



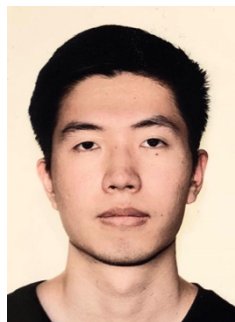
02312 62531 62532

INDLEDENDE PROGRAMMERING, UDVIKLINGSMETODER TIL IT-SYSTEMER OG VERSSIONSTYRING
OG TESTMETODER

CDIO 1

Gruppe 22

Andreas Vilholm Vilstrup
s205450



Chenxi Cai
s205420

Alexander Solomon
s201172



Oliver Fiedler
s205423

Isabel Grimmig Jacobsen
s205473



Ahmad Shereef
s173750

Afleveret d. 2/10-2020

Resumé

I denne rapport løses en opgave stille af spilfirmaet IOOuterActive. Opgaven går ud på at producere et terningespil, som efterlever nogle prædefinerede krav. Spillet bliver programmeret i IntelliJ, og testes hen over 1000 kast, for at kvalitetssikre produktet.

Det færdige resultat er et terningespil, som kan spilles af 2 spillere. Det virker på windows maskiner i Databaren på DTU. Spillerne skiftes til at slå med to terninger. Der tælles point undervejs, idet resultatet vises med det samme og vinderen er den spiller der først opnår 40 point. Desuden nulstilles den pågældende spillers point, hvis han/hun slår to 1'ere. Ved slag med to ens gives en ekstra tur til pågældende spiller.

Indholdsfortegnelse

RESUMÉ	2
INDLEDNING	4
KRAV / PROBLEMFORMULERING	4
VURDERING AF KUNDENS KRAV/ VISIONER	4
PROJEKTPLANLÆGNINGEN	5
ANALYSE	6
AKTØR MODEL	6
USE CASE OVERSIGT / DIAGRAM	6
DOMAIN MODEL	7
FLOWCHART	7
DESIGN	9
TEST	10
RESULTAT	11
IMPLEMENTERING	11
EN GUIDE TIL NÅR MAN MODTAGER FILEN OG VIL KØRE DEN:	11
BESKRIVELSE AF VORES KLASSER	12
KONKLUSION	12
TIME REGNSKAB	12
LITTERATUR- OG KILDELISTE	12

Indledning

I denne rapport dokumenteres løsningen af en opgave, stillet af spilfirmaet IOOuterActive. Opgaven går ud på at udvikle og designe et terningspil. Ud fra kundens visioner skal vi udvikle et spil, som indeholder to spillere og to terninger (som spillerne skiftes til at slå med). Der vil være én vinder, som er den spiller, der først slå øjne til en værdi af 40 point. Vi har valgt, i samarbejde med IOOuterActive, at opstille nogle forhindringer i spillet, for at gøre det sværere at vinde og for at give spillet en dybde. Bl.a har vi indført, at man mister alle sine point, hvis man slår to 1'ere. Derudover får man en ekstra tur, hvis man slår to ens tal (dette gælder også hvis man slår to 1'ere). Spillet kræver ikke nogen yderligere brugsanvisning, da det undervejs vil blive forklaret, hvad den enkelte spiller skal gøre. Rapporten indeholder også modeller og diagrammer, som giver et overblik over, og en forståelse for, hvordan spillet virker.

Krav / problemformulering

Der er opstillet nogle krav i samarbejde med kunden og deres visioner, for selve programmet. Det første krav er at spillet skal kunne køres på en af windows maskinerne i Databarerne på DTU og spilles af alle almindelige mennesker.

Derudover er også opstillet følgende krav:

- To spillere
- To terninger, som kan slås med
- Resultatet skal vises med det samme
- Der tælles point
- To 1' restarter dine point
- To ens giver ekstra tur
- Først opnået 40 point, vinder
- Teste terningernes "random" funktion af 1000 kast

Vurdering af kundens krav/ visioner

Forventningsafstemning med kunden.

Vi har i samarbejde med kunden vurderet og gennemgået deres visioner for dette program/spil. Vi er nået frem til at basisdelen af spillet fungerer fint og at følgende visioner/krav for ekstra opgaver er mulige og giver mening i forhold til det færdige spil:

Spilleren får en ekstra tur hvis spilleren slår to ens.

Denne opgave fungerer fint i praksis og var nem at implementere.

Spilleren mister alle sine point hvis spilleren slår to 1'ere.

Dette har vi løst således, at man også stadig får en ekstra tur, efter man har fået fratrasket alle sine point. Det er dog uklart beskrevet af kunden om man bør få en ekstra tur efter denne straf.

Vi er nået frem til at følgende visioner/krav for ekstra opgaver, ikke er mulige eller giver mening i forhold til det færdige spil:

Man kan vinde spillet ved at slå to 6'er to runder i træk, uanset antal point spilleren måtte have;

Vi havde ikke tid/ressourcer til at fuldføre denne opgave inden for tidsfristen. Med mere tid/ressourcer ville denne opgave godt kunne løses og blive en fin del af spillet.

For at vinde skal man slå to ens, efter de 40 point er opnået;

Vi havde ikke tid/ressourcer til at fuldføre denne opgave inden for tidsfristen. Med mere tid/ressourcer ville denne opgave godt kunne løses, men noget der er uklart, som kunne blive beskrevet bedre, er hvis man slår 2 1'ere. Vinder man så stadig eller mister man sine point og skal starte forfra igen?

Projektplanlægningen

Vi har valgt at bruge følgende programmer med henblik på at møde kundens krav:

IntelliJ:

IntelliJ er blevet benyttet som program til kodning i .java. Derudover tilbyder IntelliJ en platform som gør det nemt at arbejde flere på samme fil, qua deres integration af .git.

GitHub:

For at vi kan arbejde på kodningen hver for sig, har vi brugt GitHub til at dele og hente nye versioner. Dette gør, at vi ikke behøver sidde og arbejde på samme computer, men kan sidde og arbejde individuelt på vores egen computer. Derudover var der også mulighed for, at kunden kunne gå ind og kigge/rette på den kode, vi har lavet.

Domain diagram:

Vi har valgt at lave et domain diagram, for at kunne få et overblik over hvad programmet indeholder, både for programmører og kunden.

Aktør oversigt:

Kunden
Spillere
Udviklere

Flow-chart:

Vi har benyttet et Flow-chart til at skabe overblik over hvordan programmet egentlig skal designes. Dette har vi brugt til at analysere, styre og dokumentere de processer, som vi har i systemet.

Use case beskrivelse:

Dette er en oversigt over funktioner som brugeren kan udføre i systemet, som er med til at give brugeren et klart billede af hvad de kan.

Use case diagram:

Vi bruger "use case diagrammet" til at illustrerer aktørernes/brugernes forhold til use casene.

Analyse

I dette afsnit har vi valgt at benytte analysemodeller, der vil give indblik i forskellige aspekter af projektet. Vi vil beskrive og gennemgå modellerne, så det kan tydeliggøres over for kunde og projektleder, hvordan vores fremgangsmåde og forståelse til udviklingen af systemet er forekommet.

Aktør model

Nedenfor har vi opstillet en liste af vores aktører, hvor de er blevet inddelt, om de har store indflydelse eller lille indflydelse på projektet, og om de bliver påvirket af projektet.

	Lille indflydelse på projektet	Stor indflydelse på projektet
Bliver påvirket af projektet	Spillere	Kunder
Bliver ikke påvirket af projektet		Udvikler

Figur 1 ; Aktør model

Use case oversigt / diagram

Tast spillere navne:

Spiller 1 skriver først sit navn på keyboardet, derefter skal spiller 2 skrive sit navn på keyboardet.

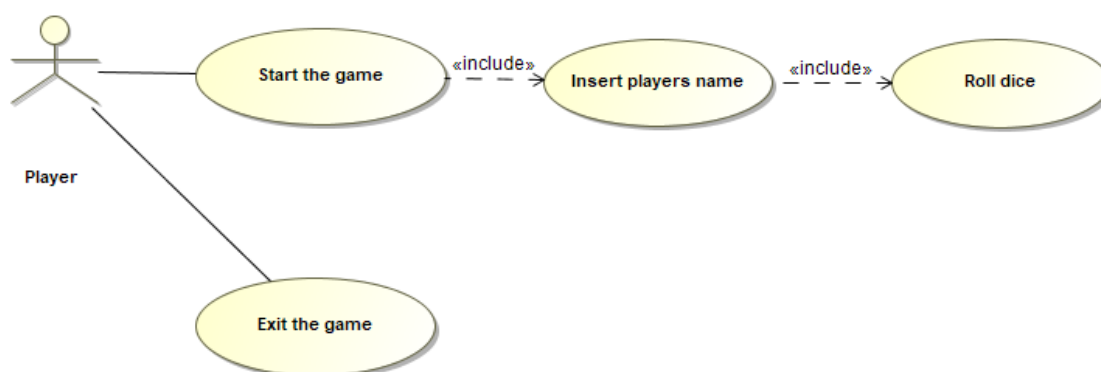
Kast terninger:

Spiller 1 starter med at trykke enter, hvorefter terningerne bliver kastet. Terningernes værdier bliver lagt sammen til en sum, som derefter bliver gemt til spillerens point.

Spiller 2 trykker derefter enter og samme proces gentages.

Use case-diagram

Her ses en use case, hvor der kan ses forholdet mellem aktør, som er i dette tilfælde spilleren, og dens forhold til use cases. For eksempel har *Player* relationer til *Start the game* og *Exit the game*, hvor *Start the game* inkluderer *Insert players name* og *Roll dice*.



Figur 2; Use case diagram

Domain model

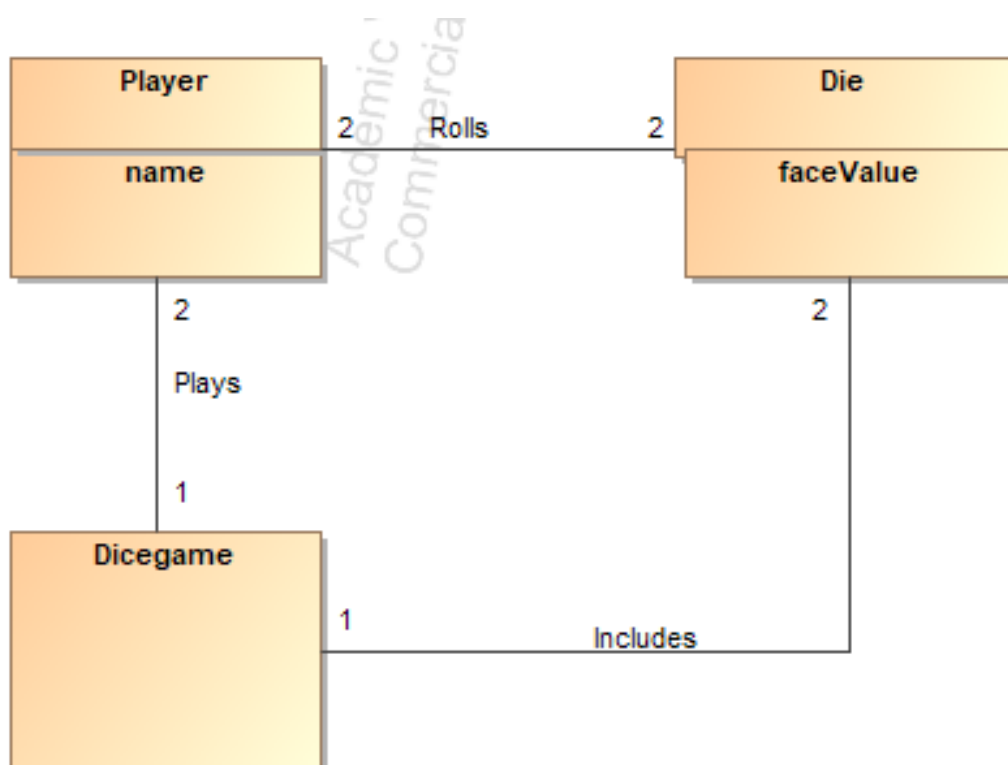
Nedenfor ses en domain model, der viser objekter fra vores dice game og deres attributer. Mellem objekterne vises hvad deres tilhørsforhold er til hinanden.

Modellen viser at et dice game der kan have 2 players/spiller tilknyttet.

Objektet player har attributten name samt point. Disse er henholdsvis en String og en int.

1 spiller kan bruge 2 terninger.

Selve dice game'et indeholder 2 terninger, der hver kan vise en face value mellem 1 og 6.



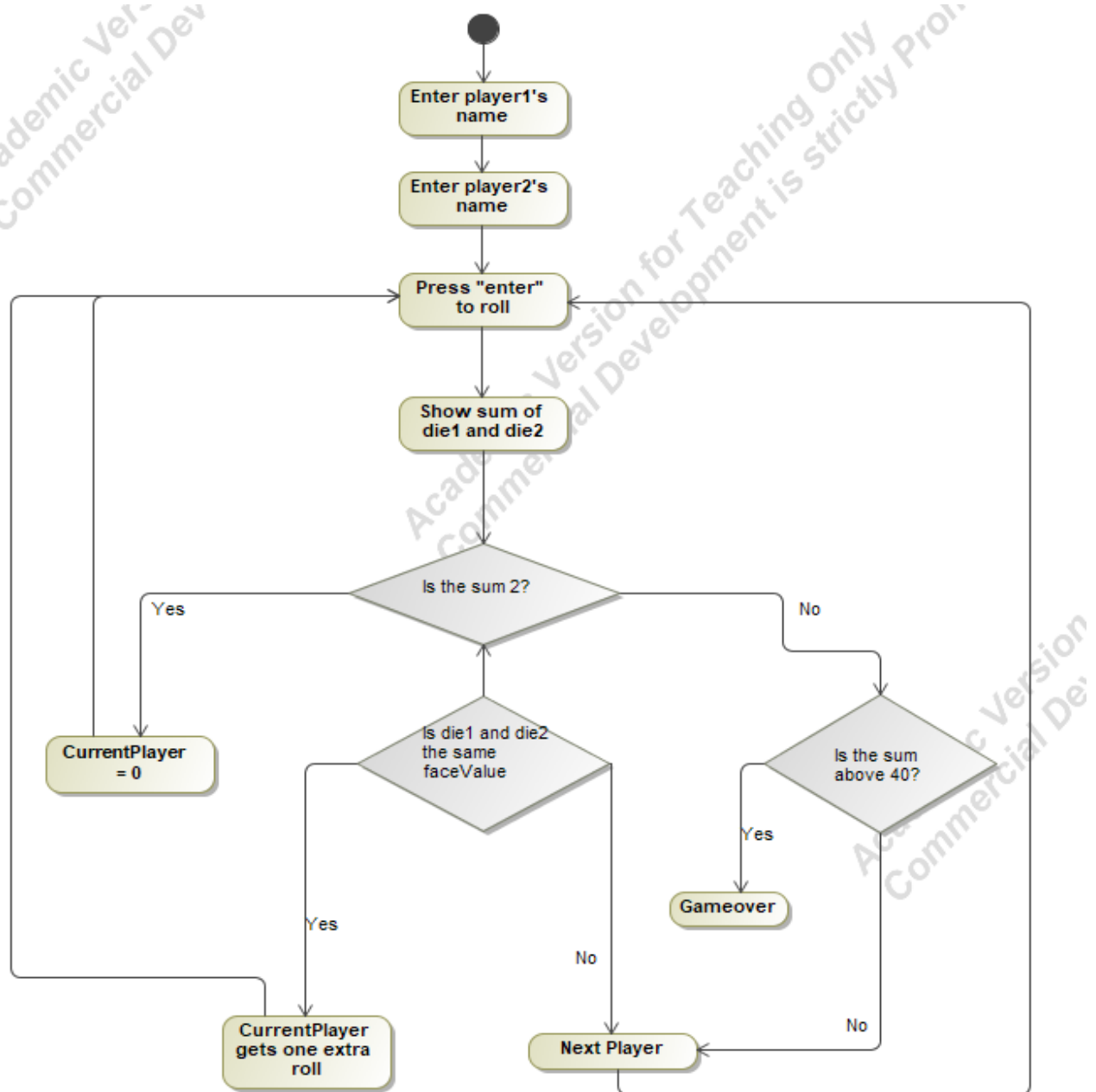
Figur 3; Domain model

Flowchart

Nedenfor ses flowchart over vores programmet, hvor det kan aflæses hvordan systemet kommer til at virke, og dens processer.

Der kan ses, systemet starter med indtastning af spillernavne, hvorefter systemet vil bede en spiller om at trykke på "enter", for at slå terningerne. Hver gang terningerne bliver kastet, så vil systemet tjekke følgende ting: er summen er 2, har man slået to tal med samme "face value", eller om en af spillers point er over 40. Hvis summen er 2, vil "currentPlayer" miste alle sine pointer og få en ekstra tur.

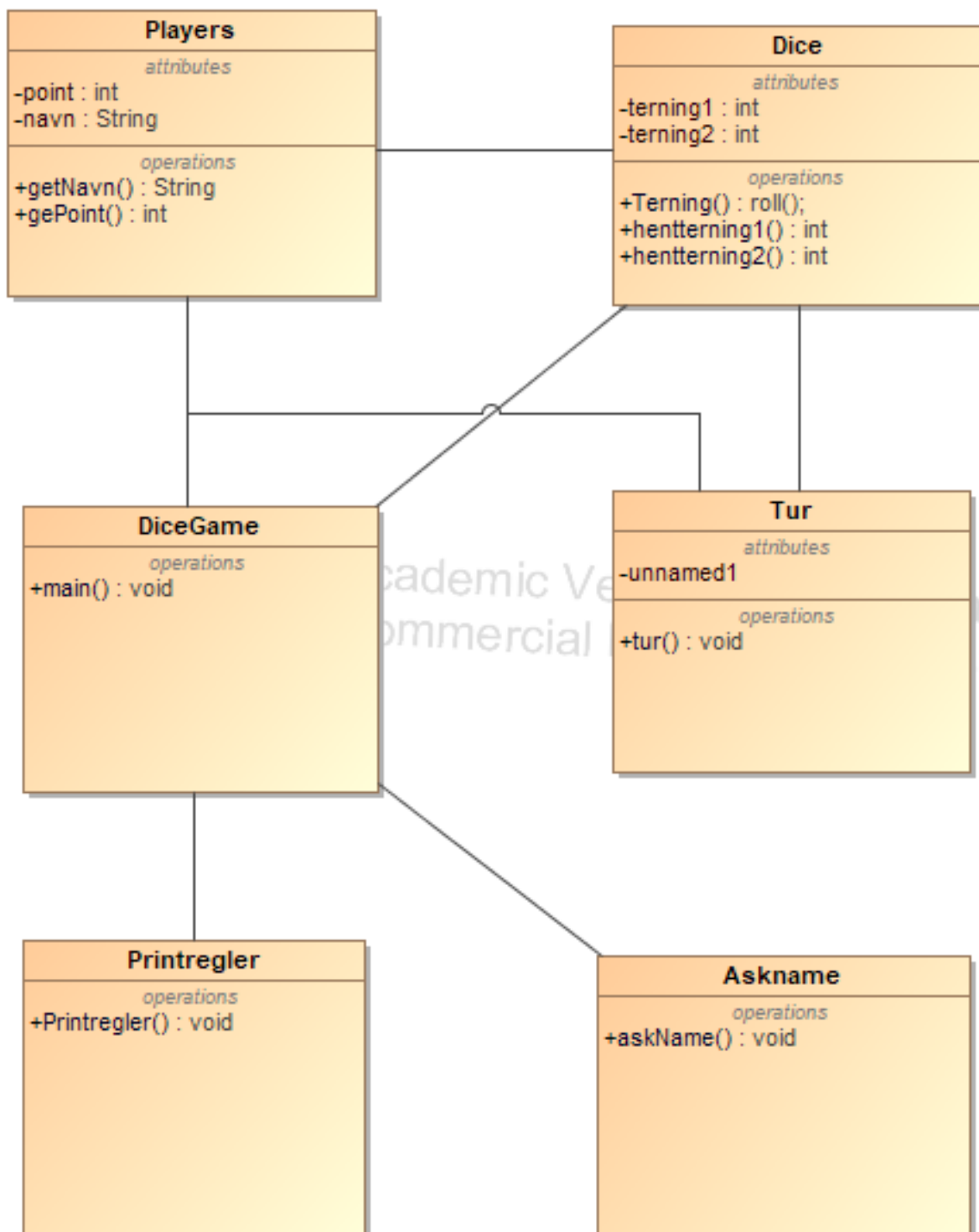
Hvis man slår to af tal af samme value, vil "currentPlayer" få en ekstra tur, og når en af spiller opnå 40 point, slutter spillet.



Figur 4; Flowchart diagram

Design

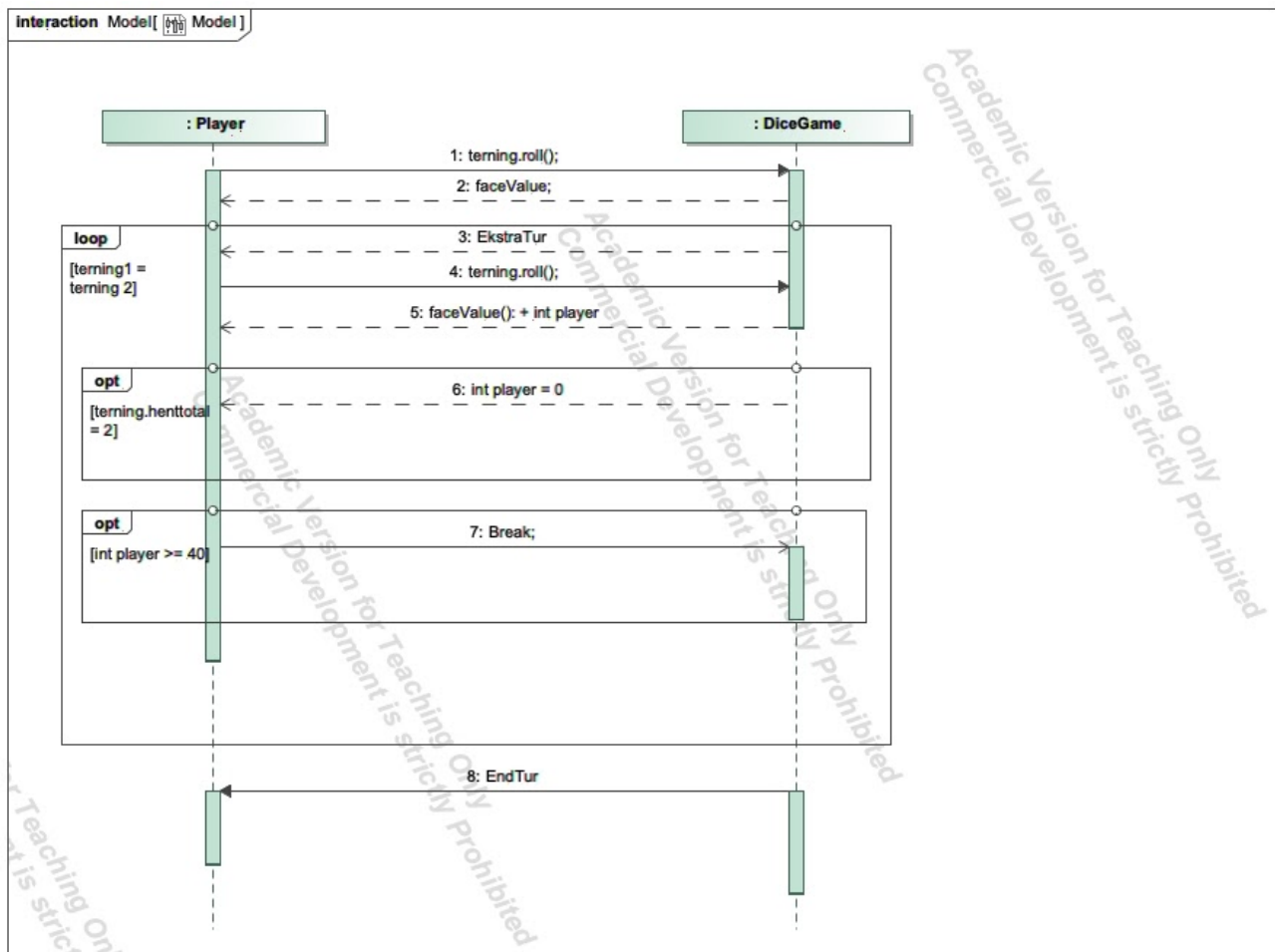
Nedenfor ses et domain software model, som tager udgangspunkt i vores kodning i IntelliJ. Dette gør at vi som programmører får et overblik over hvordan softwaren virker. Hver del er delt op i en Class, Attributes og Operations. Der er fx blevet oprettet nogle attributter i vores class Spiller, som "point" og "navn", hvor point skal være et heltal, derfor har vi brugt "int" til det, hvorimod "navn" har vi brugt "String", da den bruges til tekst.



Figur 5; Domain software diagram

Der er blevet benyttet et sequence diagram:

Her ses en billedlig afbildning af hvordan en tur bliver kaldt og hvad der sker når koden bliver initialiseret.



Figur 6; Sequence diagram

Test

Kunden anmodede om at den udviklede kode skulle testes. Dette skulle gøres ved at tjekke terningernes tilfældighed over 1.000.

Til forklaringen henvises til Terningtest.java under src i det færdige projekt hvor kunden også selv kan foretage testen.

For at gøre testen så nem som mulig har vi oprettet en ny main string og en ny pakke hvor vi kun har importeret den del af projektet der skal til for at kunne teste terningerne. Der ses derfor at der er importeret terningerne fra vores Terning.java class.

Dernæst skaber vi en række int som har til formål at tælle antallet af forekomster ved at benytte vores terning.roll funktion.

Dernæst skaber vi en while-løkke som relaterer sig til vores tæller int t som betyder at denne kode vil blive gentaget 1000 gange.

Dernæst benyttes en print line kommando til at vise resultatet af testen.

Resultat

Resultatet af testen viser at vores terninger agerer som forventet, ved at følge den forventede normalfordeling med mindre udsving som man kan forvente fra terninger.

For at foretage testen selv, skal du blot åbne Terningtest.java og klikke på den grønne pil på linje 5 og derefter klikke på "Run "Terningtest.main()".

Implementering

En guide til når man modtager filen og vil køre den:

Når man modtager zipfilen downloader man den til et sted hvor man kan finde den igen. Så går man ind på IntelliJ's startside (hvor man kan create new project eller import project osv). Så finder du din zip fil i stifinder og lægger den ud på fx skrivebordet (så den bliver pakket ud). Derefter laver du et "drag and drop" ved at venstreklikke og holde den inde på mappen du har lagt på skrivebordet, og så trækker du den over i IntelliJ's startvindue. Derefter burde programmet automatisk lægges ind og så skal du bare åbne klasserne.

For at åbne klasserne skal du trykke på pilen ved siden af mappen med navnet CDIO1, så klikke på pilen ved siden af mappen Terningsprogram og så på pilen ved siden af mappen src. Herfra kan du se pdf filen hvor vores rapport ligger. Hvis du så vil videre til programmeringsklasserne skal du trykke på pilen ved siden af com.company. Derfra burde du kunne se alle klasserne og så dobbeltklikker du for at åbne dem.

For at køre programmet skal du så åbne den klasse der hedder Main og trykke på den grønne pil der står ud for public class Main på linje 3.

Her kan du støde ind i et problem hvor den siger at du bruger en for gammel version af SDK/JDK. Hvis det er problemet skal du trykke på File->Project Structure og så tjekker du om du har en version som er 12 eller højere oppe i højre hjørne lige under et + og et -. Hvis ikke skal du gå på google og søge "java jdk" og gå ind på det første link og downloade en nyere JDK der. Så går du ind på Project structure igen og trykker på + og finder din downloadede JDK og apply den så fjerner du din gamle version så kun den nye står tilbage. Så trykker du apply.

Næste gang du så kører programmet siger den nok stadig fejl og der kommer en meddelelse oppe i toppen hvor den beder dig opdatere. Tryk på den, lad den opdatere og så burde det hele virke.

Beskrivelse af vores klasser

Vi har oprettet klasser til at kategorisere vores kode. Klasserne er også oprettet for at holde vores main til et minimums antal kodet linjer. Vi har i alt oprettet fem klasser, som er "AskName", "PrintRegler", "Tur", "Spiller" og "Terning".

"Tur-klasse" som gøre det nemt at holde en god rækkefølge og orden i spillets struktur

"Terning" som spiller en afgørende rolle for spillernes point og for hvilken spiller der først gennemfører spillet med højst antal point.

Klassen "Spiller" som vi definerer i starten af vores program, der angives 2 spillere, hvor hver bruger af programmet får den mulighed for at tilføje sit eget navn.

derefter skaber vi "Print-regler" klassen, til nemt at guide brugeren igennem spillet uden behov for brugsanvisning.

Vi benytter også klassen "AskName" til brugerens mulighed for at indtaste sit navn.

I "main" klassen printer vi vores regler i en rækkefølge efterfulgt af "løkker" og pauser.

Konklusion

I denne rapport har vi løst opgaven, stillet af spillefirmaet IOOuterActive, om at udvikle og designe et terningspil. Vi har sikret at kundens krav er efterkommet, ved at eftertjekke at spillet kan køres på windows maskinerne i Databaren på DTU og at det kan spilles af almindelige mennesker, uden forudgående kendskab til spillet. De øvrige af kundens krav, er også efterkommet. Vi har opstillet nogle forhindringer i spillet, for at øge sværhedsgraden en anelse. Bl.a har vi indført, at man mister alle sine point, hvis man slår to 1'ere. Derudover får man en ekstra tur, hvis man slår to ens tal (dette gælder også hvis man slår to 1'ere).

Vi har desuden prøvet, så vidt muligt, at imødekomme kundens visioner, men der er visse visioner, der ikke har været mulighed for at efterleve, grundet tidspres og manglende ressourcer.

Time regnskab

Navn	Antal arbejdes timer
Isabel	17 timer
Oliver	17 timer
Andreas	17 timer
Ahmad	17 timer
Alexander	17 timer
Chenxi	17 timer

Litteratur- og kildeliste

- <https://www.w3schools.com>
- Bogen "projekter & rapporter på tekniske uddannelser" af Samuel Brüning Larsen.