

02121 Introduktion til Softwareteknologi - Projektplan

Anders Reher - s194587
Anders Gad - s204496
Magnus Siegumfeldt - s204472
Kevin Moore - s204462

08/01-2021

Problemanalyse

Vi skal i dette projekt udarbejde to varianter af det kendte Minesweeper spil, hvor spilleren spiller mod sig selv (ofte på tid) på en firkantet bane. Spilleren forsøger at undvige en række forudplacerede miner, ved hjælp af numrene på spillepladen.

I projektet skal vi som nævnt ovenfor udarbejde et basalt og et avanceret Minesweeper spil, hvor det basale spil, blot indeholder de mest nødvendige funktioner. Det avancerede spil derimod skal indeholde en række ekstra 'features' og funktioner, som beskrives nærmere i afsnittet 'Beskrivelse af delopgaver'.

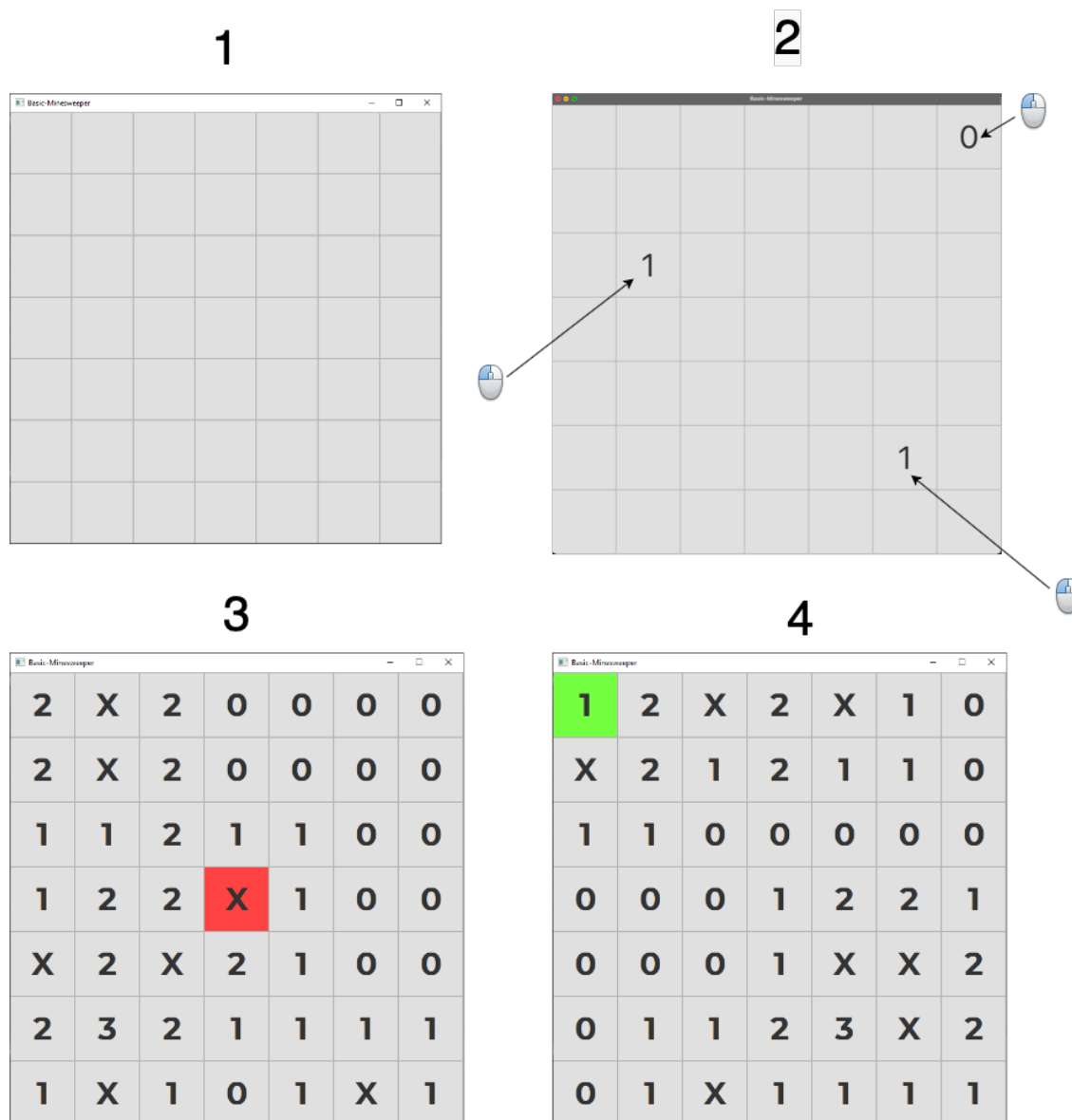
Da vi skal udvikle et større spil, med en række ekstra funktioner, er det nødvendigt at strukturen af programmet overvejes og designes, med udvidelsesmuligheder, og skalerbarhed i fokus.

Vi har tænkt os at opbygge vores spil vha. UI-biblioteket JavaFX, og benytte Model-View-Controller konceptet (MVC), til at strukturere vores program.

Vi forventer undervejs, at støde på en række udfordringer, af forskellige varianter. Nogle udfordringer vil ligge på det tekniske niveau, hvorimod andre udfordringer kan ligge på mere overordnede områder. Vi er undervejs, meget opmærksomme på at struktur af program, og samspillet mellem de enkelte dele, fungerer 100% efter planen, da vi hver især vil sidde med forskellige dele af koden. Derfor er det også vigtigt at vi har præcise beskrivelser af de enkelte dele, samt grænsefladerne imellem dem, så der ingen misforståelser sker.

Beskrivelse af pladen

Vi har tegnet en skitse over hvordan den basale version af spillet kommer til at fungere. Den primære måde brugeren interagerer med programmet på bliver ved hjælp af musen.



Skitse 1 Pladen er repræsenteret af en n gange m stor firkant, hvor hvert kvadrat på pladen er et felt brugeren kan interagere med. Når spillet starter ville værdierne være "gemt".

Skitse 2 Herfra vil brugeren have mulighed for at se hvilken værdi der ligger bagved ved at venstreklikke på dem. Værdierne der er bestemt af hvor mange bomber der er i de omkringliggende felter, således at hvis der er 2 miner så vil værdien være 2.

Skitse 3 og 4 Klikker brugeren på en mine viser alle felterne i spillet sig og feltet klikket på blive rødt. Man vinder spillet ved at have trykket på alle ikke-mine felterne hvor det sidste felt trykket på derfra bliver grønt.

Beskrivelse af delopgaver

Til at starte med har vi lavet en overordnet strukturering af programmet, hvor det er bestemt hvilke klasser programmet skal indhold, og hvilke metoder der skal kunne kaldes fra disse klasser. Der laves først en basal version af programmet, som overholder følgende krav.

Krav til det basale spil

- Kunne starte programmet fra terminalen og samtidig bestemme en n gange m bane-størrelse og antallet af bomber, x , hvor $4 \leq n, m \leq 100$ og $5 \leq x \leq 2000$.
- Have mulighed for at klikke på felter med en reaktion fra programmet. (bombe, tal og tomt felt)

Vi har til det avancerede spil, opstillet følgende prioriteringer af de brainstormede udvidelser i gruppen. Vi vil dog se på disse prioriteringer med en hvis fleksibilitet og ændringer i disse vil formentlig ske undervejs som programmet udvikles.

Krav til det avancerede spil

1. Prioritet

- Generer banen efter første tryk
- Brugermenu som spillet starter med hvor man kan ændre på indstillingerne og trykke start for at starte spillet.
- Flag til at markere hvor man tror minerne er. Dette sker når brugeren højreklikker.
- Figur
- At kunne genstarte spillet efter slut.

2. Prioritet

- Tilføjelse til indstillinger: Forud defineret sværhedsgrad som brugeren kan vælge imellem.
- Timer og tidsfrist på spillet for at tilføje en ekstra sværhed.

- Lave værdier for beregning af High Score for at kunne konkurrere med andre som kan gemmes. Alle indstillinger som f.eks. tidsfrist, banestørrelse og sværhedsgrad som gør spillet sværere giver en bonus til High Scoren.

3. Prioritet

- En fortryd knap i tilfælde af man taber spillet og vil et træk tilbage i spillet.

4. Prioritet

- Gem/hent en bane samt indlæsning af bane gennem fil

5. Prioritet

- En algoritme der kan løse en vilkårlig bane
- Lydeffekter

Tidsplan

Prioriteringen af udvidelserne er vist med tal fra 1 til 5, hvor 1 viser de udvidelser, som vil blive implementeret først, mens 5 viser de udvidelser, som evt. laves, hvis det er muligt inden for tidsrammen. Ellers følger tidsplanen af det vist billed.

Tidsplan:

Arbejdsopgaver	Dato
Påbegyndelse af rapporten: <i>skriv rapport til basic-versionen af spillet</i>	08/01
Færdig med basic-versionen af spillet	08/01
Arbejde med at udvide spillet begynder: <i>Startende med de første prioriteter</i>	11/01
Færdig med de første prioriteter af spillet, og videre arbejde med prioriteter 2 og 3.	13/01
Arbejde med spillets UI og prioriteter 4 og 5, hvis det tidsmæssigt er muligt.	15/01
Simultan med kodeskrivningen, arbejde på rapporten i den rækkefølge som giver mening.	08/01-15/01
Udfører opgaver, som mangler i rapport eller program	18/01-20/01
Færdig med spil og rapport	20/01
Små rettelser og buffer i tidsplanen	21/01-22/01