

# Danmarks Tekniske Universitet



02314 – 62531 - 62532

Introductory Programming, Version Control and test Methods and Development Methods for IT Systems

---

## CDIO 0

---

### AUTHORS

Jeppe Knud Oswald Fischer  
S235440



Subeer Abdirahman Awil Mohamed  
S235088



Sakariya Abdulqadir Ahmed Ali  
S235100



September 29, 2023

## Summary

Dette projekt går ud på et spil, som vi som spilfirmaet IOOuterActive udvikler et brugervenligt spil til vores kunde, der har bedt os om at lave et terningspil. Vi startede med at analysere vores kundens krav og vision. Vi har blandt andet analyseret risikoerne, der kan forekomme i projektperioden, hvor alvorligt det kan være, hvilken betydning det kan have for projektet og hvorvidt vi kunne løse risiciene.

## Hourly Accounting

*Sammen:* 8 timer

*Hver for sig:*

Jeppe: 6 timer

Sakariye: 5 timer.

Subeer: 3,5 timer

## Table of Contents

<i>Summary</i> .....	2
<i>Hourly Accounting</i> .....	2
1. <i>Introduction</i> .....	4
2. <i>Project planning</i> .....	5
3. <i>Requirements and analysis</i> .....	6
3.1 Kravspecifikation .....	6
3.2 Vision.....	6
3.3 Stakeholders og Actors .....	7
3.4 Use Case .....	7
3.5 Risikoanalyse .....	8
4. <i>Design</i> .....	10
4.1 Domain Model.....	10
4 .2 Use Case model:.....	10
5. <i>Implementation</i> .....	11
6. <i>Testing</i> .....	12
7. <i>Conclusion</i> .....	13
8. <i>Appendix</i> .....	14
8.1 Literature .....	14
8.2 Code .....	14

## 1. Introduction

Terningspillet vi har lavet, er lavet til at kunne spilles to personer hvor den der først når 40 point har vundet. Vi har prøvet at arbejde med at få forskellige features lagt ind i spillet som kan læses om i vores kravspecifikation.

## 2. Project planning

Vi har planlagt, at vi vil fokusere på den analyserende del af projektet, som artifacts in requirements specification i den første uge i projektet. Hereafter, tænker vi på at begynde at kode ugen efter, hvor vi udvikler spillet og tester den løs igennem.

### 3. Requirements and analysis

#### 3.1 Kravspecifikation

##### **Functional:**

Det skal være et spil mellem 2 personer og skal foregå med et raffelbære (det kan jo ikke rigtig lade sig gøre?) med 2 terninger, der er blevet sagt god for at det må være 6 sidet terning. Summen af ens slag, bruges til at give points til hver spiller op til 40 eller derover, dette er også godkendt af kunden. Systemet skal kunne gemme data fra spillet, dvs. Summen af point fra hver spiller.

Der er fire ekstra opgaver som går ud på:

1. Spilleren mister alle sine point hvis spilleren slår to 1'ere.
2. Spilleren får en ekstra tur hvis spilleren slår to ens.
3. Spiller kan vinde spillet ved at slå to 6'ere, her er der blevet rettet at du ikke vil kunne bruge slaget for forrige kast da dette ikke vil være en mulighed, OK med kunde også.
4. Spilleren skal slå to ens efter at have opnået 40 point eller mere, også godkendt af kunde.

Skal kunne bruges af DTU studerende uden større forklaring.

##### **Usability:**

Det skal kunne fungere på Windows

Det skal kunne køre i DTU databar.

Det skal kun kunne bruges mellem to personer.

##### **Reliability:**

Spillet skal kunne køre uden større nedbrud.

##### **Performance**

Spillet skal kunne give svar hurtigst muligt, projektlederen mener at dette er 333 millisekunder.

##### **Supportability**

Kunden ønsker at der skal kunne videreudvikles på spillet.

Derfor skal der være adgang til kode og branches så der kan videreudvikles. (Github adgang)

Kunden ønsker her ud over udført test af spillet hvor der undersøges forkomst af mulige værdier og antallet af kast hvor der vil være 2 ens.

#### 3.2 Vision

IOOuterActive har lavet et brugervenligt terninge spil som er nemt for alle at spille og skal kunne køres på DTU databars computere. Spillet spilles mellem to spillere og den, der først når 40 point

eller derover, vinder spillet. Spillet går i sin enkelthed ud på at to spillere slår to terninger hvor summen udgør ens point. Herudover er der lagt ekstra features ind i det som f.eks. At hvis man slår to 1'ere så vil man miste sine point.

### 3.3 Stakeholders og Actors

Spillet består af 2 stakeholders, personerne der vil spille og producenten der laver spillet. Primary actor vil være spiller da han direkte forbundet til spillet.

Stakeholder:

Spiller 1

IOOuterActive (Producenten af spiller)

Actors

Spiller

Da der skal to spillere til at spille, vil disse to spillere være vores primære actors.

### 3.4 Use Case

*IndsætNavn UC\_1 (sakariye)*

1. System viser siden frem
2. spilleren indtaster sit navn.
3. Så længer navnet eksister
  - 3a. Systemet spørger spilleren at indtaste navnet igen.
  - 3b. Systemet validere spillerens detaljer.
4. Systemet godkender spillernes navn.

*StartSpillet. UC\_2 (Subeer).*

1. Spilleren starter spillet
2. Systemet starter spillet op.
  - 2a. Hvis det er spillerens runde, slår spillet terningerne.
  - 2b. Hvis det ikke er spillerens runde venter spilleren til det er sin runde.
3. Spillet fortsættes indtil der findes en vinder eller taber.
4. Efter en af spillene har vundet eller tabt,
  - 4a. Kan startet op igen efter ønske.

*SlåTerninger UC\_3 Jeppe*

1. Spillet slår terningerne.
2. Systemet printer resultatet.
  - 2a. Hvis spilleren får to 1'er taber spilleren alle ens point.

- 2b. Hvis spilleren slår to ens får spiller en ekstra tur
- 2c. Hvis spilleren slår to 6er, får den samme spiller lov til at slå igen.
- 2d. Hvis spilleren slår to 6er igen, vinder spilleren spillet.
- 3. Summen af terningerne bliver lagt sammen og gemt
- 4. Runden går videre til næste spiller
  - 4a. Fortsættes indtil en til en af spillerne får 40 point eller mere.
- 5. Den første spiller med 40 point eller mere, vinder spillet hvis der slås 2 ens.
  - 5a. Hvis der slås to 1er resetter dette spillet og spilleren starter forfra. (skal lige tjekkes op på)

### 3.5 Risikoanalyse

#### **Estimation:**

Tiden, der skal bruges til at udvikle spillet, er blevet undervurderet.

Modarat (3) – Serious (4)

Ud fra modellen som er vist nedenunder, får denne risiko samlet værdi 12. Dette vil være et problem, da der er mulighed for at vi ikke kommer i mål med opgaven til tiden. Tiden skal genovervejes.

#### **People:**

Udviklere er ikke uddannet nok.

High (5) – Catastrophic (5)

Denne værdi fra modellen bliver 25 og derfor skal der søges hjælp inden det bliver for sent.

Sygdom eller andre årsager til udviklere ikke kan være til stede.

Moderate (3) – Tolerable (3)

Værdi 9, dette kan godt gå hen at blive et problem, da det vil forsinke projektet endnu mere.

#### **Requirements:**

Udviklere forstår ikke kravene til spillet:

Moderate (3) – Tolerable (3)

Værdi 9, denne burde være løst i forhold til kravspecifikationerne.

Ikke alle features vil komme med:

Significant (4) – Serious (4)

Værdi 16, hovedopgaven er at lave et terninge spil, og features er ekstra. Derfor vil vi prioritere hovedopgaven af spillet og prøve efter bedste mulighed for at lave alle features.

#### **Tools:**

Spillet virker ikke på DTUs databar.

Moderate (3) – Catastrophic (5)

Værdi 15, dette vil være et problem, men burde kunne løses i form af test.

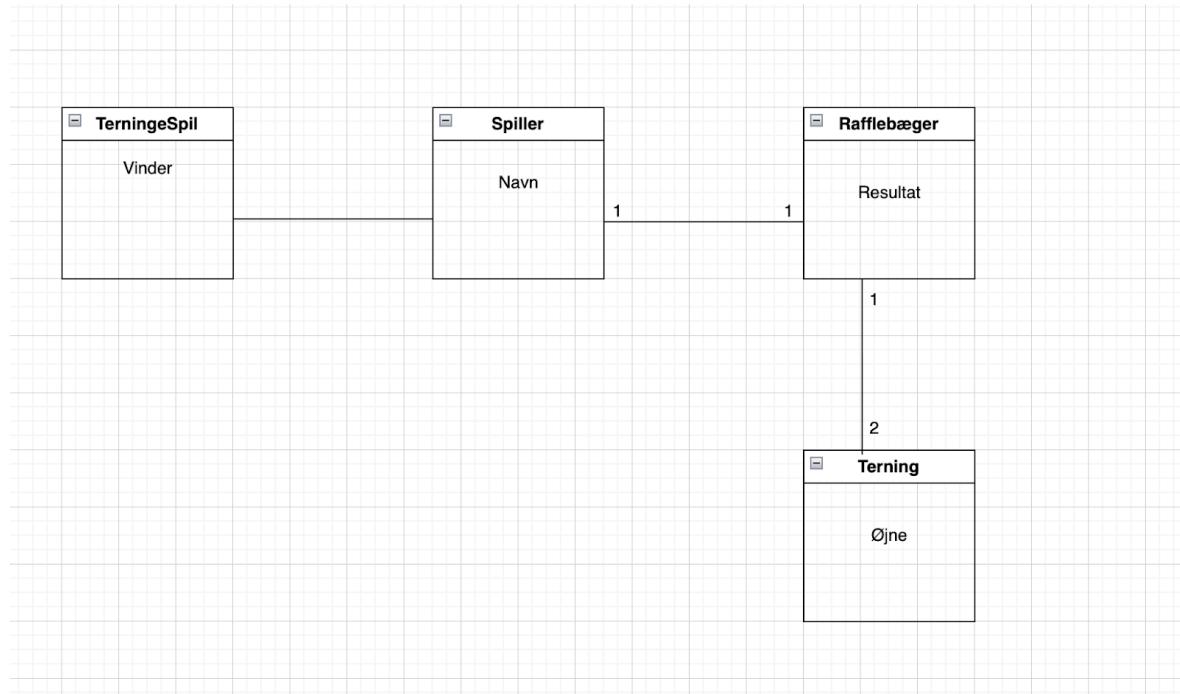
***Risiko plan:***

Vi vil acceptere risiciene, da vi mener vi kan arbejde rundt omkring dem og stadig løse opgaven, dog vil der være en mulighed for at de ekstra features der skal være i spillet ikke kommer med, hvilket vil være ærligt. Til gengæld burde vi aflevere et spil der kan videreudvikles på.

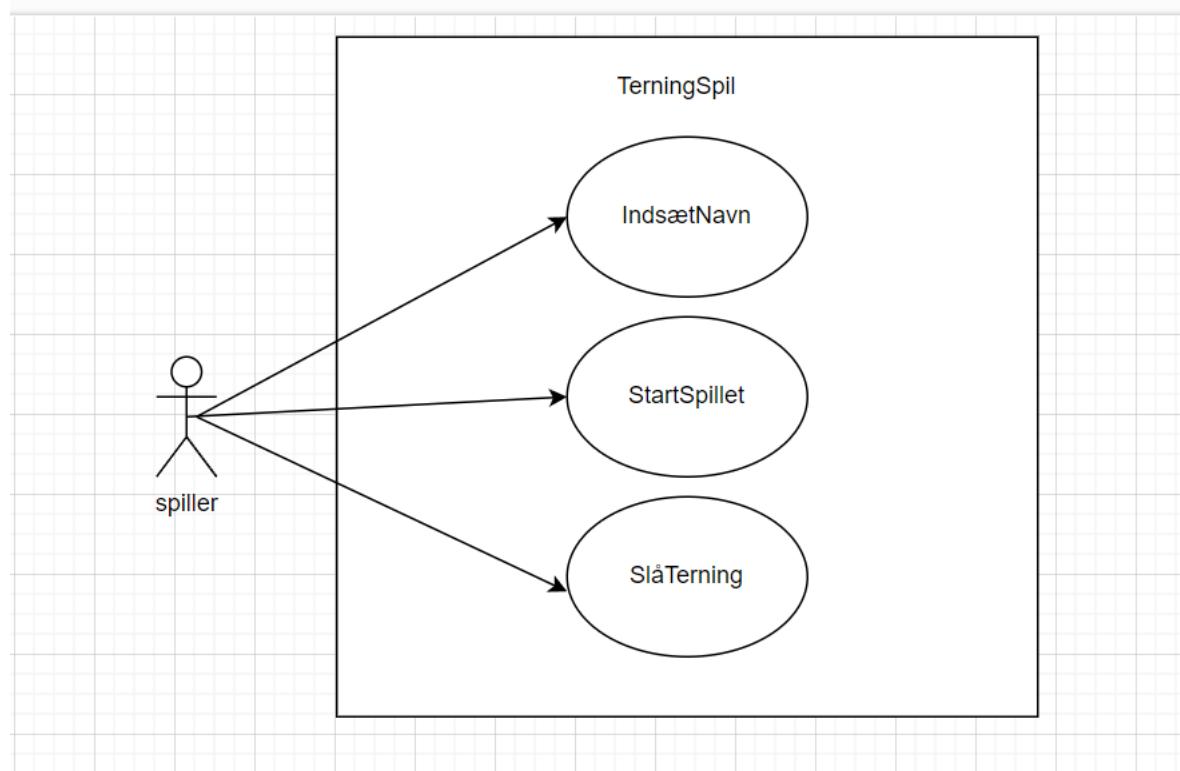
## 4. Design

### 4.1 Domain Model

Her har vi tegnet et domain model der forklarer hvordan spillet kommer til at virker.



### 4.2 Use Case model:



## 5. Implementation

```
if (player1 >= 40) {  
    System.out.println("player1 er vundet");  
} else if (player2 >= 40) {  
    System.out.println("player2 er vundet");  
}
```

Her har vi lavet en kode der bestemmer hvilken spiller der vinder og opnår først 40 point

```
int dice1 = rand.nextInt(6) + 1;  
System.out.println("du har scoret: " + dice1);  
// programmet vælger et tilfældige tal som er mellem 1 og 6  
  
int dice2 = rand.nextInt(6) + 1;  
System.out.println("du har scoret: " + dice2);  
// programmet vælger et tilfældige tal som er mellem 1 og 6  
  
int sum = dice1 + dice2;
```

Her vælger vores program to tilfældige tæller og summer de sammen.

## 6. Testing

Vi har lavet test undervejs for at sikre at de funktioner vi har lavet virker. Derudover har vi testet på en person som ikke har haft noget med udviklingen at gøre for at sikre os at det virket efter hensigten og andre der skal bruge det kunne finde ud af det, hvilket vi har haft succes med.

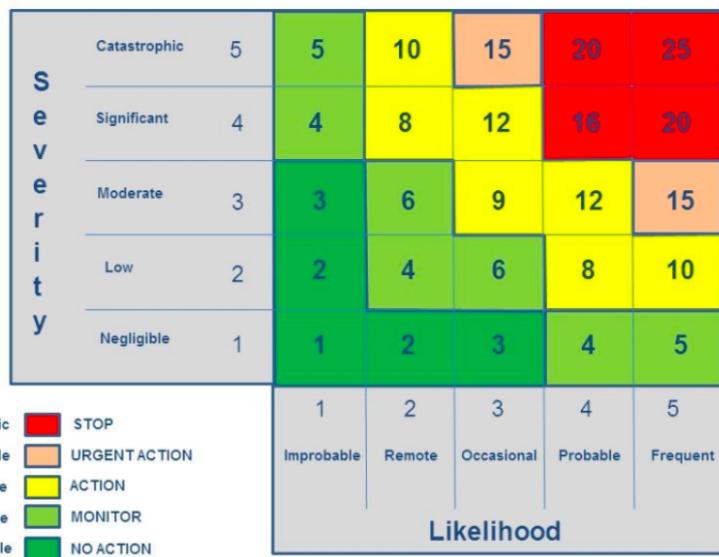
## 7. Conclusion

Vi kan konkludere at vi har fået lavet et terrninge spil som virker. Dog må vi erkende at der er visse funktion vi ikke har fået til at virke som vi ønsket det er blandt andet: En ekstra tur ved to 6'ere, at en spiller kan slå to 6'ere i træk og vinde, og at man skal slå to ens for at vinde. Dette er selvfølgelig et problem, da det var en del af kravene.

## 8. Appendix

Figur fra: Deena Francis - Power Point - Lecture 4

### **Probability \* Impact matrix:**



## 8.1 Literature

### *Bøger:*

Larman, Craig. Applying UML and Patterns - Pearson Education 2005

### *Web:*

<https://www.uml-diagrams.org/>

## 8.2 Code

### *GitHub:*

[https://github.com/KnudOswald/CDIO\\_1](https://github.com/KnudOswald/CDIO_1)

Se vedhæftning.