

02313 Udviklingsmetoder til IT-systemer

## CDIO 1

Mads Glæsner: s215704

Mette L.B. Andersen: s172840

Josef baraki: s205112

Albert M. Halkjær s215718

Hans Andersen: s215705



**Danmarks  
Tekniske  
Universitet**

## Indholdsfortegnelse

<b>1. Resumé .....</b>	<b>2</b>
<b>2. Timeregnskab .....</b>	<b>2</b>
<b>3. Indledning .....</b>	<b>2</b>
<b>4. Projekt-planlægning .....</b>	<b>2</b>
4.1 Iterationsplan .....	2
Fredag 24/9 .....	2
Mandag og tirsdag 27-28/9 .....	2
Onsdag og Torsdag 29-30/9 .....	3
Fredag 1/10 .....	3
<b>5. Krav .....</b>	<b>3</b>
5.1 Use case .....	3
Primary actors .....	<i>Error! Bookmark not defined.</i>
Preconditions .....	3
Main success scenario .....	3
Alternative scenarios .....	3
Postconditions .....	4
Special requirements .....	4
5.2 Non-funktionelle krav .....	4
<b>6. Analyse .....</b>	<b>4</b>
<b>7. Design .....</b>	<b>5</b>
<b>8. Implementering .....</b>	<b>6</b>
8.1 GUI .....	6
<b>9. Test .....</b>	<b>7</b>
9.1 Terninger .....	7
9.2 Responstid .....	7
<b>10. Konklusion .....</b>	<b>7</b>

## 1. Resumé

IOOuterActive har fået til opgave at skabe et terningspil. Spillet **skal** kunne køre på computerne i Databaren. Derudover har kunden en række ønsker til spillets regler og rammer. Disse skal også indbefattes i programmet. For at løse opgaven har vi valgt at gå iterativt til værks og vi har brugt artefakter fra RUP.

## 2. Timeregnskab

27/9 3 timer

28/9 3 timer

29/9 1 time

30/9 2 timer

01/10 1 time

## 3. Indledning

Vi har fået til opgave at udvikle et terningspil mellem to personer. Det er vigtigt, at der bliver lagt fokus på kundens krav og hvilke regler de gerne vil have implementeret. I denne rapport fremlægger vi vores process, analyse og resultat.

## 4. Projekt-planlægning

Projektet er planlagt i iterativ stil trods dets beskedne omfang

### 4.1 Iterationsplan

Iterationsplanen er skrevet således at planen for en iteration er skrevet under afholdelsen af en tidligere iteration.

#### Fredag 24/9

Inception med fokus på at udvikle en grundlæggende forståelse for vores domæne, samt opstart på forskellige artefakter som use case, grundet programmets simplicitet vil det kun indgå en use case i udviklingen, den skal laves som en brief use case.

#### Mandag og tirsdag 27-28/9

Elaboration 1. Forslag til fully dressed use case ønske med inddragelse af projektleders krav, der opnås gennem uddybende spørgsmål. I iterativ stil ønsker vi at bruge disse dage til at udføre en hel elaboration iteration og her er målet at få styr på implementeringen af den største risikofaktor nemlig implementeringen af det simpleste spil der kan spilles. Til slut af iterationen ønsker vi at udforske nogle af de andre regler der yderligere kan implementeres så de kan blive målet i næste iteration.

Onsdag og Torsdag 29-30/9

Elaboration 2. Vi ønsker at vi får en gui på vores program. Samtidig bliver det med stor sandsynlighed nødvendigt at tilpasse det arbejde vi allerede har lavet en smule da vi opdager nye sammenhænge i opgaven.

Fredag 1/10

Oprydning og sidste tilpasning af projektet.

## 5. Krav

De funktionelle krav kan bedst ses i use casen. De non-funktionelle krav er så sparsomme at vi ikke har valgt at lave yderligere artefakter, såsom Glossary osv., men i stedet har vi bare skrevet dem kortfattet i afsnittet Non-funktionelle krav. For at sikre en fælles referenceramme for de krav der arbejdes med i projektet er der løbende sket forventningsafstemning af de udarbejdede kravspecifikationer med projektleder der også agerer kunden repræsentant

### 5.1 Use case

#### **Spil spillet**

##### **Primary actors**

- Spillere

##### **Preconditions**

- Der skal være 2 spillere
- Man har fundet ud af hvem der er spiller 1 og 2

##### **Main success scenario**

1. Spillerne slår hver med to terninger
2. Summen af øjnene på terningerne gemmes
3. Øjnene på terningerne lægges til total score
4. Step 1-3 gentages indtil en af spillerne opnår 40 point, og herefter slår 2 ens. Herefter er spillet vundet

##### **Alternative scenarios**

1a. Spiller slår 2 ens

1. Hvis en spiller slår to seksere
  1. Hvis spilleren har 40 eller flere point har spilleren vundet
  2. Hvis der i sidste runde, eller i samme runde med ekstra tur, er slået to sekser så vinder man spillet.
2. Hvis spilleren slår to 1'ere
  1. Spiller mister alle sine point
3. Summen af øjnene på terningerne gemmes
  1. Medmindre man er nået til 40 point
4. Terningers værdi lægges til spillers point

## 5. Spiller slår igen

### Postconditions

- Vinder findes

### Special requirements

- Kastet må ikke tage mere end 333 millisekunder

Under udviklingen af programmet har vi gjort en række antagelser for at skabe et fyldestgørende program. Herunder bl.a at spillerne skal følge programmets process flow. Derudover antager vi at brugeren kender til reglerne. Vi har snakket med vores projektleder, som har været i kontakt med kunden, for at bekræfte nogle af vores spørgsmål angående spillet. Vi var blandt andet i tvivl om nogle af reglerne, herunder hvordan du vinder spillet. Vi fik bekræftet at man **ikke** vinder spillet, hvis man slår to 1'ere efter at have opnået de nødvendige 40 point, men blot vender tilbage til 0 point. Vi fik også bekræftet, at man på forhånd, spillerne imellem, finder ud af, hvem der er spiller 1 og spiller 2. Ligeledes har vi antaget at spillet forløber uden uforudsete exceptions da de ikke håndteres i kode.

### 5.2 Non-funktionelle krav

Programmet skal kunne køre på computerne i Databaren, som benytter styresystemet Windows.

## 6. Analyse

Vi har udarbejdet en domain model som main artifact. Ud fra førnævnte gik vi i dialog med projektlederen. Samtalen gav gruppen en bedre forståelse af projektet og det tiltænkte resultat. Undervejs har vi udforsket domænet og gradvist udvidet vores model til det den er blevet til nu

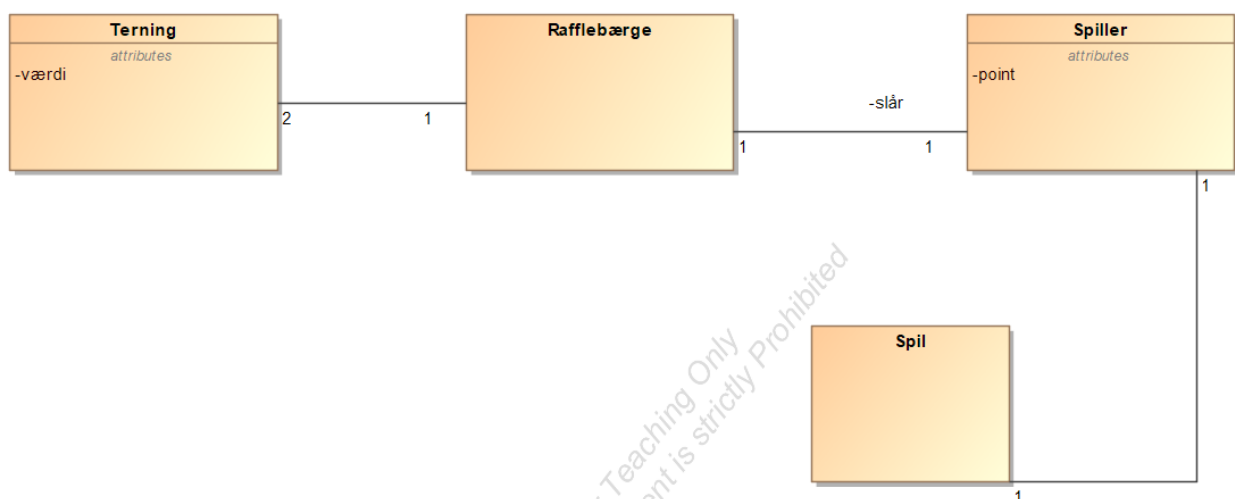


Fig 1.Domænenemodel

## 7. Design

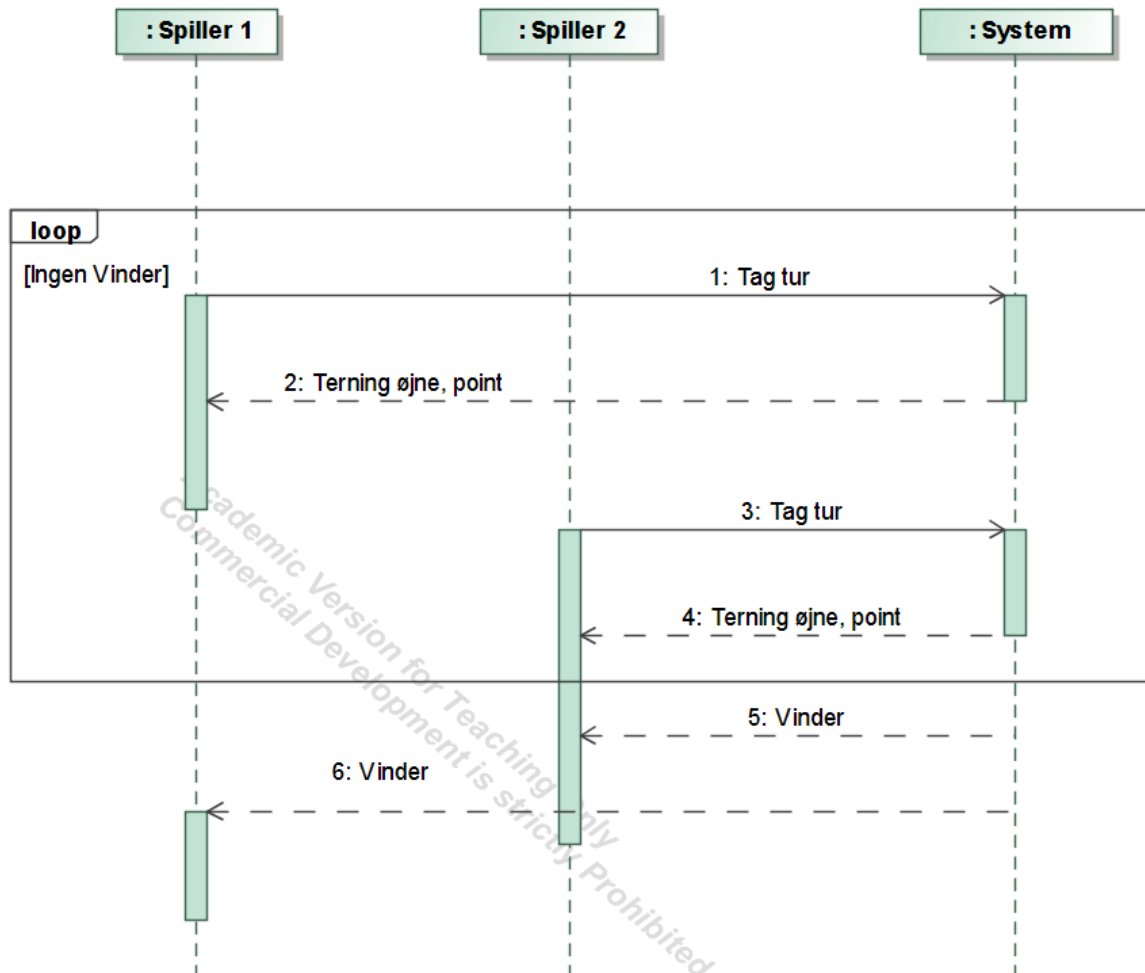


Fig. 2 SSD, som viser hvordan bruger skal interagere med systemet.

Vi har valgt at repræsentere begge spillere som aktører der interagerer med systemet. De får også begge svar på hvem der er vinder.

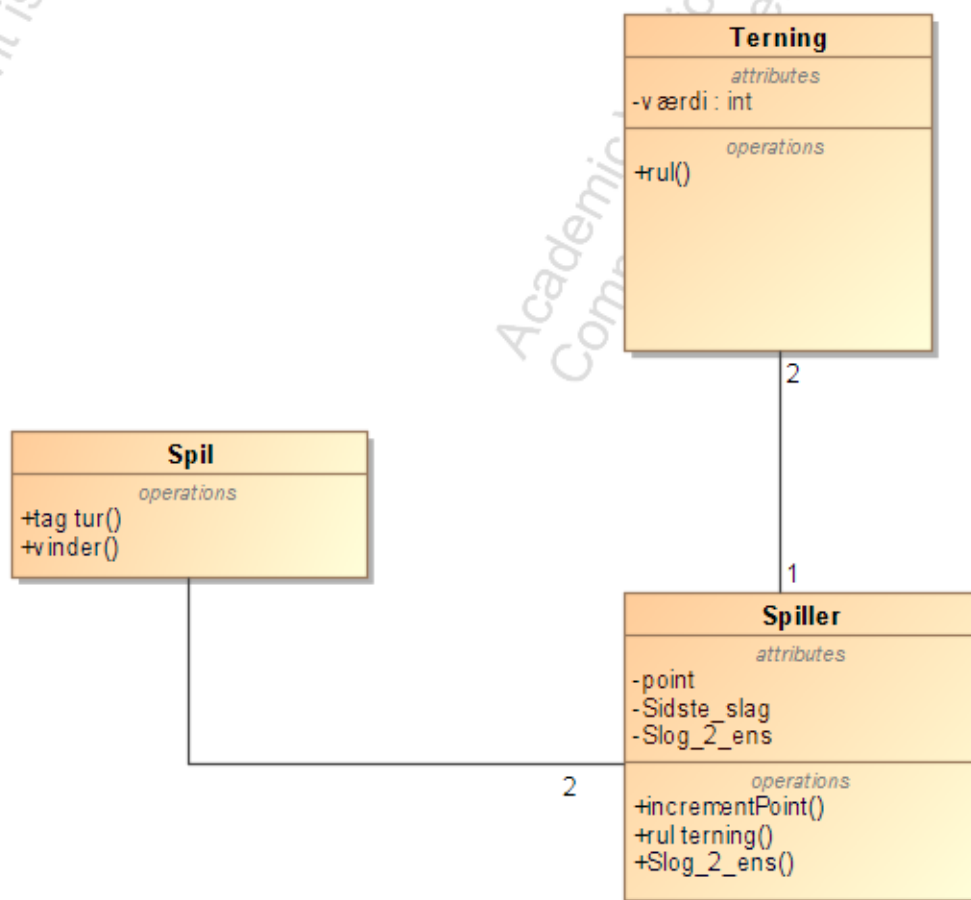


Fig 3.Klassediagram

Klasse diagrammet viser hvordan programmet er struktureret, det skal bemærkes at vi har valgt at undlade raflebæger som ellers er med i vores domain model. Dette er tilfældet fordi vi mener at det ikke udfylder en rolle som de andre klasser ikke gør i forvejen. Klassediagrammet viser vores domain layer. Vi har yderligere vores GUI, der har en klasse der hedder GameController der virker som forbindelsen mellem spil og GUI. Vi har valgt ikke at have et application layer mellem vores GUI og vores Domain. Det har vi valgt fordi vores program er så simpelt at vores GUI kan kommunikere med vores Spil-klasse.

## 8. Implementering

Programmet er skrevet i Java. Da computerne i Databarene har installeret Windows som styresystem, har vi udviklet programmet til at fungere med netop det styresystem. Programmet kommer som jar-fil, som kan køres fra terminalen.

### 8.1 GUI

I forhold til GUT'en til spillet, har vi eksperimenteret og prøvet at lave vores egen. Derfor har vi ikke brugt de GUI'er som var præsenteret til os. Det skal noteres at vi har en "ekstra klasse" med nemlig Main\_1. Den er kun med af implementerings grunde da vi ikke kunne få

lov til at lave en .jar-fil hvis vi ikke også havde den med en main-metode. Den har ikke nogen indflydelse udover den muliggørelse af en .jar-fil og alt logikken ligger i de tidligere nævnte klasser. Der er derfor to main-metoder i projektet, en i GameController og en i Main\_1, dette er ikke best practice men den måde vi kunne opnå den ønskede funktionalitet på.

## 9. Test

### 9.1 Terninger

I kvag af kundens krav har vi testet terning-klassen 1000 gange. Vi undersøgte om terningen leverede resultater uden for intervallet [1:6]. Det kan vi afkræfte, da fordelingen af udfaldene er testet. Vi har valgt, at udfaldene skal være inden for 10% af det teoretiske udfald, for at testen er godkendt. Tilsammen viser de to test at terninge-klassen giver normalfordelte resultater, og dermed virker efter hensigten.

### 9.2 Responstid

Vi har testet responstiden på vores gui ved at måle hvor lang tid der går fra man klikker på rul-knappen til opdateringerne er printet. Det har vist sig at der maksimalt går 2 millisekunder og det er inden for kravene fra kunden.

## 10. Konklusion

Igennem hele processen er domænemodel og use case, er blevet benyttet til at få et bedre overblik over programmet, inden vi begyndte at programmere det. Herudover fandt vi det nødvendigt at opremse samtlige regler, som kunden ønskede, for at få det fulde overblik over hvad spillet skulle indeholde. Gennem processen havde vi få problemer, men vi fik dem afklaret med vores projektleder, som var i dialog med kunden. Afslutningsvis kan vi nu konkludere at der er skabt et terningspil som opfylder kundes krav og regler. Hele projektet har været en iterativ process, det vil sige at vi har testet igennem hele udviklingen, og dermed kan vi sige at der er afleveret et fyldestgørende program.