Dato:</b> | [dd.mm.åå] |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Utført/ikke utført |              |            |

#### 4.1.1.1.1 Figur 1: Mal for detaljert testbeskrivelse

Tabellen består av fire hovedgrupper; *Identifisering*, *Kryssreferanse*, *Testing* og *Teststatus*. Identifisering er forklart tidligere i Overordnet testbeskrivelse.

Kryssreferanse omhandler hvilke krav, use-caser og risikoer som testen er knyttet mot. Testing handler om hvilke kriterier som settes før testing, hvordan testen skal utføres, og

hva som er forventet resultat, samt det faktiske resultatet.

Teststatus handler om hvem som har lagt til testen, hvem som har utført testen og om testen er utført eller ikke.

## 4.2 Funksjonstester

### 4.2.1 TF01 – Bemanne systemet

#### Identifisering:

|                            |                  |
|----------------------------|------------------|
| <b>Testidentifikasjon:</b> | TF01             |
| <b>Beskrivelse:</b>        | Bemanne systemet |
| <b>Type:</b>               | Funksjonstest    |

#### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 1F  |
| <b>Use Case:</b> | UC1 |
| <b>Risiko:</b>   | R1  |

#### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være inne i VBS2, og det må være et kjøretøy med RWS montert i nærheten. |
| <b>Fremgangsmåte:</b>      | Lokalisere RWS-kjøretøyet, gå bort til kjøretøyet, bemanne «turret» posisjonen.      |
| <b>Forventet resultat:</b> | Brukeren får opp korrekt RWS GUI, og kan starte å bruke treneren                     |
| <b>Faktisk resultat:</b>   |  |

#### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 10.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.1.1.1 Tabell1: TF01

## 4.2.2 TF02 – Sikte: elevasjon og rotasjon

### Identifisering:

|                            |               |
|----------------------------|---------------|
| <b>Testidentifikasjon:</b> | TF02          |
| <b>Beskrivelse:</b>        | Sikte         |
| <b>Type:</b>               | Funksjonstest |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 2F  |
| <b>Use Case:</b> | UC2 |
| <b>Risiko:</b>   | R1  |

### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være inne i VBS2, og det må være et kjøretøy med RWS montert i nærheten.   |
| <b>Fremgangsmåte:</b>      | Bevege på joystickken.   |
| <b>Forventet resultat:</b> | Siktet skal bevege seg i samme retning som joystickken peker mot. En skal kunne snu skyttertårnet 360 grader uten stoppunkt.<br><br>Elevasjon skal gå fra -30 til 60 grader. |
| <b>Faktisk resultat:</b>   |  |

### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 10.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.2.1.1 Tabell2: TF02



### 4.2.3 TF03 – Armering på

#### Identifisering:

|                            |               |
|----------------------------|---------------|
| <b>Testidentifikasjon:</b> | TF03          |
| <b>Beskrivelse:</b>        | Armering på   |
| <b>Type:</b>               | Funksjonstest |

#### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 3F  |
| <b>Use Case:</b> | UC3 |
| <b>Risiko:</b>   | R1  |

#### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.        |
| <b>Fremgangsmåte:</b>      | Bruke mus eller touch screen til å trykke på armeringsknappen. |
| <b>Forventet resultat:</b> | Systemet skal nå være armert og brukeren skal kunne skyte.     |
| <b>Faktisk resultat:</b>   |  |

#### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 10.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.3.1.1 Tabell3: TF03

#### 4.2.4 TF04 – Armering av

##### Identifisering:

|                            |               |
|----------------------------|---------------|
| <b>Testidentifikasjon:</b> | TF03          |
| <b>Beskrivelse:</b>        | Armering av   |
| <b>Type:</b>               | Funksjonstest |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 3F  |
| <b>Use Case:</b> | UC3 |
| <b>Risiko:</b>   | R1  |

##### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.                   |
| <b>Fremgangsmåte:</b>      | Bruke mus eller touch screen til å trykke på armeringsknappen.            |
| <b>Forventet resultat:</b> | Systemet skal nå være desarmert, og brukeren skal ikke kunne skyte igjen. |
| <b>Faktisk resultat:</b>   |   |

##### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 10.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

##### 4.2.4.1.1 Tabell4: TF04

## 4.2.5 TF05 - Skyte

### Identifisering:

|                            |                   |
|----------------------------|-------------------|
| <b>Testidentifikasjon:</b> | TF05              |
| <b>Beskrivelse:</b>        | Skyte med våpenet |
| <b>Type:</b>               | Funksjonstest     |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 4F  |
| <b>Use Case:</b> | UC4 |
| <b>Risiko:</b>   | R1  |

### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet. |
| <b>Fremgangsmåte:</b>      | Trykke på avtrekker knappen på joysticken               |
| <b>Forventet resultat:</b> | Ett skudd blir avfyrt.                                  |
| <b>Faktisk resultat:</b>   |   |

### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 10.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.5.1.1 Tabell5: TF05

## 4.2.6 TF06 – Velge skuddhastighet

### Identifisering:

|                            |                      |
|----------------------------|----------------------|
| <b>Testidentifikasjon:</b> | TF06                 |
| <b>Beskrivelse:</b>        | Velge skuddhastighet |
| <b>Type:</b>               | Funksjonstest        |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 5F  |
| <b>Use Case:</b> | UC5 |
| <b>Risiko:</b>   | R2  |

### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.                                |
| <b>Fremgangsmåte:</b>      | Trykke på "Rate"-knappen på GUI'et.  |
| <b>Forventet resultat:</b> | Systemet skal bytte mellom Singelskudd, Serie og Mitraljøse hver gang knappen trykkes. |
| <b>Faktisk resultat:</b>   |  |

### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 11.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.6.1.1 Tabell6: TF06

## 4.2.7 TF07 – Simulering av servoer

### Identifisering:

|                            |                       |
|----------------------------|-----------------------|
| <b>Testidentifikasjon:</b> | TF07                  |
| <b>Beskrivelse:</b>        | Simulering av servoer |
| <b>Type:</b>               | Funksjonstest         |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 6F  |
| <b>Use Case:</b> | UC8 |
| <b>Risiko:</b>   | R2  |

### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.  |
| <b>Fremgangsmåte:</b>      | Bevege på joysticken.  |
| <b>Forventet resultat:</b> | Siktet skal ha en tregere start og stopp for å simulere servoene. Det skal også ha en maks hastighet slik at det går en gitt tid for å snu 360 grader. |
| <b>Faktisk resultat:</b>   |  |

### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 11.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.7.1.1 Tabell7: TF07

#### 4.2.8 TF08 – Zoome inn

##### Identifisering:

|                            |               |
|----------------------------|---------------|
| <b>Testidentifikasjon:</b> | TF08          |
| <b>Beskrivelse:</b>        | Zoome inn     |
| <b>Type:</b>               | Funksjonstest |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 7F  |
| <b>Use Case:</b> | UC7 |
| <b>Risiko:</b>   | R2  |

##### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet. |
| <b>Fremgangsmåte:</b>      | Trykke på Zoom-In knappen                               |
| <b>Forventet resultat:</b> | Siktet skal zoome inn ett steg fra 1x-52x zoom.         |
| <b>Faktisk resultat:</b>   |   |

##### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 11.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

##### 4.2.8.1.1 Tabell8: TF08

## 4.2.9 TF09 – Zoome ut

### Identifisering:

|                            |               |
|----------------------------|---------------|
| <b>Testidentifikasjon:</b> | TF09          |
| <b>Beskrivelse:</b>        | Zoome ut      |
| <b>Type:</b>               | Funksjonstest |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 7F  |
| <b>Use Case:</b> | UC7 |
| <b>Risiko:</b>   | R2  |

### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet. |
| <b>Fremgangsmåte:</b>      | Trykke på Zoom-ut knappen                               |
| <b>Forventet resultat:</b> | Siktet skal zoome ut ett steg i området 1x-52x zoom.    |
| <b>Faktisk resultat:</b>   |   |

### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 11.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.9.1.1 Tabell9: TF09

## 4.2.10 TF10 – Nattmodus på

### Identifisering:

|                            |               |
|----------------------------|---------------|
| <b>Testidentifikasjon:</b> | TF10          |
| <b>Beskrivelse:</b>        | Nattmodus på  |
| <b>Type:</b>               | Funksjonstest |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 8F  |
| <b>Use Case:</b> | UC6 |
| <b>Risiko:</b>   | R3  |

### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.   |
| <b>Fremgangsmåte:</b>      | Trykke på "Day/Night"-knappen på GUI'et.  |
| <b>Forventet resultat:</b> | Skjermen med siktet skal gå over fra vanlig til nattmodus.<br>Zoom-stegene i nattmodus skal være annerledes enn dagmodus. |
| <b>Faktisk resultat:</b>   |   |

### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 11.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.10.1.1 Tabell10: TF10



#### 4.2.11 TF11 – Nattmodus av

##### Identifisering:

|                            |               |
|----------------------------|---------------|
| <b>Testidentifikasjon:</b> | TF11          |
| <b>Beskrivelse:</b>        | Nattmodus av  |
| <b>Type:</b>               | Funksjonstest |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 8F  |
| <b>Use Case:</b> | UC6 |
| <b>Risiko:</b>   | R3  |

##### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.          |
| <b>Fremgangsmåte:</b>      | Trykke på "Day/Night"-knappen på GUI'et.                         |
| <b>Forventet resultat:</b> | Skjermen med siktet skal gå over fra nattmodus til vanlig modus. |
| <b>Faktisk resultat:</b>   |  |

##### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 11.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.11.1.1 Tabell11: TF11

#### 4.2.12 TF12 – Sensitivitet

##### Identifisering:

|                            |                             |
|----------------------------|-----------------------------|
| <b>Testidentifikasjon:</b> | TF12                        |
| <b>Beskrivelse:</b>        | Endre joystick sensitivitet |
| <b>Type:</b>               | Funksjonstest               |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 9F  |
| <b>Use Case:</b> | UC9 |
| <b>Risiko:</b>   | R3  |

##### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.  |
| <b>Fremgangsmåte:</b>      | Trykke på "+/-"-knappen under "SPEED" på GUI'et.   |
| <b>Forventet resultat:</b> | Joystickens bevegelser skal gjøre større/mindre utslag enn før, avhengig av hvilket nivå det er på sensitiviteten. |
| <b>Faktisk resultat:</b>   |  |

##### Teststatus:

|                     |                      |              |          |
|---------------------|----------------------|--------------|----------|
| <b>Lagt til av:</b> | Ole-Martin Grøtterud | <b>Dato:</b> | 29.04.10 |
| <b>Utført av:</b>   |                      | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                      |              |          |
| <b>Status:</b>      | Ikke utført          |              |          |

##### 4.2.12.1.1 Tabell12: TF12

#### 4.2.13 TF20 Nattmodul & Thermal vision

|                            |                            |
|----------------------------|----------------------------|
| <b>Testidentifikasjon:</b> | TF20                       |
| <b>Beskrivelse:</b>        | Nattmodus & Thermal vision |
| <b>Type:</b>               | Funksjonstest              |

#### Kryssreferanse

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 8F  |
| <b>Use Case:</b> | UC6 |
| <b>Risiko:</b>   | R3  |

#### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være inne i VBS2, systemet må være bemannet.   |
| <b>Fremgangsmåte:</b>      | Testeren trykker på Day/Night knappen.<br>Testeren trykker på Day/Night knappen igjen.<br>Testeren trykker på Day/Night knappen en gang til.   |
| <b>Forventet resultat:</b> | Siktet endrer seg fra dag til nattmodus ved første trykk.<br><br>Siktet endrer seg fra nattmodus til thermal vision ved andre trykk.<br><br>Siktet endrer seg fra thermal vision ved tredje trykk. |
| <b>Faktisk resultat:</b>   |  |

#### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 19.05.10 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.13.1.1 Tabell 1.

#### 4.2.14 TF22 Granatkaster

|                            |                            |
|----------------------------|----------------------------|
| <b>Testidentifikasjon:</b> | TF22                       |
| <b>Beskrivelse:</b>        | Granatkaster / smokescreen |
| <b>Type:</b>               | Funksjonstest              |

#### Kryssreferanse

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | N/A |
| <b>Use Case:</b> | N/A |
| <b>Risiko:</b>   | N/A |

#### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være inne i VBS2, og kjøretøyet med RWS-CROWS skal være bemannet og armert.   |
| <b>Fremgangsmåte:</b>      | Under "GRENADES"<br><br>Testeren velger "UPPER" eller "LOWER" ved å trykke og dra bryteren opp eller ned, deretter flippes sikkerhetsvippen opp, og testeren trykker på bryteren under denne. |
| <b>Forventet resultat:</b> | Testeren får opp teksten "Granat avfyrt" i en hint melding på skjermen. Det er kun to granater som totalt kan avfyres, en i hver av de armerte posisjonene.                                   |
| <b>Faktisk resultat:</b>   |   |

#### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 19.05.10 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.2.14.1.1 Tabell 1.

## 4.3 Brukergrensesnittet

### 4.3.1 TB01 – GUI i henhold til spesifikasjon

#### Identifisering:

|                            |                                  |
|----------------------------|----------------------------------|
| <b>Testidentifikasjon:</b> | TB01                             |
| <b>Beskrivelse:</b>        | GUI i henhold til spesifikasjon. |
| <b>Type:</b>               | Ikke funksjonelt                 |

#### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 1IF |
| <b>Use Case:</b> | N/A |
| <b>Risiko:</b>   | R1  |

#### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.                  |
| <b>Fremgangsmåte:</b>      | Sammenligne utseende på GUI'et på skjermen mot GUI'et fra oppdragsgiver. |
| <b>Forventet resultat:</b> | GUI'et skal være likt det vi har fått fra oppdragsgiver.                 |
| <b>Faktisk resultat:</b>   |  |

#### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 10.11.09 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

#### 4.3.1.1.1 Tabell13: TB01

### 4.3.2 TB02 - Meny

|                            |                      |
|----------------------------|----------------------|
| <b>Testidentifikasjon:</b> | TB02                 |
| <b>Beskrivelse:</b>        | Meny                 |
| <b>Type:</b>               | Brukergrensesnittest |

#### Kryssreferanse

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | n/a |
| <b>Use Case:</b> | n/a |
| <b>Risiko:</b>   | n/a |

#### Testing:

|                            |   |                           |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
|----------------------------|---|---------------------------|----------|-----------------|-------------------|--------------|------------|------------------------|-------------------|----------------------|-------------|--|---------------------------|------------------|--|--|--------------------|--|--|-----------------|--|--|
| <b>Testkriterier:</b>      | Brukeren må være inne i VBS2 og systemet må være bemannet.  |                           |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| <b>Fremgangsmåte:</b>      | <p>Under "MENU"</p> <p>Trykke "MENU On/Off" for å få opp menyen</p> <p>Navigere med "L" og "R" knappene på GUI'et for å velge undermeny</p> <p>Trykke "SEL/ZERO" for å velge undermeny</p> <p>Navigere med "U" og "D" i undermenyen</p> <p>Trykke "SEL/ZERO" for å velge funksjon i undermenyen</p> <p>Testereren skal gå gjennom samtlige undermenyer, og velge alle funksjoner, i tillegg skal en kunne gå ut fra menyen ved å trykke på "MENU On/Off" knappen.</p>   |                           |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| <b>Forventet resultat:</b> | <p>Når testereren trykker på "MENU On/Off" skal menyen komme opp med 3 menyer og deres undermenyer å velge mellom:</p> <table><tr><td>1 1 SETTING</td><td>2 0 AMMO</td><td>3 0 MAINTENANCE</td></tr><tr><td>1 2 LRF Min Range</td><td>2 1 M33 Ball</td><td>3 1 Status</td></tr><tr><td>1 3 Restore Bore Value</td><td>2 2 No Ballistics</td><td>3 2 Cal. Pos Sensors</td></tr><tr><td>1 4 Zeroing</td><td></td><td>3 3 Adjust Reticle centre</td></tr><tr><td>1 5 Boresighting</td><td></td><td></td></tr><tr><td>1 6 Graphics level</td><td></td><td></td></tr><tr><td>1 7 CG Sharping</td><td></td><td></td></tr></table> <p>Når MENU trykkes skal bare de 3 hovedmenyene komme opp, de skal velges med "L" og "R" knappene, "L" knappen velger SETTING uansett hvor på menyen testereren befinner seg.</p> <p>Ved å velge SETTING, AMMO eller MAINTENANCE og trykke "D" skal undermenyen til den valgte dukke opp.</p> <p>For å gå tilbake til hovedmenyen skal "U" knappen benyttes helt til undermenyen forsvinner og testereren befinner seg i hovedmenyen igjen,</p> | 1 1 SETTING               | 2 0 AMMO | 3 0 MAINTENANCE | 1 2 LRF Min Range | 2 1 M33 Ball | 3 1 Status | 1 3 Restore Bore Value | 2 2 No Ballistics | 3 2 Cal. Pos Sensors | 1 4 Zeroing |  | 3 3 Adjust Reticle centre | 1 5 Boresighting |  |  | 1 6 Graphics level |  |  | 1 7 CG Sharping |  |  |
| 1 1 SETTING                | 2 0 AMMO  | 3 0 MAINTENANCE           |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| 1 2 LRF Min Range          | 2 1 M33 Ball  | 3 1 Status                |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| 1 3 Restore Bore Value     | 2 2 No Ballistics   | 3 2 Cal. Pos Sensors      |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| 1 4 Zeroing                |   | 3 3 Adjust Reticle centre |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| 1 5 Boresighting           |   |                           |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| 1 6 Graphics level         |   |                           |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |
| 1 7 CG Sharping            |   |                           |          |                 |                   |              |            |                        |                   |                      |             |  |                           |                  |  |  |                    |  |  |                 |  |  |

|                          |  |
|--------------------------|--|
|                          | eventuellt kan "L" brukes for å velge seg direkte tilbake til SETTING. |
| <b>Faktisk resultat:</b> |  |

**Teststatus:**

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 14.05.10 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke utført        |              |          |

**4.3.2.1.1 Tabell 14 TB02**

## 4.4 Ytelsesprofilering

[Ingen tester utarbeidet]

## 4.5 Belastningstest

### 4.5.1 TBL01 - Armering

#### Identifisering:

|                            |  |
|----------------------------|--|
| <b>Testidentifikasjon:</b> | TBL01                                  |
| <b>Beskrivelse:</b>        | Armering                               |
| <b>Type:</b>               | Belastningstest, brukergrensesnitttest |

#### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 3F  |
| <b>Use Case:</b> | UC3 |
| <b>Risiko:</b>   | R1  |

#### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.   |
| <b>Fremgangsmåte:</b>      | Under "SYSTEM"<br>Bruke mus eller touch screen til å slå armeringen på og av i varierende hastigheter over lengre tid. Ha med stoppeklokke for å dokumentere tiden. |
| <b>Forventet resultat:</b> | Armeringsknappen forblir funksjonell selv etter stor belastning.  |
| <b>Faktisk resultat:</b>   |   |

#### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 29.04.2010 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |            |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Ikke Utført        |              |            |

#### 4.5.1.1.1 Tabell14: TBL01



## 4.5.2 TBL02 – Armering: Safe Remote

### Identifisering:

|                            |  |
|----------------------------|--|
| <b>Testidentifikasjon:</b> | TBL02                                  |
| <b>Beskrivelse:</b>        | Armering: Safe Remote                  |
| <b>Type:</b>               | Belastningstest, brukergrensesnitttest |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 3F  |
| <b>Use Case:</b> | UC3 |
| <b>Risiko:</b>   | R1  |

### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.  |
| <b>Fremgangsmåte:</b>      | Under "SYSTEM"<br>Bruke mus eller touch screen til å slå "Remote Safe" knappen på og av i varierende hastigheter over lengre tid. Ha med stoppeklokke for å dokumentere tiden. |
| <b>Forventet resultat:</b> | "Remote Safe"-knappen forblir funksjonell selv etter stor belastning.  |
| <b>Faktisk resultat:</b>   |  |

### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 29.04.2010 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |            |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Ikke Utført        |              |            |

#### 4.5.2.1.1 Tabell15: TBL02

### 4.5.3 TBL03 - Nightvision

#### Identifisering:

|                            |  |
|----------------------------|--|
| <b>Testidentifikasjon:</b> | TBL03                                  |
| <b>Beskrivelse:</b>        | Nightvision                            |
| <b>Type:</b>               | Belastningstest, brukergrensesnitttest |

#### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 8F  |
| <b>Use Case:</b> | UC6 |
| <b>Risiko:</b>   | R3  |

#### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.   |
| <b>Fremgangsmåte:</b>      | Under "SIGHT"<br>Bruke mus eller touch screen til å slå Day/Night på og av i varierende hastigheter over lengre tid. Ha med stoppeklokke for å dokumentere tiden. |
| <b>Forventet resultat:</b> | Day/night-knappen forblir funksjonell selv etter stor belastning.   |
| <b>Faktisk resultat:</b>   |   |

#### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 29.04.10 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke Utført        |              |          |

#### 4.5.3.1.1 Tabell16: TBL03

#### 4.5.4 TBL04 - Zoom

##### Identifisering:

|                            |  |
|----------------------------|--|
| <b>Testidentifikasjon:</b> | TBL04                                  |
| <b>Beskrivelse:</b>        | Zoom                                   |
| <b>Type:</b>               | Belastningstest, brukergrensesnitttest |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 7F  |
| <b>Use Case:</b> | UC7 |
| <b>Risiko:</b>   | R2  |

##### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.  |
| <b>Fremgangsmåte:</b>      | Under "SIGHT"<br>Bruke mus eller touch screen til å zoome inn og ut i varierende hastigheter over lengre tid. Ha med stoppeklokke for å dokumentere tiden. |
| <b>Forventet resultat:</b> | Zoom-knappene forblir funksjonelle selv etter stor belastning.   |
| <b>Faktisk resultat:</b>   |  |

##### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 29.04.2010 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |            |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Ikke utført        |              |            |

##### 4.5.4.1.1 Tabell17: TBL04

#### 4.5.5 TBL05 – Rate of fire

##### Identifisering:

|                            |                                       |
|----------------------------|---------------------------------------|
| <b>Testidentifikasjon:</b> | TBL05                                 |
| <b>Beskrivelse:</b>        | Rate of fire                          |
| <b>Type:</b>               | Belastningstest, brukergrensesnittest |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 5F  |
| <b>Use Case:</b> | UC5 |
| <b>Risiko:</b>   | R2  |

##### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.  |
| <b>Fremgangsmåte:</b>      | Under "SYSTEM"<br>Bruke mus eller touch screen til å endre skytemodusen i varierende hastigheter over lengre tid. Ha med stoppeklokke for å dokumentere tiden. |
| <b>Forventet resultat:</b> | "Rate"-knappen forblir funksjonell selv etter stor belastning.   |
| <b>Faktisk resultat:</b>   | Den fungerte som den skulle frem til 4min og 20sek, da kræsjet programmet  |

##### Teststatus:

|                     |                    |              |          |
|---------------------|--------------------|--------------|----------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 29.04.10 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |          |
| <b>Kommentar:</b>   |                    |              |          |
| <b>Status:</b>      | Ikke Utført        |              |          |

##### 4.5.5.1.1 Tabell18: TBL05

## 4.5.6 TBL06 - Skyte

### Identifisering:

|                            |  |
|----------------------------|--|
| <b>Testidentifikasjon:</b> | TBL06                                  |
| <b>Beskrivelse:</b>        | Skyte                                  |
| <b>Type:</b>               | Belastningstest, brukergrensesnitttest |

### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 4F  |
| <b>Use Case:</b> | UC4 |
| <b>Risiko:</b>   | R1  |

### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.   |
| <b>Fremgangsmåte:</b>      | Testerens skal trykke på avtrekker knappen på joysticken i varierende hastigheter over lengre tid. Ha med stoppeklokke for å dokumentere tiden. |
| <b>Forventet resultat:</b> | Systemet skal fremdeles virke slik det skal, selv etter stor belastning på skytefunksjonen.   |
| <b>Faktisk resultat:</b>   |   |

### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 29.04.2010 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |            |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Ikke Utført        |              |            |

Tabell19: TBL06

#### 4.5.7 TBL07 - Sikte

##### Identifisering:

|                            |                                       |
|----------------------------|---------------------------------------|
| <b>Testidentifikasjon:</b> | TBL07                                 |
| <b>Beskrivelse:</b>        | Sikte                                 |
| <b>Type:</b>               | Belastningstest, brukergrensesnittest |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 2F  |
| <b>Use Case:</b> | UC2 |
| <b>Risiko:</b>   | R1  |

##### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.  |
| <b>Fremgangsmåte:</b>      | Testerens skal bruke joysticken til å se rundt omkring over en lengre periode. Ha med stoppeklokke for å dokumentere tiden.<br>Testerens skal også teste ved forskjellig sensitivitet på joysticken.<br>Husk å dokumentere tiden på hvert sensitivitetsnivå. |
| <b>Forventet resultat:</b> | Siktet fortsetter å virke slik det skal selv etter stor belastning.  |
| <b>Faktisk resultat:</b>   |  |

##### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 29.04.2010 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |            |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Ikke Utført        |              |            |

##### 4.5.7.1.1 Tabell20: TBL07

#### 4.5.8 TBL08 - Sensitivitet

##### Identifisering:

|                            |  |
|----------------------------|--|
| <b>Testidentifikasjon:</b> | TBL08                                  |
| <b>Beskrivelse:</b>        | Sensitivitet                           |
| <b>Type:</b>               | Belastningstest, brukergrensesnitttest |

##### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | 9F  |
| <b>Use Case:</b> | UC9 |
| <b>Risiko:</b>   | R3  |

##### Testing:

|                            |  |
|----------------------------|--|
| <b>Testkriterier:</b>      | Brukeren må være i VBS2 og systemet skal være bemannet.  |
| <b>Fremgangsmåte:</b>      | Under "SPEED".<br>Testeren skal bruke musen til å trykke på "+" og "-" knappene i varierende tempo over en lengre tidsperiode. |
| <b>Forventet resultat:</b> | Sensitiviteten skal fremdeles kunne byttes mellom 4 nivåer etter stor belastning.  |
| <b>Faktisk resultat:</b>   |  |

##### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 30.04.2010 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |            |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Ikke utført        |              |            |

##### 4.5.8.1.1 Tabell20: TBL08

## **Konfigurasjonstest**

[Ingen tester utarbeidet]

### **4.6 Stresstest**

[Ingen tester utarbeidet]

### **4.7 Volumtest**

[Ingen tester utarbeidet]

### **4.8 Sikkerhetstest**

[Ingen tester utarbeidet]

### **4.9 Gjenopprettingstest**

[Ingen tester utarbeidet]

### **4.10 Installasjonstest**

[Ingen tester utarbeidet]



## 4.11 Kodeinspeksjon

### 4.11.1 TKI01 - Kodestandard

#### Identifisering:

|                            |                |
|----------------------------|----------------|
| <b>Testidentifikasjon:</b> | TKI01          |
| <b>Beskrivelse:</b>        | Kodestandard   |
| <b>Type:</b>               | Kodeinspeksjon |

#### Kryssreferanse:

|                  |     |
|------------------|-----|
| <b>Krav:</b>     | n/a |
| <b>Use Case:</b> | n/a |
| <b>Risiko:</b>   | n/a |

#### Testing:

|                            |   |
|----------------------------|---|
| <b>Testkriterier:</b>      | Testeren må ha tilgang til Visual Studio 2008 og VBS2 Plugin kildekode.   |
| <b>Fremgangsmåte:</b>      | Gå gjennom alle klassene og lese kode for å se om den følger kodestandard. Det legges trykk på innrykk, blokker, mellomrom, variabelnavn, klassenavn og metodenavn. |
| <b>Forventet resultat:</b> | All kode er i henhold til kodestandard  |
| <b>Faktisk resultat:</b>   |   |

#### Teststatus:

|                     |                    |              |            |
|---------------------|--------------------|--------------|------------|
| <b>Lagt til av:</b> | Zoran Vukobratovic | <b>Dato:</b> | 21.04.2010 |
| <b>Utført av:</b>   |                    | <b>Dato:</b> |            |
| <b>Kommentar:</b>   |                    |              |            |
| <b>Status:</b>      | Ikke utført        |              |            |

#### 4.11.1.1.1 Tabell21: TBL08

#### **4.12 Referansetest**

[Ingen tester utarbeidet]

#### **4.13 Data- og databaseintegritetstest**

[Ingen tester utarbeidet]

#### **4.14 Regresjonstest**

[Ingen tester utarbeidet]



# KDA RIVET



## Teststrategi

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth

## Innhold

---

|       |  |   |
|-------|--|---|
| 1     | Introduksjon.....                            | 4 |
| 1.1   | Hensikt med dokumentet.....                  | 4 |
| 1.2   | Definisjoner, akronymer og forkortelser..... | 4 |
| 1.3   | Dokumenthistorie.....                        | 4 |
| 1.4   | Ansvarlig .....                              | 4 |
| 2     | Innledning .....                             | 5 |
| 2.1   | Krav til testingen .....                     | 5 |
| 3     | Teststrategier .....                         | 6 |
| 3.1   | Begreper .....                               | 6 |
| 3.1.1 | White Box .....                              | 6 |
| 3.1.2 | Black Box .....                              | 6 |
| 3.1.3 | Statisk Testing/Kodelesing .....             | 7 |
| 3.1.4 | Dynamisk Testing .....                       | 7 |
| 3.2   | Kvalitetsmønster.....                        | 7 |
| 3.2.1 | Pålitelighet.....                            | 7 |
| 3.2.2 | Funksjonalitet .....                         | 7 |
| 3.2.3 | Ytelse .....                                 | 7 |
| 3.3   | Faser.....                                   | 7 |
| 3.3.1 | Integrasjonstest.....                        | 8 |
| 3.3.2 | Integrasjonstest.....                        | 8 |
| 3.3.3 | Systemtest.....                              | 8 |
| 3.3.4 | Godkjennelsestest.....                       | 8 |
| 3.4   | Typer .....                                  | 8 |
| 3.4.1 | Data- og database integritetstesting.....    | 8 |
| 3.4.2 | Funksjonstesting.....                        | 8 |
| 3.4.3 | Brukergrensesnitttesting .....               | 9 |
| 3.4.4 | Ytelsesprofilering.....                      | 9 |
| 3.4.5 | Referansetesting .....                       | 9 |
| 3.4.6 | Belastningstesting .....                     | 9 |

|        |                              |    |
|--------|------------------------------|----|
| 3.4.7  | Volumtesting .....           | 9  |
| 3.4.8  | Stresstesting .....          | 9  |
| 3.4.9  | Sikkerhetstesting .....      | 9  |
| 3.4.10 | Konfigurasjonstesting .....  | 10 |
| 3.4.11 | Gjenopprettingstesting ..... | 10 |
| 3.4.12 | Installasjonstesting .....   | 10 |
| 3.4.13 | Kodeinspeksjon .....         | 10 |
| 3.4.14 | Regresjonstesting .....      | 10 |
| 4      | Ressurser og verktøy .....   | 11 |
| 4.1    | Arbeidere/roller .....       | 11 |
| 4.1.1  | Testmanager .....            | 11 |
| 4.1.2  | Testanalytiker .....         | 11 |
| 4.1.3  | Testdesigner .....           | 12 |
| 4.1.4  | Tester .....                 | 12 |
| 4.2    | Programvareverktøy .....     | 12 |
| 4.3    | Maskinvareverktøy .....      | 12 |
| 5      | Testdokumenter .....         | 13 |
| 5.1    | Testspesifikasjon .....      | 13 |
| 5.2    | Testlogg .....               | 13 |
| 5.3    | Testrapport .....            | 13 |
| 5.4    | Feillogg .....               | 13 |
| 5.5    | Kodeinspeksjon .....         | 13 |

## Figurer

|                                  |   |
|----------------------------------|---|
| Figur 1: White Box Testing ..... | 6 |
| Figur 2: Black Box Testing ..... | 7 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å gi leseren en innsikt i testing og ulike måter å teste et produkt på. I dette dokumentet beskrives hvilke teststrategier vi kommer til å bruke, hvilke ressurser som kommer til å bli brukt og hvilke "artifacts" som vi kommer til å lage mens testingen foregår.

## 1.2 Definisjoner, akronymer og forkortelser

- Artifact:** En "artifact" er informasjon i form av f. eks et dokument, en modell, kildekode og lignende.
- RUP:** Rational Unified Process.
- GUI:** Graphical User Interface.
- VBS2:** Virtual Battle Space 2.

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                          | Redaktør                                |
|---------------|----------|------------------------------------|---|
| 1.1           | 05.05.10 | Oppdatering og ny dokumentstandard | Zoran Vukobratovic                      |
| 1.99          | 27.05.10 | Korrekturlest                      | Ole-Martin Grøtterud og Thomas Transeth |
| 2.0           | 28.05.10 | Klargjort for innlevering          | Thomas Transeth                         |

## 1.4 Ansvarlig

Zoran Vukobratovic

## 2 Innledning

---

Testing er noe de aller fleste tenker på som en sluttprosedyre, men slik er det ikke. Testing starter tidlig og fortsetter å spille en viktig rolle gjennom hele prosjektet. Formålet med testing er å bestemme og sikre kvaliteten til produktet eller systemet som blir utviklet.

Testing innebærer å sørge for at alle krav har blitt oppfylt i følge spesifikasjonen, at alle komponenter i et system samarbeider riktig, at alle komponenter er implementert på riktig måte, og sist men absolutt ikke minst, at alle eventuelle feil blir oppdaget tidlig og rettet før systemet eller produktet blir tatt i bruk.

Kvaliteten av systemet blir ikke bare bestemt av fraværet av feil, men også virkningsgraden. Det hjelper lite om et system gjør alt riktig, hvis det tar for lang tid å gjøre det, da oppfylles ikke de forventningene oppdragsgiveren har hatt.

### 2.1 Krav til testingen

Hovedformålet med testing er å verifisere at alle krav som er satt opp i kravspesifikasjonen er implementert og godkjent.

Når testingen begynner er det viktig at personen som utfører testen ikke er den samme som da har skrevet koden eller laget enheten som testes. Dette er fordi personen som har laget det som skal testes ofte er "blind" på sitt eget verk, og ikke ønsker å finne feil i det. Hvis personen ikke har laget verket som testes, er personen mer objektiv, og kan tenke på å teste ut ting som utvikleren hadde ignorert, fordi han eller hun var sikker på at det virket.

Testeren bør også kunne skille mellom den negative/destruktive og positive/konstruktive tilnærmingen til testen. Testeren bør tenke på hvordan enheten kan feile, men ikke være så kritisk at enheten aldri holder mål og blir sett på som uakseptabel.

## 3 Teststrategier

---

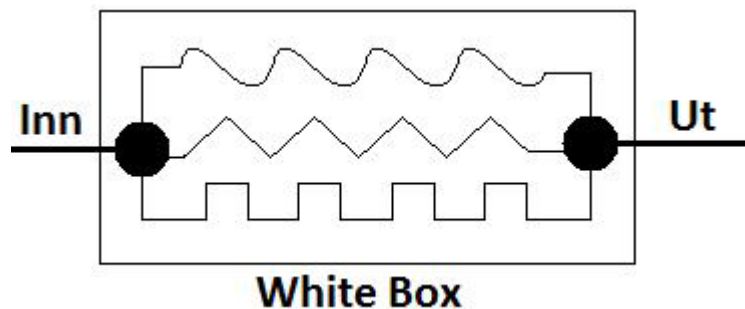
I forbindelse med testing finnes det en del ulike begreper, faser og testtyper. Det er viktig å forstå disse for å kunne forstå formålet med testingen, og for å kunne utføre effektive tester. I kapitlene under finnes en kort beskrivelse av hver av disse.

### 3.1 Begreper

#### 3.1.1 White Box

White Box testing bruker et internt perspektiv på systemet for å utvikle testcaser basert på den interne strukturen. White Box testing blir også kalt for Glass Box, Clear Box og strukturell testing.

White Box testing krever at testereren har kunnskap innen programmering og systemets struktur for å kunne identifisere alle ulike veier i systemet. Testereren tar for seg ulike testdata for å velge spesifikke veier gjennom systemet, og følger dem gjennom systemet, gjerne ved hjelp av debugging, for å bestemme passende utgangsdata.

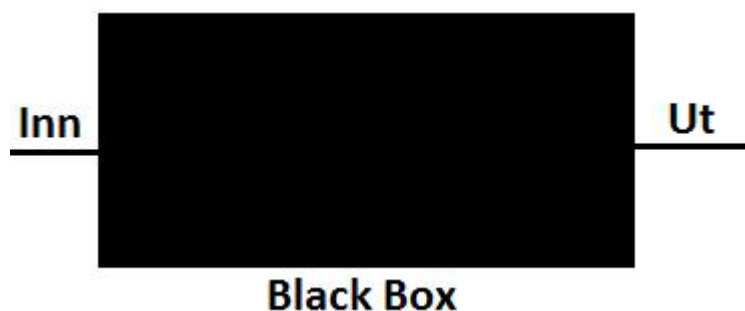


Figur 2: White Box Testing

#### 3.1.2 Black Box

Black Box testing er annerledes i og med at testereren ikke har behov for å ha kunnskap om systemets oppbygging eller programmering. Denne type testing bruker et eksternt perspektiv for å utvikle testcaser. Testereren her tar for seg bare inn-data og ut-data, og testene kan både være funksjonelle og ikke-funksjonelle.





**Figur 3: Black Box Testing**

### **3.1.3 Statisk Testing/Kodelesing**

Statisk testing er en form for testing der eksekverbar kode ikke blir compilert og kjørt. Testeren leser gjennom koden for å sjekke syntaks i kode og algoritmer. Ofte er dette en veldig god metode for å finne feil som ellers blir oversett.

### **3.1.4 Dynamisk Testing**

Dynamisk testing er den aller mest vanlige form for testing. Her blir eksekverbar kode kjørt, og testeren bruker systemet for å lete etter eventuelle feil.

## **3.2 Kvalitetsmønster**

I forbindelse med kvalitet er det vanlig at problemer som oppstår i ulike systemer ofte gjentar seg. For å kunne løse disse problemene må vi se nærmere på påliteligheten, funksjonaliteten og ytelsen til systemene.

### **3.2.1 Pålitelighet**

Systemet er motstandsdyktig til feil under eksekveringen. Dette inkluderer blant annet sammenbrudd/kræsj, minnelekasje, og system som henger seg opp/fryser.

### **3.2.2 Funksjonalitet**

Systemet utfører alle use-caser og har ønsket oppførsel etter de kravene som er satt opp i kravspesifikasjonen.

### **3.2.3 Ytelse**

Systemet utfører prosesser og svarer innenfor rimelig tid. Systemet fortsetter også å holde et akseptabelt nivå på arbeidsinnsats selv om det blir utsatt for varierende mengder av data og ressurser.

## **3.3 Faser**

Som tidligere nevnt er ikke testing en separat aktivitet der all testing blir eksekvert etter hverandre. I stedet er testing delt opp mot ulike formål og faser i utviklingen av et system. Fasene fokuserer på alt fra små fragmenter til testing av hele systemet. Testfasene er delt i fire, der alle har ulike formål som blir beskrevet i kapitlene under:

- Enhetstest
- Integrasjonstest
- Systemtest
- Godkjennelsestest

Enhetstest og integrasjonstest er basert på White Box testing, mens systemtest og godkjennelsestest er basert på Black Box testing.

### 3.3.1 Integrasjonstest

De minste testbare delene av et system blir testet individuelt, ofte på samme tidspunkt som de blir implementert. Det er viktig at feilene blir oppdaget her, ellers overlever de videre og forårsaker feil og forsinkelser.

### 3.3.2 Integrasjonstest

Hele systemet deles opp i mindre deler, kalt moduler. Modulene består av de minste testbare delene fra enhetstesting som til sammen gir en "output". Hensikten med integrasjonstesting er å avdekke feil som finnes i samarbeidet mellom moduler, som for det meste er GUI og kommunikasjonsfeil.

### 3.3.3 Systemtest

Systemtesting er testing av systemet opp mot kravspesifikasjonen. Som nevnt i *kapittel 3.3, Faser*, er dette en Black Box test, og krever ingen kunnskap om oppbyggingen til systemet.

### 3.3.4 Godkjennelsestest

Det ferdige produktet testes av representanter fra oppdragsgiveren, med formålet om enten å godkjenne eller avvise systemet.

## 3.4 Typer

I forbindelse med testing finnes det mange ulike typer testing. Under kan dere lese om noen av de mest kjente.

### 3.4.1 Data- og database integritetstesting

Data og database integritetstesting har som hensikt å verifisere at systemet lagrer data på en slik måte at dataen ikke utsettes for fare i forbindelse med uthenting, oppdateringer eller gjenopprettinger. Denne testen avdekker feil som kan ende opp med korrupsjon av data, generelle feil med databasen, og uautorisert datatilgang.

### 3.4.2 Funksjonstesting

Denne testen er den vanligste type testing. Her testes systemet opp mot kravene som er satt i kravspesifikasjonen for å se at alle krav har blitt oppnådd. Funksjonstesting er ofte brukt som en demonstrasjon for å bevise at systemet virker etter behov.

### 3.4.3 Brukergrensesnitttesting

Hensikten med denne testen er å verifisere brukerens grensesnitt med systemet. Her er det viktig å være sikker på at brukeren kan bruke systemet på en riktig måte, i vårt tilfelle blir dette bruken av mus, tastatur og joystick.

### 3.4.4 Ytelsesprofilering

Ytelsesprofilering tester systemets responstider, reaksjonstider, overføringshastighet og alle andre tidskrevende krav.

### 3.4.5 Referansetesting

Referansetesting tester systemet opp mot lignende kjente systemer, denne type testing er sterkt knyttet til ytelsesprofilering.

### 3.4.6 Belastningstesting

Under belastningstesting påføres systemet varierende mengder av data over lang tid for å se hvordan systemet håndterer det. Hensikten med denne testen er å se om systemet kan arbeide over den tiltenkte maksimum arbeidsmengden. Denne testen evaluerer også ytelseskarakteristikken som testes i ytelsesprofilering.

### 3.4.7 Volumtesting

Volumtesting er nesten det samme som belastningstesting, men her blir systemet testet over en viss tidsperiode.

### 3.4.8 Stresstesting

Stresstesting er en type ytelsestesting der systemet testes med ekstreme tilstander eller betingelser som for eksempel veldig mange brukere eller minimale ressurser.

### 3.4.9 Sikkerhetstesting

Sikkerhetstesting er en test som går ut på å teste systemets sikkerhet. Sikkerhetstesting deles inn i to nøkkelområder:

- Applikasjonsnivå
- Systemnivå

#### 3.4.9.1 Applikasjonsnivå

Applikasjonsnivåsikkerhet ivaretar at en aktør/bruker med sikkerhetsbegrensninger kun har tilgang til spesifikke funksjoner, data som er tilgjengelig eller use-caser. Eksempelvis kan man se på en vanlig bruker som aksesserer en database. Denne brukeren henter ut informasjon fra databasen, men får da begrenset informasjonsmengde. For å få resten av informasjonen må brukeren ha høyere sikkerhetstilgang.

### **3.4.9.2 Systemnivå**

Systemnivåssikkerhet sørger for at kun de brukerne som har relevant tilgang til systemet er i stand til å aksessere systemet. Det er også viktig at aksesseringen kun kan skje gjennom de "rette kanalene", som for eksempel ved fjernkontrollering fra en spesiell datamaskin eller gjennom en gitt port.

### **3.4.10 Konfigurasjonstesting**

Konfigurasjonstesting har som hensikt å verifisere operasjonen av systemet på ulike programvare- og maskinvarekonfigurasjoner. Det kan være mange forskjeller mellom ulike maskiner som kan brukes, og det er derfor viktig å produsere et system som kan tilvenne seg ulike teknologier.

### **3.4.11 Gjenopprettingstesting**

Gjenopprettingstesting sikrer at systemet kan feile for så gjenopprette seg selv. De ulike systemfeilene kan variere fra maskinvare-, programvare- eller nettverksfeil. Det er viktig at når en systemfeil forekommer, skal et avlørsystem ta over kontrollen fra systemet og sørge for at tapet av data forblir minimalt.

### **3.4.12 Installasjonstesting**

Installasjonstesting har to formål, den første gjelder installasjon ved forskjellige forhold og den andre gjelder verifisering av systemet etter installasjon. I tillegg kan installasjonstesting referere til grensesnittet til maskinvaren.

### **3.4.13 Kodeinspeksjon**

I forbindelse med programmeringsdelen av prosjektet er det viktig at koden holder en viss standard. For å overholde dette er det laget en intern kodenstandard av gruppen som vi til enhver tid skal følge. Det er derfor viktig jevnlig å kontrollere syntaks og andre forhold som kodenstandardens spesifiserer. Kodeinspeksjonen utføres manuelt av bruker og inngår i statisk testing.

### **3.4.14 Regresjonstesting**

Regresjonstesting er en type test som har til hensikt å avdekke systemregresjon. Regresjon forekommer når funksjoner i et system slutter å fungere, når denne funksjonen tidligere har fungert. Hensikten med regresjonstesting er:

- Å sikre at tidligere identifiserte feil har blitt tatt hånd om.
- Å sikre at endringen til tidligere identifiserte feil ikke har introdusert nye feil eller mangler i systemet.

## 4 Ressurser og verktøy

---

I forbindelse med testingen får vi bruk for å definere ressurser og verktøy som skal benyttes under arbeidet:

- Arbeidere/roller
- Programvareverktøy
- Maskinvareverktøy

### 4.1 Arbeidere/roller

Det er fire typer roller som er definert i RUP i forbindelse med testrelaterte aktiviteter:

- Testmanager
- Testanalytiker
- Testdesigner
- Tester

Disse rollene er gitt på bakgrunn av ferdigheter og ansvarsområder. I en stor organisasjon eller et prosjekt er det naturlig å la hver rolle være fylt av en individuell arbeider.

I et mindre prosjekt er det ofte vanlig at alle rollene fylles av en og samme person. Unntaket er rollen "Tester", som oftest er en rolle for alle gruppe-medlemmene.

På grunnlag av dette vil testansvarlig derfor fylle rollene "testmanager", "testanalytiker" og "testdesigner", mens "tester" er en rolle som vil fylles av samtlige gruppe-medlemmer.

#### 4.1.1 Testmanager

Testmanager har som hovedansvar å administrere testingen under prosjektet. De viktigste ansvarsområdene for en testmanager er:

- Administrere og planlegge testingen.
- Identifisere hvilke "artifacts" som skal produseres.
- Definere hvilke testfaser og teststrategier som er nødvendige.
- Definere hvor mye av arbeidet som skal testes.
- Innhente nødvendige ressurser.
- Verifisere testresultater sammen med gruppe-medlemmer og oppdragsgiver.

#### 4.1.2 Testanalytiker

Testanalytiker har som hovedansvar å identifisere og definere de ulike testene som skal utføres. De viktigste ansvarsområdene for en testanalytiker er:

- Identifisere testidé.
- Definere testdetaljer ved hjelp av testcaser.
- Avgjøre testresultater.
- Evaluere produktets kvalitet.
- Identifisere verktøy som behøves for å gjøre testene.

### 4.1.3 Testdesigner

Testdesigner har som hovedansvar å definere den tekniske tilnærmingen til implementasjonen av en test. De viktigste ansvarsområdene for en testdesigner er:

- Definere tilnærmingen til testing.
- Sikre en vellykket testimplementasjon.
- Definere testbare elementer.
- Verifisere testteknikker.

### 4.1.4 Tester

Tester har som hovedansvar å implementere og eksekvering av tester. De viktigste ansvarsområdene for testeren er:

- Implementere og eksekvere tester.
- Loggføre resultater.
- Analysere og dokumentere resultat.
- Dokumentere feil i egen feilrapport.

## 4.2 Programvareverktøy

Følgende programvareverktøy skal benyttes under testingen:

- Microsoft Visual Studio 2008 SP1
- VBS2

## 4.3 Maskinvareverktøy

Følgende maskinvareverktøy skal benyttes under testingen:

- Brukerens egen datamaskin.
- Gruppe-datamaskinen som vi har fått fra oppdragsgiver.

## 5 Testdokumenter

---

I prosjektperioden vil vi produsere en del ulike testdokumenter. Disse dokumentene er viktige for å styrke systemets kvalitet gjennom hele prosjektarbeidet. Dokumentene vil også gi en god indikasjon på hvor mye tid vi har brukt for å kvalitetssikre systemet i henhold til kravspesifikasjonen.

### 5.1 Testspesifikasjon

Testspesifikasjonen har som hensikt å gi en god beskrivelse av hvilke tester som skal gjøres. Testspesifikasjonen inneholder informasjon om hver enkelt test og en kryssreferanse til alle dokumenter som er relevante.

### 5.2 Testlogg

Hensikten med testloggen er å dokumentere resultatet fra en test. Testloggen skal inneholde informasjon om hvilken test som er utført og av hvem. Testloggen skal i utgangspunktet ikke kreve noe teknisk innsikt eller kunnskap om systemet. Testloggen konkluderer kun om testen er godkjent eller ikke ved å se på det faktiske resultatet.

### 5.3 Testrapport

Hensikten med testrapporten er å gi en overordnet beskrivelse av en del av systemet. Testrapporten kan inneholde mange testlogger som omhandler samme del av systemet. På bakgrunn av testloggene skrives det da en testrapport om den delen av systemet, og dette vil være et mer teknisk dokument. Det er vanlig at testrapporten skrives av en person som har god kunnskap om den relevante delen av systemet.

### 5.4 Feillogg

I forbindelse med feildeteksjon skal det skrives en logg på den aktuelle feilen. Alle feil som blir funnet i systemet skal dokumenteres i henhold til mal for feillogg.

### 5.5 Kodeinspeksjon

Hensikten med kodeinspeksjonsdokumentet er å beskrive en strukturert og god planlegging, gjennomføring og kontroll av programkode skrevet av gruppemedlemmene. Prosessen vil ikke bare gagne systemet, men alle parter involvert. Programmererne vil lære av hverandres feil og mangler, i tillegg til å lære alt det positive en medprogrammerer kan gjøre.

# KDA RIVET



## Utvidelse av oppgaven

---

Lone Knutsen

---

Arne Kristian Amellem

---

Zoran Vukobratovic

---

Ole-Martin Grøttenud

---

Thomas Transeth



## Innhold

---

|       |   |   |
|-------|---|---|
| 1     | Introduksjon.....                                 | 3 |
| 1.1   | Hensikt med dokumentet.....                       | 3 |
| 1.2   | Definisjoner, akronymer og forkortelser.....      | 3 |
| 1.3   | Dokumenthistorie.....                             | 3 |
| 1.4   | Ansvarlig .....                                   | 3 |
| 2     | Ønskede utvidelser .....                          | 4 |
| 2.1   | Skifte crosshair.....                             | 4 |
| 2.2   | Watch face.....                                   | 4 |
| 2.3   | Menysystem .....                                  | 4 |
| 2.4   | Mulighet for å velge FCU .....                    | 4 |
| 2.5   | Implementere laser på sikte .....                 | 4 |
| 2.6   | Slå av auto-reload.....                           | 4 |
| 3     | Gjennomførte og ikke-gjennomførte utvidelser..... | 5 |
| 3.1   | Gjennomførte utvidelser: .....                    | 5 |
| 3.1.1 | Menysystem .....                                  | 5 |
| 3.1.2 | Mulighet for å velge FCU .....                    | 5 |
| 3.2   | Ikke-gjennomførte utvidelser .....                | 5 |
| 3.2.1 | Skifte crosshair .....                            | 5 |
| 3.2.2 | Watch face.....                                   | 5 |
| 3.2.3 | Laser på sikte .....                              | 5 |
| 3.2.4 | Slå av auto-reload.....                           | 5 |
| 4     | Konklusjon.....                                   | 6 |

# 1 Introduksjon

---

## 1.1 Hensikt med dokumentet

Hensikten med dokumentet er å dokumentere ønskede utvidelser av oppgaven, men som samtidig ikke blir sett på som krav.

## 1.2 Definisjoner, akronymer og forkortelser

## 1.3 Dokumenthistorie

| Utkast nummer | Dato     | Aktivitet                 | Redaktør                                       |
|---------------|----------|---------------------------|--|
| 0.1           | 05.05.10 | Førsteutkast              | Lone Knutsen                                   |
| 0.99          | 27.05.10 | Korrekturlest og endret   | Arne Kristian Åmellem<br>og Zoran Vukobratovic |
| 1.0           | 28.05.10 | Klargjort for innlevering | Thomas Transeth                                |

## 1.4 Ansvarlig

Lone Knutsen

## 2 Ønskede utvidelser

---

Etter ønske fra KDA så var det noen ekstra utvidelser som de gjerne ville ha med dersom vi skulle bli ferdig med oppgaven før tiden. Det var i hovedsak snakk om tre utvidelser vi skulle se på og vurdere om ville være mulig å gjennomføre. Disse er:

### 2.1 Skifte crosshair

I VBS2 eksisterer det allerede et crosshair, men dette er ikke likt det i Protector treneren. Det er da ønskelig å se på om det er mulig å bytte ut dette med to egendefinerte som finnes i Protector, og å lage en mekanikk for å switche mellom disse.

### 2.2 Watch face

I Protector treneren er det et slags kompass som hele tiden viser hvordan RWS'en står i forhold til bilen den er montert på. Det er ønskelig å se på muligheter for å implementere denne i VBS2, slik den er i Protector.

### 2.3 Menysystem

I Protector er det implementert et menysystem som det er mulig å manøvrere i ved hjelp av knappene på GUI'et og deretter velge ønskede funksjoner. Det er ønskelig at vi ser på muligheten for å implementere menyfunksjonen, slik at menyen er på plass, men uten de funksjonene som er i den opprinnelige menyen i Protector.

### 2.4 Mulighet for å velge FCU

Etttersom det finnes flere typer RWS, så var det ønskelig å se på muligheten for å bytte mellom forskjellige brukergrensesnitt avhengig av hva slags våpen som er i bruk. Vi valgte derfor å se på muligheten for å sette opp en bil med granatkastere, med tilhørende brukergrensesnitt og muligheten for å veksle mellom bilene.

### 2.5 Implementere laser på sikte

Dersom det viser seg mulig å gjøre, så er det ønskelig å få implementert laser på sikte.

### 2.6 Slå av auto-reload

Dersom det viser seg mulig er det ønskelig å slå av auto-reload av RWS'en slik at soldaten selv må venne seg til å lade våpenet når det er tomt for ammunisjon.

## 3 Gjennomførte og ikke-gjennomførte utvidelser

---

Etter å ha gått gjennom hva som støttes av VBS2 og hva som ikke støttes, så er det noen utvidelser som er gjennomført og noen som ikke har blitt gjennomført. Grunnen til at noen av utvidelsene ikke ble gjennomført er at noe var det manglende støtte for i VBS2 eller dem er allerede styrt av VBS2 og ikke lar seg overstyre.

### 3.1 Gjennomførte utvidelser:

#### 3.1.1 Menysystem

Via script var det mulig å kode menyen inn i selve VBS2 og å gjøre kall til denne via brukergrensensnittet vårt. Menyene er manøvrerbar på lik linje som i Protector treneren, men er uten funksjoner. Dette var i henhold til KDA sine ønsker.

#### 3.1.2 Mulighet for å velge FCU

Vi valgte å lage et brukergrensensnitt, beregnet på granatkastere og legge dette til en ny bil med granatkaster på. I tillegg gjorde vi det mulig å veksle mellom de to bilene som finnes i scenarioet, slik at man kan bemanne den bilen som er mest hensiktsmessig.

### 3.2 Ikke-gjennomførte utvidelser

#### 3.2.1 Skifte crosshair

Ettersom vi ikke fant noen informasjon angående om det er mulig å bytte ut det eksisterende siktet med egendefinerte sikter, så tok vi kontakt med Bohemia Interactive. Svaret vi fikk tilbake var at det var mulig dersom vi bygget vår egen modell. Siden prosjektet var i en avsluttende fase valgte vi ikke å gjennomføre denne utvidelsen da tiden det ville ta å bygge en ny modell fra bunn av ikke var tilstrekkelig.

#### 3.2.2 Watch face

Etter å ha gjort noen tester så fant vi ut at vi fikk inn bildet av "kompasset" på samme måte som vi fikk inn menyen, men at eventuell korreksjon av kompass i forhold til posisjon på bilen ville skape konflikter mellom script og VBS2. Dette vil si at vi hadde valget mellom å få systemet til å skyte eller å få systemet til å vise riktig retning. Vi valgte da å la systemet beholde skytefunksjonen sin.

#### 3.2.3 Laser på sikte

Pga manglende støtte for laser i VBS2 så ble ikke denne gjennomført.

#### 3.2.4 Slå av auto-reload

Ikke gjennomført pga manglende mulighet for å slå av auto-reload.

## 4 Konklusjon

---

Det vi har lagt vekt på i hvilke utvidelser som har blitt gjort og hvilke vi valgte og ikke gjennomføre er hva det var støtte for i VBS2 og hvor ressurskrevende utvidelsene ville være. Vi har valgt å gjennomføre de utvidelsene det var støtte for, og som var mulig å rekke innen fristen for levering av oppgaven.

De gjennomførte utvidelsene er utvidelser som er med på å forbedre brukeropplevelsen og ytelsen til systemet.