

01/04/2019



**Gruppe:**

Andreas Rømer  
Ali Al-Sharefi  
Wajid Ahmad  
Leopold Nguessan

**Undervisere:**

Jarl Tuxen  
Asger B. Clausen  
David H. Ema

## Indholdsfortegnelse

Indledning	4
Afgrænsning	5
Requirements	5
IT organisation	11
Risikoanalyse	13
FURPS+	16
Functionality:	16
Usability:	16
Reliability:	16
Performance:	16
Supportability:	16
Use Cases:	17
Use case Diagram	23
Domæne model:	24
Entity Relationship Diagram	24
SSD UC4 - Skibs bevægelse:	25
SD UC2 - Valg af scenarie:	26
Klassediagram	27
Konklusion	28
Bilag:	29
Kanban/ Trello	32

## Indledning

Spil er et marked som er blevet rigtig stort, især digitalt. Der flere og flere mennesker der foretrækker det digitale frem for det fysiske brætspil, vi tror årsagen til det er blandt andet at man nu kan spille på tværs af lande og kontinenter. Vi oplever at flere og flere mennesker blandt andet har en Computer, Smartphone eller en spillekonsol. Det er efterhånden blevet et lukrativt marked inden for digitale spil.

Vi har til opgave fået stillet at lave et digitalt sejlads spil, vores tanker omkring spillet er at det skal kunne spilles på tværs af computere i første omgang og skulle henholdsvis bestå af.

Spillet består af:

- 3 forskellige former af skibe
- 2 forskellige størrelser af baner
- 3 former af våben
- Turbaseret-Spil
- Man skal kunne udveksle kamp dueller
- En database der holder skibene og deres værdier.

Måden vi kommer til at arbejde på er i forhold til UP-Process som er agilt(Mulighed for at gå frem og tilbage i projektet) med hjælp af diverse UML-Analyseværktøjer

Når vi arbejder med UP-Processen, så er der 4 faser vi skal igennem

1. Inception - I denne fase overvejer vi hvad der skal med i projektet, krav og brainstormer.
2. Elaboration - Denne fase laver vi en mere detaljeret plan over projektet som vi måske ikke kommer til at holde 100%, vi prøver at lave nogle deadlines som vi selvfølgelig vil prøve at overholde.
3. Construction - Fasen hvor vi begynder med at kode og laver nye deadlines
4. Transition - Her finder vi en deadline for hvornår projektet er klar til brug

Vi bruger Trello som et Kanban værktøj, som vi hjælpe os med at have et overblik over deadlines, prioriteringer samt hvor langt vi er med vores projekt. Se mere i afsnittet "Kanban".

## Afgrænsning

I front-end kunne vi sagtens få noget op på skærmen som havet og skibene. Der er desværre bare ikke noget at interagere med. Så angående interface mangler der evnen til at indtaste ordre.

I forhold til back-end er der ikke meget der holder os tilbage. Der mangler nogle få funktioner angående critical hit med hvilken type ammunition skibene skyder med og hvilken skade den så påfører på modstanders skibe. Der mangler dog også en movement modul der godkender spillernes indtastning af koordinater når de skal bevæge deres skibe.

Årsagerne til vi blandt ikke når i mål, er blandt andet hvordan vi skulle gribe opgaven an da der var en del tvivl til at starte med. Vi startet blandt andet med at kode i javascript og fik da en flot UI frem med både skibe og våben der kunne bevæge sig frit frem og tilbage, da det ikke kunne anvendes i thyme leaf/spring blev det dermed skrottet og det var tid til at starte på ny. Det var først længere hen i forløbet at vi fik en udvidet forståelse med spillets regler, opbygning og arbejdsmetoder.

## Requirements

### Krav omkring Server/Communication Controller

REQ no.	Requirement:	Kommentarer
KR001	Spillet spilles af 2 spillere - En server og en client der kobler sig på serveren.	

KR002	<p>Spillet skal håndtere turns. Hvert turn indeholder følgende i følgende rækkefølge:</p> <ol style="list-style-type: none"> <li>1. Vejr determination KR003</li> <li>2. Send / Modtag via TCP / IP. <ol style="list-style-type: none"> <li>a. Vejret (sendes fra serversiden)</li> <li>b. de førhenværende moves og attacks?</li> </ol> </li> <li>3. Spillet bestemmer movement damage (KR206)</li> <li>4. Spillet bestemmer attack damage (KR303)</li> <li>5. UI skal vise aktioner foretaget i runden.</li> <li>6. UI modtager nye aktioner fra brugeren.</li> <li>7. Spillet checker om brugernes aktioner er ok (validering) (KR207)</li> </ol>	
KR003	<p>Vejrets retning bestemmes af serveren og skal være tilfældigt fra turn til turn. Vejret har betydning for skibenes hastighed, se (KR203).</p>	
KR004	<p>Spilleregler:</p> <ul style="list-style-type: none"> <li>- Vinderen af spillet er den som har det sidste flydende skib eller det den spiller der har taget mindst skade på skibe efter et fixed antal turns.</li> </ul>	Hører den til her i Controller?

## Krav omkring User interface

REQ no.	Requirement:	Kommentarer
KR101	Brugeren kan se hvilke skibe han/hende har tilgængelig og hele banen er synlig men det ikke muligt at se fjenden skibe.	Deres attributter vil også være tilgængelig for brugeren til at kunne se.
KR102	Brugeren kan give ordre til skibene om hvilke aktioner de skal tage.	Vejret påvirker brugerens skibe.
KR103	Brugerne giver deres ordre på den samme tur. Aktionerne kommer først frem næste tur.	
KR104	Hvis brugeren indtaster aktioner der ikke godkendes af movement modul bliver de afvist.	Der skal indtastes korrekte koordinator.

## Krav omkring Movement

REQ no.	Requirement:	Kommentarer
KR201	Alt bevægelse er styret af et hexagons koordinatsystem.	
KR202	Hvert skib har sin egen hastighed og den kan sejle fra 0 til max.	
KR203	Skibets hastighed bliver bestemt af vindhastigheden. <ul style="list-style-type: none"><li>- Modvind direkte = no speed</li><li>- Modvind højre/venstre = Max speed - 2</li><li>- Medvind = Max speed.</li></ul>	
KR204	Collision sker når 2 eller flere skibe sejler ind i den samme Hex på samme tur.	
KR206	Et skib må kun dreje et vis antal gange.	Hvert skib har sit egen movement pattern
KR207	Tjek for legal movement efter brugeren har lavet sit move.	

## Krav omkring Gun Control

REQ no.	Requirement:	Kommentarer
KR301	et vis antal af våben pr. række er monteret på begge sider af skibet, dvs. halvdelen af nummeret på hver side af skibet. 3 mænd er nødvendige pr. enhed til støtte for reload og affyring af dem under en kamp	
KR302	Skibet har våben på begge sider Skibet kan skyde med ammunition hver anden runde Du kan skifte "våben", men det koster en runde!	
KR303	3 former af våben: <b>Kanonkugle:</b> de kan beskadige skroget, og i tilfælde af et kritisk slag kan sprænge hele skibet. <b>Reach:</b> Kort til lang rækkevidde.  <b>Kæder:</b> Kæder med en mindre bold i begge ender. De kan beskadige riggen alvorligt. <b>Reach:</b> Gennemsnitligt afstand  <b>Grape Shot/drue skud:</b> Masser af løse stykker af jern. Det kan decimere antallet af sejlere. <b>Reach:</b> Kort rækkevidde.	for at se hvor meget de forskellige våben skader  Se: KR306



KR304	3 Forskellige former af skibe med forskellige krav for at kunne føre og gå til slag	Se skibs modellerne S.4
KR305	<p>Kollisionsskade, sker når et eller flere skiv prøver at indtage samme hex under bevægelse. Når kollisionen sker, er hastigheden 0. Skaden tager udgangspunkt i 2 formler</p> <p>Hull skade = <math>\frac{1}{3}</math> af den faktiske skibsværdi af det andet skib</p> <p>Actual hull = Forrige Faktisk Hull - Hull Damage</p>	
KR306	Skibene har mulighed for at skade 40% eller 10% afhængende af koordinatsystemet.	

# IT organisation

## SWOT analyse

Swot-analysen tager udgangspunkt i Ships and Sails, som er et spil under udvikling.

Den er udarbejdet på baggrund af spillets kommende udviklingsfaser

(Visse elementer i analysen er opfundet af os)

<b>Interne situation</b>	
<i>(Styrker)</i>	<i>(Svagheder)</i>
<i>Digitalt spil. Multiplayer. Interaktive spil. Gen-spilbart. Rangliste.</i>	<i>Grafik. UI. Pc-spil. Ingen mulighed for singleplayer.</i>
<b>Eksterne situation</b>	
<i>(Muligheder)</i>	<i>(Trusler)</i>
<i>Gratis for alle. Samarbejde med virksomheder. Events. Reklamer.</i>	<i>Ingen interesse. Allerede udviklet.</i>

Det er en styrke, at spillet har et digitalt interface som blandt andet brugerne har mulighed for at kunne integrere med, da planen med spillet er at det skal kunne spilles online. Det skal være et interaktivt spil, hvor man både skal anvende taktik og spille strategisk. Vi tager blandt andet højde for at spillet skal kunne spille igen og igen, og for at ikke at miste brugeren vil vi blandt andet indføre en rangliste så man kan komme til at spille med dem der er på niveau med en selv.

Nogle af de svagheder vores spil ville stå over for, vil vi selvfølgelig prøve at tackle så godt som muligt. Grafikken & UI vil vi blandt andet investere i multimediedesigner og 3D grafiker, som vil give vores spil et markant boost. Til at starte med vil spillet kun være tilgængelig for pc-bruger, her vil vi meget gerne ud og konkurrencere på flere platforme. Her håber vi på at komme ud på tablets og telefoner hvor man så også vil have mulighed for at spille på tværs af enheder. Er man her som bruger ikke i humør til et dybt seriøst spil, vil vi i fremtiden lave en singleplayer del hvor man så har mulighed for at spille mod en AI.

Umiddelbart ser vi ikke ingen trusler på markedet, den eneste trussel vi vil mærke er hvis spillet ingen interesse fanger.

Spillet vil være tilgængelig for alle voksne, unge og børn. Vi vil blandt andet optræde til diverse events for at reklamere for vores spil, i den forbindelse håber vi undervejs at vi får et samarbejde med virksomheder som måske ville investere i os.

Vores umiddelbare indtjeningskilde vil være reklamer til at starte med, vi vil efterfølgende med tiden tilføje plugins hvor du som bruger har mulighed for at købe nye baner, skins og nye typer af våben.

## Risikoanalyse

Denne risikoanalyse er udarbejdet sådan, som man ville forvente at gøre det i en rigtig situation med en virksomhed eller kunde.

### Ordforklaringer:

**Sandsynlighed S** = hvor sandsynligt det er, at risikomomentet indtræffer på en skala fra 1 til 5:

1. Meget lav
2. Lav
3. Moderat
4. Høj
5. Meget høj

**Konsekvens K** = hvor alvorlige konsekvenser det har for projektet, hvis risikomomentet indtræffer på en skala fra 1 til 10:

1. Ubetydelig
3. Tålelig
7. Alvorlig
10. Katastrofal

Risikotal:  $R = S * K$

Aktionsplan: hvordan risikomomentet undgås, og hvad der skal gøres, hvis det alligevel indtræffer. Der rettes særlig opmærksomhed på alle risikomomenter med risikotal på over 9 i overensstemmelse med aktionsplanen. Der foretages løbende risikovurdering, og aktionsplanen justeres efter behov.

Risiko	S	K	R	Beskrivelse
Spring/java	4	7	28	Vi har ikke arbejdet ret meget med Spring/java, så der vil højst sandsynligt komme nogle tekniske problemer
MVC	3	7	21	Vi har ikke arbejdet ret meget med mvc, så der vil højst sandsynligt komme nogle tekniske problemer
Git	1	3	3	Der kan godt opstå konflikter, når vi bruger versionsstyring, og det er værd at tage højde for.
Sygdom	3	3	9	Vi har også valgt at tage sygdom med i risikoanalysen.

Java/Spring samt MVC er 2 teknologier, som vi ikke har så meget kendskab til, da vi ikke har så meget med det før. Vi har derfor vurderet at der er 50% sandsynlighed for at der vil opstå tekniske problemer. Det er især i starten når man lærer nye teknologier, det medføre så risiko for tidstab.

Vi bruger github som versionsstyring, da alle fra gruppen har erfaring med det. Der kan dog stadig opstå problemer idet vi alle arbejder med filerne og der hyppigt vil komme ændringer. Vi har derfor med lav sandsynlighed og god kommunikation formentlig kunne komme til at undgå disse problemer, sygdom er også noget vi har taget højde for da det kan ske for alle og enhver.

## **Business mission**

- Hvad er det virksomheden vil opnå?
  - o Et underholdende spil, som er tilgængelig for alle unge, voksne og de ældre.
- Hvad vil den levere til sine omgivelser (kunder)?
  - o Et spil hvor man har mulighed for at kunne konkurrere mod venner
  - o Et digitalt spil
- Virksomhedssiden af programmet
  - o Markedsføring og strukturering
  - o Micro-transaktioner: da spillet er gratis
  - o Micro-transaktioner: kunne f.eks være nyt udstyr såsom våben til skibe
- Hvilke behov har kunderne?
  - o Krigs fuldt spil
  - o forskellige årstider med informationsrigt og lærerigt
  - o Nemt at starte og afslutte.

## **Business vision:**

- Multiplayer
  - o Spillet skal kunne spilles af flere personer
  - o Taktiske turneringer
- Grafisk design
  - o Man skal have mulighed for at bevæge sig rundt på oceanet
  - o Vi vil gå fra 2d til 3d animation
- Udvidelsespakker
  - o Flere og større baner, hvor der kræves en større indsats af spilleren for at kunne vinde
  - o Nye historier
  - o Udvidet sortiment af våben og skibe samt opgraderinger
  - o World league, her vil kun de bedste konkurrere om priser

## **FURPS+**

### ***Functionality:***

Dette program/spil vil kunne implementere et skibs spil, hvor du kæmper mod en anden modstander. Spillet vil foregå sådan at brugeren kontrollerer en gruppe skibe gennem en konsol, hvor de giver hvilke ordrer de vil have deres skibe skal foretage. Spillet vil blive kodet gennem Java hvorefter det så vil blive vist på en hjemmeside via HTML, Spring og UI via JavaScript. Egenskaberne for skibene og scenarier vil blive gemt i en Database skabt via MySQL.

### ***Usability:***

Programmet vil fokusere på kun at tage de nødvendige oplysninger som brugeren har brug for. Derfor vil brugeren ved hjælp af 'user interface' kunne se deres skibe på banen, deres modstander og har mulighed for valg af scenarie.

### ***Reliability:***

Både programmet og filerne for spillet, hvor alt vores data ligger vil blive uploadet på Github og gemt på hver af gruppens computer.

### ***Performance:***

Da der skal være mere end en klient for at kunne spille spillet har vi valgt at kode programmet server i multithreaded, sådan at den kan acceptere forbindelsen fra en klient/bruger, starte en thread for den kommunikation og fortsat tage 'requests' fra other klienter/bruger.

### ***Supportability:***

Programmet vil blive støttet af Java, Spring, HTML, JavaScript og MySQL. Både spillet og programmet vil foregå på engelsk.

Use Cases:

**Use case oversigt:**

**UC1:** venter på modstander

**UC2:** valg af scenarie:

**UC3:** Kort(Koordinatsystem)

**UC4:** Skibs bevægelse:

**UC5:** Kollision:

**UC6:** Runder/træk

**UC7:** Vind og hastighed:

**UC8:** Ammunition/skibsmænd:

**UC9:** Reload:

**UC10:** Skade delegering:

**UC11:** Slutspillet

**UC1:** venter på modstander

Overview: Brugeren åbner programmet for at køre spillet

Actors: Brug, System

Preconditions: forbindelse til databasen.

Main Success Scenario:

1. bruger starter spillet
2. bruger venter på forbindelse til modstander
3. når forbindelsen træder i kraft vil den vise en besked med  
"Connection successful"



**UC2:** valg af scenarie:

Overview: Bruger har en aktiv forbindelse til modstander

Actors: Bruger, System.

Preconditions: en aktiv forbindelse til modstanderen, forbindelse til DB.

Main Success Scenario:

1. Brugeren har nu mulighed for at vælge forskellige scenarier
2. Der kan vælges mellem flere forskellige nationaliteter, og forskellige placeringer på kortet.
3. Hver nation har 3 forskellige sejl Brig, Ship of the line og man of war
  - a. Se Bilag 1 vedrørende egenskaber for de forskellige skibe

**UC3:** Kort(Koordinatsystem)

Overview: Brugeren har valgt scenarie

Actors: Bruger, System.

Preconditions: forbindelse til modstander og forbindelse til DB.

Main Success Scenario:

- 1: Kortet er bygget op omkring et koordinatsystem. Felterne er sekskantet
- 2: Brugeren har mulighed ved hjælp af musen til at flytte sig rundt på tværs af havet. Se use case 4
- 3: Kortet er begrænset, det betyder at hvis en brugeren bevæger ud for grænsen. Så vil skibet/sejlet forsvinde.

## Fully Dressed Use Case

Use case name:	<b>UC4:</b> Skibs Bevægelse
Scope	Sails & Ships spillet.
Level	Brugerniveau
Primary actor	Bruger, System.
Stakeholder and interests	Alle der vil spille vores spil.
Preconditions	1. forbindelse til modstander 2.Valg af scenariet
Succes Guarantee	Skibet bevæger sig fra A til B på kortet.
Main Success Scenario	<ol style="list-style-type: none"> <li>1. Brugeren vælger et skib</li> <li>2. Brugeren foretager sit træk, ved at bruge musen til at klikke på pile på skærmen.</li> <li>3. Brugeren vælger et nyt skib og starter forfra.</li> <li>4. Brugeren kan nu vælge om han vil reload eller skyde (USE CASE 9 og 10)</li> </ol>
Extensions	<p>1a. bruger vælger et skib som allerede har rykket:</p> <ul style="list-style-type: none"> <li>- Systemet meddeler at skibet ikke er tilgængeligt.</li> </ul> <p>2a. Brugeren er begrænset af moves:</p> <ul style="list-style-type: none"> <li>- Brugeren får lov til at vælge et andet skib.</li> </ul> <p>2b. Brugeren bevæger sig uden for kortet:</p> <ul style="list-style-type: none"> <li>- Skibet forsvinder</li> </ul> <p>3a. Der er ikke flere skib</p> <ul style="list-style-type: none"> <li>- Går videre reload/skyd for</li> </ul>

	hvert skib. Use case 9 + 10
Special Requirements	Spring, Java.
Frequency of Occurrence	Hver runde.
Miscellaneous	

#### **UC5: Kollision:**

Overview: To eller flere skibe på det samme koordinat(hex).

Actors: Bruger, System.

Precondition: to eller flere skibe prøver at gå ind i det samme hex ved movement/bevægelse.

Main success scenario:

1. Alle skibene i en kollision har med det samme 0 i hastighed og påfører: skrogskader =  $\frac{1}{3}$  af den aktuelle skrog værdi af de andre skibe. Aktuelle skrog = tidligere skrog - skrogskade.
2. Hvis det aktuelle skrog ender med at være lige med eller mindre end 0, synker skibet.
3. Hvis mere end to skibe er involveret, tage alle skibe skade fra hinanden.

#### **UC7: Vind og Hastighed:**

Overview: Vinden skal påvirke hvor hurtigt de forskellige skibe kan sejle.

Medvind giver max speed, sidevind giver max speed - 1 og modvind giver no speed.

Actors: System.

Preconditions: Server laver vind.

Main success scenario:

1. Vinden bliver random genereret efter et dice slag.
2. Skibene forholder sig til vindens retning.

### **UC8:** Ammunition/skibsmænd:

Overview: Brugeren er i kamp mod modstanderen

Actors: Brugere, System:

Precondition: UC10

Main success scenario:

- 1: Der kræves minimum 3 mænd for at kunne affyre skud hvis færre mænd vil skaden reduceres væsentlig. Samt har skibene mulighed for at kunne affyre ammunition på hver side af skibet.
- 2: Der findes 3 former af ammunition
  - a. de kan beskadige skroget, og i tilfælde af et kritisk slag kan sprænge hele skibet. kort til lang rækkevidde
  - b. Kæder med en mindre bold i begge ender. De kan beskadige riggen alvorligt. Gennemsnitlig afstand
  - c. Grape Shot/drue skud: Masser af løse stykker af jern. Det kan decimere antallet af sejlere. kort rækkevidde

### **UC9:** Reload (Skift våben):

Overview: Brugeren kan tage beslutning for at skifte våben

Actors: Brugere, System.

Precondition: UC4, UC8.

Main success scenario:

1. Brugeren bliver vist en menu med våben.

2. Brugeren vælger med musen et nyt våben.
3. Brugeren vælger et nyt skib og kan vælge reload/skyd igen.

#### **UC10:** Skade delegering:

Overview: Brugeren tager en beslutning om at skyde.

Actors: Bruger, System.

Precondition: UC4

Skibe kan sigte mod en bestemt sekskant, hvis et skib gennemløber den sekskant tjekkes den for den skade, hvis den gennemløber flere tage den skade for dem alle sammen(se bilag 2 og UC3).

Main success scenario:

1. Brugeren vælger et skib af gangen.
2. Brugeren vælger en hexagon han vil skyde på med musen.
3. Systemet godkender trækket. Se bilag 4
4. Brugeren starter forfra indtil alle skibe er 'brugt'
5. Brugeren trykker end turn.
6. Systemet udregner damage. se UC8/KR306

#### **UC11:** Slutspillet

Overview: Brugeren skal have udgjort skade på modstanderen

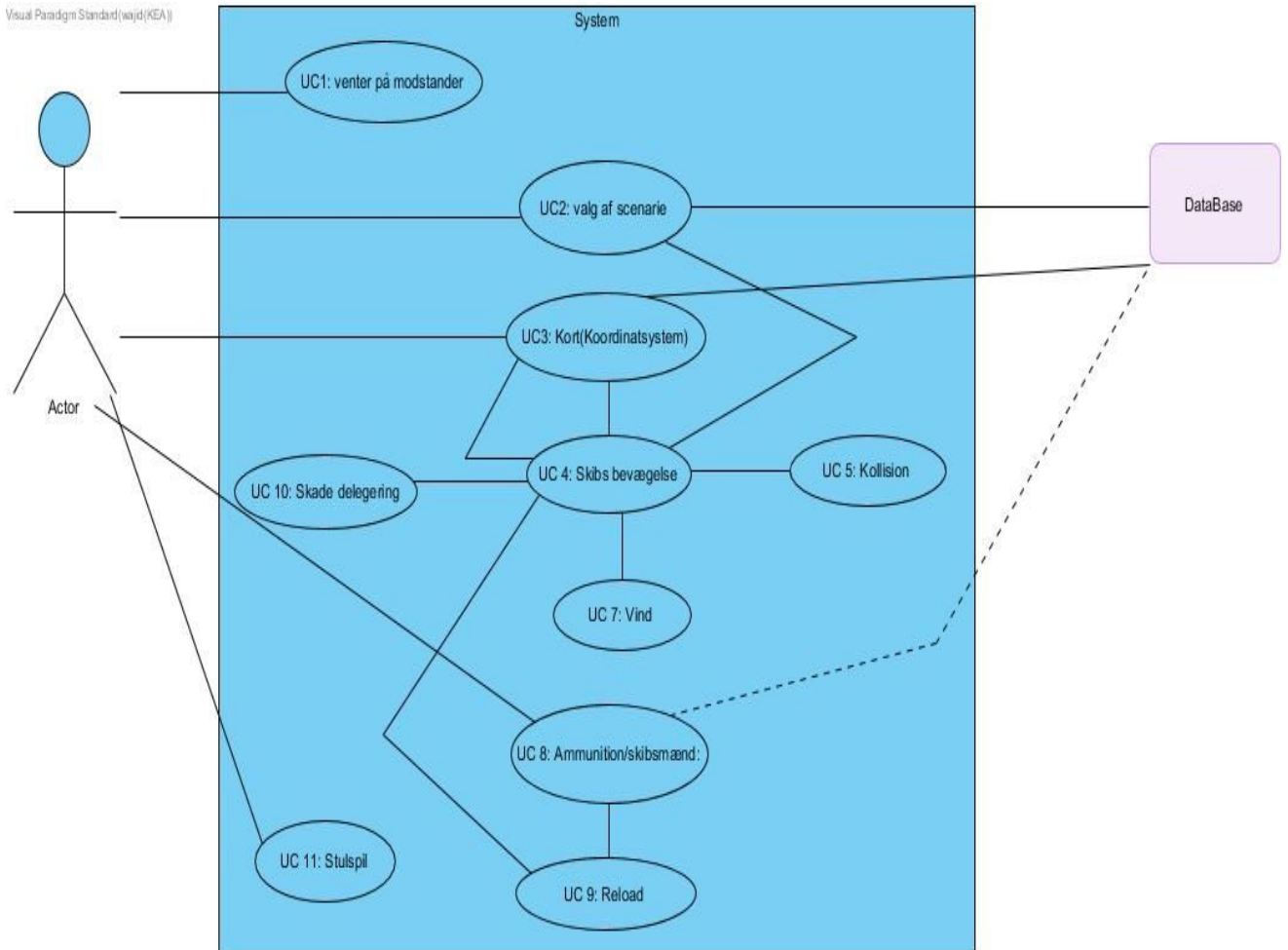
Actors: Bruger, System.

Main success Scenario:

1. Den sidste spiller med mindst et skib, der flyde.
2. Den sidste spiller med et manøvrerbart skib.
3. har lidt mindst skade efter et fast antal omgange.

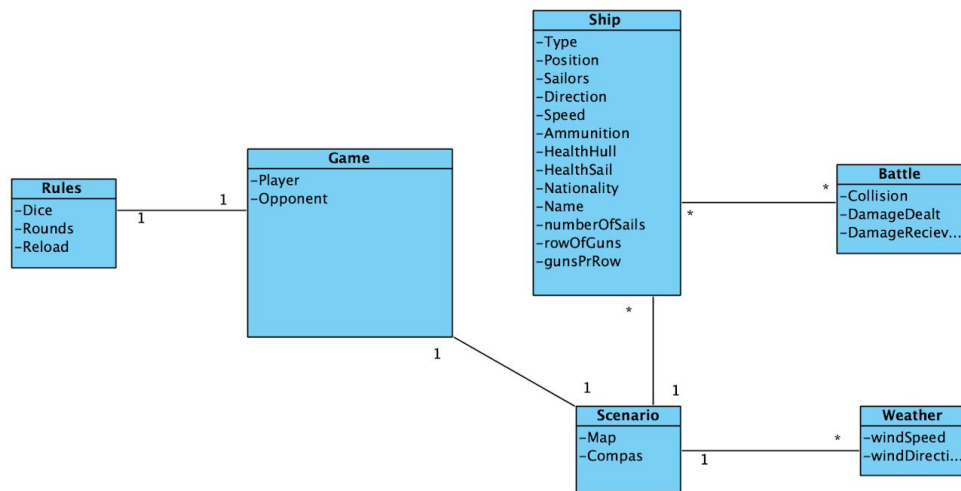
## Use case Diagram

Visual Paradigm Standard (vaidj(KEA))

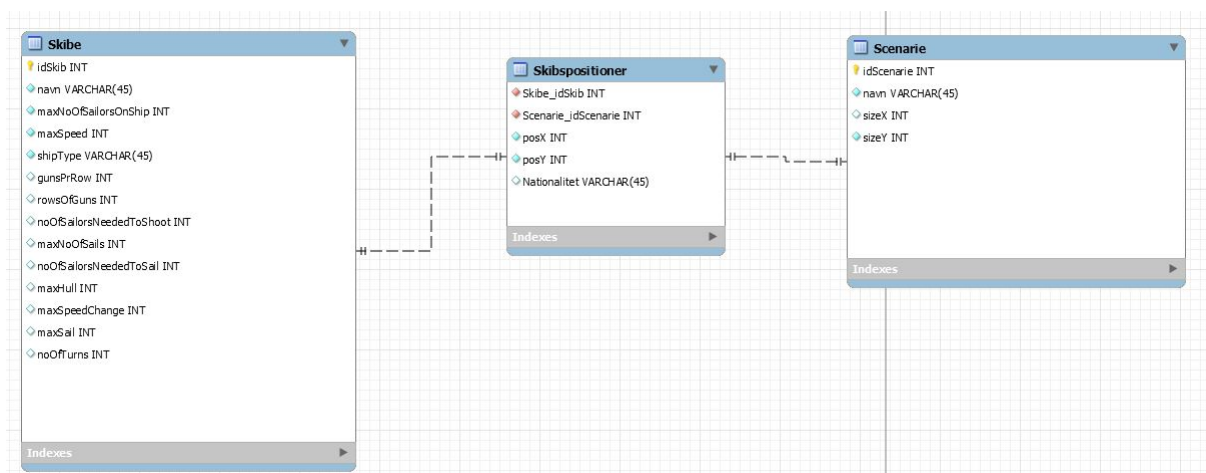


## Domæne model:

Domænemodellen viser, hvilke oplysninger vi har brug for i vores system og hvad relationerne mellem disse oplysninger er. Domænemodellen dækker samtlige use cases i vores system illustreret af konceptuelle klasser og deres indbyrdes relationer.



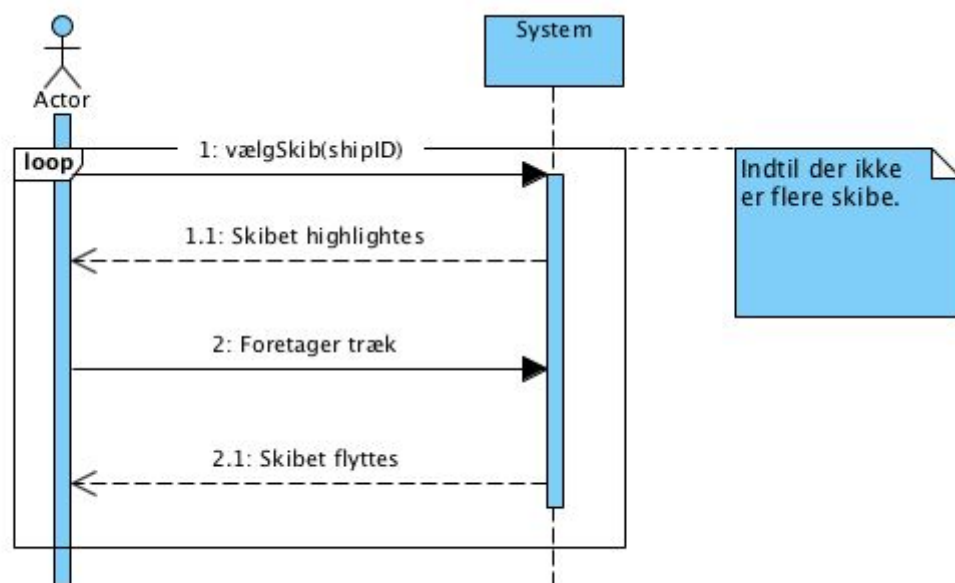
## Entity Relationship Diagram



## SSD UC4 - Skibs bevægelse:

System Sequence Diagram (SSD) er med til at vise et flow. Formålet er at illustrere de input og output, der er imellem brugere (aktør) og systemet for en bestemt begivenhed i et visuelt format. Man læser det oppefra, hvor man øverst har placeret aktøren og systemet. Under dem finder man lange stiplede linjer, som bliver kaldt livlinier. De linjer der strækker sig imellem dem, kaldes aktionslinjer, og de viser interaktionen mellem dem. Efter hver handling er blevet udført, bliver der vist svaret eller den næste handling.

Vores SSD tager udgangspunkt i vores use 4- Skibs bevægelse.

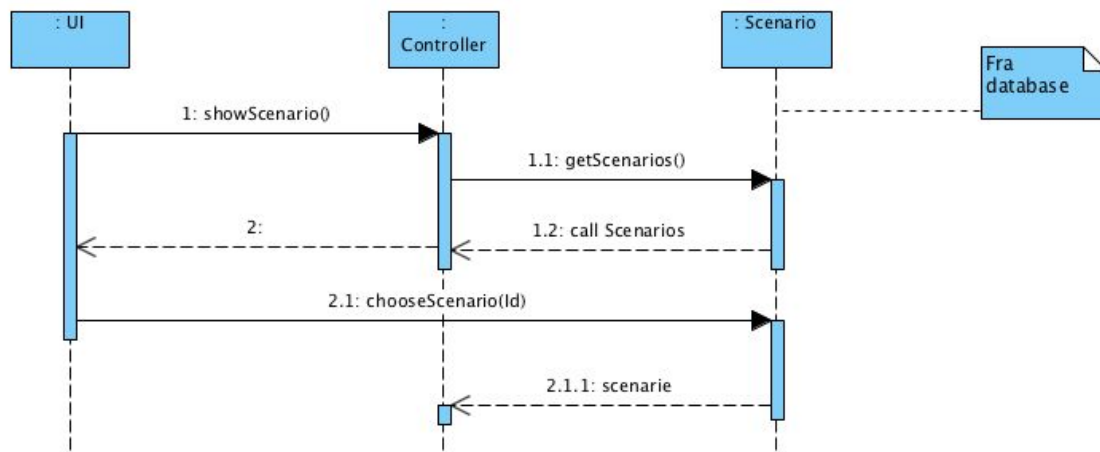




## SD UC2 - Valg af scenarie:

Et Sequence diagram (SD) illustrerer en sekvens af handlinger, der sker i et system. Den viser de objekter og klasser, der er involveret, når en given use case bliver kørt med hvilken rækkefølge metode kaldene foretages i og hvilken retur-typer, de har. De parallelle vertikale linjer (livliner), viser hvor længe objektet lever, og de vandrette pile, er meddelelser der bliver udvekslet mellem dem.

Man kan bruge SD til en slags sti for at illustrere, hvordan systemet skal respondere, og det gør det muligt at specificere et flow på en grafisk måde. Vi har taget udgangspunkt til vores UC2 for at demonstrere, hvordan det virker i vores system.



Klassediagrammet er en visuel præsentation af vores kode. Der er blandt andet dele som ikke fremgår i vores klassediagram, men kan dog findes i vores domænemodel. Vi har valgt ikke eksplicit at vise getters og setters i klassediagrammet, da de er standard i objektorienteret programmering med private attributter. Metoderne equals og toString er kun brugt i de klasser, hvor de er relevante i forhold til, hvor der håndteres oplysninger i vores system.



## Konklusion

I dette projekt har formålet været at oprette et sejlads spil, vi har i den forbindelse udarbejdet analyser og diagrammer. Udgangspunktet for projektet var, at lave et sejlads spil som skulle tage sted i historien. I spillet skulle man have mulighed for blandt andet at vælge nationalitet, skifte mellem flåder og skibe og vælge forskellige former af ammunition.

Missionen for spillet var på sigt at kunne skabe et sjovt og hyggeligt spil som både voksne, unge og børn ville have mulighed for at kunne spille. planen var selvfølgelig at skabe interesse for historien der engang fandt sted, samt stimulere den logiske tænkning i de taktiske baner. Denne mission udmøntede sig i en mere konkret business vision, hvor vi havde sat nogle mål for spillet. Visionen var blandt andet at spillet skulle kunne spilles af flere personer på tværs af kontinenter.

indledningsvis valgte vi i vores projektgruppe at afgrænse projektet. Afgrænsningen var baseret på gruppens vurdering af, hvor meget tid og hvor mange ressourcer (Kompetencer og arbejdstimer) vi havde til rådighed til opgaven. Ud fra denne vurdering valgte gruppen af fokusere på design, og spillets logik.

En mere detaljeret afgrænsning findes i afsnittet "Afgrænsning"

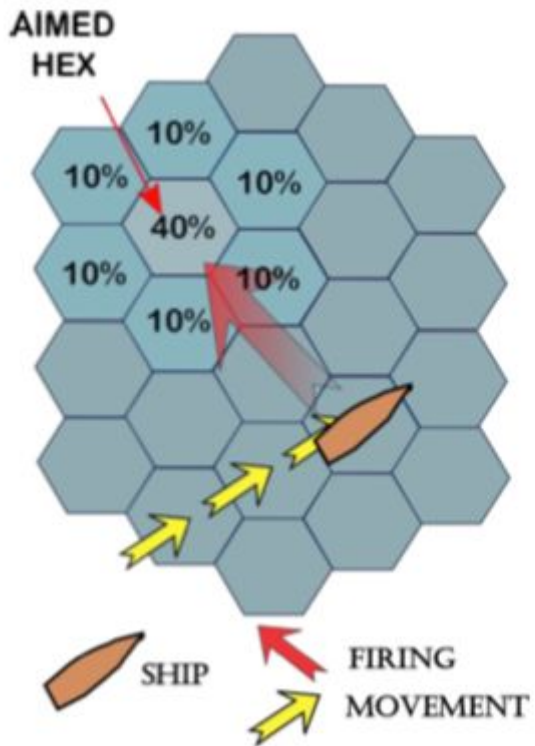
Bilag:

Bilag 1:

<b>Ships type</b>	<b># rows of Guns</b>	<b># of Guns pr. Row</b>	<b># of Sailors needed</b>	<b>MAX # of Sails</b>	<b># of Sailors needed</b>	
Brig	1	8	24	4	24	
Ship of the Line	2	16	48	10	60	
Man at War	3	28	84	24	144	

<b>Ships type</b>	<b># rows of Guns</b>	<b>MAX # of Sailors</b>	<b>MAX Hull</b>	<b>MAX Speed</b>	<b>MAX Speed Change</b>	<b>MAX Sail</b>	<b># of turns/ turn</b>
Brig	1	60	25	2	1	30	1
Ship of the Line	2	160	50	5	2	40	2
Man at War	3	340	140	4	1	80	1


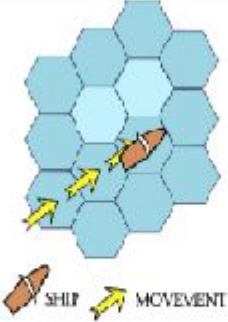








## Bilag 2:



## Bilag 3:

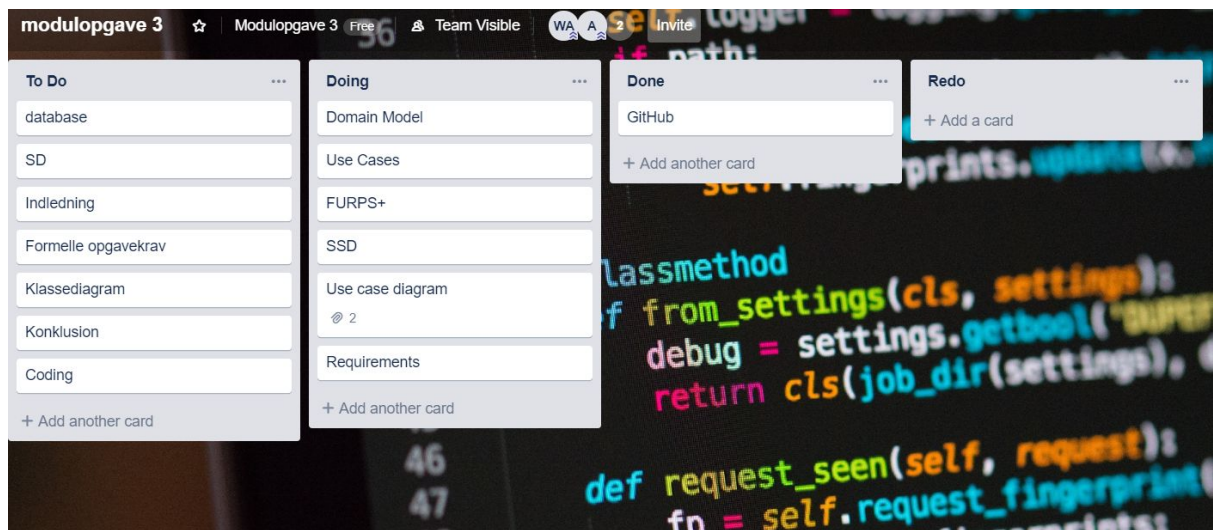
Hit Dice 1 – 20	Cannon Ball	Chain Shot	Grape Shot
1	Ship explodes	All rigging lost (ship breaks 1 /turn)	–50% of Sailors
2	Rudder lost (no future turning)	33% of rigging lost	–25% of Sailors
3	---	---	---
4 – 20	---	---	---

# Bilag 4:

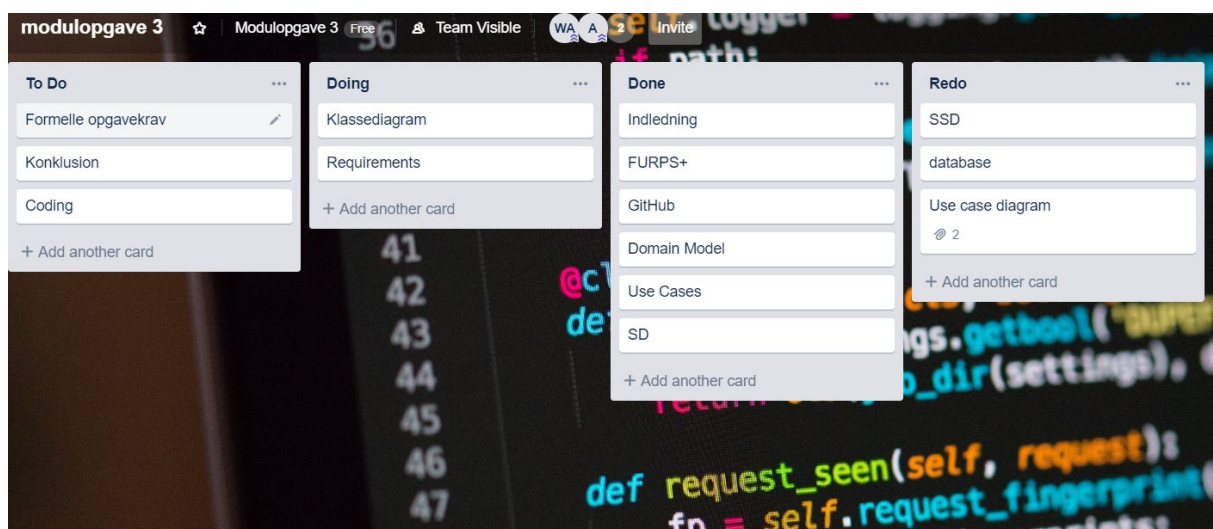
<div> <div>HEKES TO A M AT</div>  </div>	Cannon Ball	Chain Shot	Grape Shot
Brig			
Ship of the Line			
Man at War			

## Kanban/ Trello

### Bilag 5



### Bilag 6





## Bilag 7

