

*Bachelor Thesis*

Reinforcement Learning for board game Crypt

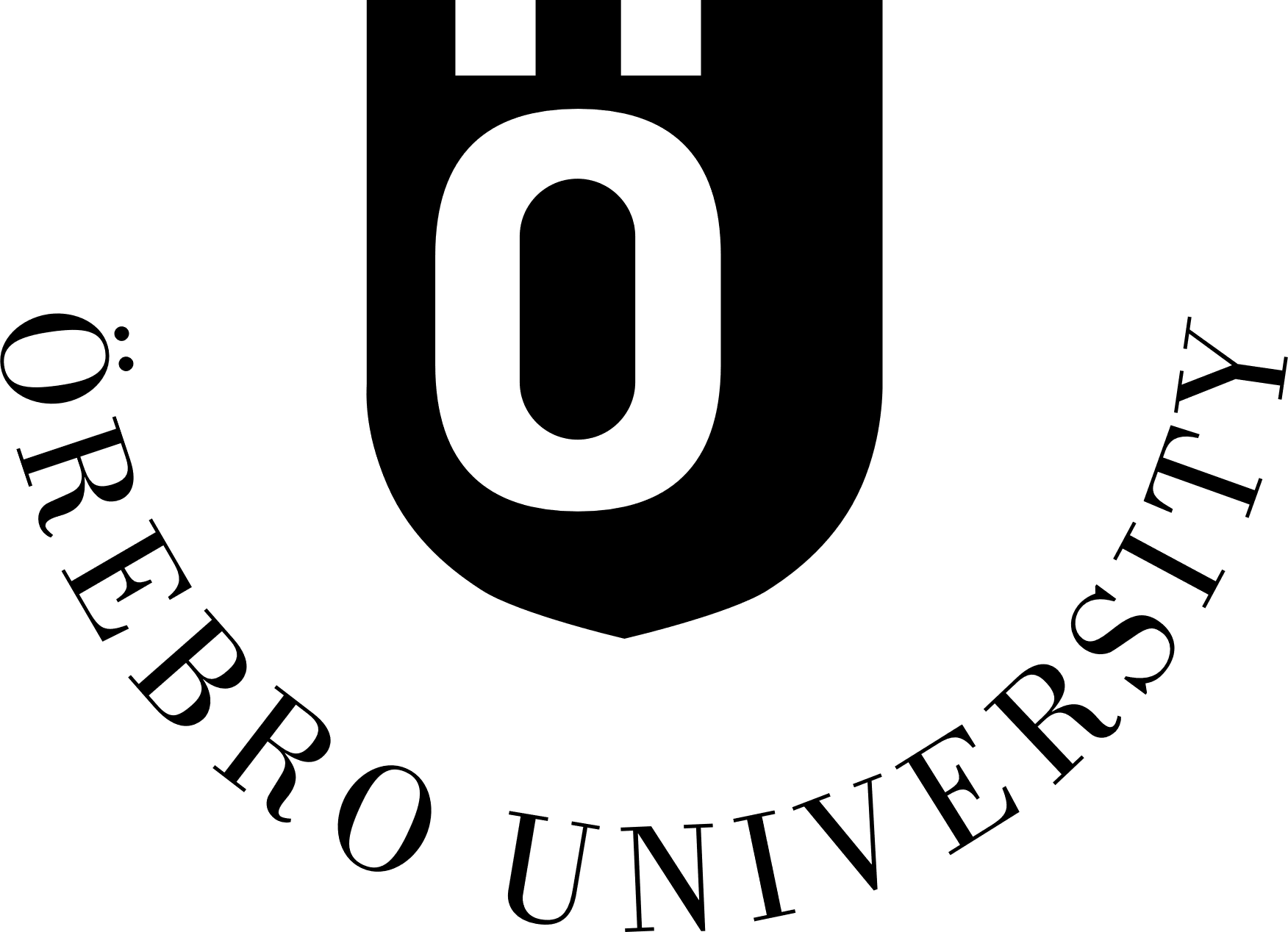
Pontus Wallquist

*Computer Science*

Studies from the School of Science and Technology at Örebro University

Örebro 2022

*Studies from the School of Science and Technology at Örebro University*



Pontus Wallquist

Reinforcement Learning for board game Crypt

**Supervisors: Simon Johansson, Piktiv AB**

**Fabien Lagriffoul, Örebro University**

**Examiner: Stephanie Lowry, Örebro University**

© Pontus Wallquist, 2022

# Abstract

Abstract is a short overview of the content of the work. The main goal of abstract is to give a reader an idea about the work, without the need to read it all. Thus, the abstract should be concise, but at the same time concrete on the content of the work.

Typically, abstract is not more than one page long, and presents the work in a brief and concise way. Often abstract follows the outline of the work presents a problem, method, and results.

### Keywords

Template, BSc Thesis, Computer Science, Computer Engineering

# Contents

[Abstract 1](#_Toc122020413)

[Contents 2](#_Toc122020414)

[1 Introduction 3](#_Toc122020415)

[1.1 Problem Formulation 3](#_Toc122020416)

[1.2 Outline 4](#_Toc122020417)

[2 Background 5](#_Toc122020418)

2.1 Reinforcement Learning

[3 Implementation 7](#_Toc122020419)

[4 Results 8](#_Toc122020420)

[5 Conclusion 9](#_Toc122020421)

[References 10](#_Toc122020422)

# 1 Introduction

The purpose of this project is to examine the viability of artificial players in stochastic board games. Mainly the board game Crypt [1] created by Andrew Nerger and Jeff Chin. This thesis will document the implementation of the game as well as the agents that will learn to play it.

## 1.1 Problem Formulation

The idea for this project comes from a company called Piktiv AB. They specialize in game development and software development. They see a potential for creating tougher adversaries in video games and board games through machine learning methods. They would like to see artificial players be able to play a videogame on a near human-level in the future.

The goals of this project are as follows:

1. Build the chosen board game as a video game. However due to time limitations, the goal is not to create a fun or good-looking game. The focus will instead be on implementing the game logic so an artificial player can play it.
2. Build a Reinforcement Learning Agent that will learn to play the game through self-play.
3. Evaluate the agent by playing games against random agent, another RL agent and human opposition.
4. Document the development and results in the form of a bachelor thesis.

## 1.2 Outline

The rest of this thesis is organized as follows:

* Chapter 2 will give an overview of the background work that led up to this project and introduce the reader to reinforcement learning concepts. This chapter will also provide the rules and details of the board game Crypt.
* Chapter 3 contains the development and implementation of the game as well as the reinforcement learning agent. At the end of this section the reader will understand the game and the chosen algorithm Q-Learning.
* Chapter 4 will present the results of the training and testing of multiple agents in the game environment.
* Chapter 5 contains the conclusions drawn from the results and a discussion on the future viability of reinforcement learning methods in board game AI.

# 2 Background

**2.1 Reinforcement Learning**

Reinforcement learning is the practice of learning by trial and error [2]. An agent can learn to navigate environments without any previous knowledge of the domain. It will map specific situations to maximize a certain reward signal. However, the agent must discover for itself which actions return the highest reward by trying them. Reinforcement learning is well suited to stochastic problems or simply problems with an immense search space far too big to be solved with dynamic programming [3].

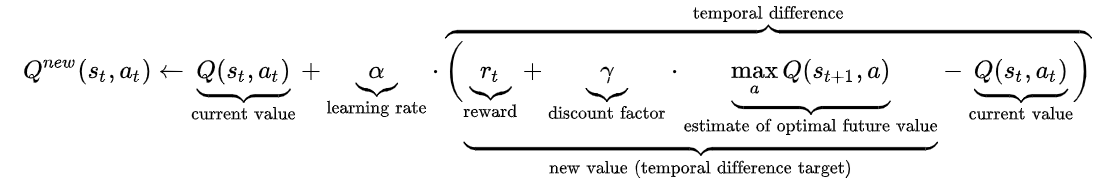
In 2017 the Google DeepMind team made a breakthrough in reinforcement learning and its use in board games with challenging domains. They created an agent known as AlphaGo [4]. Which became famous for defeating the world champion in the Chinese board game of Go. It achieved a level of superhuman performance previously unheard of. The game of Go is notorious for having one of the biggest board game domains in the world. Since then, there has been a growing popularity in creating reinforcement learning agents for more complex video games and board games. In 2019 the team at OpenAI created an agent capable of reaching superhuman performance in competitive esports games [5]. The game they chose was Dota 2, which is a multiplayer real-time strategy game. Their agent was able to defeat the current Dota 2 world champions in a best-of-three match.

The OpenAI team explains rather well in the introduction of [5], that the long-term goal of artificial intelligence is to solve advanced real-world challenges. They also mention that games have served as steppingstones to this for decades and I believe it will continue to do so for many years to come.

These are some of the most successful AI agents out today and they have helped inspire more students and researchers to go into the field of artificial intelligence and reinforcement learning.

The agent created for this project will be using an iteration of the Q-learning algorithm first introduced in 1989 by Watkins [6]. The algorithm works by building a map between rewards and actions in an iterative manner. The action values will be called Q-values. These values are then stored in a Q-table. Good actions will obtain high Q-values and vice versa for bad actions. The Q-table will be saved and updated throughout the training of the agent, further improving the predictions for each iteration.

The algorithm works by adding the current Q-value with the chosen learning rate *α*, multiplied by the reward *r* from the last action, added with a discount factor **γ,** that’s multiplied with an estimate of the optimal future value max*Q (s, a),* subtracted by the current Q-value. The equation can be seen below in figure 2.1.



*Figure 2.1 Q-learning algorithm*

In 2010 Hasselt published a paper called Double Q-Learning [7]. He explains that the original Q-learning algorithm has poor performance when training agents in stochastic environments. Instead, he suggests the use of two Q-value functions, which was proven to increase the performance, especially in stochastic environments.

Hasselt later reiterated on his algorithm in [8]. He added Artificial Neural Networks to combine with the Q-learning algorithm, which in turn improved the estimation of actions and the convergence rate. It works by implementing two neural networks that work together. One network estimates the Q-value of all actions while the other network evaluates which action to take based on the first networks predictions.  
Since this paper was published, the Double Deep Q-Network algorithm has been the go-to for implementing reinforcement learning agents with Q-learning. This is also the algorithm used for the agent in this project.

**2.2 Crypt**

# 3 Implementation

**3.1 Crypt implementation**

Environment architecture and design choices.

**3.2 Agent implementation**

Encoded environment (inputs).

Action space (outputs).

DQN agent architecture (input, hidden, output).

Rewards system.

Training the agent.

# 4 Results

Results from games (charts and tables).

# 5 Conclusion

Conclusion from results.

Discussion regarding future RL methods for stochastic board games.

# References

1. Andrew Nerger and Jeffrey Chin. R2i Games. Crypt. 2018 <https://www.kickstarter.com/projects/roadtoinfamy/crypt-1>  
   Rules: <https://roadtoinfamy.com/rules/Crypt_booklet_printerFriendly.pdf>
2. Richard S. Sutton and Andrew G. Barto. Reinforcement Learning, An Introduction. MIT Press, 1998 <https://redirect.cs.umbc.edu/courses/graduate/678/spring17/RL-3.pdf>
3. Ronald A Howard. Dynamic programming and markov processes. 1960.
4. David Silver et al. Mastering the game of Go without human knowledge. 2017 Nature, 550:354–359
5. OpenAI, Christopher Berner et al. Dota 2 with Large Scale Deep Reinforcement Learning. 2019 <https://doi.org/10.48550/arXiv.1912.06680>
6. Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989
7. Hado van Hasselt. Double Q-learning. 2010 <https://papers.nips.cc/paper/2010/hash/091d584fced301b442654dd8c23b3fc9-Abstract.html>
8. Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning, 2015 <https://arxiv.org/abs/1509.06461>