

LEARNMARK HORSSENS

PROGRAMERING C

2. MFI

Røvhul

Skrevet af:

Konrad CHRISTENSEN

Marcus LANGFELDT

Indhold

1	Abstract	2
2	Problemformulering	2
3	Dokumentation	3
3.1	Reglerne	6
3.2	Rust	6
3.3	C++	6
4	Konklusion	7

1 Abstract

Igennem denne opgave har vi arbejdet med at lave kortspillet "Røvhul", samt vi har arbejdet med at lave to AI's som er i stand til at spille spillet, på den mest optimale måde. Opgaven vil gå i gennem vores begrundelse for projektet i sektionen "Problemformulering", samt hvad vores plan at gøre med selve projektet, og hvordan vi i grove træk tænker det skal gøres.

Derefter vil opgaven komme ind på hvordan de forskellige funktioner i programmerne fungerer sammen med eksempler på hvordan de kan bruges. Der vil i kapitlet "Dokumentation", også blive snakket om hvordan projektet fungerer, ud over de forskellige funktioner som bliver brugt. Efter følgende vil vi så konkludere om det er lykket at opfylde vores krav til det fuldente program resultat, som vi satte som mål i problemformuleringen.

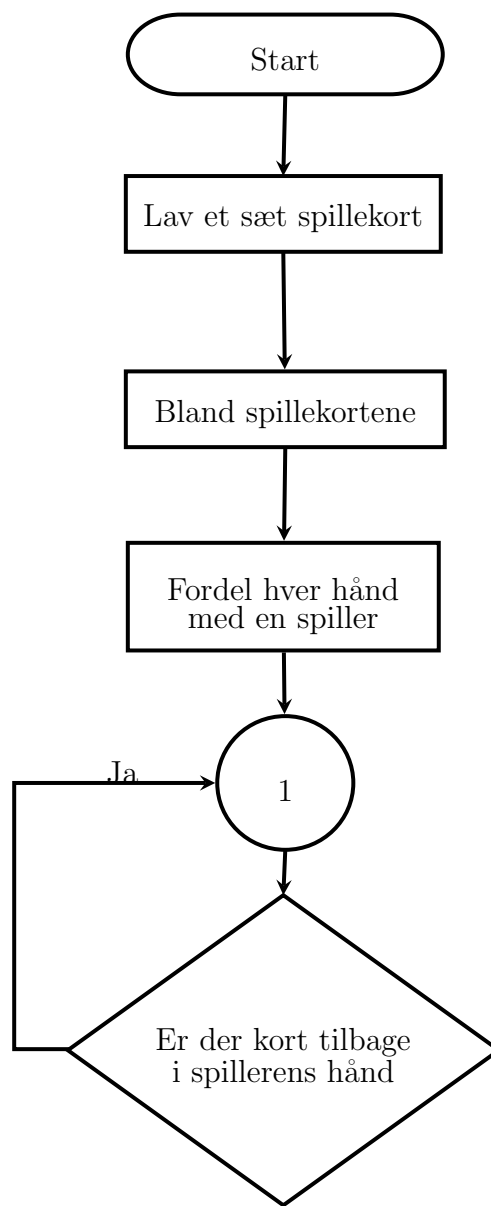
2 Problemformulering

Der er mange forskellige strategier i kortspillet Røvhul. Ved at programmere forskellige AI med forskellige strategier kan vi spille dem mod hinanden og finde den bedste strategi.

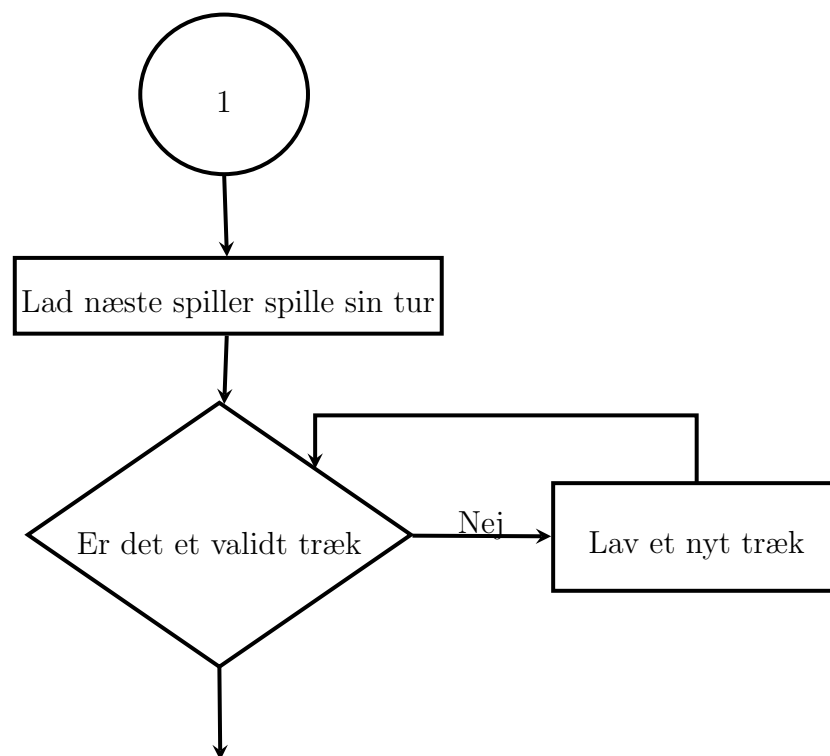
For at kunne dette skal vi først lave en implementering af Røvhul og derefter skal der identificeres og implementeres strategier. Vi forventer at vi med denne metode kan lave en AI der kan vinde over mennesker.

3 Dokumentation

Selve kortspils delen af projektet er lavet i Rust, hvor af spillets regler også er implementeret. For at undgå at de forskellige AI's overholder reglerne, er der også blevet implementeret en funktion som skal tjekke om de forskelle træk som bliver lavet er lovlige. Og hvis de ikke er vil spil delen bede den specifikke AI om at lave et nyt træk. Vi har valgt at lave de forskellige AI's i Rust og C++,



Figur 1: Diagram 1



Figur 2: Diagram 2

Diagram 1, er en visualisering over selve spillet samt loopet, som tjekker om der er flere kort der kan spilles.

Diagram 2, er en visualisering over det loop som står for at næste spiller får sin tur, samt at trækkene er valide.

3.1 Reglerne

I spillet røvhul er der mange regler som gør at forskellige strategier kan have sin fordel i forskellige scenarier. For eksempel, når man starter et spil, så spilleren som har klør 3, er den spiller som starter. Den der starter behøves dog ikke at smide klør 3, da det kan være smartere at lave et træk som gør at man kan komme af med flere kort på en gang. Andre regler er for eksempel

- Rydde bunken
- Pas eller at skippe runden
- Højere eller lig med sidst lagte kort

Når man ryder bunken vil den eksisterende bunke blive lagt til side og et nyt kort lægges af den der sidst har lagt et kort. Bunken i vores udgave af "Røvhul" kan rydes af 10'ere, ved at 4 ens kort er blevet lagt i streg, eller ved at alle har sagt pas. Hvis det sker at der ikke kan lægges flere kort i bunken da ingen spillere kan lægge et kort, vil spillerene blive tvunget til at sige pas, og bunken vil blive ryddet.

Når en spiller siger pas, vælger spilleren at de først kan lægge et kort igen når bunken er blevet ryddet. Det betyder at hvis en spiller har en 2 som de ikke kan ligge, og dermed siger pas. Og hvis de så kan lægge den næste gang rækkefølgen kommer til dem, så må de ikke lægge kortet da bunken ikke er blevet ryddet endnu.

Spillerene må kun lægge kort der enten er lig med den øverste kort i bunken eller kort som er højere end det øverste kort i bunken.

3.2 Rust

```
fn main() {  
    println!("Fart")  
}
```

3.3 C++

4 Konklusion