

LEARNMARK HORSSENS

PROGRAMERING C

2. MFI

Røvhul

Skrevet af:

Konrad CHRISTENSEN

Marcus LANGFELDT

Markus JAKOBSEN

Indhold

1	Abstract	2
2	Problemformulering	2
3	Dokumentation	3
3.1	Reglerne	6
3.2	Rust	7
3.3	C++	7
4	Konklusion	8

1 Abstract

Igennem denne opgave har vi arbejdet med at lave kortspillet "Røvhul". Til spillet har vi identificeret 3 strategier hvortil vi har lavet 3 forskellige AI's som hver følger deres strategi. Ud fra det har vi identificere hvilken af de 3 strategier der er bedst. Opgaven vil gå i gennem vores begrundelse for projektet i sektionen "Problemformulering", samt vores plan for selve projektet, og vores tre strategier til at spille spillet.

Derefter vil vi i kapitlet "Dokumentation" komme ind på eksempler fra koden og forklare hvordan forskellige funktioner i koden fungerer og hvordan de bliver brugt. Vi vil også i dokumentationen beskrive hvordan de tre AI's er udført i koden.

Til sidst vil vi konkludere om det er lykket at opfylde vores krav til det fuldente program resultat, som vi satte som mål i problemformuleringen, og hvilken af vores AI's der er bedst til at spille spillet.

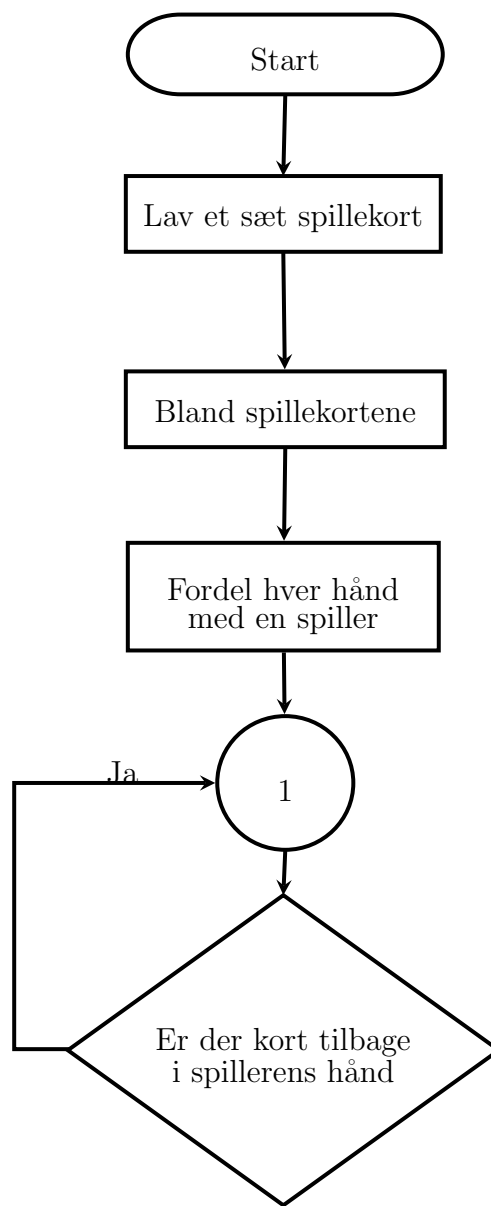
2 Problemformulering

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.

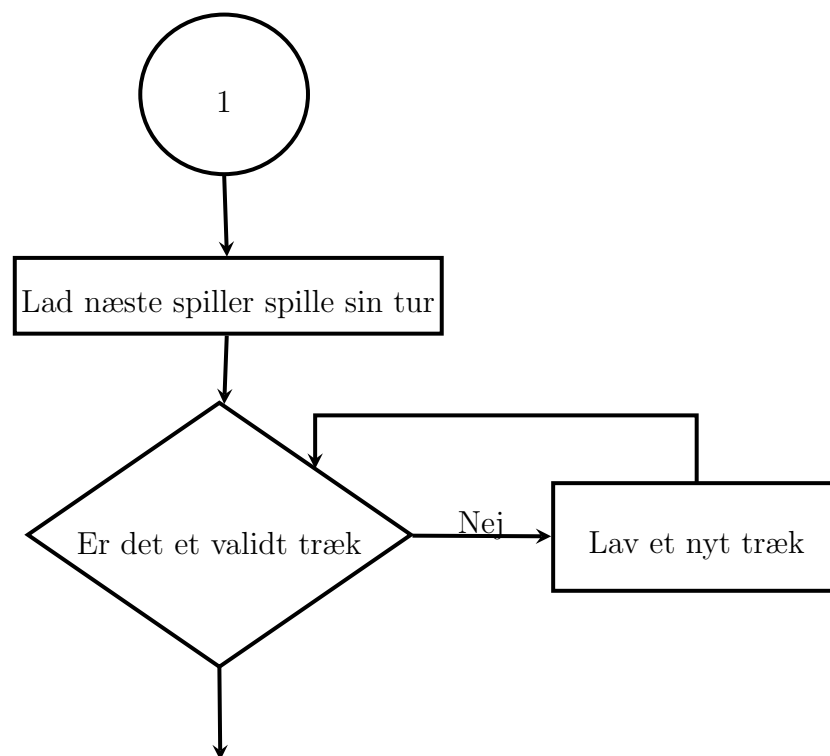
For at kunne dette skal vi først lave en implementering af Røvhul og derefter skal der identificeres og implementeres strategier. Vi forventer at vi med denne metode kan lave en AI der kan vinde over mennesker.

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 forskellige træk som bliver lavet er lovlige. Og hvis de ikke er lovlige, så vil spillet bede den AI som lavede det ulovlige træk, om at lave et nyt træk, i et håb om at det så er et lovligt træk. Vi har valgt at lave de forskellige AI's i Rust og C++. Neden under er der et digrammer over hvordan spillet fungere, samt også et diagram over hvordan de forskellige AI's fungere samt at der bliver tjekket trækket som AI'en laver er et validt træk.



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

iiiiii HEAD Reglerne i røvhul er forholdsvist simple, men de skaber mulighed for et rigt og avanceret strategisk kortspil, hvor man ofte er nød til at nare sin modstander til at tro noget, så man kan få dem til at hjælpe en med at afslutte sin egen hånd så man kan vinde. Det at der er forskellige regler betyder at forskellige strategier, som hver kan have sine fordele og ulemper i forskellige scenarier. ===== Reglerne i røvhul er forholdsvist simple, men de skaber mulighed for et rigt og avanceret strategisk kortspil. De forskellige regler betyder at forskellige strategier kan have sine fordel i forskellige scenarier. llllllll master Der er et par forskellige regelsæt og der kan til tider være uenigheder om enkelte regler. Vi har derfor defineret det regelsæt som vi har lavet spillet ud fra.

- Målet i røvhul er at spille alle sine kort så man har nul kort tilbage i hånden.
- Den første person som har spillet alle sine kort og vinder spillet bliver kåret præsident. Den spiller som sidst har kort i hånden bliver kåret røvhul. Yderligere hvis mere end 3 spillere spiller på en gang bliver den anden første og anden sidste henholdsvis kåret visepræsident og viserøvhul.
- I første spil er det spilleren med klør tre som starter, dog behøver de ikke spille klør tre i første runde. I følgende spil er det røvhullet fra det tidligere spil som starter.
- Hver spiller skiftes til at ligge et kort ned på toppen af bunken i urets retning. Kortet man ligger ned skal altid være lig med eller højere end det øverste kort i bunken.
- Det højeste kort er to efterfulgt af es, konge osv. ned til tre. Det gælder dog ikke 10 som ryder bunken og derfor er højere end to.
- Når bunken rydes kasseres alle kort i bunken og spillet fortsætter. Det er altid den spiller som sidst har lagt et kort i bunken som starter med at ligge et kort.
- Hvis der ikke er nogle kort i bunken må man spille et hvilket som helst kort. Yderligere er det tiladt at ligge to eller flere af det samme kort eks. to konger eller tre es. Hvis to eller flere kort bliver lagt i bunken på denne måde skal alle følgende spillere ligge den samme mængde kort på bunken som også er ens og er lig med eller højere.
- Hvis to spiller ligger to ens kort i streg er det ikke tiladt at den tredje

spiller ligger to ens kort i bunken.

- Hvis der på noget tidspunkt ligger fire ens kort i streg i bunken rydes bunken.
- Hvis man ikke kan, eller ikke vil spille et kort er det tiladt at sige pas, men hvis en spiller siger pas og bliver sprunget over vil de automatisk blive sprunget over ind til den nuværende bunke rydes.
- Når man begynder et spil efter det første spil eller efterfulgte spil er afsluttede skal røvhullet give sine to bedste/højeste kort hvor ti er højest så to, es osv til præsidenten. Præsidenten skal derefter give to af sine kort til røvhullet og præsidenten må selv vælge hvilke to kort de giver. Alle kort gives med bagsiden opad. Hvis man spiller visepræsident og -røvhul gælder samme regel for dem, dog bytter de kun et kort.

Vi har analyseret reglerne og udvalgt enkelte regler som vi føler ikke vil passe ind i en digitalisering af spiller. For det første giver det ikke mening at rækkefølgen af spillere er givet ud fra hvordan man sidder rundt om et bord. Derfor har vi i valgt at tilfældigt lave rækkefølgen før spillet starter og derved simulere ture rundt om et bord. Da der kun er tre spillere med i spillet spiller vi ikke med visepræsident og -røvhul reglen.

3.2 Rust

```
fn main() {  
    println!("Jeg er fart")  
}
```

3.3 C++

4 Konklusion