

**Honours Project - MHW225671**

**INTERIM REPORT**

**2023-2024**

**Department of Computing**

**Submitted for the Degree of: BSc Computer Games (Software Development)**

**Project Title: Development of an AI agent using Reinforcement Learning to optimise Rocket League team behaviour play.**

**Name: Ektor Zoidis**

**Programme : Computer Games (Software Development)**

**Matriculation Number: S2222427**

**Project Supervisor: Hamid Homatash**

**Second Marker:**

**Word Count:**(***excluding contents pages, figures, tables, references and Appendices***)

**“Except where explicitly stated, all work in this report, is my own original work and has not been submitted elsewhere in fulfilment of the requirement of this or any other award”**

**Signed by Student: Ektor Zoidis                     Date: 20/11/2023**

# Contents

[Contents 2](#_Toc153141453)

[1 Introduction 3](#_Toc153141454)

[1.1 Background 3](#_Toc153141455)

[1.2 Project Overview 4](#_Toc153141456)

[1.2.1 Project Outline 4](#_Toc153141457)

[1.2.2 Project Objectives 4](#_Toc153141458)

[1.2.3 Hypothesis 5](#_Toc153141459)

[2 Literature and Technology Review 5](#_Toc153141460)

[2.1 Investigating Reinforcement Learning algorithms 5](#_Toc153141461)

[2.2 Examining how Rocket League works as a game 8](#_Toc153141462)

[2.3 Defining the attributes that contribute to teamwork in Rocket League 8](#_Toc153141463)

[2.4 Identifying the tools that will be required to achieve this project 8](#_Toc153141464)

[3 Methods 8](#_Toc153141465)

[4 References 9](#_Toc153141466)

# Introduction

This section will outline the project along with its scope and objectives. It will describe what Reinforcement Learning (RL) is, how it is being used and explain how it can be applied in this project with the goal of creating an effective AI agent at learning to play Rocket League with a focus on team behaviours. It will also delve into a brief history of previous projects that used RL and present the achievements this technology has been able to achieve. Additionally, it will explain the general gameplay and concept of Rocket League and how the game works.

## Background

Reinforcement Learning (RL) is one of the three fundamental pillars of machine learning which centres on the improvement of performance through reinforcement. This is the iterative process of actions and rewards obtained from an environment to reach a set of goals (Barto, 1997). The basic idea is that with each action taken by the agent its state changes and a reward is given to it for that given action, positive or negative, it is with this method that we are able to communicate what is right and wrong and train it to choose the most effective action for a given state (Kaelbling, Littman, & Moore, 1996).

A white paper with black text

Description automatically generated

Figure 1 – The relation between the agent and its environment (Kaelbling, Littman, & Moore, 1996)

Figure 1 (Kaelbling, Littman, & Moore, 1996) shows an intuitive example of how the agent and the environment communicate with one another. The agent’s goal is to maximise the total amount of positive reward it receives and with a well-defined reward system the agent will be able to complete the desired objectives.

Previously RL has been used to master many board games such as TD-Gammon (Tesauro, 1994), Go (Silver, et al., 2018), Shogi (Silver, et al., 2018) and Chess (Silver, et al., 2018), as well as many older Atari games (Kaiser, et al., 2013). In more recent years, however, RL has been used for more complicated games where it managed to defeated the world champions in Dota 2 (OpenAI, et al., 2019), achieve grandmaster levels in StarCraft 2 (Vinyals, et al., 2019) and play Minecraft (Guss, et al., 2021). All these games present unique challenges with lengthy game durations, complex rules and large number of actions and states which proves how effective RL is at solving challenging and complicated problems.

Rocket League is a fast-paced, physics, online multiplayer, vehicular soccer video game made by Psyonix in 2015. It uses rocket-powered cars to play soccer and score goals and features unique mechanics that require high precision and fast reflexes like aerial manoeuvres, wall and ceiling plays, realistic ball physics and many more that add a layer of complexity and dynamism to the gameplay making it unpredictable. These complexities make it a challenging and distinctive title for AI to excel in, yet there have been attempts already made to achieve superhuman performance. The most notable ones are the agents Necto and its newer and improved version Nexto which are considered the best Rocket League bots, being able to win against top ranked players and in bot championships. Part of why these bots are so successful is due to their reward shaping functions, well-defined action spaces and the vast number of network parameters (Moschopoulos, Kyriakidis, & Lazaridis, 2023). There are more recent attempts that have been made, however, and one of them was able to outperform and surpass Nexto: Lucy-SKG (Moschopoulos, Kyriakidis, & Lazaridis, 2023) proving yet again how effective RL can be for training any agent.

While an API with a number of different bots exists and there has been a lot of engagement from the community to create the best possible bot for Rocket League the existing bots do not show sophisticated team behaviour, particularly intentional passing behaviour which is one of the most basic forms of teamwork in any ball game (Verhoeven & Preuss, 2020). This creates a notable gap for research and arises the question of how this can be achieved.

## Project Overview

### Project Outline

Due to the lack of studies on achieving bots that perform team behaviour in Rocket League, this project aims to investigate the development of an AI agent that will learn to collaborate and perform team interactions. Therefore, the research question for this project is:

**How effectively can an AI agent trained through Reinforcement Learning perform team behaviours and interactions in Rocket League.**

### Project Objectives

The primary aim of the project is to develop an AI that will be able to perform team behaviours and interactions effectively using Reinforcement Learning in Rocket League. In order to develop the AI agent a list of Literature and Primary Objectives have been defined and listed below.

#### Literature Objectives

* Investigating Reinforcement Learning algorithms

Look into existing algorithms/methods used in Reinforcement Learning and investigate which ones would be most suitable for this project.

* Examining how Rocket League works as a game

As Rocket League is a complicated game understanding the elements that make it and breaking them down is a crucial part that must be performed to develop the AI and create a well structure reward system.

* Defining the attributes that contribute to teamwork in Rocket League

As this project focuses on team behaviours recognising which parts make up teamwork and the attributes used to measure it is an important goal for the literature.

* Identifying the tools that will be required to achieve this project

Identifying which tools have been previously used for related projects and learning how to use them effectively is vital for the success of this project.

#### Primary Objectives

* Prototyping the AI

As a starting point the AI must first learn to play the game in general to then be able to focus on the goals of this project. A prototype will be made that will create the basic framework to begin working on.

* Developing the AI

As soon as the AI is able to perform the very basic functionalities of the game the full development will begin using the knowledge gained from the literature review to ensure that the agent is as effective it can be.

* Training the AI

Once the AI has been successfully developed time must be arranged for it to train. Training is a very important part of RL that takes time so having time after its development is essential.

* Carrying out testing

When the AI has been developed and trained for some time testing will be conducted in a suitable method to gather results.

* Evaluating its performance

When the results are out the evaluation will be undertaken to assess the AI’s effectiveness at performing team behaviours and interactions. The results will be analysed and a conclusion will be drawn.

### Hypothesis

To make sure that the AI’s team behaviour capability can be tested and that a conclusion can be drawn team behaviour will be assessed through the observation of replays. Given this the proposed hypothesis for the project is:

The developed AI agent will be able to perform team behaviours in Rocket League effectively with the use of Reinforcement Learning.

# Literature and Technology Review

## Investigating Reinforcement Learning algorithms

As described in the introduction Reinforcement Learning (RL) allows agents to automatically determine the ideal behaviour that must be performed for a given state which has been previously used to deal with large and complex problems while having only partial information regarding it. In order to understand how it is able to solve these problems it is necessary to break down the foundation. The main mathematical framework used to describe RL problems is referred to the Markov Decision Process (MDP) which essentially is the model of the environment and agent, and it ensures that the future state and reward depend only on the current state and action. It is defined by a tuple *M = (S, A, p, γ, R)*, where *S* is the state space, *A* is the action space, *p* is the transition probability of moving to another state, *γ* is the discount factor of future rewards and *R* is the reward function, a list of desired and undesired behaviours. (Sutton & Barto, 2018). The goal is to learn a policy (*π*) that maximises the expected cumulative reward over time. In the context of MDPs, policies are sets of rules that specify how the agent must behave in different states. In a single time step (iteration) as briefly mentioned the environment tells the agent what the state is, the agent decides about what action to perform based on the state and then the environment transition to its next state. This cycle keeps going with this back-and-forth interaction between the environment and agent.

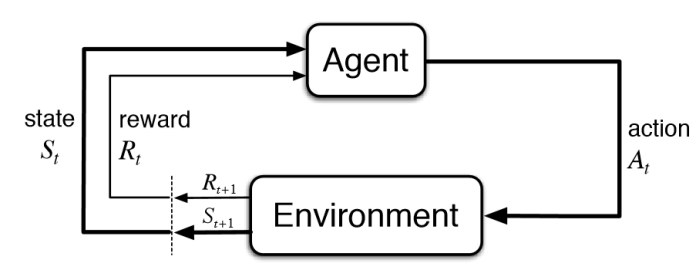


Figure 2 – The Reinforcement Learning Loop (Bhatt, 2018).

One of the biggest challenges faced in RL is finding the balance between trying new actions to discover potentially better strategies (exploration) vs exploiting known actions for immediate rewards (exploitation), this is known as the Explore-Exploit Dilemma (Sutton & Barto, 2018). This problem becomes apparent in Rocket League where the agent may exploit known techniques to achieve reliable performance but miss out on the opportunity to explore new mechanics to score more goals and receive higher rewards. There are two common approaches for addressing this problem: model-based and model-free algorithms. Model-based methods attempt to learn the underlying MDP and establish the transition probabilities and reward function, and then applying the discovered knowledge for decision-making once the MDP is adequately explored (Brafman & Tennenholtz, 2002). With Model-free methods, the focus is to determine the optimal policy directly, without requiring explicit knowledge of the underlying MDP.

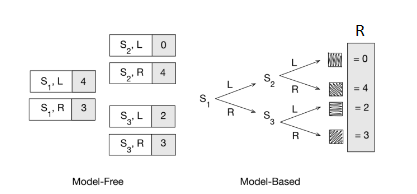


Figure 3 – Model-Based and Model-Free strategies to solve a hypothetical sequential action-selection problem (Sutton & Barto, 2018).

These can be further divided into on-policy and off-policy algorithms. In an on-policy method data is collected using the current policy, giving the agent full control over what information to collect and allowing it to decide whether to explore or exploit. In an off-policy method there is no control over how data is acquired which prevents the trade-off between exploration and exploitation and allows the use of different policies for data collection.

A study made by Daanesh Ibrahim et al. looks into the effectiveness of different RL algorithms to autonomously learn and optimise Rocket League play (Ibrahim, Stacy, & Stroud, 2021). It constructs a framework to identify the best performing method to complete this task and concludes that Proximal Policy Optimisation (PPO) an on-policy model-free algorithm achieved the best results. Other methods such as Deep Q-Learning and Twin Delayed DDPG (TD3) were tested but were proven to be unsuccessful due to their design. Q-Learning uses a discrete action space (finite possible actions) which contrasts Rocket League’s continuous action space (infinite possible actions) and TD3 is fragile to large updates resulting in drastic behaviour changes, making it unable to discover behaviours that are crucial for the agent’s learning. PPO performed best due to its ability to handle continuous action spaces, achieving long-term goals and performing gradual policy updates. These properties allowed the agent to develop beneficial behaviours that led to positive outcomes and rewards but also fundamental behaviours that play a vital role in the agent’s ability to learn during training.

A blue line graph with numbers

Description automatically generated

Figure 4 – Average Cumulative Rewards vs. Timesteps Trained using PPO (Ibrahim, Stacy, & Stroud, 2021).

Another study made by Chao Yu et al. looks into the effectiveness of PPO in a variety of cooperative multi-agent games (Yu, et al., 2022). It concludes that PPO can be a competitive baseline for cooperative tasks and comparable to the state-of-the-art methods in various cooperative multi-agent scenarios and challenges. It identifies that there are 5 main factors that are especially influential for its performance: the use of value normalisation which mitigates the issue of drastic changes in the value targets and often improves the final performance or reduces the training variance; the use of fewer epochs per update in the value function inputs to limit the change in the agent’s policies inputs and improve stability; policy/value clipping to prevent the policy and value functions from drastically changing; and utilizing specific batch size settings to optimise for sample-efficiency.

PPO is part of the policy gradient algorithm category which means that the policy is updated in such a way that increase the likelihood that actions taken currently will yield grater rewards in the future. The agent explores the state and the algorithm monitors the agent’s actions and how the resulting state changes – these interactions are known as trajectories. Once the algorithm accumulates a set of trajectories, it will analyse them to determine which actions resulted in a positive or negative reward and the policy is then modified based on these observations (Schulman, Wolski, Dhariwal, & Radford, 2017). In PPO the advantage function is able to predict the benefit of the action the agent is about to perform and modify the results by implementing weights (gradients). The resulted outcome is registered and compared with the advantage estimate, if the outcome is better than the estimate the gradients increase but if it is worse than the gradients will decrease making favourable outcomes more desired and unfavourable less likely (Schulman, Wolski, Dhariwal, & Radford, 2017). PPO’s unique probability ratio that factors into updating the policy is also something beneficial for this case as it restrains large updates to the policy from happening which prevents the risk of missing reward windows and hurting the policy. As Rocket League is a game of unpredictability all these factors make PPO the perfect candidate for an environment as complex and continuous as Rocket League (Ibrahim, Stacy, & Stroud, 2021).

PPO has been previously used for other related studies that managed to develop top-performing bots for Rocket League. In particular the papers of Neville Walo (Walo, 2022) and Vasileios Moschopoulos et al. (Moschopoulos, Kyriakidis, & Lazaridis, 2023) which as previously mentioned were able to outperform the best performing bot Nexto.

## Examining how Rocket League works as a game

## Defining the attributes that contribute to teamwork in Rocket League

## Identifying the tools that will be required to achieve this project

# Methods

# References

Barto, A. G. (1997). Reinforcement Learning. *Elsevier eBooks*, pp. 7–30. doi:https://doi.org/10.1016/B978-012526430-3/50003-9

Bhatt, S. (2018). *5 Things You Need to Know about Reinforcement Learning.* N/A: KDnugget. Retrieved from chat.openai.com/?model=text-davinci-002-render-sha

Brafman, R. I., & Tennenholtz, M. (2002). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research, 3*(Oct), 213-231.

Guss, W. H., Castro, M. Y., Devlin, S., Houghton, B., Kuno, N. S., Loomis, C., . . . Nakata, K. a. (2021). The minerl 2020 competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv:2101.11071*, pp. 37. doi:https://doi.org/10.48550/arXiv.2101.11071

Ibrahim, D., Stacy, J., & Stroud, D. a. (2021). Rocket Learn. *SMU Data Science Review, 5*(2), 12.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: a survey. *Journal of Artificial Intelligence Research, 4*, pp. 237-285. doi:https://doi.org/10.1613/jair.301

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., . . . Michalewski, H. (2013). Model-Based Reinforcement Learning for ATARI. *arXiv preprint arXiv:1312.5602*, 'p. doi:https://doi.org/10.48550/arXiv.1903.00374

Moschopoulos, V., Kyriakidis, P., & Lazaridis, A. a. (2023). Lucy-SKG: Learning to Play Rocket League Efficiently Using Deep Reinforcement Learning. *arXiv preprint arXiv:2305.15801*, p. 24. doi:https://doi.org/10.48550/arXiv.2305.15801

OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., . . . P, H. (2019). Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv preprint arXiv:1912.06680*, 'p. doi:https://doi.org/10.48550/arXiv.1912.06680

Schulman, J., Wolski, F., Dhariwal, P., & Radford, A. a. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 11-21.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., . . . Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science, 362*(6419), pp. 1140-1144. doi:https://doi.org/10.1126/science.aar6404

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction.* Cambridge: MIT press.

Tesauro, G. (1994). TD-Gammon, a Self-Teaching backgammon program, achieves Master-Level Play. *Neural Computation, 6*(2), pp. 215-219. doi:https://doi.org/10.1162/neco.1994.6.2.215

Verhoeven, Y., & Preuss, M. (2020). On the potential of Rocket League for driving team AI development. *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 2335-2342. doi:10.1109/SSCI47803.2020.9308248

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., . . . Ewalds, T. a. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature Publishing Group UK London, 575*(7782), pp. 350-354.

Walo, N. (2022). Seer: Reinforcement Learning in Rocket League. *SDSC*, 11-21.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 24611-24624.