

# Introduktion til Ingeinørarbejde – Efterår 2020

## Projekt 2020/21

**Dato:** 04. 01. 2021, **Slut dato:** 22. 01. 2021, 23:59

### Projektopgave: Minestryger

## Om opgaven

Projektopgaven er obligatorisk og indgår i kursets samlede bedømmelse. Opgaven skal laves i grupper af **4 personer**. I skal såvidt muligt selv danne grupperne. Grupperne skal **registreres** på DTULearn in starten af 3-ugers-perioden. Opgaven går overordnet ud på at designe og dokumentere et spil baseret på det klassiske Minestryger computerspil. Se Figur 1. Spillet skal implementeres i **Java** ved brug af **JavaFX**, rapporten skal være i **LaTeX**. Detaljerede beskrivelser af spillet, varianter, historie, strategier, etc kan, bl.a., findes på hjemmesiderne

[https://en.wikipedia.org/wiki/Minesweeper\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Minesweeper_(video_game))

<http://www.minesweeper.info/>.

Som udgangspunkt for opgaven anvender vi en simpel version af minestryger som foregår på følgende måde. Spilleren præsenteres for et  $n \times m$  rektangulært gitter af felter,  $4 \leq n, m \leq 100$ . Et tilfældigt udvalg af disse felter gemmer på miner. Resten af felterne er tomme. Til at begynde med er indholdet af alle felter gemt for spilleren. Størrelsen af gitteret og fordelingen af miner er afgørende for spillets sværhedsgrad. Spilleren skal nu vælge felter et ad gangen ved at klikke på dem med musen. For hvert felt, som spilleren klikker på, er der to tilfælde. Hvis feltet gemmer på en mine er spillet tabt. Ellers vises antallet af miner i de højst 8 nabofelter som et tal i feltet (se Figur 1). Spilleren vinder hvis han/hun formår at klikke på alle de tomme felter uden at klikke på et felt med en mine.

Opgaven består af en indledende del og en avanceret del. Alle grupper skal først lave den indledende del. Efter dette har I et simpelt Minestrygerspil. Efterfølgende kan I gå videre til den avancerede del. Her kan I fortsætte i mange forskellige retninger afhængig af interesse og kompetencer. En ikke-udtømmende liste af muligheder er givet nedenfor.

## Indledende del

I den indledende del af opgaven skal konstrueres et simpelt minestrygerspil som beskrevet ovenfor. Interaktionen med spilleren skal foregå via en simpel grafisk brugergrænseflade (f. eks. som vist på Figur 1). Størrelsen af gitteret og antallet af miner skal angives som

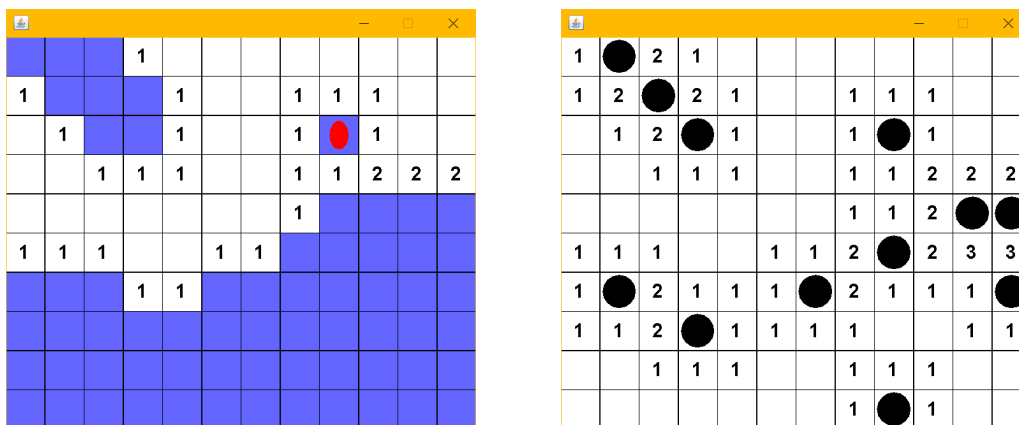


Figure 1: Eksempel på minestryger brugergrænseflade. Til venstre ses et igangværende spil med udvidelserne flag og blankfeltsløser. Til højre ses spillet når det er afsluttet.

kommandolinieparametre (som argumenter i `main`). Minerne skal fordeles tilfældigt i gitteret.

## Avanceret del

I den avancerede del bygger man videre på designet fra den indledende del. Man kan frit udvide og variere sit spil. En liste af mulige udvidelser og variationer i tilfældig rækkefølge er givet her. Nogle udvidelser er krævende.

**Flag** Tilføj flag og/eller spørgsmåltegn til spillet (se forklaringer på ovenstående hjemmesider).

**Antal** Tilføj en tæller, der viser hvor mange miner er blevet fundet og/eller der er tilbage.

**Tid** Tilføj et stopur til spillet.

**Hints** Tilføj en mekanisme som spilleren kan bruge til bede om at få hints til spillet.

**Blanktfeltsløser** Vend automatisk felter der er forbundne med det valgte felt og ikke har nogle miner som nabo.

**AI** Design en computerspiller for spillet.

**Geometri** Lav en variant baseret på ikke-firkantede felter (f. eks. trekanten, hexagoner, etc.)

**Dimension** Lav en 3D-variant.

**Startfelt** Design spillet så spilleren aldrig kan tabe på første klik.

**Sværhedsgrad** Tilføj mulighed for at variere sværhedsgraden af spillet. F. eks. ved at variere fordelingen af miner (se f. eks. 3BV-målet på hjemmesiderne).

**High score** Tilføj en liste over de bedste præstationer i spillet.

**Spilindstillinger** Tilføj indstillinger i brugerfladen så man nemt kan specificere størrelse af gitter, sværhedsgrad, etc.

**Multiplayer** Design en flerpersoners variant af spillet.

**Undo** Tilføj mulighed for at slette sidste træk.

**Gem/Hent** Tilføj funktion til at gemme og hente spil.

**Andet** Find på din egen variant/udvidelse af spillet.

## Aflevering

I skal aflevere en rapport, der dokumenterer de udviklede spilvarianter og de overvejelser I har gjort jer omkring dem. Rapporten skal skrives i LATEX. I skal også aflevere kildeteksten til spillet. Følgende filer skal uploades via DTULearn:

- En pdf-fil med jeres rapport. Rapporten skal være individualiseret, dvs., I skal kendetegne jeres individuelle bidrag. Filnavnet skal være `Gruppe<NR>-rapport.pdf`.
- En eksekverbar JAR-fil af den indledende version, der kan køre på Windows 10 med Java15+FX15, samt en kort beskrivelse af, hvordan man starter programmet (f.eks. hvilke argumenter der skal angives). Filnavnet skal være `Gruppe<NR>-simpel.jar`.
- En eksekverbar JAR-fil af den avancerede version, der kan køre på Windows 10 med Java15+FX15, samt en kort beskrivelse af, hvordan man starter programmet (f.eks. om der skal angives argumenter eller om de kan sættes i GUI-en). Filnavnet skal være `Gruppe<NR>-avanceret.jar`.
- JAR-filerne skal også indeholde Java-kildeteksten, dvs. `*.java`-filer. Der kommer en vejledning om JAR-filer.

Den endelige karakter gives på baggrund af en helhedsvurdering af jeres tidligere afleveringsopgaver og projektopgaven. I vurderingen lægges vægt på følgende.

- Analysen af problemet.
- Kvalitet af dokumenterede overvejelser og valg i forbindelse med projektet (afgrænsning, design, implementation, evalueringen, etc.).
- Teknisk kvalitet af programmet (korrekthed, robusthed, funktionalitet, effektivitet, etc.).
- Programmets struktur. Klasseopdeling, programmeringsparadigmer (f.eks. MVC).
- Kvalitet af beskrivelse af programmet (virkemåde, opbygning, etc.).
- Kvalitet af projektrapport (præcision, struktur, etc.).
- Kvalitet af brugergrænseflade.
- Kvalitet af tidligere obligatoriske opgaver.

## Frister

- På fredag, 8.1., **kl 12:00** afleveres en projektplan på max 2 sider, der indeholder følgende:
  - En problemanalyse, som indeholder udfordringer og hvordan I har tænkt jer at løse dem.
  - En grov beskrivelse af repræsentationen af problemet, især “bordet”.
  - En grov beskrivelse af delopgaver/arbejdspakker og en foreløbig tidsplan.

Feedback gives senest tirsdag 12.1.

- Den 10.1.2021, afleveres Studieplanen.
- Opgaven skal afleveres senest fredag d. 22.01.2021, kl. 23:59 på DTULearn. Karakteren forventes cirka 2 uger senere.

## Noter, uddybning og hints

Der bliver lagt materiale om rapportskrivning på DTULearn i starten af 3-ugersperioden. Nogle råd i forvejen:

- Brug god tid på at tænke den overordnede struktur af jeres program igennem inden i begynder implementationen.
- Noter alle overvejelser, argumenter, resultater, alternativer, etc. undervejs i processen. (Hvorfor har vi valgt denne løsning/værktøj og ikke en alternativ).
- Tag udgangspunkt i Model-View-Controller paradigmet når I skal designe og implementere programmet. Beskriv jeres implementation i rapporten i forhold til Model-View-Controller paradigmet.
- Lav den indledende del færdig og skriv rapporten til den, inden I kaster jer over de videregående ting
- Implementer og dokumenter hver enkelt videregående udvidelse/variant hver for sig, så I ender med en række gradvist mere avancerede minestrygerspil I kan sammenligne med hinanden.
- Brug figurer og metakommunikation i rapporten.
- Skriv kort og præcist.
- Undgå knudrede sætninger og fyldord.
- Det vigtigste i opgaven er ikke de færdige spil, men de overvejelser og analyser laver undervejs. Sørg for at få dem alle med i rapporten. I skal vise, at I har en ingeniørmæssig tilgang til udviklingsprocessen.