Projektplan för examensarbetet

Preliminär titel: Reinforcement Learning Agent for boardgame Crypt.

Namn: Pontus Wallquist

Uppdragsgivare: Piktiv AB

1 Projektidé

Idén handlar om att implementera ett brädspel som heter Crypt så det kan spelas virtuellt på en dator. Det ska även skapas en maskininlärnings-agent baserat på reinforcement-learnings metoder som kan spela/lösa spelet på en nästan-mänsklig nivå.

2 Bakgrund

2.1 Uppdragsgivare

Piktiv är ett konsultbolag som jobbar inom systemutveckling, spelutveckling och growth management. Det är idag en trend inom spelbranschen att utveckla AI agenter för att skapa nya spelupplevelser till konsumenterna. Det finns också ett intresse att konvertera spel och populära IP till brädspel eller spel med brädspels liknande mekanik. Med dagens teknologi kan man relativt snabbt applicera AI till ett spel och få ett "human-level" motstånd i dessa spel. Det finns inte många studios som gör detta idag och därför ser Piktiv ett intresse i att ge sig in i detta område. Projektet är viktigt för Piktiv då det ger företaget ett första steg in i detta område.

2.2 Projektbakgrund

AI för spel och brädspel har ökat i popularitet sedan Google DeepMinds AlphaGo agent 2016 nått en ”super-human” nivå på det kinesiska brädspelet Go. Agenten har sedan dess blivit känd världen över som en slags oslagbar motståndare. Det har startat en trend att använda Reinforcement Learning och artificiella neurala nätverk i allt fler spel och brädspel. Piktiv har sedan ett par år tillbaka arrangerat dessa typer av arbeten för studenter. Målet med arbetet är att implementera ett stokastiskt, diskret brädspel samt någon ML modell som kan visa sig vara en värdig motståndare för en mänsklig spelare. Piktiv är en bra kandidat till att skapa dessa brädspelskonverteringar, tex till mobiler. I ett senare skede kan dessa arbeten förädlas till en prototyp och användas som koncept i företagets portfölj. Senaste arbetet gjordes under våren 2022 med brädspelet Ticket-To-Ride.

Board Game AI Using Reinforcement Learning. Linus Strömberg, Viktor Lind.
Örebro Universitet, Institutionen för naturvetenskap och teknik.
https://www.diva-portal.org/smash/record.jsf?pid=diva2:1680520

2.3 Teoretisk bakgrund

Ämnet handlar till största del om vilka RL metoder och algoritmer som kan lära sig spela diskreta, stokastiska brädspel på ett människoliknande sätt. Från en informationssökning har jag hittat några bra algoritmer som utgår från Q-Learning som först introducerades av Watkins 1989, och som bygger på att optimalt lösa en Markov Decision Process (MDP) som låter agenter utföra sekvenser av aktioner utan att kartlägga hela spelets domän. Senare forskning har dock visat en brist på prestanda i stokastiska miljöer. Hado van Hasselt publicerade 2010 en fortsättning på Q-Learning algoritmen som han kallar Double Q-Learning. Där algoritmen sparar två separata Q-funktioner som i sin tur uppdaterar varandra. Metoden har lett till förbättringar inom konvergens, teori och praktik. Van Hasselt har sedan tillsammans med Google DeepMinds team vidareutvecklat algoritmen med hjälp av artificiella neurala nätverk i deras publikation Deep Reinforcement Learning with Double Q-Learning från 2015. I den rapporten kombinerar författarna Double Q-Learning med Deep Learning. De introducerar ett koncept som består av två neurala nätverk. De två nätverken hanterar valet av en aktion och värdet på aktionen separat. Författarna har valt att kalla denna algoritm för Double DQN. Den nya iterationen har gett mycket bättre resultat och konvergens än de tidigare iterationerna av algoritmen när modellen får spela klassiska Atari spel.

Utvecklingen av Q-Learning algoritmen har visat stora framsteg för användningen av RL modeller i spel och brädspel. Det fick mig att välja Double DQN algoritmen för mitt eget projekt.
Ett delmål med mitt arbete är att fortsätta denna trend och försöka avgöra om algoritmen faktiskt är ett bra val för diskreta, stokastiska spel och brädspel. Det här kan möjligen utvecklas eller byggas på ytterligare i framtiden.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Hado van Hasselt. Double Q-learning. 2010
https://papers.nips.cc/paper/2010/hash/091d584fced301b442654dd8c23b3fc9-Abstract.html

Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning, 2015.
https://arxiv.org/abs/1509.06461

3 Projekt

3.1 Beskrivning

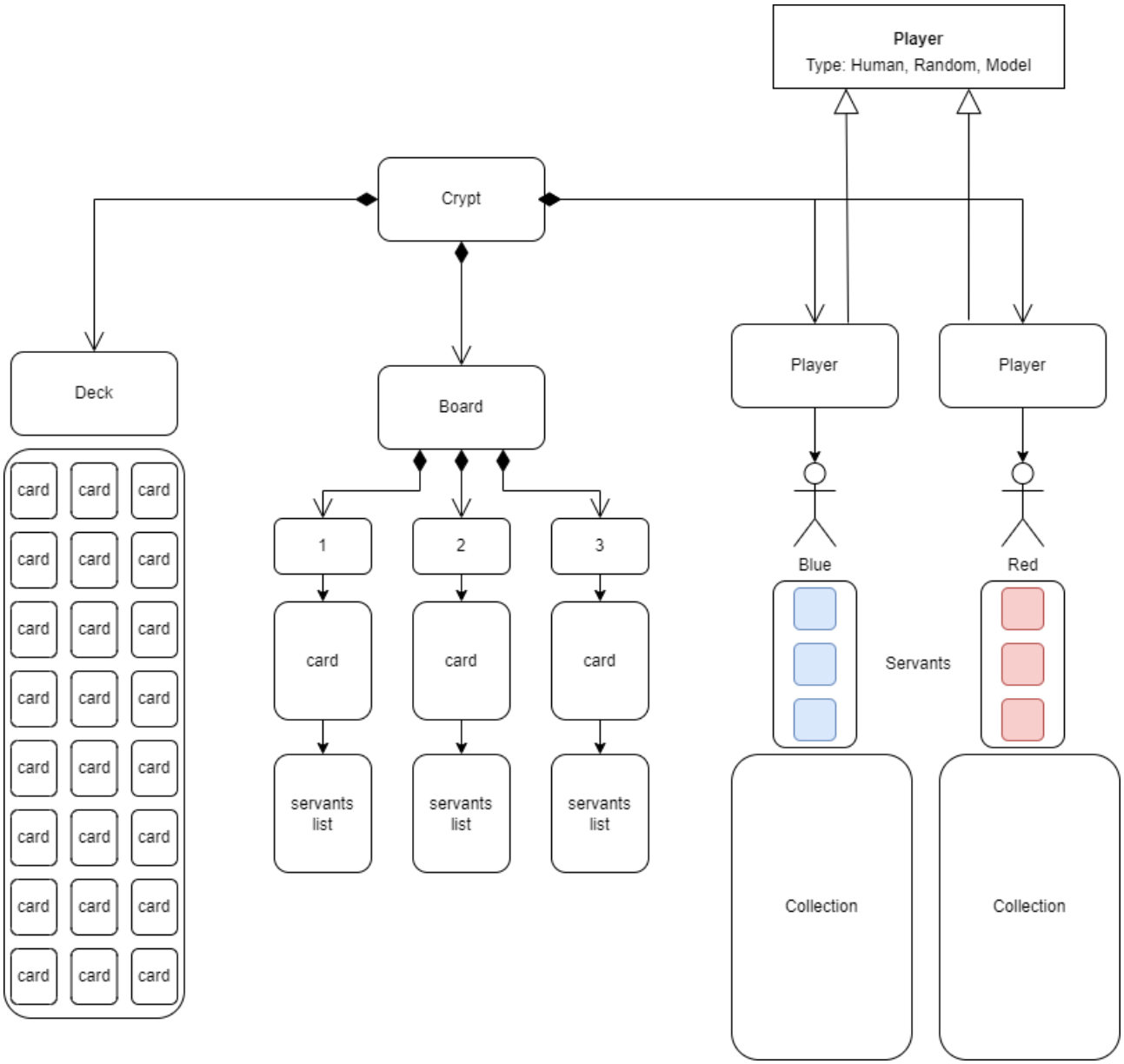
Arbetet är ett utredningsjobb. Det som ska utredas är att först välja en RL algoritm baserat på egen research och avgöra om det är ett bra val för att lösa denna typ av stokastiska spel. Samt om metoden är ett bra val för framtida arbeten att bygga vidare på.

3.2 Krav

Vid projektets slut ska det finnas en virtuell implementation av brädspelet Crypt samt minst en tränad modell som kan agera som en motståndare. Modellen ska utvärderas för att avgöra om den är ett bra val för spelet eller inte. Det kan även implementeras ett GUI i Pygame om det finns tid, men det är inget krav.

3.3 Domänmodell, Arkitekturutkast, Mönster

Systemet kommer till största hand bestå av spelkomponenten Crypt. Det är viktigt att spelet är väldefinierat så modellen har bäst chans att lyckas. Arkitekturen för Crypt kan ses nedan.

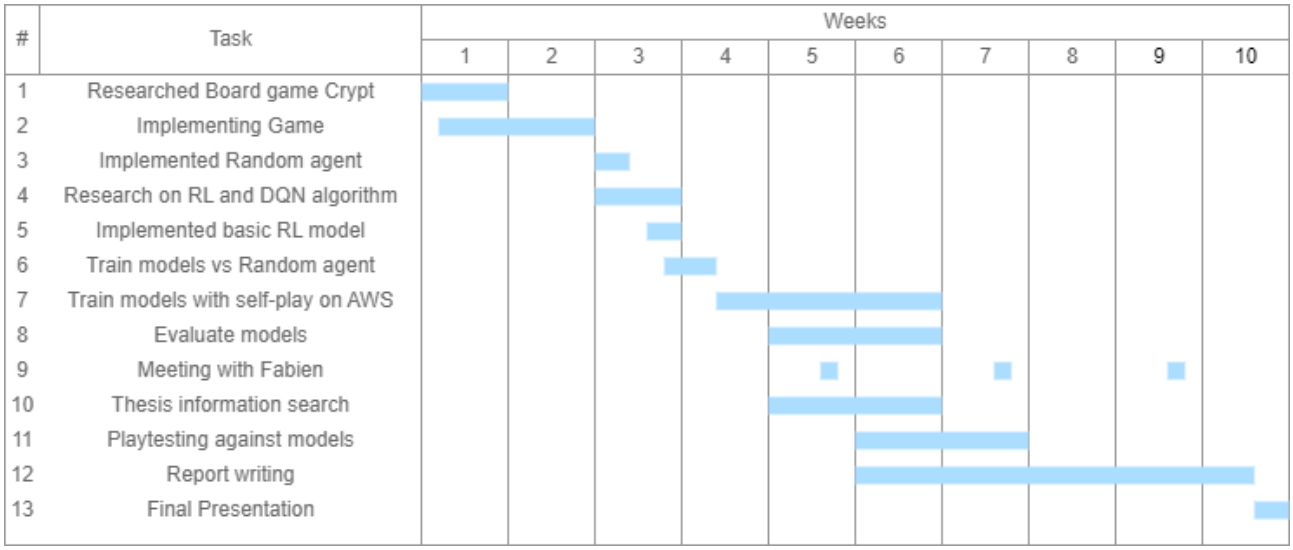


Komponenten Crypt innehåller spelets kortlek på 24 unika kort, en bräda där nya kort dras och läggs fram och två spelare. Det ska gå enkelt till att byta mellan mänsklig, slumpmässig och ML modell. Då spelet är ganska unikt så har jag valt att inte välja något välkänt design-koncept.

4 Metoder, Språk eller Verktyg

Allt arbete kommer göras av mig själv. Systemet kommer byggas med en agil metod genom att först bygga en version av spelet som sedan itereras över och byggs vidare. Det gör det lätt att implementera nya idéer genom arbetets gång. Spelet och ML modellerna kommer byggas i språket Python med hjälp av ramverken Tensorflow, Keras och Numpy. Jag kommer även använda Docker för att enkelt kunna träna modeller på olika datorer utan att behöva installera alla ramverk på förhand. Träningen kommer ske på CPUs och Piktiv kommer bistå med cloud-tjänsten AWS för att träna modeller.

5 Tidsplan



Jag har planerat arbetet så att det praktiska jobbet och byggandet av systemet sker de första fem veckorna. De sista fem veckorna är det mer intensivt fokus på rapporten.