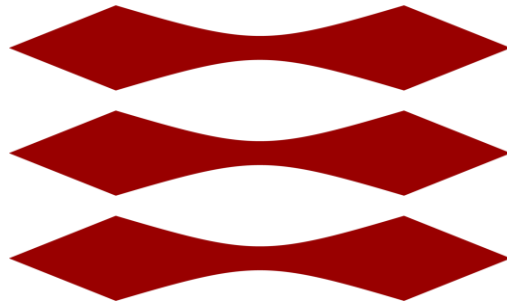


DTU



Indledende programmering (02312)
1. Semester (Gruppe - 29)



Fathi Benigen Muhammad
s215770



Ali Majdl
s216252

MATADOR



Resume:

Vi har i forvejen lidt erfaring med at skabe nogle forbedrede versioner af forskellige spille. Den her gang bliver vi bedt om at skabe en forbedret version af Matador Spillet. Alle Monopoly elskere ved hvad spillet "Matador" går ud på. At købe, udleje eller sælge ejendomme. Når man bliver den rigeste spiller, er man dermed spillets eneste matador. Denne her gang udvikler vi spillet, så den indeholder flere korte, flere regler osv....

Analyse og Design:

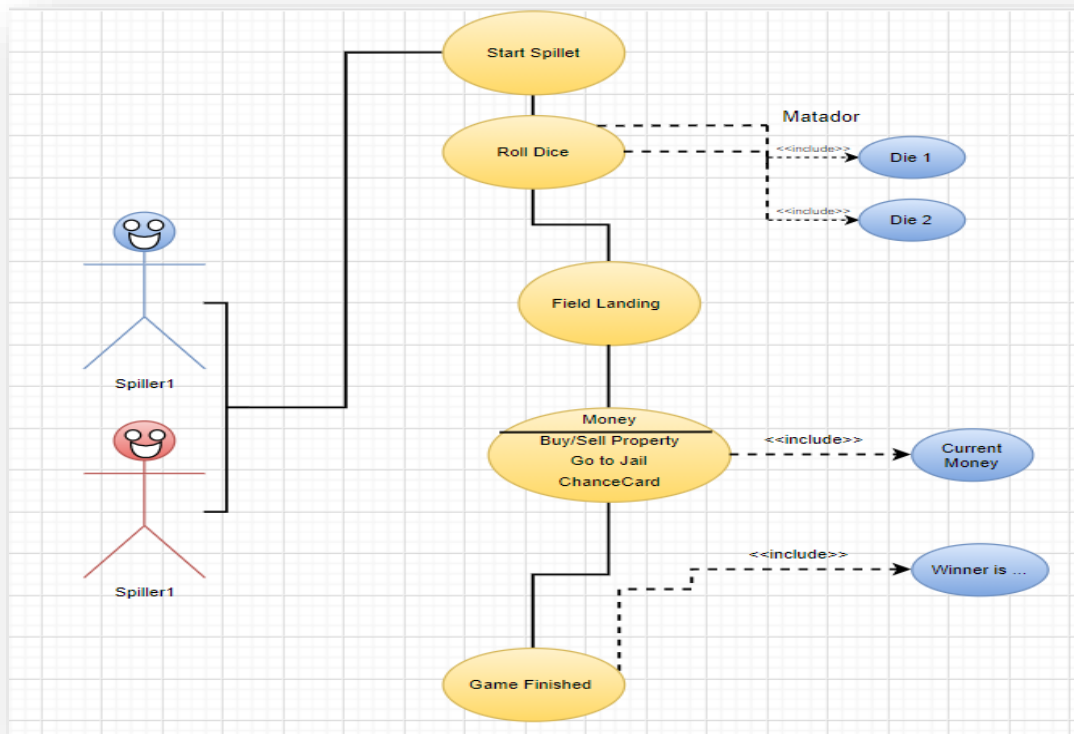
Der skal udarbejdes følgende artefakter:

• Kravliste

ID	Funktionelle Krav
Must have	
M01	Spillet skal kunne lade brugeren vælge antal spillere, og næve dem.
M02	Spillet skal kunne lade spillere starte spillet.
M03	Spillet skal kunne lade spillerne slå terninger.
M04	Spillers brik (bil), skal rykkes ifølge terningekast (Summen af terningers værdi).
M05	Spillet skal lade spilleren kunne købe/sælge ejendomme.
M06	Spilleren skal betale 10% (4000 kr.), hvis han lander på brættet "indkomstskat".
M07	Spillet skal kunne annoncere matadoren (Den rigeste spiller) i enden af spillet.

ID	Ikke-funktionelle Krav
IF01	Spillet skal være versioneret (GUI).
IF02	Spillet skal have automatisk deploy.
IF03	Spillet skal være en Single Page Application.
IF04	Spillet skal have tekstsporing.

- Use case diagram (Spil Matador)



- Her er et brugsscenarie af et stykke af det software vi kommer til at kode, som design til beskrivelse af samspillet relation mellem en aktør og et system.

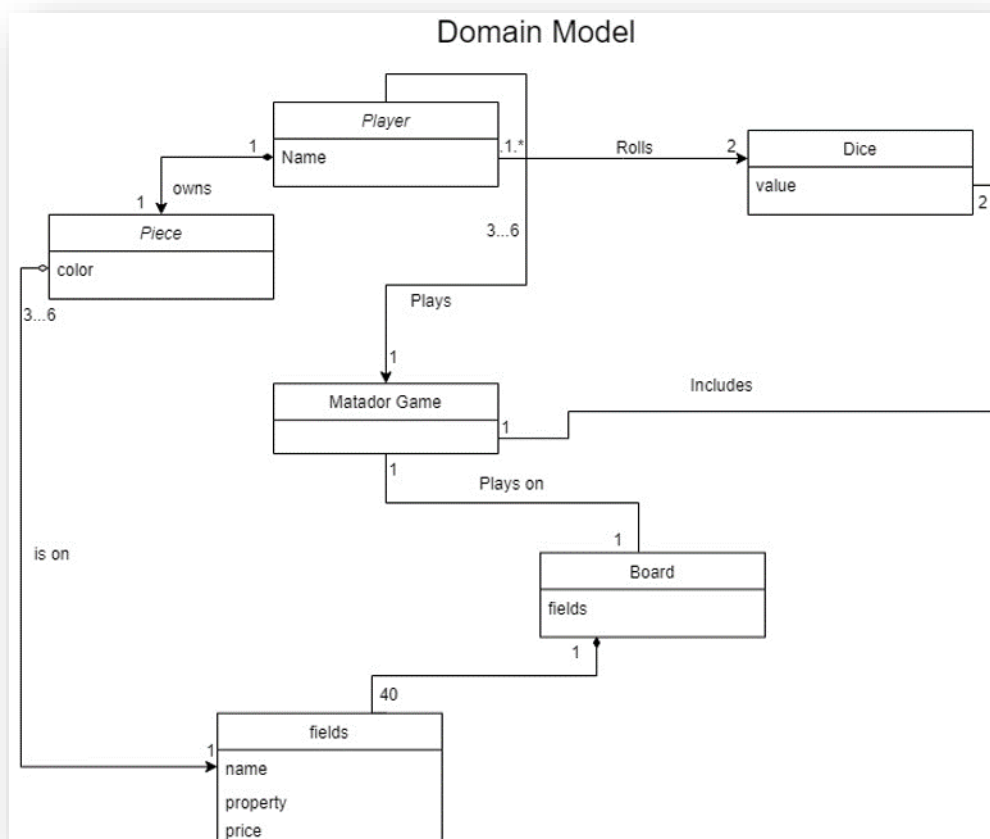
Use case Description

Use case Section	Comment
Use case name	Matador
Scope	Slå to terninger samtidigt, landing på en felt vil give , trække point eller give 0 point og den første spiller der bliver den rigeste , vinder spillet
Level	Nå at blive den rigeste for at vinde spillet
Primary Actor	Spiller 1
Secondary Actor	Spiller 2
Stakeholders and Interests	Kunderne og udviklerne.
Preconditions	Enhver spiller har 30000 dkk i starten.
Postconditions	Spillet slutter, når man bliver den rigeste
Main Success Scenario	1-Brugeren starter spillet , indtaster antal af Spillere med navne 2-Terninger rulles og brikkerne rykkes ifølge terningkast. 3-der købes og sælges egendomme mellem spillerne. 4- Den rigeste Spiller er Matador(vinder Spillet).
Extensions	Spillet Fejler , når antal af Spillere er større end 6 1- Brugeren genstarter systemet . 2- Spillet nulstilles.
Special Requirements	En grafisk brugerflade (GUI) med tekst sporing (registrering).

- **Sporbarhed**

Use Case / Krav	M01	M02	M03	M04	M05	M06	M07
UC01	X	x					
UC02			X	X			
UC03						X	
UC04					X	X	X
UC05							X

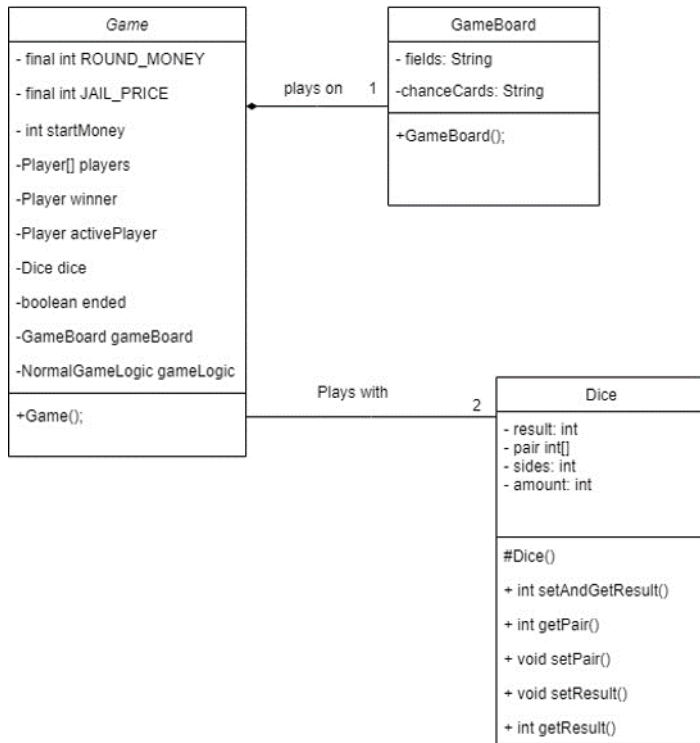
- **Domænemodel**



Vores Domain Model Her viser forløbet af spillet og sammenhængen mellem de forskellige handlinger i selve spillet.

- Klasse Diagram

Design class Diagram



- Her viser vi et designklassediagram, bestående af klasser med stort samspil imellem og fællesskab om at udføre bestemte slags opgaver, og som eksempel tog vi samspillet mellem vores spil, brættet og vores terninger.

- Test

Positiv test:

Krav	Status
M01	Bestået
M02	Bestået
M03	Bestået
M04	Bestået
M05	Bestået
M06	Bestået
M07	Bestået

Github repo

<https://github.com/fathionsons/DTU-gruppe29FathiAli.git>

- **Konklusion**

Formålet med projektet er at udvikle en fuld version af spillet "Matador".

Vi har startet med at designe spillet, så vi kan vise et tydeligt samspil mellem en aktør og spillet, og uddybet designet ved at skrive en beskrivelse af vores use-case, som en fortælling til kunden for en nemmere forståelse af spillet, og hvordan vi har udviklet det. Vi har også skrevet en liste af de vigtigste krav nummererede, og har lavet en tabel til at vise sammenhængen mellem vores design og krav. Derudover har vi også kodet vores spil færdigt. Vi har sørget for at alle de krav som vi prioriteret, stemmer med vores design. Vi synes altid, at vi kunne klare det meget bedre med mere design, implementering osv, men sådan er det bare når der gives et stort projekt til en gruppe af to.

Vi kunne desværre ikke udføre vores Junitest 5 process, og vi prøvede meget på det, men kunne ikke få det til at virke, da det allerede var hårdt at hænge vores koder og klasser sammen, at det blev for kompleks at lave koderne om, så har vi besluttet at ikke ændre noget som helst fra vores koder til lavning af spillet. Vi er ret tilfredse med hvad vi har udført og har fået det til at virke på den bedste måde.