TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A REPORT
ON
SnakeAI

SUBMITTED BY:

Bigyapti Bashyal

SUBMITTED TO:

Department of Electronic and Computer Engineering

Acknowledgement

We would like to express our gratitude to all those who have assisted or motivated us in completing this project. Our special thanks go to our teachers for their guidance, counselling, and constant supervision. He provided us with necessary information and valuable suggestions which helped us to learn the importance of collaboration and innovation.

We also commend the Department of Electronics and Computer Engineering for their initiative to involve students in practical education, enabling us to familiarise ourselves with real-world problem-solving and teamwork.

We are sincerely thankful to everyone who has helped us to turn our ideas into a tangible project. We especially appreciate the support and guidance from our parents and friends, without whom this project would not have been possible.

We welcome any suggestions for the improvement of this project and would appreciate any feedback.

Abstract

This Project focuses on simulating garbage collection robots using the Deep Q-Learning algorithm, which is a modified and upgraded version of Q-Learning algorithm.In this project, we propose a simulation environment for training artificial intelligence (AI) agents to collect garbage scattered around the environment. Our Garbage Collector simulation AI project is built using Deep Q-Learning, a type of reinforcement learning algorithm. The objective of our project is to train an AI agent that can navigate a digital environment, collect Garbage, and avoid obstacles, all while avoiding its body. We evaluate our model's performance against a range of benchmarks, including traditional rule-based approaches, and demonstrate that our Deep Q-Learning Garbage Collector Simulation AI outperforms these methods. We also explore the impact of various hyperparameters on the performance of our model, showing that careful tuning can lead to significant improvements. Our findings suggest that Deep Q-Learning is an effective method for training intelligent agents capable of playing complex games like Snake.

Table of Content

Introduction

Background

SNAKEAI is an innovative project aimed at improving waste management and reducing the negative impact of garbage on the environment. The project leverages the power of Deep Q-learning, a popular machine learning algorithm, to train an intelligent garbage collection system that can make optimal decisions about when and where to collect garbage from. By using this cutting-edge technology, SNAKEAI seeks to enhance the efficiency of waste management, minimise fuel consumption and travel time, and ultimately contribute towards building a more sustainable future.

The problem of waste management is a critical issue that affects the well-being of communities around the world. SNAKEAI addresses this issue by developing an intelligent system that can collect garbage more effectively and efficiently. The simulation environment will provide a platform for testing different scenarios and parameters, allowing us to fine-tune the system to optimise its performance in different situations.

Motivation

Our motivation for working on the garbage simulation project stemmed from our interest in Reinforcement Learning, which we learned about through an online course. One of the initial examples presented in the course was an AI agent that collected garbage scattered around a room using Reinforcement Learning. We found this idea to be not only fascinating but also very socially relevant. We realized that there was a significant potential to develop an automated garbage collection system that could be both efficient and smart.

We strongly believe that automated garbage collection is a pressing need, and that there is a lot of scope for improving the existing systems. In particular, we feel that the integration of artificial intelligence and machine learning techniques could enable more sophisticated garbage collection methods that are better suited to the needs of today's world.

Overall, we are motivated to explore the potential of using Deep Q-Learning to build a garbage collector that can operate more intelligently and efficiently. We hope that our work can contribute towards developing better solutions for managing waste, and ultimately lead to a cleaner, healthier, and more sustainable environment.

Problem Statement

Inefficient garbage collection is a significant problem in places around the world, leading to increased pollution and negative impacts on the environment. Current garbage collection systems often follow a fixed schedule, resulting in wasteful trips to empty bins that are not yet full, and missed pickups from overflowing bins. In addition, many garbage collection vehicles have a large carbon footprint due to inefficient routing.

To tackle the problem of inefficient garbage collection, we need a smarter system that can adapt to changing conditions and minimize waste. That's where the SNAKEAI project comes in, with the aim of developing a Deep Q-Learning-based simulation for garbage collection. By creating an efficient and adaptable garbage collection system, SNAKEAI aims to help reduce the environmental impact of waste management, improve resource efficiency, and make our living environment cleaner and healthier for all.

Objectives

- Develop a simulation environment for garbage collection using Deep Q-Learning.
- Design an intelligent garbage collection agent that can make optimal decisions about which locations to collect garbage from and in which order, based on the current state of the environment.
- Train the garbage collection agent using Deep Q-Learning to maximize the overall efficiency of garbage collection while minimizing travel time and fuel consumption.
- Evaluate the performance of the garbage collection agent using various metrics, such as the total distance travelled, fuel consumption, and the time taken to collect all the garbage.
- Compare the performance of the Deep Q-Learning-based garbage collection agent with that of traditional fixed-schedule garbage collection systems.
- Identify potential challenges and limitations of the Deep Q-Learning approach to garbage collection, and propose possible solutions or improvements.

Literature Review

Deep Q-learning is a popular reinforcement learning algorithm that is used to solve complex decision-making problems. The algorithm uses a neural network instead of a table of Q-values like in Q-Learning to represent the expected future reward of taking a particular action in a given state. Deep Q-learning works by iteratively updating the Q-values based on the rewards received from the environment, with the goal of maximizing the expected cumulative reward over time. Deep Q-learning is widely used in applications for robotics, and autonomous systems, and has been shown to be effective in a wide range of scenarios.

Existing Systems

During our development we found out that the development of garbage collector simulation is very similar to simulating a snake game using Deep Q-Learning. It is interesting to note that the development of garbage collector simulation shares many similarities with the development of a snake game using Deep Q-learning like in both cases, the agent's goal is to navigate through an environment and collect items, whether it is garbage or food. The agent must make decisions about where to move and what actions to take in order to achieve this goal.

Roomba

Roomba is a line of autonomous robotic vacuum cleaners manufactured by iRobot, introduced in 2002. These robots are equipped with sensors that enable them to navigate the floor area of a home, avoiding obstacles such as furniture and detecting dirty spots and steep drops to avoid falling down stairs. As of 2022, iRobot markets models of their 600, i, j, Combo, and s9 series, which have different features, such as tangle-free brushes, separate sweep canisters, more powerful vacuums, obstacle avoidance, and performance maps displayed via smartphone apps.

Roomba has become a popular choice for home cleaning, thanks to its compact design, ease of use, and advanced sensors and software that enable it to navigate around obstacles and detect and clean dirt and debris from floors. The robot can be programmed to clean on a schedule or controlled remotely using a mobile app. iRobot continues to support and sell accessories for their previous series while introducing new models with advanced features. It uses sensing and pathfinding algorithms to clean the bot without using any AI features.

Field of Research

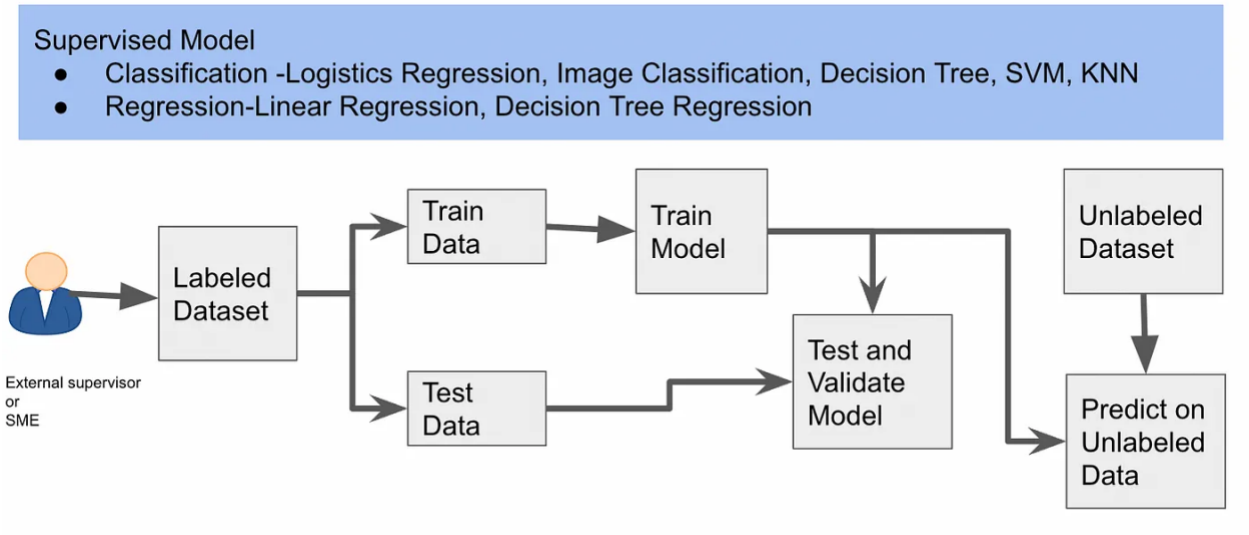
1. "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto:
   This book provides a comprehensive overview of reinforcement learning, including Q-learning, and covers both the theory and practical implementation of the algorithms.
2. "Deep Reinforcement Learning" by Volodymyr Mnih et al.:
   This paper introduces the concept of deep Q-networks (DQNs), which combine Q-learning with deep neural networks, and shows that DQNs can achieve state-of-the-art performance in various game playing tasks.

Theoretical Background

Types of Machine Learning:

Supervised Learning

Supervised learning uses a labeled dataset, typically labeled by an external supervisor, subject matter expert(SME), or an algorithm/program. The dataset is split into training and test dataset for training and then validating the model. The supervised learned model is then used to generate predictions on previously unseen unlabeled data that belongs to the category of data the model was trained on.



Examples of Supervised Learning are Classification and Regression. Classification is used in applications like Image Classification and K- Nearest Neighbors for identifying customer churn. Regression algorithms are used to predict sales, home prices, etc.

Unsupervised Learning

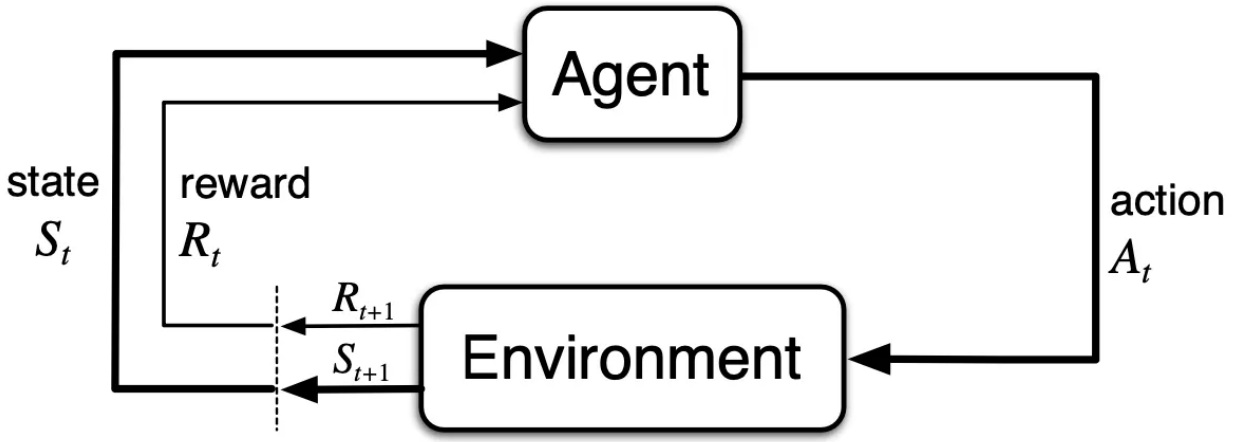
This learning algorithm is completely opposite to Supervised Learning. In short, there is no complete and clean labeled dataset in unsupervised learning. Unsupervised learning is self-organized learning. Its main aim is to explore the underlying patterns and predicts the output. Here we basically provide the machine with data and ask to look for hidden features and cluster the data in a way that makes sense.

Unsupervised Model

- Clustering-K-Means Clustering, Hierarchical Clustering, Agglomerative Clustering
- Association - Recommendation system
- Dimensionality Reduction - Principal Component Analysis(PCA)
- Anomaly detection

Reinforcement Learning

In Reinforcement learning an agent interacts with the environment by sensing its state and learns to take actions in order to maximize long-term reward. As the agent takes actions it needs to maintain a balance between exploration and exploitation by performing a variety of actions using trial and error to favor the actions that yield the maximum reward in the future.



The goal of RL is for the agent to learn to make sequential decisions in an uncertain environment by mapping the different environment states to actions to maximize long-term rewards.

The two main characteristics of RL are trial and error search and delayed rewards like delayed gratification.

Reinforcement learning is applied in Robotics, Self-driving cars, evaluating trading strategies and adaptive controls.

Q-Learning

Q-learning is a model-free reinforcement learning algorithm.

Q-learning is a values-based learning algorithm. Value based algorithms updates the value function based on an equation(particularly Bellman equation). Whereas the other type, policy-based estimates the value function with a greedy policy obtained from the last policy improvement.

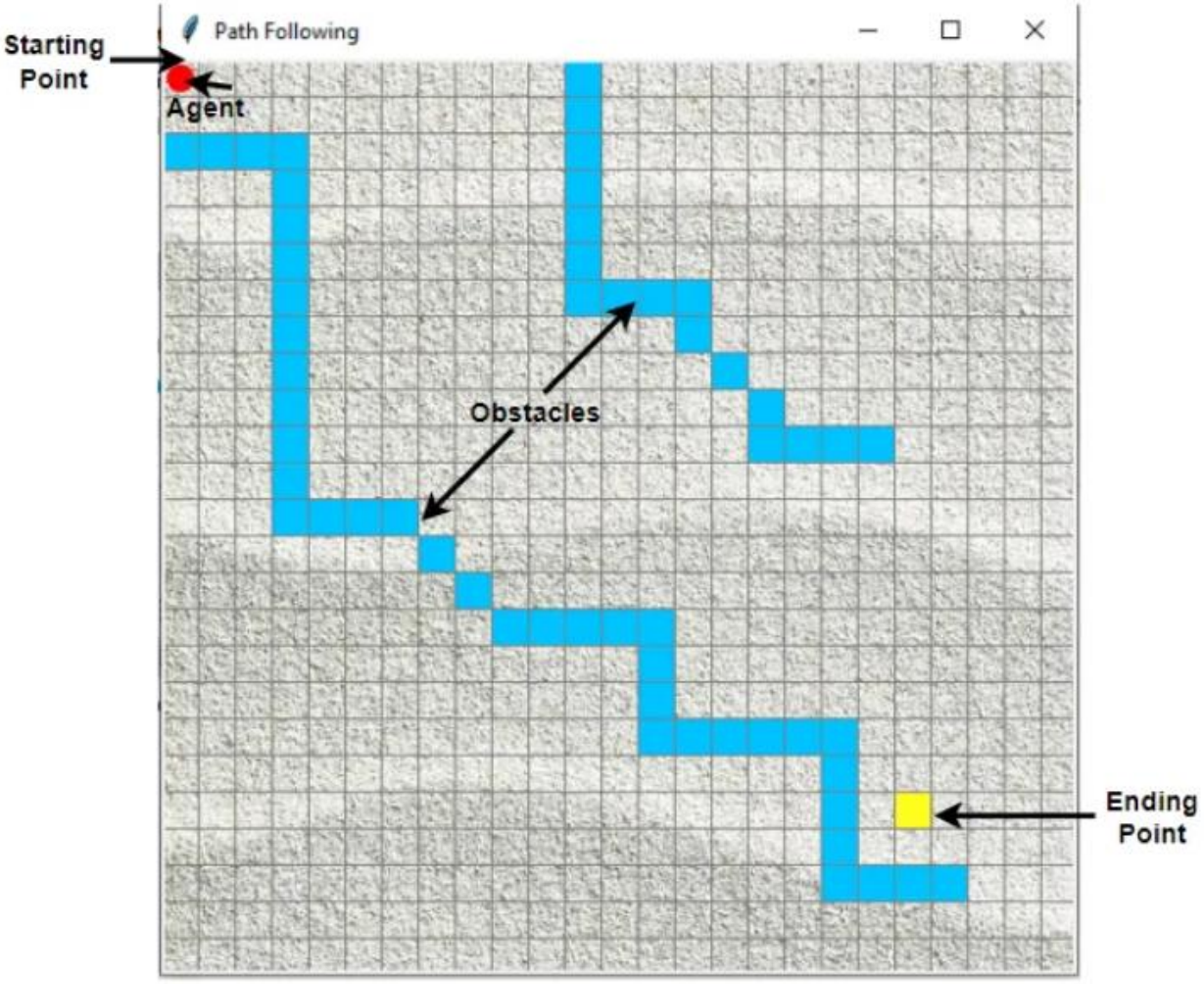
Q-learning is an off-policy learner. Means it learns the value of the optimal policy independently of the agent's actions. On the other hand, an on-policy learner learns the value of the policy being carried out by the agent, including the exploration steps and it will find a policy that is optimal, taking into account the exploration inherent in the policy.

What's this 'Q'?

The 'Q' in Q-learning stands for quality. Quality here represents how useful a given action is in gaining some future reward.

Q-learning Simple Example

Let's say an agent has to move from a starting point to an ending point along a path that has obstacles. Agent needs to reach the target in the shortest path possible without hitting in the obstacles and he needs to follow the boundary covered by the obstacles. For our convenience, I have introduced this in a customized grid environment as follows.



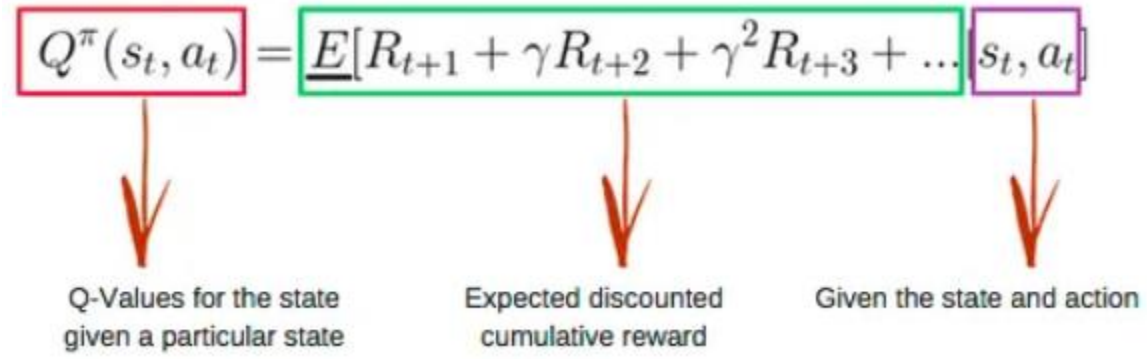
Agent and its Environment

Q-Table

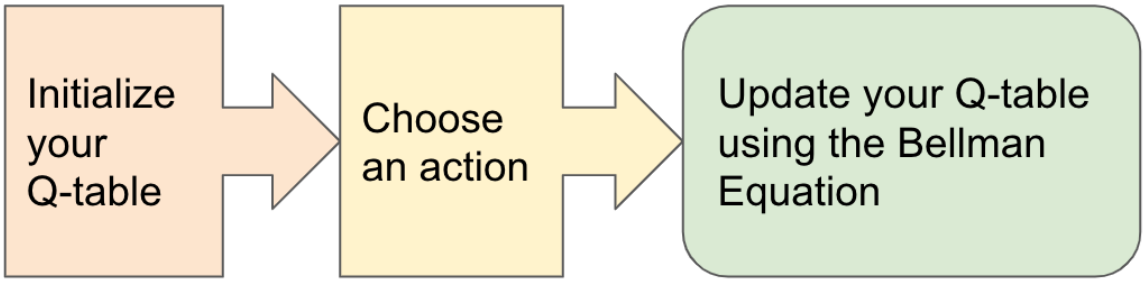
Q-Table is the data structure used to calculate the maximum expected future rewards for action at each state. Basically, this table will guide us to the best action at each state. To learn each value of the Q-table, Q-Learning algorithm is used.

Q-function

The Q-function uses the Bellman equation and takes two inputs: state (s) and action (a).



Q-learning Algorithm



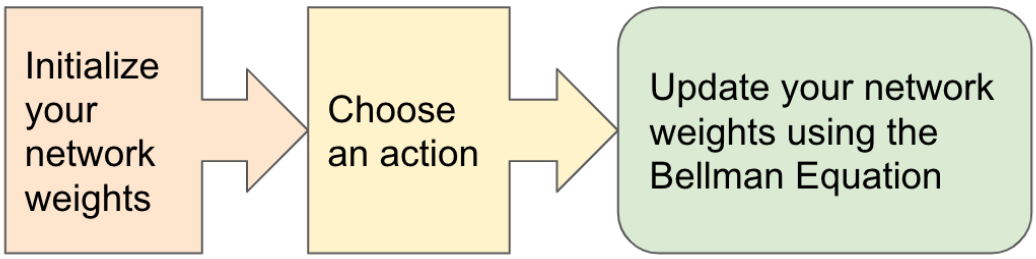
Deep Q-Learning

Although simple, Q-learning is quite a robust algorithm to create a cheat sheet for our agent. This helps the agent figure the most suitable action. In some cases, the cheat sheet is too long, having 10,000 states and 1,000 actions per state. Things will quickly get out of control with a table of 10 million cells. Thus, we can't infer the Q-value of new states from already explored states. This gives rise to two problems:

- Memory required to save and update the table increases as the number of states increase
- Amount of time required to explore each state to create the required Q-table isn't practical

A neural network is used to approximate the Q-value function in deep Q-learning. The state is taken as the input, and the Q-value of all possible actions is generated as the output. T

Deep Q-Learning Algorithm



Major Differences between Q-Learning and Deep Q-Learning

The primary reason for developing Deep Q-Learning was to handle environments that involve continuous action and states. The rudimentary Q-Learning algorithm can be used for small and discrete environments. This is because it works by maintaining a Q-table where the row encodes specific states and the columns encode the various actions that the agent can take in the environment. In a continuous environment, Q-learning can still be worked with by discretizing the states. If multiple variables are to be defined in any possible state in the environment, the Q-table becomes ridiculously large and impractical. The reason is apparent; the more number of rows and columns, the more time agents take to explore every state and update values. This is not a feasible solution but Deep Q-networks as it uses a deep neural network to approximate the Q-table.

Neural Networks

A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.

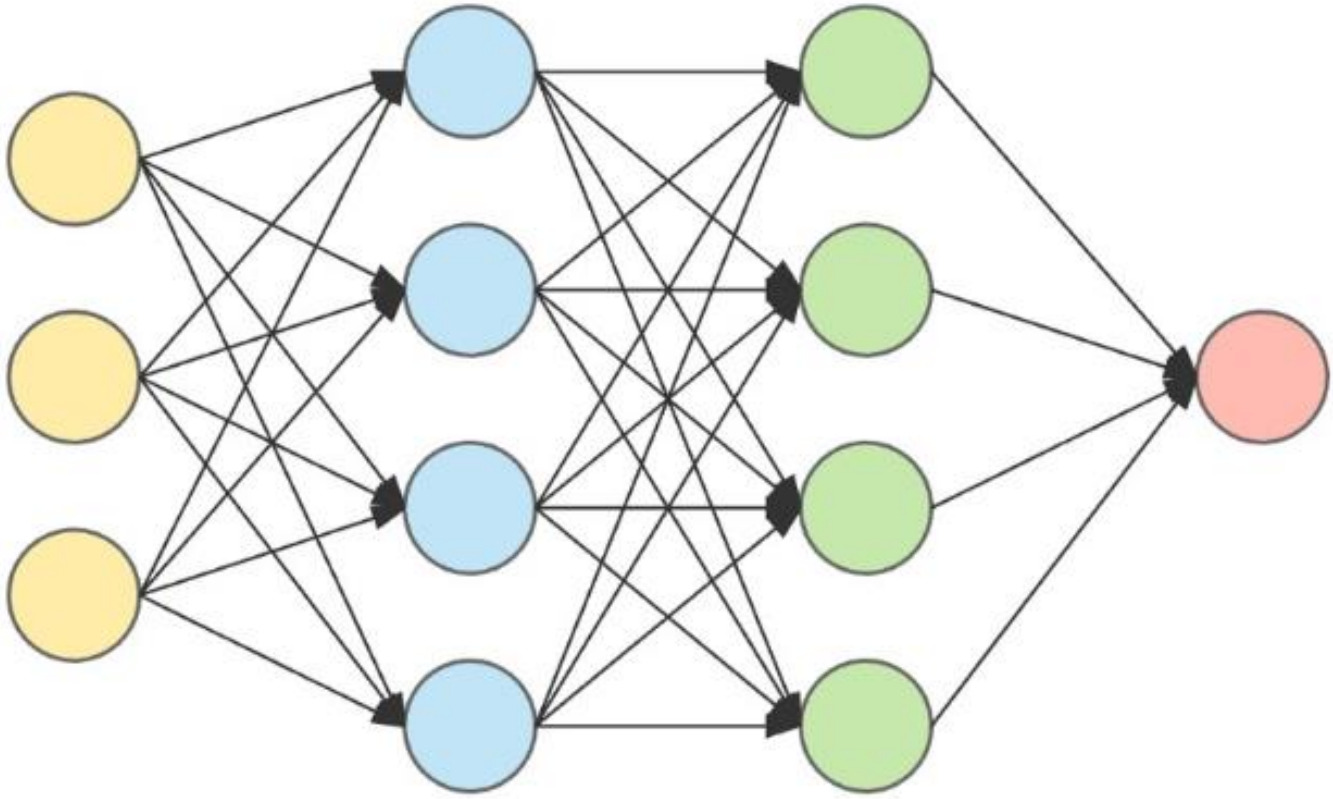
Neurons

Scientists agree that our brain has around 100 billion neurons.These neurons have hundreds of billions connections between them.

Neurons (aka Nerve Cells) are the fundamental units of our brain and nervous system. The neurons are responsible for receiving input from the external world, for sending output (commands to our muscles), and for transforming the electrical signals in between.

Neural Nets

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google's search algorithm.

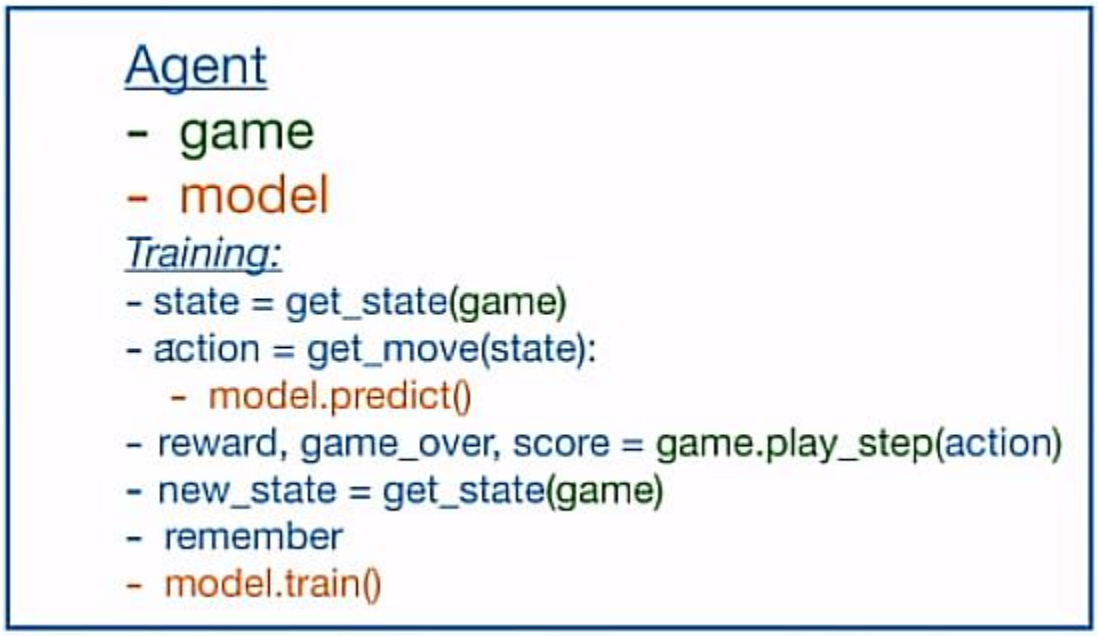


Methodology

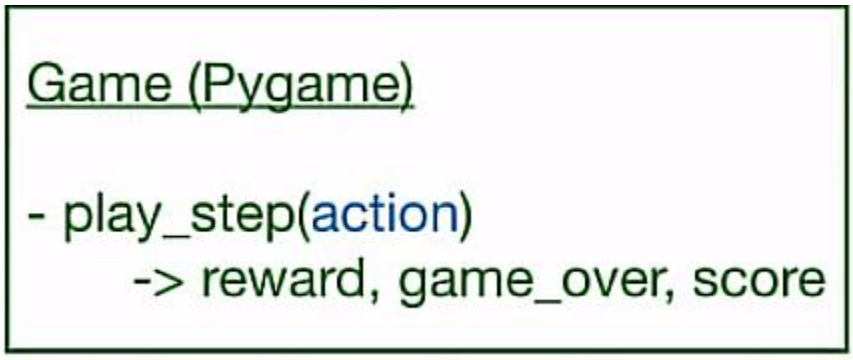
Methodology for building a parking app using the specified technologies:

1. **Develop a simulation environment:** The first step of the project will be to design and build a simulation environment that can model the behaviour of a garbage collection system. The environment will include a garbage collection vehicle, obstacles and Garbage itself
2. **Modify the environment to allow control via code:** The simulation environment is not enough as the simulation should be done via code so the environment is modified to allow control via code and also added reward function
3. **Implement Deep Q-learning algorithm:** Once the simulation environment has been developed, the next step will be to implement the Deep Q-learning algorithm for garbage collection.
4. **Implement the agent:** The agent is a critical component that connects the environment and the model. It serves as the interface between the simulation and the model, controlling the bot's actions in the environment and communicating with the model to obtain the necessary information to make the best possible moves. Without the agent, the environment and model would not be able to work together effectively.
5. **Train the garbage collection agent:** With the Deep Q-learning algorithm in place, the next step will be to train the garbage collection agent using the simulation environment. The agent will learn to make optimal decisions about which garbage bins to collect from and in which order, based on the current state of the environment.
6. **Evaluate the performance of the garbage collection agent:** After the agent has been trained, the next step will be to evaluate its performance using various metrics, such as the Total score, Average score

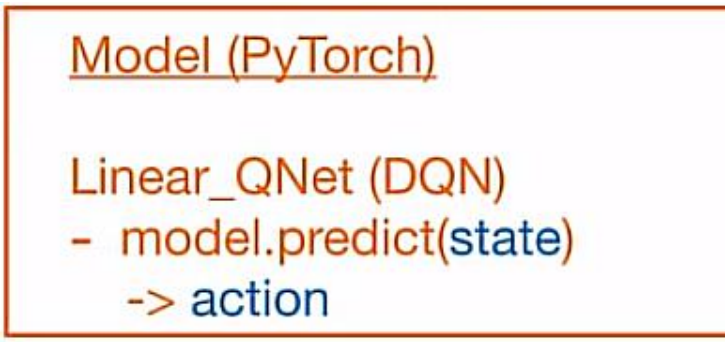
Agent



Game

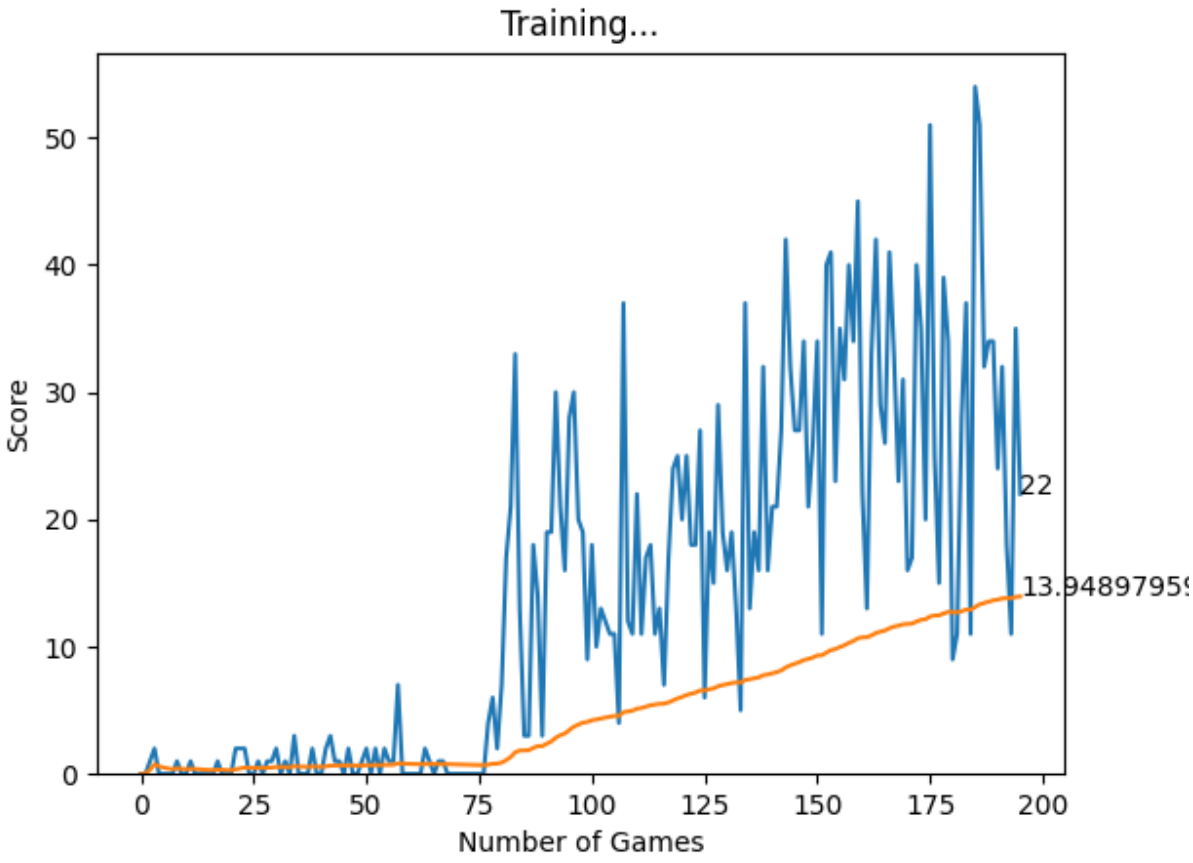


Model

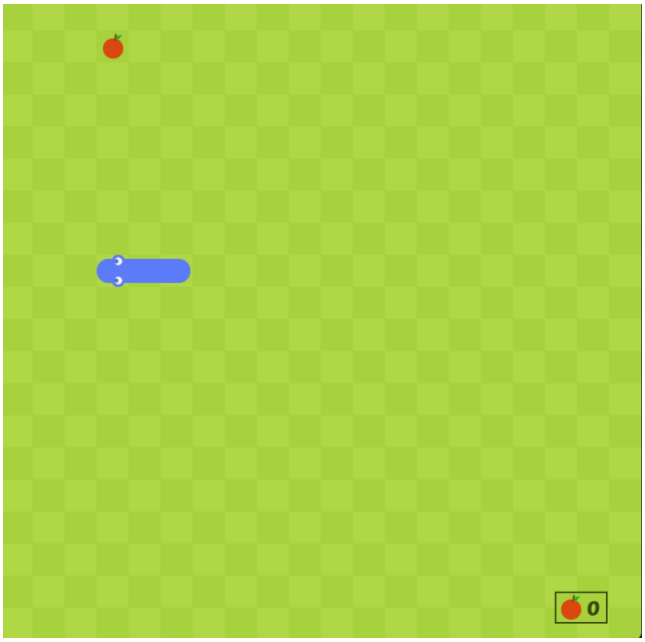


Output

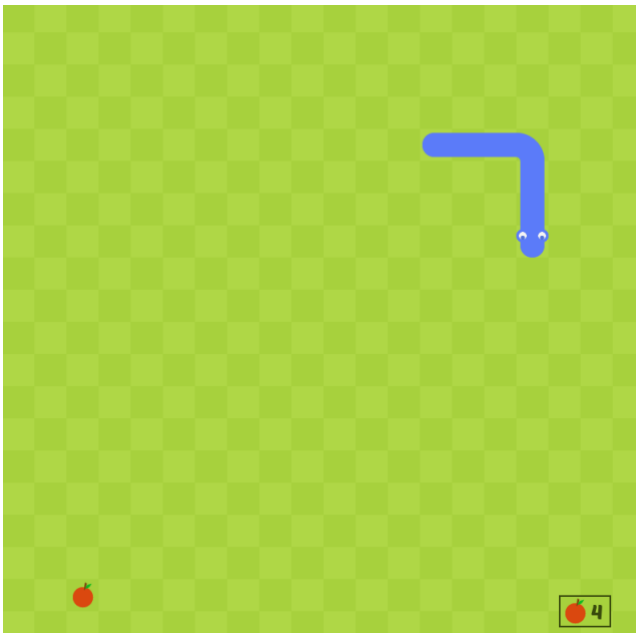
As the Game was similar to the snake game we tried and converted it into snake game for fun. The graph and games below shows the trend of increase in the ability of the snake.



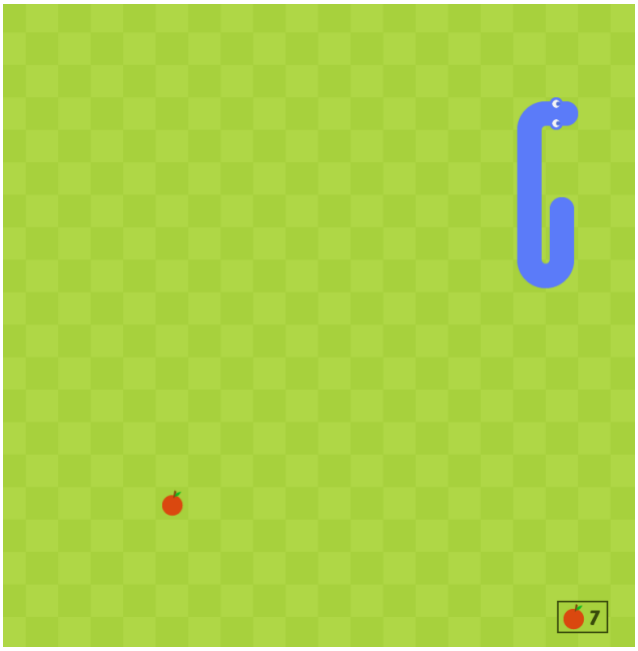
In this graph the blue line shows the score obtained by the agent and the orange line is the running average of the scores. We can clearly see that the average score of the snake is increasing as the number of games increase and the model learns to do better



Game 2

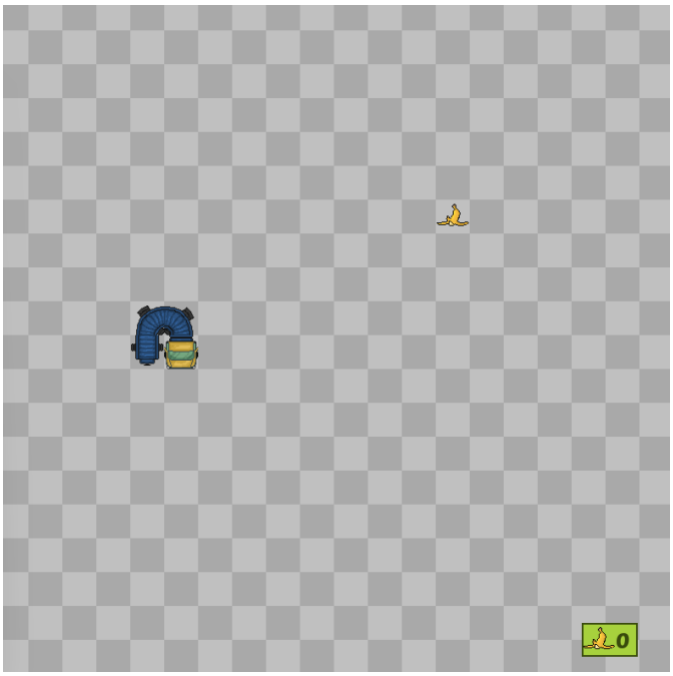


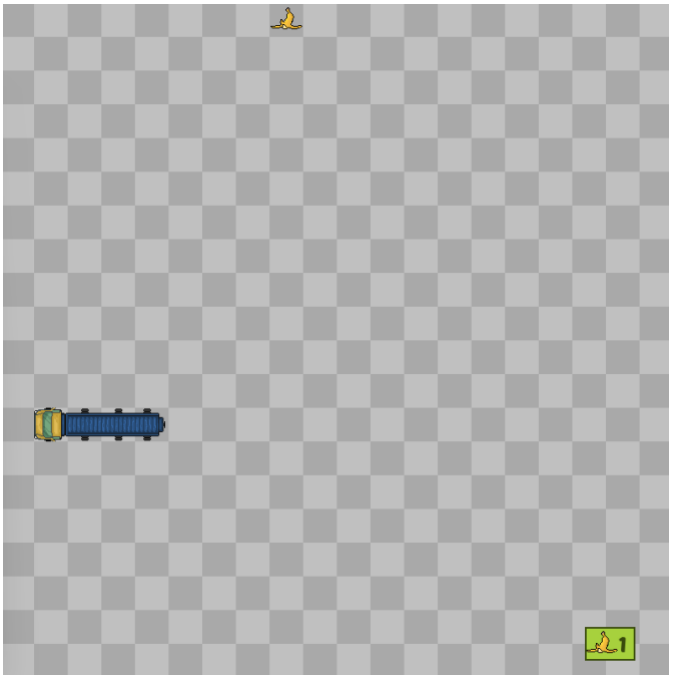
Game 50



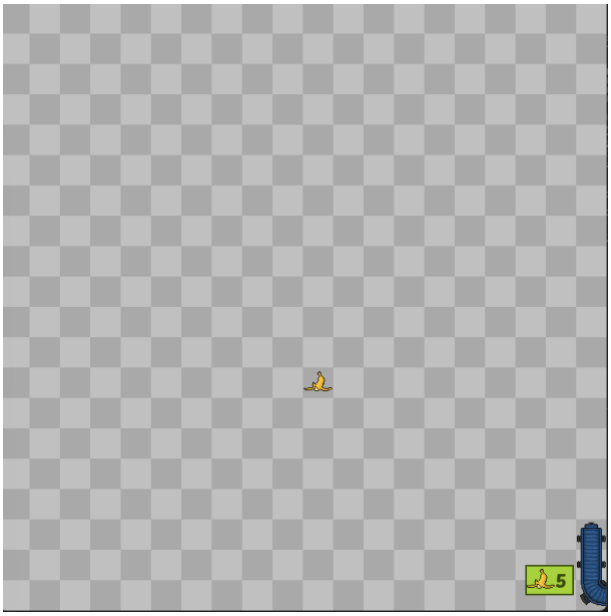
Game 94

Now as the Garbage Collector doesn't get longer as the game progresses the learning curve of the collecting agent is very steep. We think this is caused because the model has to just learn no to hit the wall and once that is achieved the rest is smooth sailing

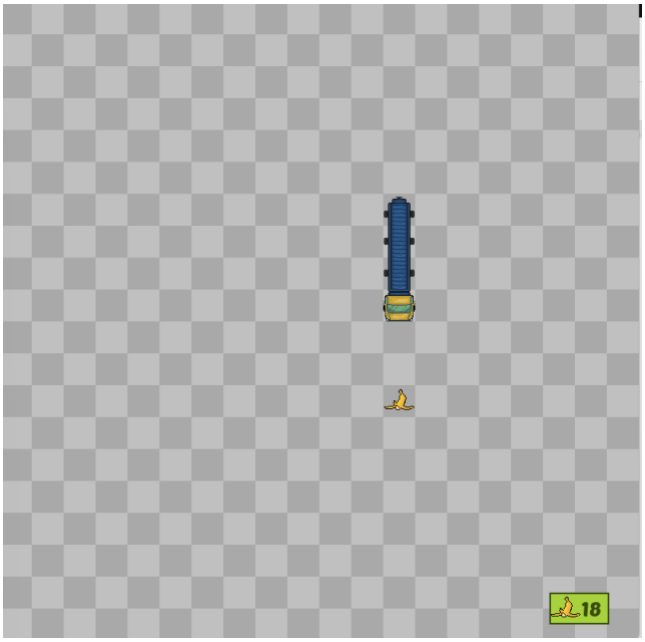




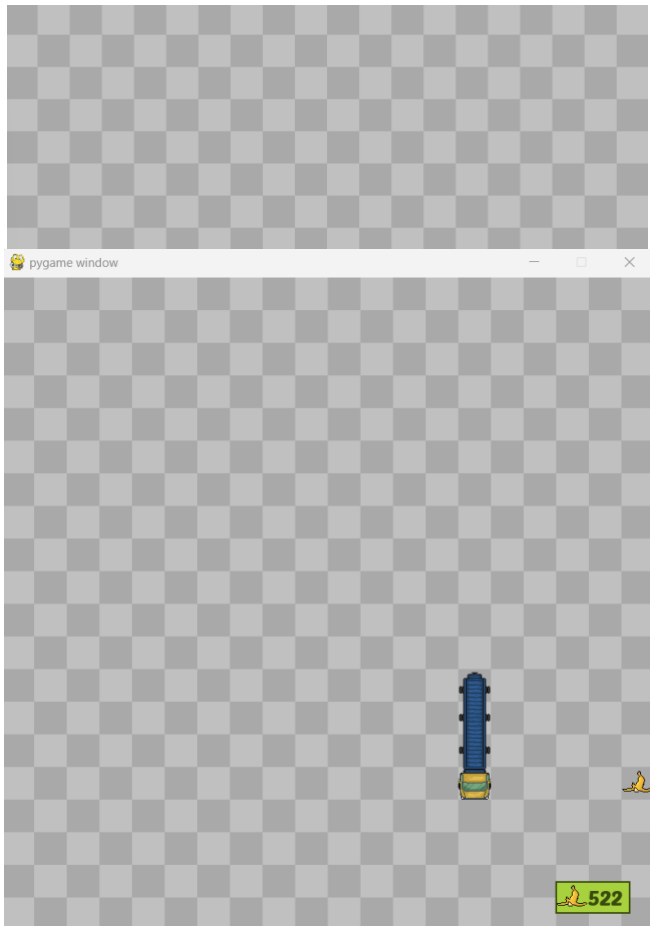
Game 3



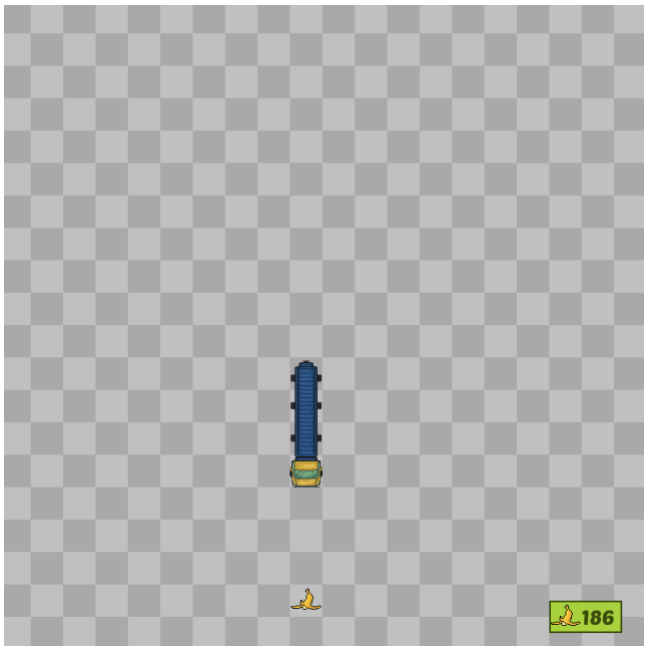
Game 46



Game 73

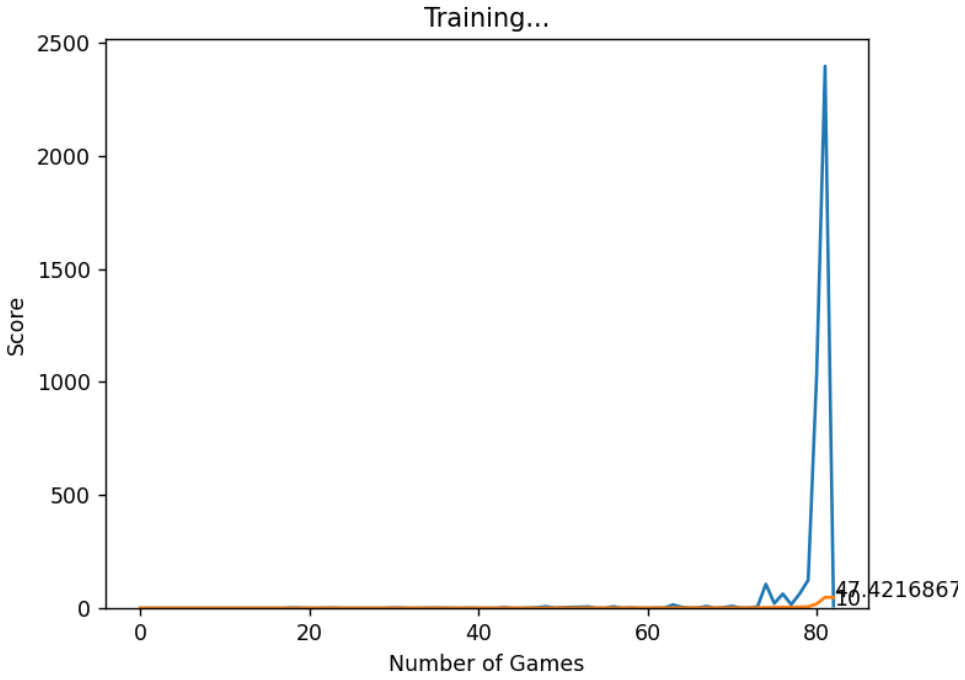


Game 84



Game 90

Game 100


Training...
Score
Number of Games
2500
2000
1500
1000
500
0
0
20
40
60
80
47.4216867
10

Future Enhancement

- Hierarchical Reinforcement Learning: Implement a hierarchical reinforcement learning system to enable the AI to learn and make decisions at multiple levels of abstraction, allowing it to learn more complex strategies and behaviours.
- Real-Time Obstacle Avoidance: Implement real-time obstacle avoidance using computer vision or sensor data to allow the AI to navigate around obstacles in real-time.
- Dynamic Difficulty Adjustment: Implement a dynamic difficulty adjustment system that adjusts the game's difficulty level based on the AI's performance, to ensure that the AI is consistently challenged but not overwhelmed.
- Transfer Learning: Enable the AI to transfer its knowledge and skills learned from Garbage collector simulation to other similar games or tasks, reducing the amount of training required for new tasks.

Conclusion

In conclusion, Deep Q-Learning has proven to be a successful method for training an AI to make a garbage collecting system. The AI is able to learn optimal strategies through trial and error, ultimately achieving high scores and outperforming human players. By leveraging the power of neural networks and reinforcement learning, the AI is able to make decisions based on the current state of the game and maximize its score.

References

Mnih, V., et. al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. https://doi.org/10.1038/nature14236

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