



JANUARPROJEKT

02121 INTRODUKTION TIL SOFTWARETEKNOLOGI

Afleveringsgruppe 16:

Abinav Aleti s224786
Yahya Alwan s224739
Aslan Behbahani s224819
Rasmus Wiuff s163977
github.com/rwiuff/Reversi 

20. januar 2023

INDHOLD

	Side
1 Introduktion & Problemanalyse	1
1.1 Arbejdsfordeling	1
1.2 Opgaven	1
1.3 Specifikationer	1
1.4 Målsætninger	1
1.5 Analyse af problemer	1
2 Design	2
3 Implementation	2
4 Test	2
5 Project Management	2
6 Konklusion	2
Listings	3
Figurer	4
Tabeller	4
Appendicer	5
A Arbejdsfordeling	5

1 INTRODUKTION & PROBLEMANALYSE

1.1. ARBEJDSFORDELING

Arbejdsfordelingen under projektet kan ses i Appendiks [A](#).

1.2. OPGAVEN

I dette projekt har vi skulle skrive et program, der starter- og kører spillet "Reversi", et brætspil der handler om kontrollere et bræt med flest mulige brikker. Programmet skal være skrevet i JavaFX. Problemet består i at, via, diverse datastrukturer, at skulle beskrive et bræt, brikkerne såvel som deres egenskaber og opførsel. Der har skulle være en grænseflade, der giver udtryk for alt det visuelle, såvel som logikken, der binder de forskellige elementer sammen. Vi har som hjælp til at designe programmet benyttet os af Model-View-Control-strukturen. Vi har skrevet modellen, som er logikken bag alle reaktioner, programmet kan have til diverse handlinger, vi har skrevet view, som er det grafiske element, der bestemmer hvordan programmet skal se ud, former, farver mm. Til sidst har vi skrevet control, som har skulle specificere, hvilket handlinger, der skal reagere på.

Et væsentligt problem, ved implementering af logik og det visuelle har været, at hvis vi har villet teste den ene, har det krævet den fuldstændige version af den anden. Hvis den del af gruppen, der har udviklet logikken, har villet teste om det virker, har de måtte udvikle en midlertidig løsning i form af en alternativ visualisering, da det grafiske stadig har været under udvikling.

1.3. SPECIFIKATIONER

Reversi er et relativt simpelt spil med få regler. Vi ved, at vi skal lavet et bræt med 8x8 felter. I begyndelsen af spillet skal der placeres 4 brikker af to forskellige farver, én farve pr. spiller. Herefter skal de to spillere placere deres brikker én efter én, og vende modstanderens brikker ved at klemme dem inde mellem ensfarvede brikker. Brikkerne kan kun stilles et sted, hvor man med sikkerhed klemmer en brik af modsat farve inde. Vinderen er den spiller, der har flest brikker på brættet, når alle 64 felter er udfyldt. Alternativt kan man vinde spillet, ved at modstanderen siger "pas" to ture i træk. Ud fra disse informationer skal vi lave spillet. Vi har haft artistisk frihed ift. hvilken farve brikkerne og brættet skulle have, hvor stort brættet skal være ift. computerskærmen og hvordan layout'et skulle se ud mht. placering af eventuelle knapper og tekstområde ift. selve brættet.

1.4. MÅLSÆTNINGER

Vores mål i dette projekt har været, at vi først og fremmest vil designe spillet i sin basisform, uden fejl og mangler. Spillet har skulle kunne køre, og været testet adskillige gange for robusthed og eventuelle små fejl. Dette har været vores minimumskrav, for at kunne kalde projektet "bestået". Vi ville gerne skrive et program, som var let læseligt, overskueligt, og så simpelt som muligt i struktur, uden unødvendig redundans. Vi har sigtet efter en simpel struktur, således at det er nemt at redigere i koden, hvis vi skulle ønske evt. ændringer i programmet undervejs, da dette kan blive en stor udfordring, skulle man have skrevet noget sammenfiltret og kompliceret. Derudover har vi villet gøre spillet så avanceret, som vi har kunnet. Dette betyder at vi har implementeret de valgfri tilføjelse, som vi idømte inde for vores faglige kapacitet. Igen hjælper det her med simpel og overskuelig kode, der gør, at man let kan genbruge noget, man tidligere har skrevet, til at udvide og forbedre det. Et andet vigtigt mål har været at arbejde som en gruppe. Med det menes der at lære, hvorledes man arbejder sammen, dele arbejdet op når det kommer til et projekt af denne størrelse, og tage stilling til hvordan at det kode, man er ved at skrive, kommer til at arbejde sammen med de andre gruppemedlemmers kode.

1.5. ANALYSE AF PROBLEMER

Projektet har budt på adskillige overvejelser ift. løsning af problemer. Som det første havde vi det visuelle, og dernæst havde vi det logiske. Vi blev enige om at dele arbejdet op i grupper af to og to. Så kunne vi spørge os selv: Hvordan vil vi gribe vores opgave an? Hvad angår den grafiske del, besluttede vi os for at bruge scene builder til at lave vores "stage", "Scene" og nogle af vores "nodes". Et alternativ til dette kunne være at skrive det hele i JavaFX. Dette ville have både fordele og ulemper. På den ene side vil det være en fordel, at være uafhængig af et fremmede program, og have al grafisk info stående på en let overskuelig måde i vores klasser. Ulempen vil dog være, at det vil tage længere tid at skrive hver enkelt ting op. Fordelen vil scene builder er, at det er hurtigt, overskueligt (inde i selve programmet), og det er let at forbinde evt. kode fra andre klasser med det, man har designet i programmet.

Dog var det ikke alt, der kunne løses i scene builder. Vi skulle også have implementeret nogle metoder, der reagerede på forskellige hændelser. Det kunne f.eks. være en metode, der kunne tilføje cirkler via. museklik, og dette kunne ikke gøres i scene builder. Dette måtte vi skrive ind i JavaFX, hvorefter vi forbandt metoden med et element.

2 DESIGN

For at løse opgaven, delte

3 IMPLEMENTATION

4 TEST

5 PROJECT MANAGEMENT

6 KONKLUSION

LISTINGS

FIGURER

TABELLER

1	Arbejdsfordeling i projektet	5
---	--	---

Appendices

A ARBEJDSFORDELING

Tabel 1: Arbejdsfordeling i projektet

Navn	Klasser/metoder	Referencer i rapporten
Rasmus Wiuff	Board.moveAnalyser(int colour) Board.getValidMoves()	