

Ships & Sails

2. Semester Modulopgave 3 Dat18C

Gruppe 5:

Thomas

Tariq

Marcus

Esben

Vejledere:

Jarl Tuxen

Asger B. Clausen

David H. Ema

Afleveringsdato:

26. April 2019

Indholdsfortegnelse

Indledning	2
SWOT-Analyse	2
Ordliste	4
Kravspecifikation	4
FURPS	4
Use Cases	6
Use Case Diagram	10
Domænemodel	11
Klassediagrammer	12
SSD	13
Konklusion	13

Indledning

Vores opgave er at kreere et spil, som simulerer et søslag mellem en anden spiller. Disse spillere skal tilslutte sig til spillet gennem egne computere. En af spillerne agerer som server og den anden som klient.

Derudover skal spillet have en database til rådighed (MySQL), som indeholder skibe, ammunition og mandskab. Spillet skal som helhed spilles gennem et web-baseret interface.

SWOT-Analyse

Interne situation	
Strengths	Weaknesses
<ul style="list-style-type: none">- Open source (adgang til spillet, så længe en server er aktiv)- Multiplayer: der er adgang for flere spillere af gangen- Simple regler/retningslinjer: nemt at forstå- Det er interaktivt	<ul style="list-style-type: none">- Der er for få valgmuligheder i forhold til angreb og bevægelse- Manglende tid til projektet- Svært at vedligeholde siden der ikke er lagt budget for projektet.
Eksterne situation	
Opportunities	Threats
<ul style="list-style-type: none">- Gode muligheder for udvidelsespakker- Der er ikke så bredt et marked for brætspils-inspirerede strategi-spil: det er nemt at komme ind på	<ul style="list-style-type: none">- Konceptet er blevet skabt før og der findes bedre udgaver

Mulige ændringer i SWOT:

Weaknesses:

- **'Der er for få valgmuligheder i forhold til angreb og bevægelse.'**
Ovenstående weakness kunne laves om til en strength ved, at gøre det muligt for fremtidige versioner af spillet at have flere former for ammunition og bevægelse. Dette vil som resultat gøre applikationen mere underholdende for spilleren, samtidigt vil det også give spillet mere dybde.
- **'Svært at vedligeholde siden der ikke er lagt budget for projektet.'**
En styrke kunne være at lægge et budget for spillet, hvilket som resultat ville give udviklerne større økonomisk frihed til, at arbejde på projektet.

En anden mulighed kunne være at tage 'crowdfunding' vejen. Dette vil også kunne garantere spillere ved lancering af spillet, eftersom kun dem som er interesseret vil støtte om projektet.
- **'Manglende tid til projektet.'**
En fast og grundig planlægning af arbejdsstrukturen ville kunne bidrage med et mere gennemført og kvalitetssikret opgave.

Threats:

- **Konceptet er blevet skabt før og der findes bedre udgaver.'**
For at kunne konkurrere mod de mange andre og/eller bedre udgaver, kræves det fra udviklerne at levere et spil, som ikke ligner de andre. Dette kan gøres ved at komme med et helt enestående funktion, som de andre spil ikke har. Udviklerne kan også lære af tidligere fejl begået af andre udviklere og derved forbedre eget projekt.

Ordliste

Ord:	Forklaring:
Hexagon	Sekskantet felt.
Crowdfunding	Økonomisk opbakning fra samfundet.
Scenarie	Opsætning for given kamp. Dette indebærer navnet på scenariet, størrelsen på kortet, hvilke nationaliteter de to spillere har og info om hvilke skibe hver spiller har, hvor de starter mm.

Kravspecifikation

Spillet skal implementeres efter følgende krav:

- 3 interfaces, der muliggør ombytning af individuelle moduler med andre grupper.
- TCP/IP protokollen til, at kommunikere med modspilleren.
- Gemme informationer om skibe, scenarier og alt de indebærer, i en database.
- Overholde reglerne for hvordan et spil afvikles, mht. bevægelse, angreb, vind, spilbræt, osv.

FURPS

Functionality:

- Spillet skal kunne spilles igennem et web-interface.
- En af spillernes computer skal fungere som server og den anden spillers computer som client.
- Spillet skal indeholde en relational database, som gemmer info om skibe, scenarier, mm.
- Et verdenskort bestående af hexagoner (sekskantede felter), hvorpå spillerne kan bevæge sig, angribe og kolliderer med andre skibe. Størrelsen på verdenskortet kan variere.
- Spillet er turbaseret.

Usability:

- Spilleren skal kunne manøvrere via pile der enten roterer skibet eller bevæge det fremad, og skifte mellem ammunitionstyper.
- Et felt på siden viser brugeren information om spillet, heriblandt status for ammunition, mandskab, mm.
- Der er en knap til at slutte en spillers tur.

Reliability:

- Hvis spillet lukkes med en fejl, mistes den aktuelle spiltilstand og spillet må startes forfra.
- Spillet er ikke udgivet som en cloud app, og skal derfor startes lokalt fra to maskiner, der har det installeret.

Performance:

- (Spillet skal performance-testes før udgivelse)

Supportability:

- Spillet kræver Java, Spring og Thymeleaf for både server og klient.
- Spillet kræver en netværksforbindelse.
- Spillet virker i macOS, Windows og Linux.
- Applikationen virker i alle nuværende versioner af de større browsere.

Afgrænsning

I forbindelse med størrelsen af opgaven har vi valgt at begrænse, hvilke diagrammer vi har lavet. Da opgaven (kode-mæssigt) er individuelt, har vi derfor valgt ikke at lave et SD, da størrelsen på selve opgaven har gjort det svært at overskue sammenhængen mellem de forskellige dele af programmet.

Samtidig har vi ikke afgrænset i koden da tanken var, at vi skulle spille sammen med andre der selv kører deres egen version af spillet samtidig, og det derfor ikke vil fungere, hvis vi spiller efter andre regler end de andre grupper.

Use Cases

Use cases markeret med rød er ikke blevet implementeret pga. tidsmangel.

UC1: Starting a game (casual)
UC2: Exiting to Game Menu (brief)
UC3: Selecting a Ship (brief)
UC4: Moving a Ship (fully dressed)
UC5: Attacking (casual)
UC6: Viewing Game Scoreboard (brief)
UC7: Finish Game (brief)
UC8: Ending the Turn (brief)
UC9: Toggle Sound On/Off (brief)
UC10: Collision (brief)
UC11: Taunt (brief)
UC12: Replay (brief)
UC13: Change ammunition

Title	UC1: Starting a Game (Casual)
Main Success Scenario	Spilleren starter programmet og vælger et scenarie.
Extensions	Med manglende modspiller har man én af to valgmuligheder: 1. Vente på klient 2. Oprette sig som klient til en server
Special Requirements	1. Internetforbindelse for både Server og Klient 2. JAVA 12 JRE 3. Terminal 4. MySQL server

Title	UC2: Exiting to Game Menu
Main Success Scenario	Det igangværende spil afsluttes og spilleren returneres til menuen.

Title	UC3: Selecting a Ship
Main Success Scenario	Spilleren trykker på et skib, som vedkommende ønsker at kontrollere. Pågældende skib vil blive markeret, og der kan afgives ordrer.

Title	UC4: Moving a Ship (Fully Dressed)
Scope	<i>Ships & Sails</i> spilapplikation
Primary Actor	Spiller
Stakeholders and Interests	<p>Brugeren: Ønsker den bedste spiloplevelse.</p> <p>Udviklere: Interesseret i at spillet fungerer optimalt. Derudover er udviklerne også interesseret i respons mht. forbedringer af programmet.</p>
Precondition	UC3: Selecting a Ship
Success Guarantees	Skibet er på det ønskede placering i banen og de ønskede handlinger er korrekt fuldført fra programmets side.
Main Success Scenario	<ol style="list-style-type: none"> 1. Spilleren vælger under sin tur at bevæge sit skib til en ønsket lokation. <ol style="list-style-type: none"> 1.2. Spilleren flytter sit skib til et felt, hvori der allerede er et skib. Spillerens skib kolliderer her med det andet. (Dette kan også være spillerens eget skib). <ol style="list-style-type: none"> 1.2.1. Efter kollision vil det kolliderende skib blive smidt tilbage på det tidligere felt.
Extensions	<ol style="list-style-type: none"> 1. Collision damage. Movement stoppes, og der bliver beregnet skade til de involverede skibe. 2. Out of bounds. Skibet forsvinder hvis det bevæger sig uden for banen.

Special requirements	JRE
-----------------------------	-----

Title	UC5: Attacking and Reloading(Casual)
Precondition	UC3: Selecting a Ship
Main Success Scenario	<ol style="list-style-type: none"> 1. Spilleren indgår en kamp mod fjendens skib. 1.2. Spillerens skib affyrer et skud og kan, derefter enten vælge at skifte ammunition eller fortsætte med samme ammunition. 1.2.1. Hvis skibet har affyret et skud, og ønsker at bruge det samme skud igen, så skal skibet vente én runde med at kunne affyre igen. 1.2.2 . Hvis skibet skifter ammunition efter at have affyret et skud, så skal skibet vente 2 runder med at kunne affyre igen.
Extensions	

Title	UC6: Viewing Game Scoreboard
Main Success Scenario	<p>Efter spillet printes resultaterne. Herunder:</p> <ul style="list-style-type: none"> - Skade gjort - Antal skibe tilbage - Antal skud brugt - Antal runder spillet - Antal overlevende sailors

Title	UC7: Finish Game
Main Success Scenario	<p>Spillet kommer til en ende som resultat af følgende:</p> <ol style="list-style-type: none"> 1. Spilleren har besejret sin modstander ved at sænke/immobilisere de fjendtlige skibe.

	2. Spilleren og/eller modstanderen vælger manuelt at afslutte spillet.
--	------------------------------------------------------------------------

Title	UC8: Ending the Turn
Main Success Scenario	Spilleren afslutter sin tur efter at have foretaget de ønskede handlinger.

Title	UC9: Toggle Sound On/Off
Main Success Scenario	Spilleren har mulighed for at tænde/slukke for lydeffekter og/eller musikken i spillet.

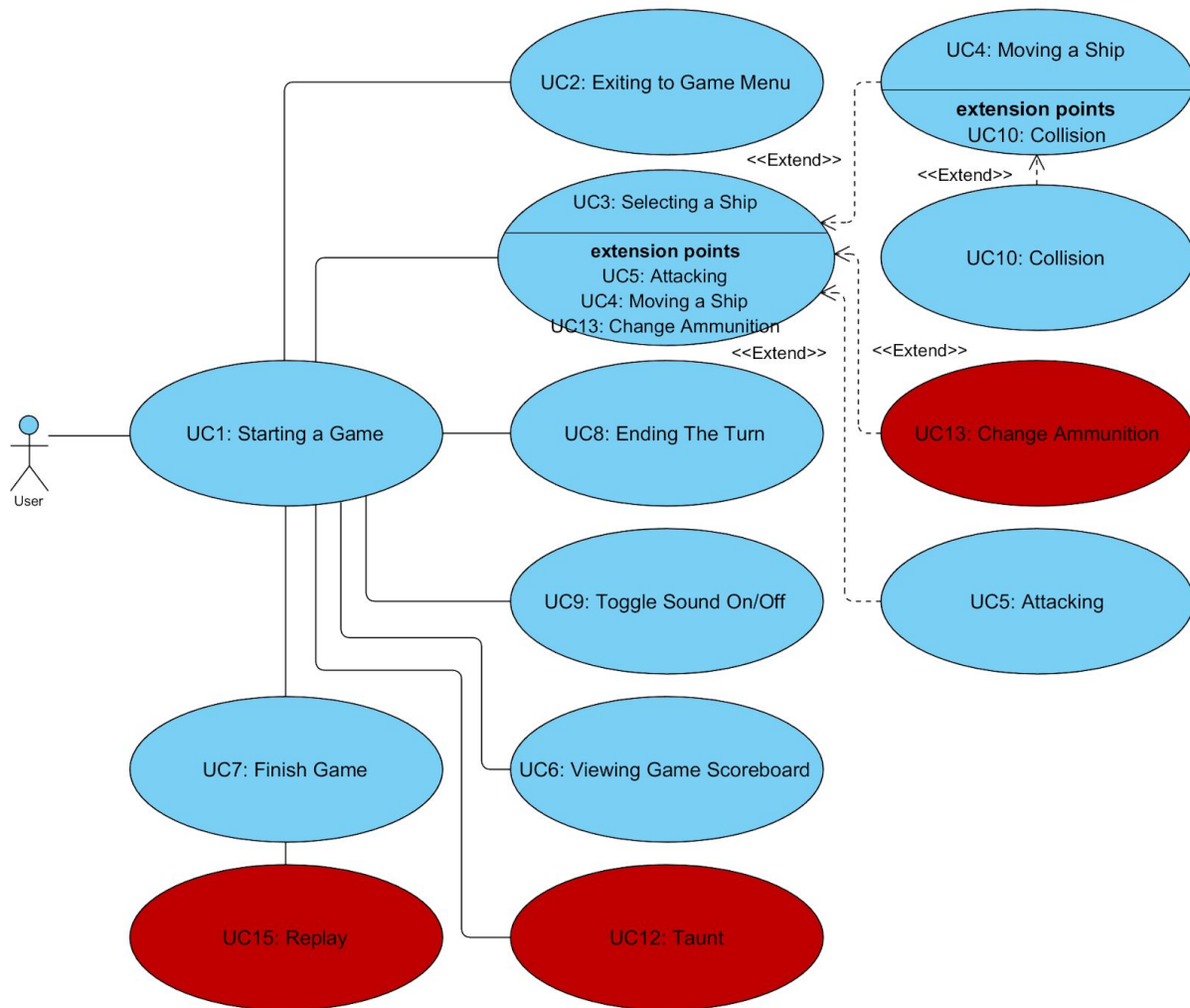
Title	UC10: Collision
Main Success Scenario	Et skib forsøges bevæget ind på en hexagon der allerede indeholder et andet skib. Begge skibe modtager skade.

Title	UC11: Replay
Main Success Scenario	Alle træk for et helt spil gemmes i databasen og kan sidenhen loades fra menuen og afspilles som et replay.

Title	UC12: Taunt
Main Success Scenario	Et skib kan taunte og dermed tvinge nærliggende skibe til at skyde imod det.

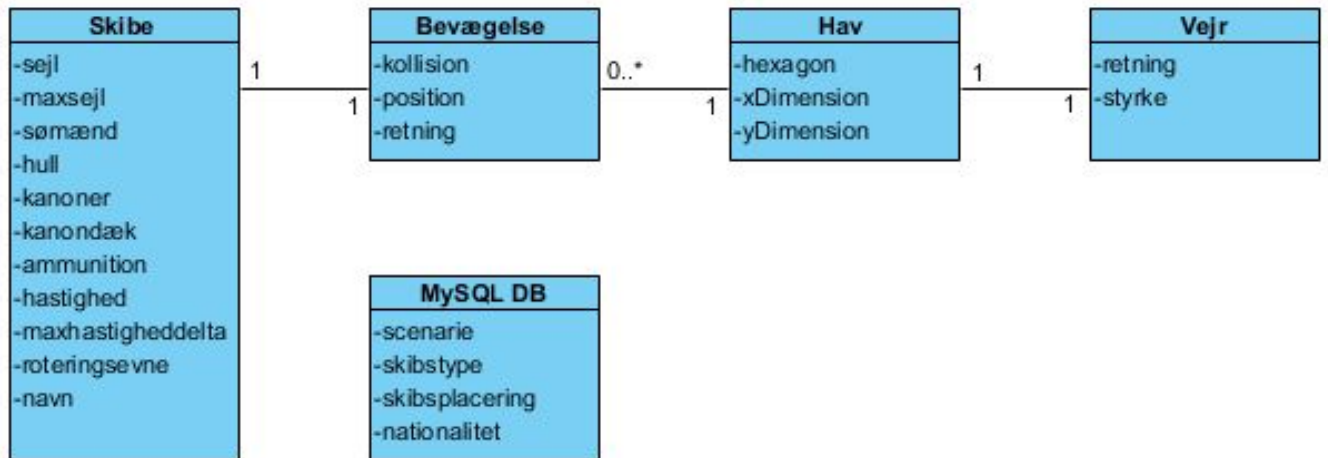
Title	UC13: Change Ammunition
Main Success Scenario	Spilleren skifter ammunition - efter at have affyret det forrige - til et af de tre typer for ammunition. Spilleren skal vente 2 runder.

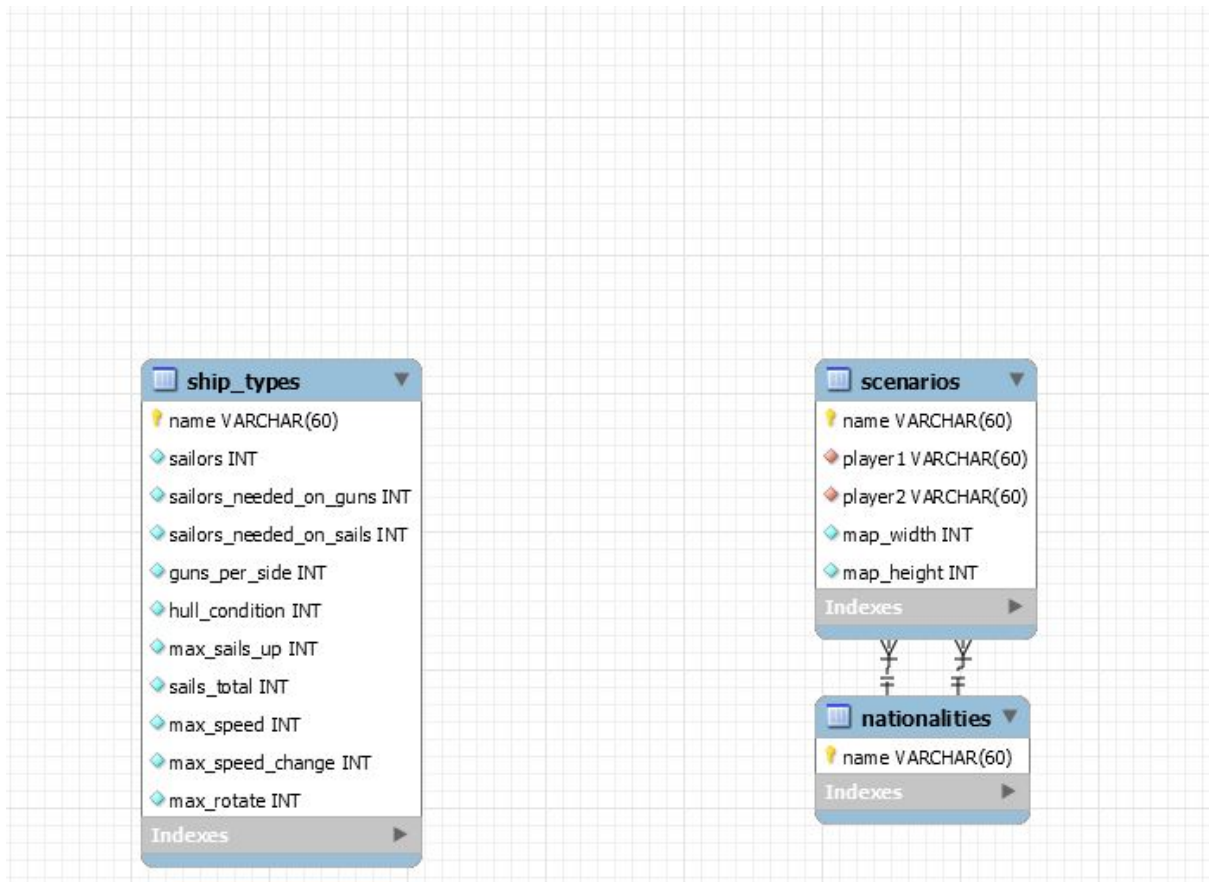
Use Case Diagram



Ovenstående usecase-diagram repræsenterer sammenhængen mellem projektets usecases. UC12, UC13 og UC15 bliver ikke anvendt og er derfor markeret med rødt. UC4 og UC5 er udvidelser af UC3, hvilket vises

Domænemodel

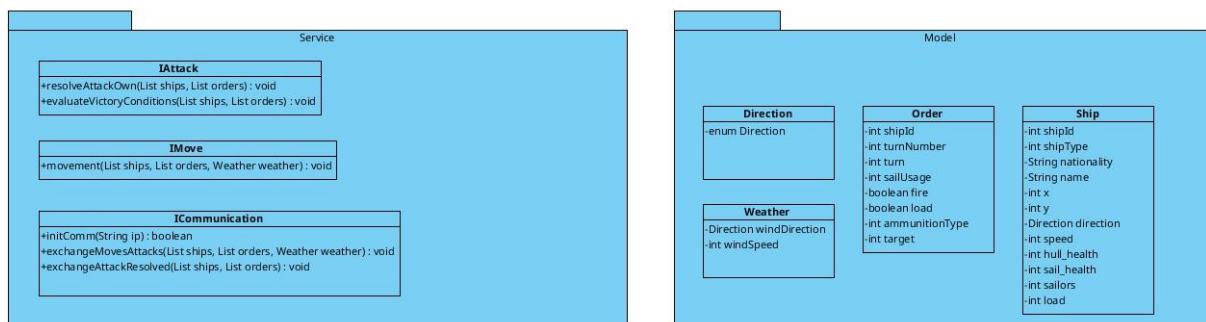




Klassediagrammer

En tidlig udgave af diagrammet, baseret på den udleverede kode.

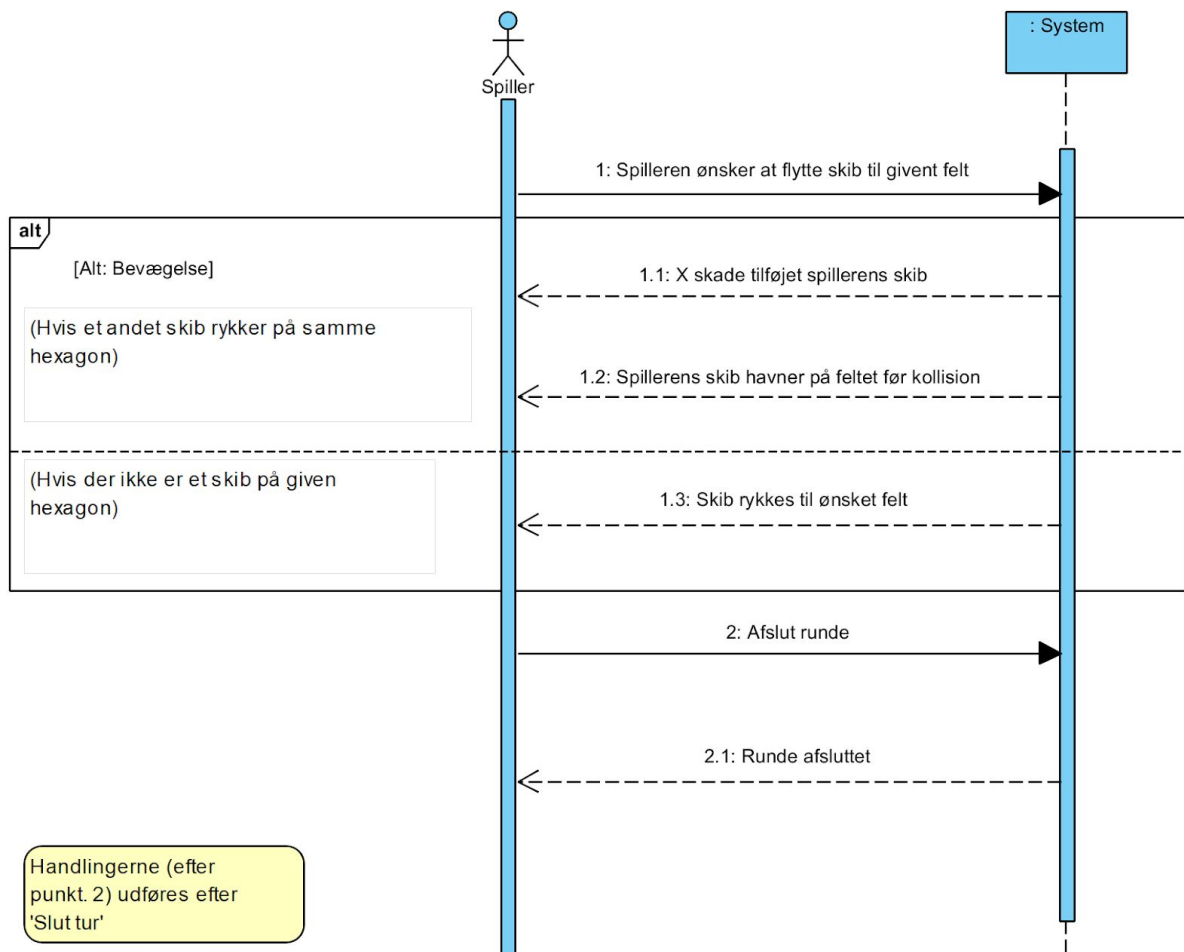
Udelader getters, setters og constructors



Genereret diagram baseret på seneste version af koden.



SSD



I ovenstående System Sequence Diagram er der taget udgangspunkt i UC4, som beskriver bevægelsen af et givent skib. I diagrammet ønsker spilleren flytte sit skib til et givent felt på banen. Undervejs kan spilleren støde ind i et andet skib, hvori vedkommendes skib tager skade og bliver smidt på feltet foran det oprindelige.

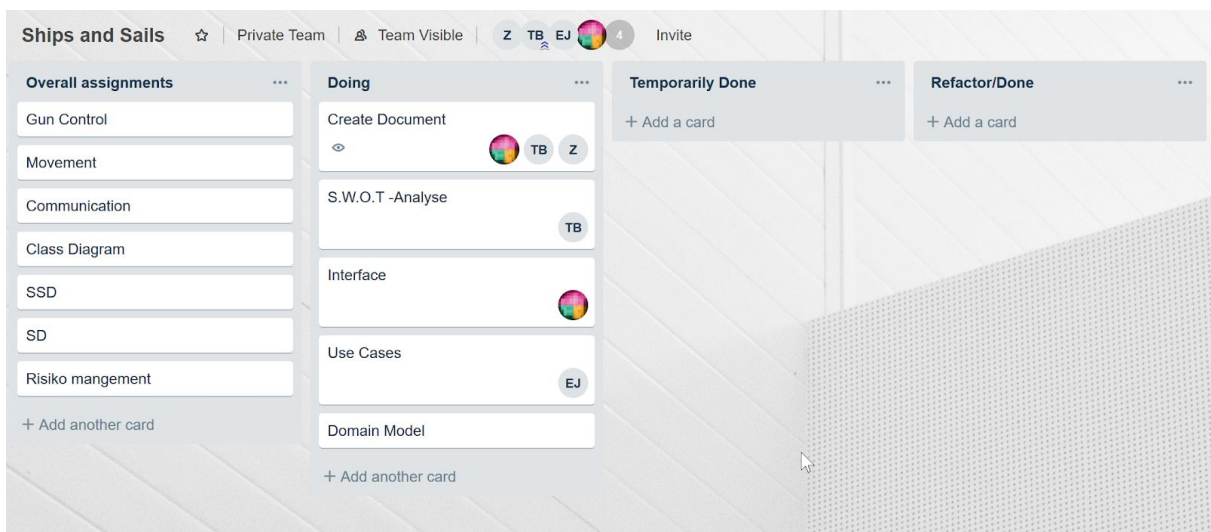
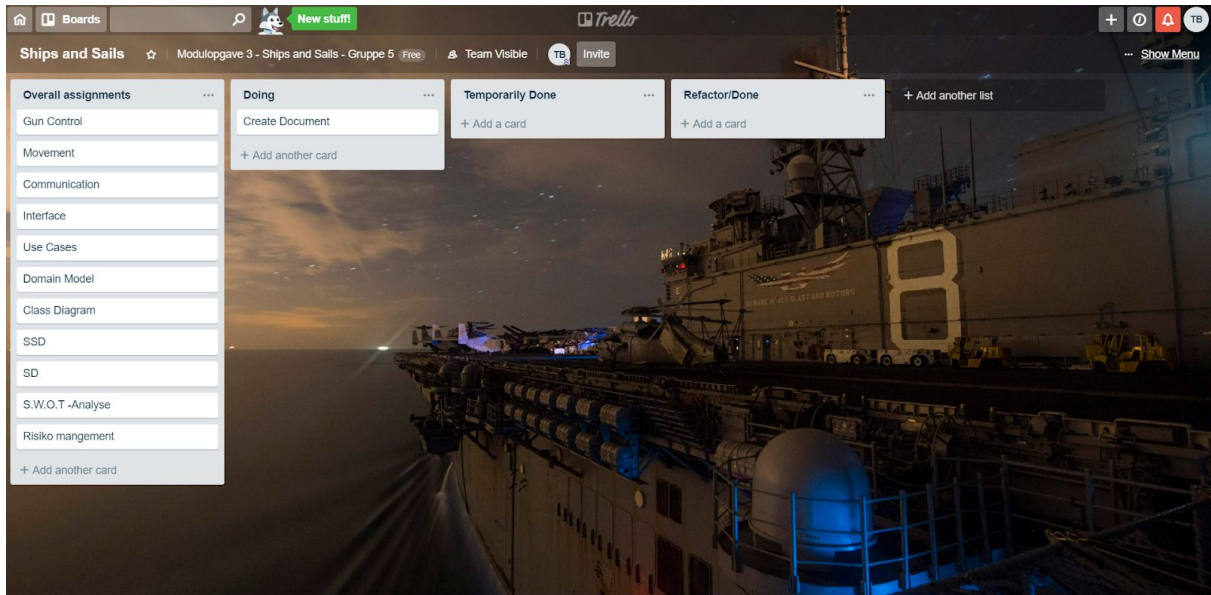
Alle brugers 'requests' træder først i kraft efter vedkommende har sluttet sin tur.

Konklusion

Omfanget af opgaven gjorde det svært at færdiggøre produktet, og det er derved blevet ufuldstændigt. En løsning en anden gang kunne være at afgrænse opgaven i fællesskab inden påbegyndelse af projektet, ift. tiden vi har til projektet. Det kræver dog en forudgående erfaring og viden til realistisk at kunne estimere, hvor meget vi kan nå, som vi endnu blot er i gang med at udvikle.

KANBAN

Uge 1:



Uge 2:

