

# Rapport - CDIO del 1

Må ikke anvendes i undervisningen

Udarbejdet af

S175219 - Thomas Løvendahl Vestergaard

S175216 - Kasper Bøgvad Nielsen

S175213 - Casper Kyster Andersen

S172139 - James Giles

S144260 - Tahany Nemer

S160435 - Cornelius Frost Schilling Hedegaard

S163157 - Abdirisaq Abdiqadir Mohamud Farah

## Timeregnskab

Dato	Deltager	Design	Impl.	Test	Dok.	Andet	Ialt
29-09-2017	Thomas	1	0	1	2	1	5
29-09-2017	Kasper	1	1	1	2	1	6
29-09-2017	Tahany	0	8	2	1	0	11
29-09-2017	James	0	0	0	0	0	0
29-09-2017	Casper	1	0	1	6	0	8
29-09-2017	Cornelius	0	5	2	0	0	7
29-09-2017	Abdirisag	0	1	0	2	1	4

## Indholdsfortegnelse

<b>Timeregnskab</b>	<b>1</b>
<b>1 Indledning</b>	<b>3</b>
<b>2 Analyse</b>	<b>4</b>
<b>3 Design</b>	<b>9</b>
<b>4 Implementering</b>	<b>12</b>
<b>5 Test</b>	<b>13</b>
<b>7 Konklusion</b>	<b>22</b>
<b>Litteraturliste</b>	<b>23</b>

## 1 Indledning

Som udviklere i spilfirmaet IOOuterActive, har vi fået sat til opgave at udvikle et terningespil, som skal bruges til computerne i DTU's databarer. Hertil vil vi starte projektet med først at analysere kundens vision og krav, med hjælp fra forskellige analysemodeller, herunder domæne modeller, use case modeller, virksomhedsmodeller. Herefter vil vi designe og implementere programmet ved hjælp af Eclipse og Github, samt herefter teste dette program med test cases, og junit tests ved hjælp af Eclipse.

## 2 Analyse

### Krav fra kunden:

#### Funktionelle krav:

K1 - Spillet skal foregå mellem 2 personer, der skifter tur.

K2 - Spillet skal efter kommando fra spilleren, der har turen, kunne slå med to 6-sidede terninger på samme tid.

K3 - Terningerne skal have samme (16.67%) chance for at slå alle tal fra 1 til 6, med en max. afvigelse på 2.5%

K4 - Spillet skal kunne lægge de "slåede" antal øjne sammen, lægge det til en spillers totale points og udskrive tallet.

K5 - Hvis en spiller slår to ens, får spilleren en ekstra tur, før den anden spillers tur.

K6 - Hvis en spiller slår to 1'ere, mister spilleren alle sine hidtil optjente points.

K7 - Hvis en spiller slår to 6'ere på en ekstra tur, efter have fået to 6'ere på forrige kast, vinder spilleren spillet uanset total points og andre regler.

K8 - Efter en spiller har opnået 40 points, skal spilleren slå to ens for at vinde spillet. Spilleren vinder stadig selvom der bliver slået to 1'ere.

#### Ikke-funktionelle krav:

K9 - Spillet skal kunne virke på DTU's databarer, som indeholder computere der bruger Windows.

K10 - Spillet skal kunne bruges uden en brugsanvisning.

K11 - Resultatet af terningekast skal vises til brugeren inden for 0,333ms.

K12 - Spillet skal være på Dansk.

## Use Cases:

Indtaste brugernavn	Spilleren kan indtaste et brugernavn, i stedet for at blive kaldt "spiller 1" og "spiller 2".
Kaste terninger	Spillet ruller 2 terninger, og viser individuelle terninge-øjne og summen af alle øjnene.
Redigere pointscore	Spillet kan lægge points til to(2) total værdier, der kun kan redigeres af spillet.
Vise pointscore	Spillet viser de to(2) pointscorer til spillerne før hvert slag.
Tjekke efter bestemte slagkombinationer	Spillet kan ved bestemte kombinationer af slag, gøre specielle ting. To ens antal øjne giver et ekstra slag, to 1'ere sætter spilleren scorer til 0.
Afgøre en vinder	Spillet kan, når krav i spillet er opfyldt, afgøre en vinder. Enten når spilleren slår to(2) ens efter at have 40 points, eller når spilleren slår to(2) 6'ere, to gange i streg.

For at forstå use case "Tjekke efter bestemte slagkombinationer" og "Afgøre en vinder" uddybes disse.

### Tjekke efter bestemte slagkombinationer:

#### Best case scenario:

Efter at en spiller har slået med 2 terninger, og resultatet er to ens antal øjne, får spilleren et ekstra kast.

#### Alternative scenario:

Hvis en spiller slår to 1'ere, mister spilleren alle points, men får stadig et ekstra slag.

Hvis en spiller slår to antal øjne, der ikke er ens, sker der intet.

Hvis en spiller på forrige kast slog to 6'ere, og på dette slag har slået to 6'ere igen, vinder spilleren spillet og får derfor ikke et ekstra kast.

### Afgøre en vinder:

#### Best case scenario:

Efter at en spiller har opnået 40 points, og slår 2 ens antal øjne (også selvom dette er to 1'ere), vinder spilleren spillet.

#### Alternative scenario:

Hvis en spiller på forrige kast slog to 6'ere, og på dette slag har slået to 6'ere igen, vinder spilleren spillet uanset pointscore.

Hvis en spiller har 40 points, men ikke slår 2 ens antal øjne, er spillerens score forsat 40 points, men spilleren vinder ikke spillet.

**Sammenhæng mellem krav og use cases:**

Use cases	↔	Krav
Indtaste brugernavn	↔	K1
Kaste terninger	↔	K2, K3
Redigere pointscore	↔	K1, K4
Vise pointscore	↔	K1, K4
Tjekke efter bestemte slagkombinationer	↔	K5, K6, K7
Afgøre en vinder	↔	K7, K8

**Aktører og interessenter:**

I forhold til aktører og interessenter, er der 4 interessenter og 2 aktive aktører.

De 2 primære aktører er brugeren, der skal spille spillet, hvor brugeren kan indtaste et brugernavn, kaste terninger og se hvor mange points brugeren har. Derudover er der også tiden, der ikke er en person, men i stedet en speciel aktør, der udfører handlinger baseret på, om der er sket noget, f.eks: Det tidspunkt en tur er slut skal spillet vise pointscoren til begge spillere.

I forhold til de sekundære aktører, vil der til systemet ikke være nogen, da alle use cases, ikke har brug for nogle supporterende aktører, da alle use cases enten kræver brugerinput eller venter på bestemte tidspunkter, hvilket tiden tager sig af.

Endelig er der off-stage aktørerne, der ikke er aktører, men i stedet agere som interessenter til det kommende system. Her vil virksomheden, herunder udvikler og projektleder, og kunden forefindes. Disse interessenter vil uddybes i visionen sammen med de andre aktører, der også virker som interessenter.

**Aktørtabel:**

	Use case 1 Indtaste brugernavn	Use case 2 Kaste terninger	Use case 3 Redigere pointscore	Use case 4 Vise pointscore	Use case 5 Tjekke efter bestemte slagkombin ationer	Use case 6 Afgøre en vinder
<b>Aktør 1</b> (Brugeren)	x	x		x		
<b>Aktør 2</b> (tiden)			x	x	x	x

## Vision:

- **Beskrivelse af området/domænet**

Projektet omhandler et terningespil, der spilles mellem 2 personer. Spillet skal kunne slå 2 terninger, og lægge summen til en spillers totalsum, før det bliver den anden brugers tur. Derefter bliver det den anden brugers tur, og samme proces som for den første bruger udføres. Derudover skal spillet tjekke efter specielle slag hvor spillet skal give spilleren ekstra slag, mulighed for vinde spillet, eller miste alle sine points.

- **Beskrivelse af interessenter, deres mål og problemer**

Projektets interessenter er folk, der er interesseret i terningespillet. Dette vil inkludere brugeren, der kan bruge spillet som underholdning, evt. i pauser eller som afslapning i mellem arbejdsopgaver. En anden interessent er kunden, der har bestilt og betalt for spillet, da kunden ikke ønsker at betale for noget, der ikke virker. Endelig har virksomheden der udvikler også en interesse i spillet, da virksomheden bliver betalt for at udvikle spillet.

- **Produktoversigt – inspireret af use case diagrammet**

Spilleren har mulighed for at indtaste et brugernavn, kaste med terningerne og se sin pointsscore. Systemet kan baseret på bestemte forudindstillede kommandoer ændre på scoren, tjekke efter bestemte slag, afgøre en vinder og vise scoren når terningerne er kastet.

- **Fordele ved brug af systemet**

Spillet har til opgave at underholde brugere i databaserne på DTU. Da spillet er lavet til 2 spillere, kan der opstå samtaler mellem brugere, således at studiemiljøet i de individuelle kursushold har mulighed for forbedring.

- **Begrænsninger**

Spillet er lavet til specifikt 2 personer, således at en gruppe på et ulige antal, vil have problemer med at bruge spillet.

- **Økonomi**

Når først spillet er lagt på computerne, kræver det ikke noget specielt vedligeholdelse og derfor vil den eneste udgift for kunden være at få lave spillet, og at få den lagt på alle computerne.

- **Licens og installation**

Spillet kræver ingen licenser, fordi det er så lille et system, der ikke kræver hverken open source- eller betalingslicenser. Installationen vil foregå ved at en admin lægger spillet på alle brugere, hvilket gør spillet ikke skal installeres og derfor ikke roder i Windows-filer. Derudover gør det også at spillet ikke skal lægges på en computer ad gangen.

- **Oversigt over systemets egenskaber**

1. Spillet kan slå 2 terninger.
2. Spillet kan holde styr på brugernavne og points og vise begge.
3. Spillet kan holde styr på bestemte terningekast og udfører handlinger derfra
4. Spillet kan afgøre en vinder baseret på egenskab 2 (points) og 3 (bestemt slag).



## Supplerende funktioner

De beskriver krav, der ikke er beskrevet i use cases, det starter i inception fasen og forfines og afsluttes i elaboration, såsom:

- **Ikke-funktionelle krav**

1. Spillet skal virke på computerne i databarerne på DTU, der bruger Windows.
2. Spillet skal kunne bruges uden en brugsanvisning.

Spillet skal forklare om reglerne enten før spillet er begyndt eller når følgende sker, 2 ens øjne giver ekstra slag, 2 ens 1'ere gør spilleren mister alle sine points og at spilleren skal slå 2 ens efter have fået 40 points for at vinde.

3. Resultatet af terningekast skal vises til brugeren inden for 0,333ms.

Dette vil svare til den tid, den tager for en bruger at lægge mærke til en ændring, og svartiden burde derfor være hurtigere.

4. Spillet skal være på engelsk.

- **Begrænsninger**

På grund af GUI'en der bliver brugt, bruges der over 0,333 ms første gang summen af de 2 terninger skal gives. Derudover er spillet kun på engelsk, hvilket begrænser brugerne til kun at kunne forstå spillet, hvis de kan forstå engelsk, hvilket nok er stort set alle, som er beregnet til at anvende systemet.

- **Juridiske regler**

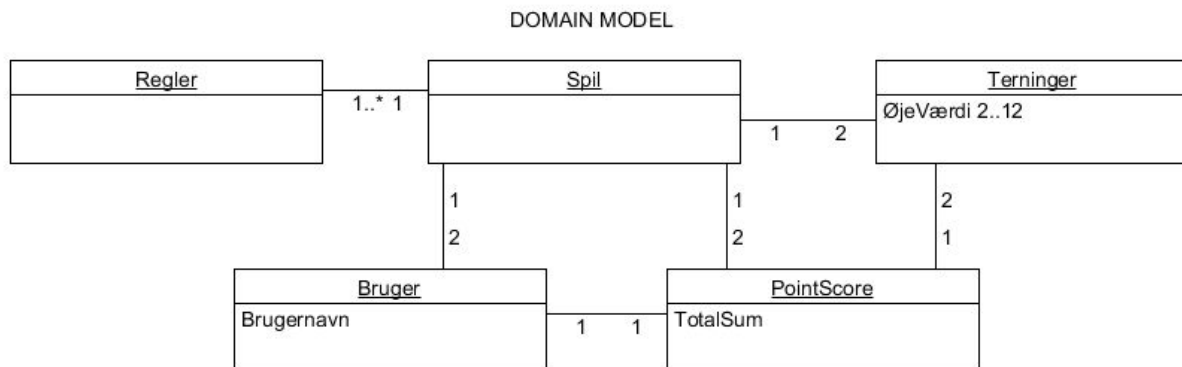
Brugeren skal selv indtaste et navn, men dette er ikke et problem, i forhold til persondataloven om almindelige ikke-følsomme oplysninger, da dataene kun bruges i programmet, og bliver slettet efter et spil er slut.

## Ordbog

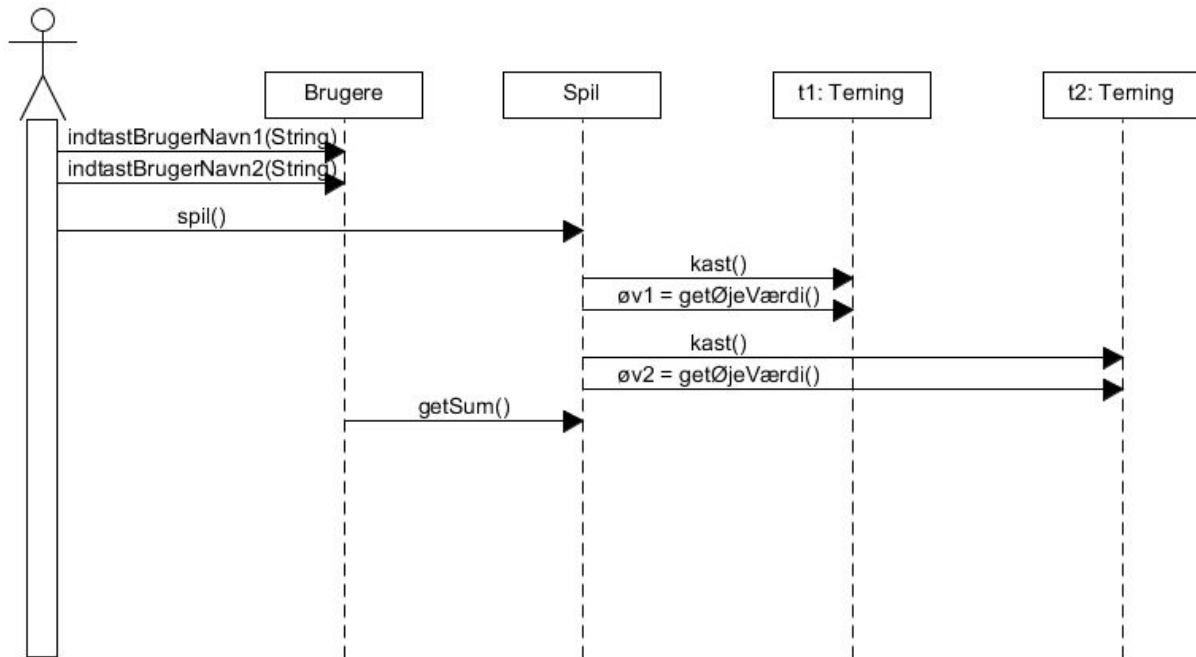
Udtryk	Definition og Information
Databar	Er et lokale udstyret med stationære computere, specifikt til brug af studerende på DTU.
GUI	Graphical user interface. Grafisk brugeroverflade. Den grafiske del af et program, som man ser på skærmen.
DTU	Danmarks Tekniske Universitet.

### 3 Design

#### Domænenemodell:

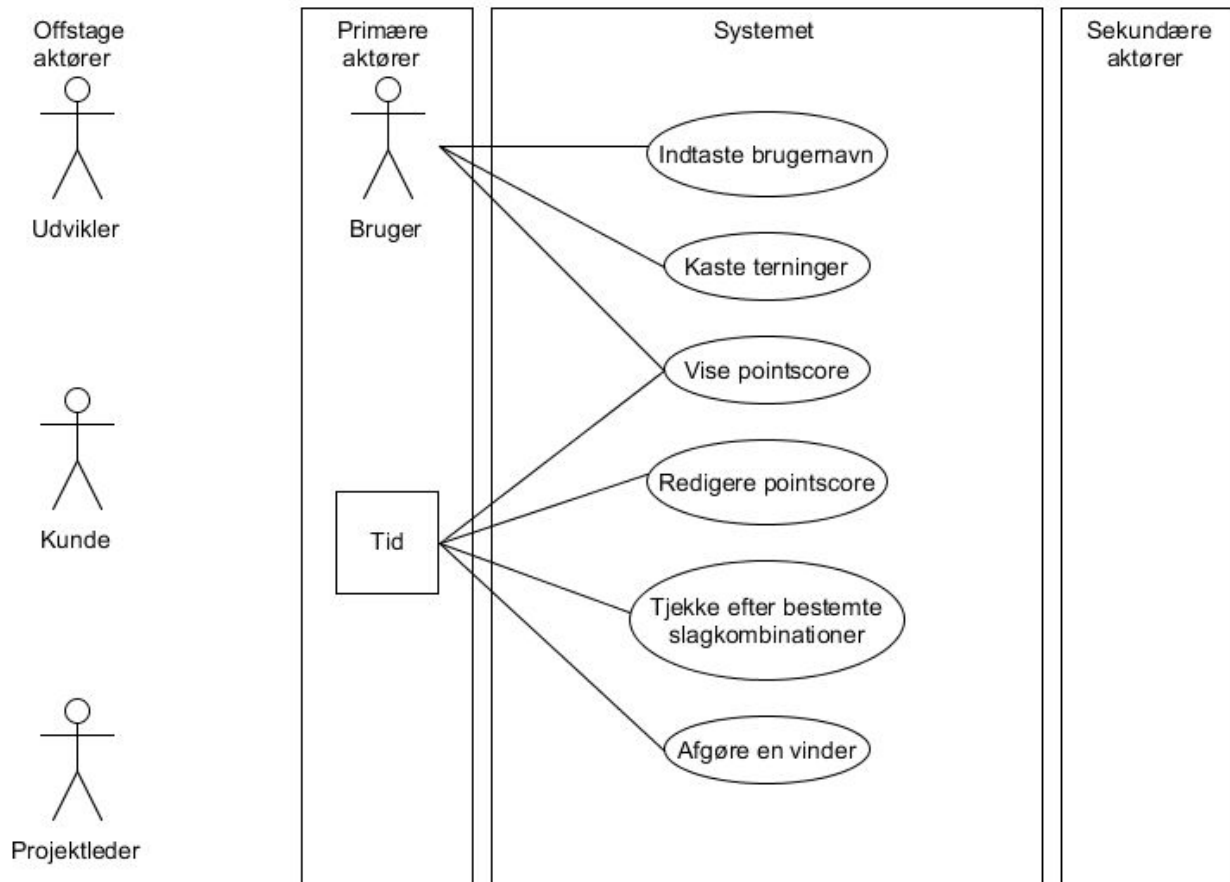


Programmet er et spil. Spillet går ud på at man kaster 2 terninger, og får en værdi mellem 2-12, når de to terninger er lagt sammen. Herefter bliver summen af dette lagt til en pointscore, som er knyttet til en bruger.

**Sekvens diagram**

Sekvensdiagrammet beskriver hvordan objekterne samarbejder i programmet. Programmet starter og brugerne indtaster deres navn og trykker spil. Herefter kører programmet med at kast terning 1 og terning 2, og derefter bruges en get-metode til at få øje værdierne og til sidst bliver værdierne lagt til spillernes score med en get-metode.

## Use case diagram



Spilleren har mulighed for at indtaste et brugernavn, kaste med terningerne og se sin pointsscore. Systemet kan baseret på bestemte forudindstillede kommandoer ændre på scoren, tjekke efter bestemte slag, afgøre en vinder og vise scoren når terningerne er kastet.

## 4 Implementering

Vores program starter op og giver en velkomstbesked. Herefter beder den om at give et navn til player1 og derefter player2. Herefter laver programmet en setName til hver bruger Strings. Herefter bliver der lavet to terninger, som giver en random værdi mellem 1-6. Herefter bliver spillerne skiftevis bedt om at indtaste et vilkårligt input for at kaste med terningerne, efter inputtet vises de to terningers værdier på skærmen. Og værdierne bliver lagt sammen en sum, som bliver lagt til scoren, til den spiller der kastede terningerne.

Den spillers score, som når 40 point først, får en besked som gratulerer spilleren for at have vundet spillet.

## 5 Test

Skal laves en junit test der beviser at rafflebægeret virker korrekt, hen over 1000 kast.

Skal laves en junit test der optæller antallet af forekomster for hver mulig værdi af summen (2-12), teste hvor mange gange terningerne er ens, og tjekke om resultaterne stemmer overens med sandsynlighederne.

Disse sandsynligheder er:

For at et par terninger slår det samme:

1 ud af 6

For at et par terninger slår summen 2 og 12:

1 ud af 36

For at et par terninger slår summen 3 og 11:

1 ud af 18

For at et par terninger slår summen 4 og 10:

1 ud af 12

For at et par terninger slår summen 5 og 9:

1 ud af 9

For at et par terninger slår summen 6 og 8:

5 ud af 36

For at et par terninger slår summen 7:

1 ud af 6

**Test cases**

Test case ID	TS00.
Summary	Terningerne skal opfylde sandsynlighedskrav for kast med 2 terninger.
Requirements	Kundekrav / K3
Preconditions	None
Postconditions	None
Test procedure	<ol style="list-style-type: none"> <li>1. Kast terningerne 1000 gange</li> <li>2. Observer resultaterne</li> <li>3. Samlign med overstående sandsynligheder</li> </ol>
Test data	
Expected result	Kastene passer med de givne sandsynligheder
Actual result	Kastene passer med de givne sandsynligheder
Status	Passed
Tested by	Tahany
Date	13/10/2017
Test environment	Eclipse Oxygen 4.7.0 i Windows 10

Test case ID	TS01
Summary	Spillet skal kunne skifte tur mellem 2 spillere.
Requirements	K1
Preconditions	None
Postconditions	Det er nu spiller 2's tur og systemet venter på input
Test procedure	1. Kast terningerne
Test data	
Expected result	Spiller 2 kan tage sin tur
Actual result	Spiller 2's tur
Status	Passed
Tested by	Kasper og Thomas
Date	13/10/2017
Test environment	Eclipse Oxygen 4.7.0 i Windows 10



Test case ID	TS02
Summary	Terninger skal kunne slås efter kommando fra spilleren.
Requirements	K2
Preconditions	None
Postconditions	Spiller 2 har sin tur
Test procedure	1. En kommando er indtastet
Test data	Kommando = a
Expected result	Terningerne kastes og der gives to værdier mellem 1-6
Actual result	Terningerne kastes og der gives to værdier mellem 1-6
Status	Passed
Tested by	
Date	13/10/2017
Test environment	Eclipse Oxygen 4.7.0 i Windows 10

Test case ID	TS03
Summary	Summen af de to terninger skal kunne lægges til brugerens score
Requirements	K4
Preconditions	Score = 5
Postconditions	Spiller 2 har nu sin tur
Test procedure	<ol style="list-style-type: none"><li>1. Sæt terningerne til 3 og 4</li><li>2. Observer pointscoren</li></ol>
Test data	SetDice(3, 4)
Expected result	Score = 12
Actual result	Score = 12
Status	Passed
Tested by	Kasper og Thomas
Date	13/10/2017
Test environment	Eclipse Oxygen 4.7.0 i Windows 10

Test case ID	TS04
Summary	Ved slag af 2 ens, skal spilleren få et ekstra slag.
Requirements	K5
Preconditions	None
Postconditions	Spilleren har sin tur igen
Test procedure	<ol style="list-style-type: none"> <li>3. Sæt terningerne til 1 og 1</li> <li>4. Sæt terningerne til 2 og 2</li> <li>5. Sæt terningerne til 3 og 3</li> <li>6. Sæt terningerne til 4 og 4</li> <li>7. Sæt terningerne til 5 og 5</li> <li>8. Sæt terningerne til 6 og 6</li> </ol>
Test data	setEyes(1,1), setEyes(2,2), setEyes(3,3), setEyes(4,4), setEyes(5,5), setEyes(6,6)
Expected result	Spilleren får en ekstra tur
Actual result	
Status	
Tested by	
Date	
Test environment	

Test case ID	TS05
Summary	Ved slag af to 1'ere skal spillerens score sættes til 0.
Requirements	K6
Preconditions	Score = 20
Postconditions	Spiller 1 har nu sin tur pga. TS04
Test procedure	9. Sæt scoren til 20 10. Sæt terningerne til 1 og 1
Test data	p1.setScore(20)
Expected result	Spilleren har mistet sine points
Actual result	
Status	
Tested by	
Date	
Test environment	

Test case ID	TS06
Summary	Ved slag af to 6'ere to gange i træk vinder spilleren.
Requirements	K7
Preconditions	None
Postconditions	Spillet slutter
Test procedure	<ol style="list-style-type: none"><li>1. Slå to 6'ere med terningerne</li><li>2. Slå igen to 6'ere med terningerne</li></ol>
Test data	setEyes(6, 6), setEyes(6, 6)
Expected result	Spilleren vinder spillet
Actual result	
Status	
Tested by	
Date	
Test environment	

Test case ID	TS07
Summary	Spilleren skal slå 2 ens efter opnåelse af 40 point for at vinde spillet.
Requirements	K8
Preconditions	Score > 40
Postconditions	Spillet slutter
Test procedure	11. Sæt scoren til 40 12. Sæt terningerne til 3 og 3
Test data	setDice(3, 3)
Expected result	Spiller 1 vinder spillet
Actual result	
Status	
Tested by	
Date	
Test environment	

## 7 Konklusion

Vi kan konkludere at processen af projektet godt kunne have været mere strømlinet. Vi glemte udtrykket "planlægning", og fik et par problemer med hvem der lavede hvad. Det leder ind til, at man kan se det i vores kode. Vi havde planer om at lave ekstraopgaverne, men måtte til sidst konkludere at vi ikke kunne nå det, da vores planlægning simpelthen ikke var til stede. Vi kan derfor konkludere til fremtidige projekter, ville det være en god idé, at bruge det første stykke tid, på at planlægge projektet, så vi ikke kommer i samme situation igen.

## Litteraturliste

*Applying UML and Patterns*. 3. ver. Craig Larman. Prentice Hall 2008. ISBN 0-13-148906-2.

Brugt løbende igennem afsnit 2 og 3.

Salomon, Inge-Lise. (u.d.). *Forlæsningsslides 3-6*. Hentet fra [cn.inside.dtu.dk](http://cn.inside.dtu.dk).

Brug løbende igennem afsnit 2 og 3.

Nyborg, Mads. (u.d.). *Tavlenoter*. Hentet fra Google Docs:

[https://docs.google.com/document/d/1nhgIjX\\_WCB6znKg1sxO3x5EeXiGf-2HLcwrOqbmebHs](https://docs.google.com/document/d/1nhgIjX_WCB6znKg1sxO3x5EeXiGf-2HLcwrOqbmebHs)

Brugt løbende i gennem udvikling af program.