

Structuur van Computerprogramma's 2

Academisch jaar 2025-2026

Examenvereiste Praktijkopdracht - Pre-project

Minesweeper

Elisa Gonzalez Boix Maarten Vandercammen Robbe De Greef

1 Inleiding

Minesweeper¹ is een spel waarbij een speler in een twee-dimensionaal veld verborgen mijnen moet vinden zonder een van deze mijnen te doen ontploffen. Om dat te doen kan de speler, als die denkt dat een bepaald vakje geen mijn bevat, een vakje *onthullen*, bv. door erop te klikken. Als het vakje inderdaad geen mijn bevatte, toont het vakje vanaf nu een getal dat aangeeft hoeveel van de (maximum) acht naburige vakjes mijnen bevatten. Als er zich wel een mijn bevond op die positie, is het spel gedaan en heeft de speler verloren. De speler wint door ofwel alle vakken die geen mijnen bevatten te onthullen, ofwel door een vlag te plaatsen bovenop alle verborgen mijnen. Het aantal vlaggen dat kan gebruikt worden is echter beperkt tot het aantal mijnen dat verborgen zijn in het veld. De afbeelding hieronder toont een voorbeeld van hoe Minesweeper gespeeld wordt: vakjes die geen naburige mijnen hebben werden grijs gekleurd.



Figuur 1: Een screenshot van *Microsoft Minesweeper*.

¹[https://en.wikipedia.org/wiki/Minesweeper_\(video_game\)](https://en.wikipedia.org/wiki/Minesweeper_(video_game))

2 Opdracht

In deze opdracht implementeer je de volgende zaken:

- De applicatielogica van Minesweeper.
- Een grafische user interface (GUI) toe waarmee het spel kan voorgesteld worden.
- Het persistent opslaan van het speelveld.
- De mogelijkheid om het programma op te starten met zowel een manier om te starten vanaf een veld met arbitraire dimensies, als een manier om te vertrekken van een reeds opgeslagen speelveld.

In de volgende secties lichten we elk van deze features toe.

2.1 Applicatielogica

De applicatielogica kan als volgt samengevat worden:

- Initieel wordt er een tweedimensionaal speelveld aangemaakt van een bepaalde grootte (zie Sectie 2.4).
- De mijnen mogen nog niet meteen tijdens het aanmaken van het veld gemaakt worden, maar er moet gewacht worden tot de speler gevraagd heeft om het eerste vakje te onthullen. Op die manier vermijd je dat de speler meteen, in de eerste beurt, op een mijn stapt en het spel verliest. Als de speler dat gedaan heeft, worden alle mijnen op willekeurige posities geplaatst. Je kan de `rand` functie² gebruiken om willekeurige posities te genereren. Het aantal mijnen dat moet geplaatst worden hangt af van de manier waarop het spel werd opgestart (zie Sectie 2.4).
- Het spel verloopt in verschillende beurten totdat de speler het spel gewonnen heeft, door een vlag te plaatsen op alle mijnen of alle niet-mijnen te onthullen, of verloren, door een vak te onthullen waar zich een mijn bevond.
- Als je een vakje onthult waarvan geen enkel naburig vakje een mijn bevat, worden automatisch al deze naburige vakjes onthuld. Dit proces herhaalt zich ook voor elk van deze burens waarvan het aantal naburige vakjes met een mijn gelijk is aan nul. Kijk bv. naar welke vakjes onthuld werden in Figuur 1.
- Het maximum aantal vlaggen dat je kan plaatsen is gelijk aan het aantal mijnen dat verborgen zijn.

²<https://en.cppreference.com/w/c/numeric/random/rand>

2.2 GUI

Je maakt in deze opdracht een GUI waarmee het spel kan gespeeld worden. Het moet via de GUI mogelijk zijn om links te klikken om een vakje te onthullen, en rechts te klikken om ofwel een vlag te plaatsen ofwel om een bestaande vlag te verwijderen. Als de speler de **P** toets indrukt, moet het veld tijdelijk helemaal onthuld worden zodat de speler kan zien waar alle mijnen zich bevinden. Als de **P** toets daarna opnieuw wordt ingedrukt, moet de GUI terugkeren naar hoe het er normaal uit ziet. Als de speler de **S** toets indrukt, wordt het spel opgeslagen (zie Sectie 2.3).

Om de GUI te implementeren zal je gebruik maken van SDL2³. SDL2 is een verzameling van libraries die je toelaten om zowel simpele als complexe spelletjes te maken.

SDL2 Installatie

Op Linux kan je de benodigde libraries via de terminal installeren door eerst het volgende uit te voeren: **sudo apt-get update**

En vervolgens: **sudo apt-get install libsdl2-dev**

Op OS X kan je deze libraries downloaden door eerst Xcode via de App Store te installeren, als je die nog niet zou hebben (dit is gratis), daarna **brew** te installeren⁴ en daarna via de terminal het volgende uit te voeren: **brew install sdl2**

SDL2 Compilatie

Om SDL2 te gebruiken in je code moet je aan de compiler duidelijk maken waar de nodige SDL2 libraries en headerfiles zich bevinden. Als je je programma compileert via de console of via een makefile kan je dit doen door je compilatiecommando als volgt op te stellen:

```
gcc codebestanden `sdl2-config --cflags --libs` -o naam_programma
```

Let op dat je de `` backticks ook gebruikt in je compilatiecommando.

Als je aparte commando's wil gebruiken om je code te compileren en te linken, kan je je code compileren via:

```
gcc codebestand -c `sdl2-config --cflags` -o naam_objectbestand
```

en linken via:

```
gcc objectbestanden `sdl2-config --libs` -o naam_programma
```

Op Canvas bevindt zich een document dat aangeeft hoe je in Eclipse en CLion kan linken naar deze library.

Gebruik

Figuur 2 toont een voorbeeld van hoe de GUI er in SDL2 zou kunnen uitzien.

³<https://www.libsdl.org/>

⁴Zie hiervoor <http://brew.sh/>



Figuur 2: Een voorbeeld van hoe het speelveld met de GUI kan afgebeeld worden.

We geven je twee bestanden `GUI.c` en `GUI.h` (in de `GUI.zip`-folder op Canvas), alsook een folder met afbeeldingen, waarin een minimalistische GUI-applicatie geïmplementeerd is die de volgende zaken al afhandelt:

1. Opstarten en afsluiten van de GUI
2. Toetsindrukken en muisklikken opvangen
3. Een afbeelding tekenen

Op het internet kan je enkele tutorials vinden over hoe je SDL2 juist gebruikt, zoals <http://lazyfoo.net/tutorials/SDL/>

2.3 Speelveld opslaan

Verder breid je het spel ook uit zodat het hele speelveld wordt opgeslagen in een bepaald bestand als de speler de `S` knop heeft ingedrukt. Dit doe je door het veld te encoderen als een tekstbestand waar je alle relevante informatie in opslaat.

Het bestand zal zeker de grootte en breedte van het speelveld moeten bevatten. Daarnaast kan je er bv. voor kiezen om voor elk vakje in het speelveld alle relevante eigenschappen van dat vakje (bv. of er zich een vlag bevindt, of het vakje al onthuld is, eventueel het aantal burens dat mijnen zijn) te encoderen als ofwel één karakter, ofwel meerdere karakters. Een speelveld inladen komt dan neer op het tekstbestand uitlezen en alle informatie te decoderen.

2.4 Spel opstarten

Als laatste feature maak je het ook mogelijk om het programma op verschillende manieren op te starten. Afhankelijk van hoe je het programma via de terminal opstart moet het zowel mogelijk zijn om een opgeslagen speelveld in te laden en daarmee verder te gaan, of om gewoon te starten vanaf een nieuw veld met

arbitraire dimensies. Als je je programma bv. `game` hebt genoemd, moet het mogelijk zijn om via het volgende commando het speelveld dat opgeslagen is in `field.txt` in te laden: `./game -f field.txt`

Anderzijds kan je ook een nieuw speelveld starten met breedte 15 en hoogte 10 dat 20 mijnen bevat, door het volgende commando uit te voeren:

```
./game -w 15 -h 10 -m 20
```

Deze drie argumenten zouden in om het even welke volgorde moeten kunnen ingegeven worden.

Zoals je kan zien wordt de keuze tussen beide mogelijkheden bepaald door of je de `-w` `-h` `-m` argumenten of het `-f` argument meegeeft.

Merk op dat in geen van beide opties de dimensies van het speelveld statisch bepaald kunnen worden. In je code zal je dus sowieso moeten gebruik maken van dynamische allocatie om het speelveld de gevraagde dimensies te geven.

3 Praktische informatie

- De deadline voor deze taak is **vrijdag 12 december, 23.59h**.
- Alle code moet samengevoegd worden in de vorm van een zip-file met als naam `<preproject_achternaam.voornaam.zip>` en geüpload worden naar de opdrachtenfolder op de Canvas cursuspagina. Bijgevoegd in deze zip moet ook een, **zelfgemaakte** makefile zitten. Het moet dus mogelijk zijn om je hele programma te compileren door enkel de zip uit te pakken en de makefile uit te voeren via het commando `make`. Voeg aan de zip ook minstens één voorbeeldbestand toe dat een opgeslagen speelveld bevat.
- Deze opdracht dient strikt **individueel** en op eigen kracht gemaakt te worden. Dat betekent dat je jouw opdracht op een zelfstandige basis moet maken en je jouw werk moet kunnen uitleggen, onder toezicht moet kunnen herimplementeren, en je werkwijze moet kunnen verdedigen. Werk (code, tekst, etc.) overnemen van of delen met derden (bv. medestudenten, websites, GitHub, generatieve AI tools, etc.) is verboden. Elektronische hulpmiddelen worden gebruikt om alle inzendingen met online bronnen en met elkaar te vergelijken, zelfs over verschillende academiejaren heen. De enige code die mag overgenomen worden is de GUI code die wordt aangeboden op Canvas, of de code die je gezien hebt in het HOC en het WPO. Als je code overneemt uit de les, moet je de bron daarvan vermelden.
- Elke handeling van een student die afwijkt van de gegeven instructies en niet in overeenstemming is met het examenreglement wordt beschouwd als onregelmatigheid. Iedere vorm van fraude die een inbreuk is op de wetenschappelijke integriteit, waaronder het gebruik van statements of teksten geproduceerd door generatieve AI-toepassingen zonder vermelding van oorspronkelijke bronnen, of het simuleren of vervalsen van onderzoeksgegevens, zijn onregelmatigheden (cf. OER, Artikel 118§2). Plagiaat

is eveneens een onregelmatigheid. Onder plagiaat wordt begrepen het gebruik maken van andermans werk, al dan niet in bewerkte vorm, zonder nauwkeurige bronvermelding (cf. OER, Artikel 118§2). Plagiaat kan betrekking hebben op verschillende vormen van werk zoals tekst, beeld, muziek, databestand, structuur, gedachtegang, ideeëngoed.

- Elk vermoeden van plagiaat zal onverwijld aan de decaan van de faculteit worden gerapporteerd. Zowel gebruiker als verlener van zulke code zullen worden gerapporteerd en zullen worden behandeld volgens de plagiaatregels van het examenreglement (cf. OER, Artikel 118). De decaan kan beslissen tot (een combinatie van) de examentuchtsancties, gaande van 0/20 op het werkstuk tot een verbod tot (her)inschrijving voor één of meerdere academiejaren (cf. OER, Artikel 118§5).
- Contacteer ons als je twijfelt of iets al dan niet als plagiaat zou beschouwd worden.

4 Evaluatie

Deze opdracht telt mee voor 15% van je eindcijfer. Het indienen van de opdracht is **verplicht**. Studenten die geen opdracht indienen, krijgen automatisch een **afwezig** voor dit vak. Zie hoofdstuk 0 voor meer informatie hierover.

Concreet zal je op de volgende zaken beoordeeld worden:

- De correctheid van alle features die in secties 1 en 2 beschreven werden.
- De structuur en kwaliteit van je code. Dit houdt ook in dat je, waar nodig, duidelijke comments bij je code schrijft.
- Of je al het geheugen dat dynamisch wordt gealloceerd tijdens het verloop van het spel ook weer correct dealloceert. Er mogen dus geen memory leaks of andere geheugenfouten ontstaan.
- De correctheid van je makefile.

Indien je een vraag hebt, aarzel dan zeker niet om me een mail te sturen (via Maarten.Vandercammen@vub.be).

Veel succes!