DEPARTMENT OF COMPUTER ENGINEERING

Mukesh Patel College Of Engineering
A.Y. 2024-25

ARTIFICIAL INTELLIGENCE PROJECT REPORT ON

Snake RL: AI-Powered Snake Game with Reinforcement Learning

Submitted By

Roll No: – B033
Name - Saakshi Jain

Under The Guidance of

Professor. Archana Nanade

TABLE OF CONTENT

LIST OF FIGURES

INTRODUCTION

Q-Learning helps an agent create a table to maximize rewards. While it works well in small environments, it doesn't scale when there are many states and actions. **Deep Q-Learning solves this by using a neural network** to estimate the values, keeping their relative importance. The network takes the initial state and gives the Q-values (possible actions) as output, making Deep Q-Learning more practical for larger environments.

In Deep Q-Learning, a combination of deep learning and reinforcement learning, the agent learns to make decisions based on the rewards received for each action. Unlike traditional supervised learning, where the algorithm needs labeled input-output pairs, in this scenario, the best action isn't predefined. Instead, the **agent explores different actions,** receiving positive or negative rewards based on the outcomes, and adjusts its predictions over time.

By feeding the game state into a deep neural network, **the agent predicts the best action to take**, optimizing for maximum rewards while avoiding actions that lead to penalties. This iterative learning process enables the agent to improve its strategy as it plays the game.

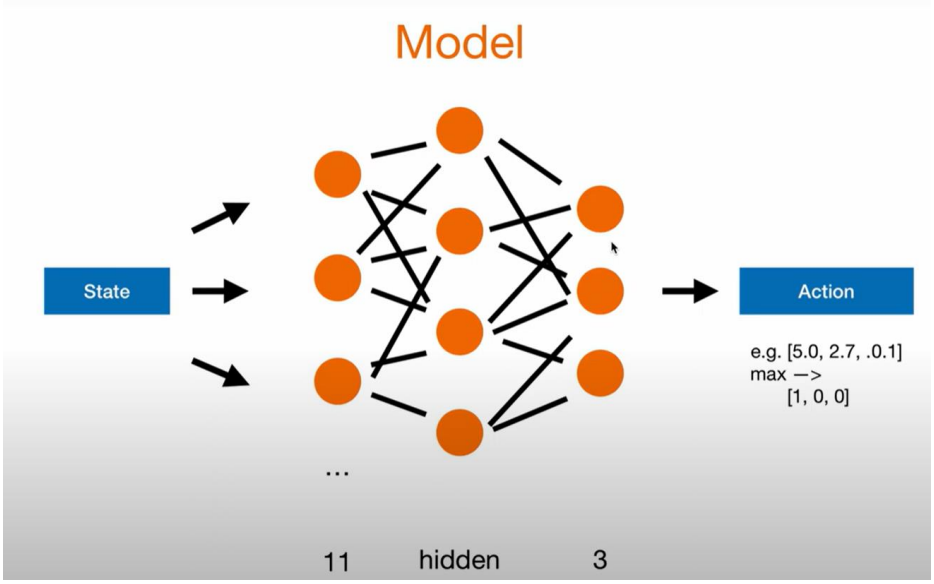

Figure 1: Deep Q-Learning

Q Learning One of many algorithms under computer science, falling into the category of reinforcement learning. The idea is to obtain the maximum reward possible given by something called an "agent" for the learner. When you give your dog a biscuit when he sits and you have taught him so, the biscuit is the reward, you are the agent, and your dog is the learner.

The general sense of the steps of reinforcement learning can be understood from the following:

- Environmental observation
- Considering which move to make
- Executing such action
- Receipt of the award or punishment
- Learn from experience. Repeat

What "Q" stands for in Q-learning is Quality, and it is an optimizing learning process that optimizes the strategy of choosing an action given different states. In a bit of technical language: the learner needs to properly map how valuable is an action A, given a state S: Q(A,S). Q Learning uses something called the **Bellman Equation** to work out the value of a state, and how good it would be to actually be in that state. The kinds of equations this expression gives are ones relating one state 1 and the value function in the next state 2, given which action was chosen in state 1.

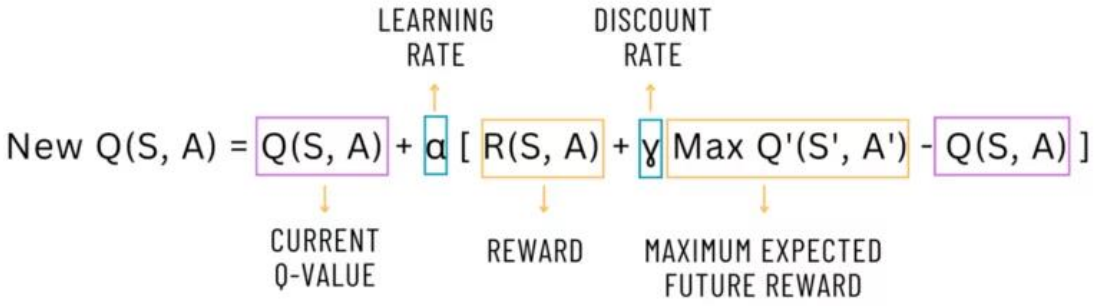

Figure 2: Formula for Deep Q-Learning

Q Update Rule Simplified:

$$Q = model.predict(state_0)$$

$$Q_{new} = R + \gamma \cdot max(Q(state_1))$$

Figure 3: Deep Q-Learning Simplified

LITERATURE SURVEY

Reference	Summary	Key Points
ResearchGate (2020)[1]	The paper provides a more accessible overview of reinforcement learning (RL), which is a type of machine learning wherein **an agent learns to decide based on observing its environment.** It emphasizes how RL can optimize long-term goals through trial and error without labeled data. The importance of experience as well as mechanisms in enhancing decision-making processes is also highlighted.	Core Concept: RL revolves around agents learning to maximize cumulative rewards through actions taken in various states. Learning Process: The agent **explores its environment, receives** feedback (**rewards** or penalties), and adjusts its strategies based on past experiences.
G. Porusniuc (2023)[2]	The thesis compares DQN and PPO reinforcement learning methods for **robotic control**, relevant for understanding how DQN could be applied to game environments.	Compares RL techniques for control tasks, including in-game behavior.
A. Finnson and V. Molnö (2019)[3]	This paper implements **deep Q-learning** to train an agent in Snake, analyzing the effects of different parameters like learning rates on performance. The importance of experience as well as mechanisms in enhancing decision-making processes is also highlighted.	Explores the **impact of parameter tuning on agent learning.**

ScienceDirect(2023)[4]	It reveals the transformative power of reinforcement learning and how it can be incorporated into **smart manufacturing,**allowing machines to make decisions based on data and optimise real-time production. The paper goes into apposite concepts of the power of technology, as well as the hurdles to apply such advanced systems into a real-world environment.	RL helps machines learn from their environment by receiving feedback based on their actions, allowing for smarter decision-making.Applications: From robotics to supply **chain management**, to **production scheduling**, RL has very diverse applications within manufacturing and can easily be changed due to shifting demands.
Ieeexplore (2011)[5]	This paper outlines the basics of reinforcement learning, including models, algorithms, and applied examples. It focuses on how RL enables **intelligent decisions through interaction feedback** with an environment.	RL improves real-valued decision policies by **trial-and-error learning**. The key algorithms used are Q-learning and deep reinforcement learning.
Springer[6]	It shows the games where RL works and fails, from simple Backgammon to much more complex Chess and Go games. The challenges that appear to be unique to the game setting, such as **state spaces of high dimension and time-varying opponent strategy.**	Games offer a multitude of challenging scenarios for applying RL, thereby improving the algorithms under different conditions.Success stories include **Backgammon** (TD-Gammon), while Chess remains challenging for RL.

D. Dwibedi (2020)[7]	Discusses using Deep Reinforcement Learning for playing various games, highlighting the application of **deep Q-networks and actor-critic models** in gaming environments.	Provides insights into different RL approaches for game environments.
Y. Z. Kun Shao et al. (2020)[8]	The survey investigates the use of Deep Reinforcement Learning in video games, exploring **how techniques like DQN** have been applied to complex gaming environments.	Analyzes RL in video games, including performance improvement techniques.
Ieeexplore[9]	This paper explains the application of DRL for enabling autonomous driving systems to make real-time decisions. This paper identifies some main challenges that DRL faces in the domain of autonomous driving: traffic complexity, safety, and interpretability. The paper reviews several DRL algorithms, identifying successes and outlining further research areas for problems such as more realistic simulation environments and improvements in safety	Complex Environments: Autonomous driving needs DRL to handle complex and unpredictable traffic scenarios. Safety Issues: Most DRL models suffer from safety issues in practical deployment. Algorithmic Improvements: Techniques like **transfer learning and hierarchical RL** are essential to advance performance.

	guarantees.	
Ieeexplore(2021)[10]	The paper offers a Deep Q-Learning approach in mastering the Snake game. The authors detail the design of the RL framework, documenting the training and testing phases. Further emphasis was placed on **fine-tuning neural network hyperparameters-essential** for optimizing the agent's performance in playing the game.	Deep Q-Learning Application: Successfully applies Deep Q-Learning for decision-making in Snake. Framework Definition: Clearly defines RL components essential for Snake gameplay. **Hyperparameter Tuning**: Emphasizes the importance of optimizing network parameters to reach target performance.
Abraham,Darryl(2023)[11]	This thesis looks at how a Deep Q-Network performs when applied to the Snake game and evaluates its generalizability to slight modifications in the rules of the game. The study considers the generalizability of DQNs, if a model trained on an original game can adapt well to slight variations such as targets moving or obstacles. Challenges in reinforcement learning are considered, such as reward structure tuning and strategy adaptation.	DQN Generalization: Evaluates how DQNs trained in a stable environment perform when introduced to variations in game settings. Reward Structure: Demonstrates **importance of reward design** to encourage optimal actions in complex environments. Adaptation Strategies: **Demonstrates the capa - bilities and short - comings of DQNs** in learning adaptive strategies in evolving situations.

PROBLEM STATEMENT

The goal of this project is to develop an AI bot that can learn to play the classic Snake game autonomously using Reinforcement Learning (RL), specifically the **Deep Q-Learning (DQL)** algorithm. In this project , the AI will begin with no predefined rules or strategies for the game. Instead, it must learn optimal behavior through rewards associated with its actions, without human guidance. The system will be trained to maximize the score by eating as many apples as possible without colliding with walls or its own body.

This model is meant to train a bot playing optimally, learning from experience to survive longer thus consuming more apples. Further developments may apply Convolutional Neural Networks, such that the AI "sees" the game better; it would prevent the tendency of self-enclosing and would make the general behavior more efficient.

The Key Expectations include:

- **Learning by Exploration:** The AI will need to try out various actions to see which ones work.

- **State Representation Handling:** The state of the snake should represent environment information such as food items, threats, and even the direction of the snake.

- **Experience Replay for Rapid Learning:** The bot should remember and replay experiences to learn quickly.

- **Performance Issues:** Avoid getting stuck in its own body, which becomes more complex as the snake grows longer.

- **Optimizing Training Parameters:** Determining the appropriate batch sizes, discount

rates, and exploration strategies to train the AI appropriately for playing multiple games.

PROPOSED SYSTEM

We should apply a 3-tier structure for the snake game based on Q-learning, for example, as shown below:

1. **Agent (Controller)**

   In turn, this system interacts through the brain with the game and model to be an agent. This agent acquires the state of a game and uses the model to decide on the kind of action to take after which it continues to operate in the environment of its game. All the agents learn based on rewards for certain actions undertaken to advance with time.

   - **State:** it would describe the game state, and the snake agent is aware of where it is against the food and obstacles.

```
State (11 values)

[danger straight,  danger right, danger left,

 direction left, direction right,
 direction up, direction down,

 food left, food right,
 food up, food down
]
```

Figure 4: Pseudo Code for State of Agent

- **Action:** Based on the agent's model that predicts where the state might go, up, down, left, or right, it applies the model.

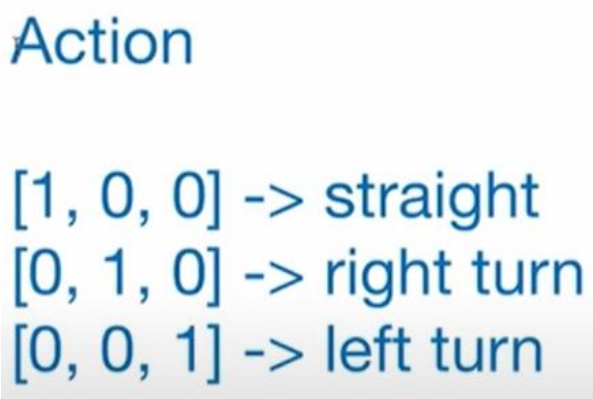

Figure 5: Pseudo Code for Actions of Agent

- **Reward:** the agent receives a reward for every action it may take-be it towards food or away from obstacles. The agent has also lost some penalty at loss; hitting into the wall or itself.

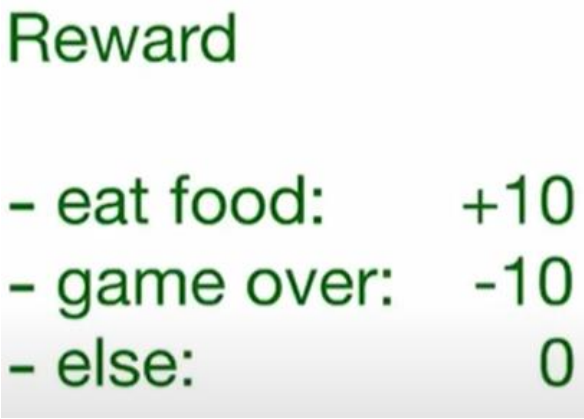

Figure 6: Pseudo Code for Reward Structure of Agent

2. **PyTorch Deep Q-Network Model**

- We approximate the optimal policy using a **Deep Q-Network** for the agent. The main input into the model is the game's current state and its corresponding output-best action to be taken.
- **Model Prediction:** Based on state, predicts the optimal possible action to maximize the reward.

- **Training:** The model is trained using experiences the agent collects throughout its play, updating its weights with an optimization decision. Use the Linear_QNet class to create the **DQN model and the QTrainer class** to train it with experience replay, allowing the agent to learn continuously by adjusting weights through rewards from gameplay. This integration ensures improved decision-making and optimal action selection over time.

3. **Front-end (Game Environment)**

A front-end will be a graphically oriented interface of a game based on the Pygame library. It will deliver the environment in which an agent operates and will let the user see the graph.

- **Pygame:** This one is used for creating the snake game by displaying the game board, snake, food, and handling the user's input.
- **Feedback Loop:** The frontend gives the agent feedback about the state, reward, and game-over status for each action.

Below is an architecture diagram to describe the interaction of an Agent, Model, and Frontend:

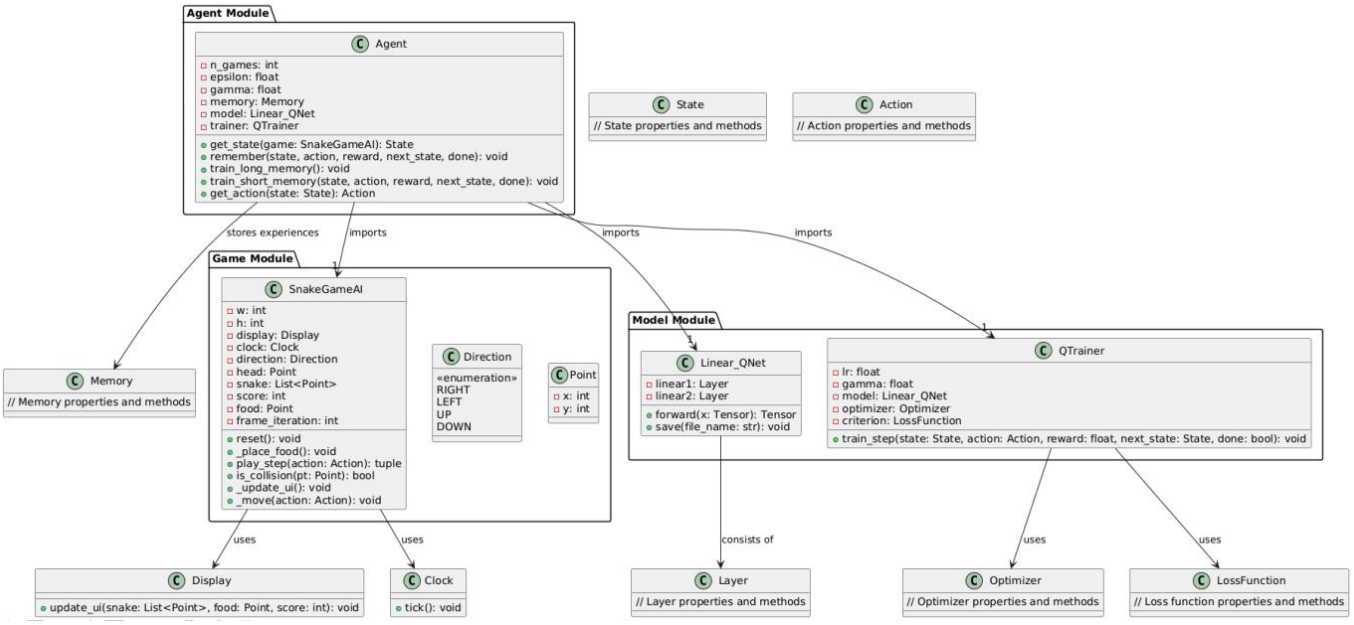

Figure 7: Pictorial Representation of the Architecture

IMPLEMENTATION

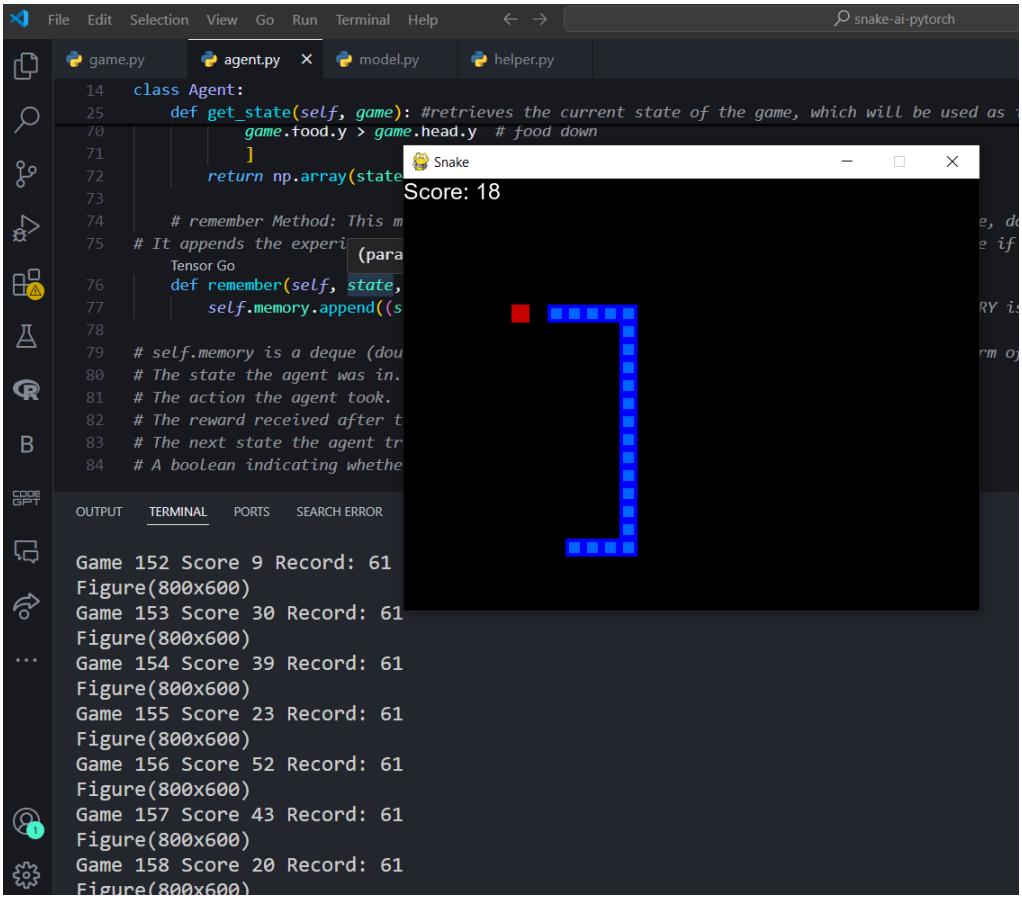

Figure 8: Implementation of the code

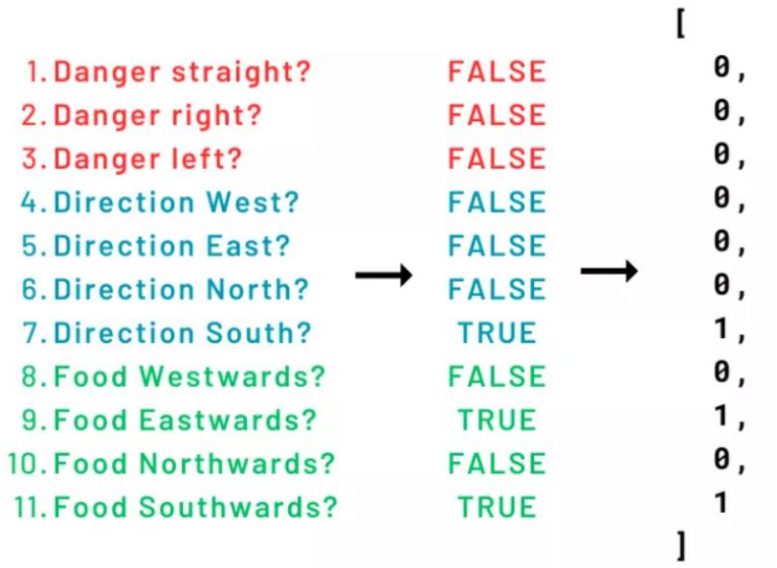

Figure 9: Conditionals of the Model

OUTPUT-Project Images

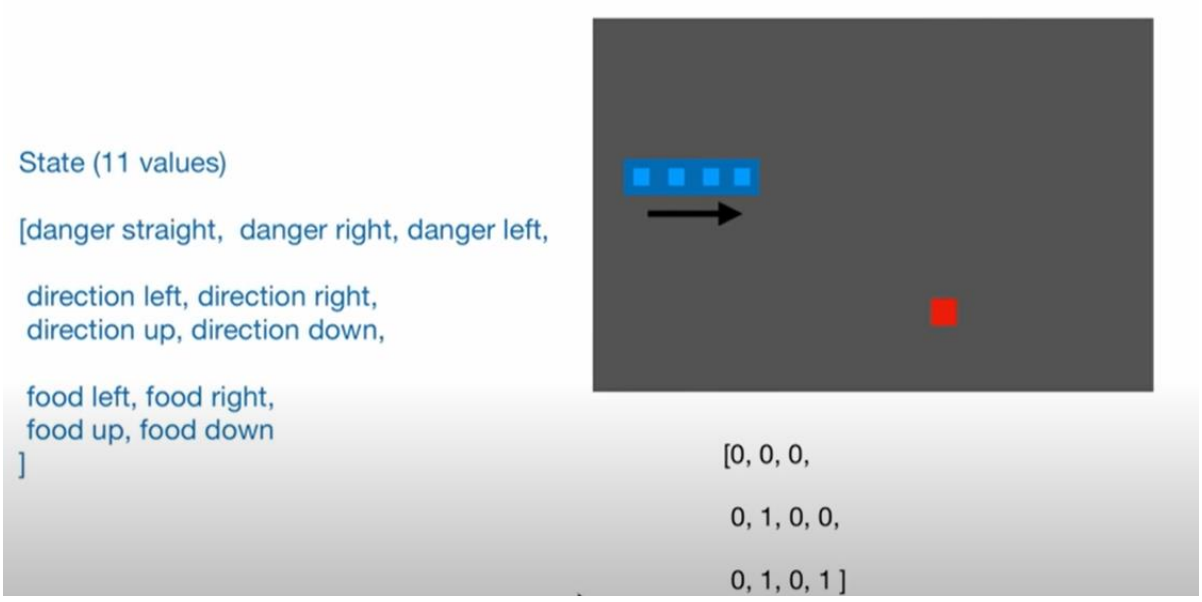

Figure 10: Output image 1

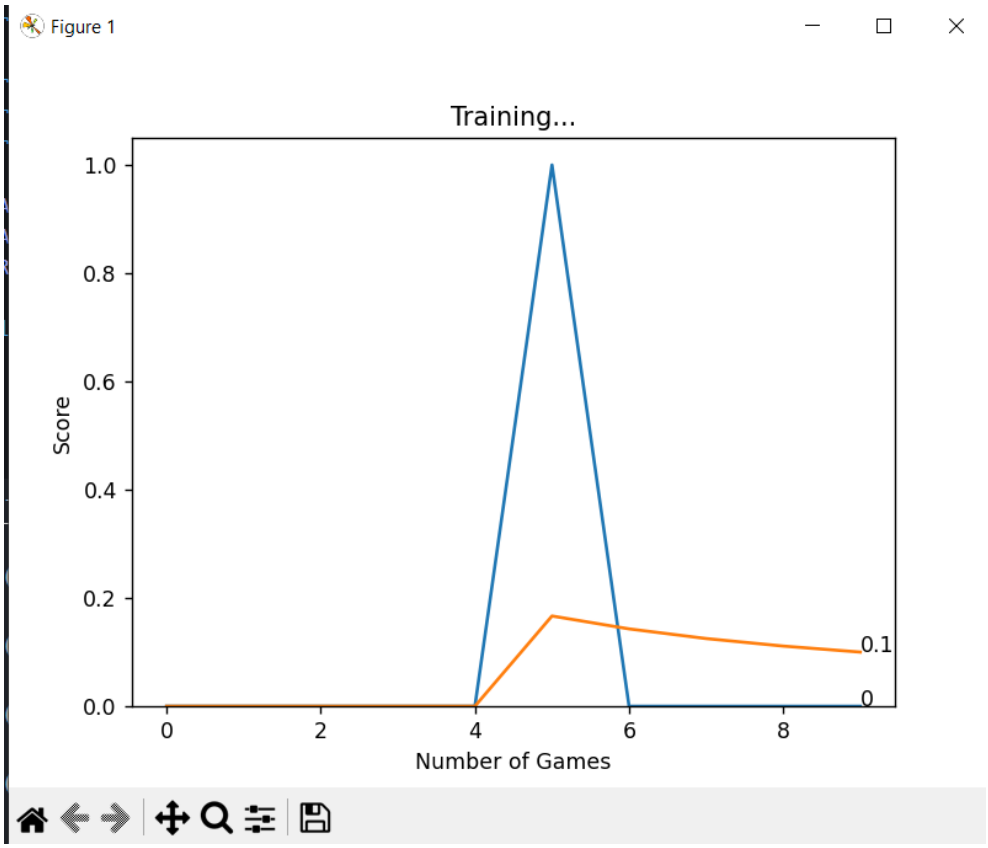

Figure 11: Output image 2

CONCLUSION

The proposed snake game using Q-learning and Deep Q-Network (DQN) reinforcement learning presents a promising approach to developing an intelligent agent capable of improving its gameplay over time. By integrating a tiered architecture consisting of the Agent, Model, and Game Environment, the system enables continuous learning through interaction, rewards, and feedback loops. The use of PyTorch-based DQN ensures adaptability and scalability, allowing the model to evolve as it trains on more experiences.

Additionally, leveraging Pygame for the frontend interface provides a seamless environment for both agent training and user interaction. This integration of a simplified yet powerful architecture allows for efficient gameplay learning, while maintaining clarity in the flow of actions, states, and rewards. The project has the potential to significantly enhance the development of AI-based gaming agents, offering insights into reinforcement learning applications beyond simple games.

Overall, this system can be a valuable tool for both educational and experimental purposes in the field of artificial intelligence, making it an ideal candidate for further research and development in autonomous gaming agents.

REFERENCES

[1] Muddasar Naeem,Syed Tahir Hussain Rizvi , Antonio Coronato ,"Understanding Reinforcement Learning: Principles and Applications" 2020. [Online].

Available:https://www.researchgate.net/publication/347004818_A_Gentle_Introduction_to_Reinforcement_Learning_and_its_Application_in_Different_Fields .

[2] G. Porusniuc, "A Comparative Study on Reinforcement Learning Methods for Learning Robot Control Behavior," Master's thesis, University of Eastern Finland, 2023. [Online]. Available: https://erepo.uef.fi/bitstream/handle/123456789/30440/urn_nbn_fi_uef-20231083.pdf?sequence=1.

[3] A. Finnson and V. Molnö, "Deep Reinforcement Learning for Snake," Master's thesis, Lund University, 2019. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1342302/FULLTEXT01.pdf.

[4] Ashish Kumar Shakya, Gopinatha Pillai, Sohom Chakrabarty, "Reinforcement learning for smart manufacturing" 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423009971 .

[5]Wang Qiang; Zhan Zhongli, "Reinforcement Learning: Models, Algorithms, and Applications

" 2011. [Online]. Available:https://ieeexplore.ieee.org/document/6025669 .

[6] Istvan Szita, "Reinforcement Learning in Games: Techniques and Challenges" [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-27645-3_17 .

[7] D. Dwibedi, "Playing Games with Deep Reinforcement Learning," Master's thesis, 2020. [Online]. Available: https://debidatta.github.io/assets/10701final.pdf.

[8] Y. Z. Kun Shao and Z. Tang, "A Survey of Deep Reinforcement Learning in Video Games,"

arXiv.org, 2020. [Online]. Available: https://arxiv.org/pdf/1912.10944.pdf.

[9] Beakcheol Jang and Myeonghwi Kim, :" Deep Reinforcement Learning for Autonomous Driving: Challenges and Future Directions" [Online]. Available:https://ieeexplore.ieee.org/document/8836506 .

[10] Alessandro Sebastianelli; Massimo Tipaldi; Silvia Liberata Ullo, "Deep Q-Learning of Snake Game AI" 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9480232 .

[11]Abraham,Darryl, "Snake Game: Evaluating Deep Q-Learning" 2023. [Online]. Available:https://pure.tue.nl/ws/portalfiles/portal/307051253/thesis_BDS_Juli_D._Abraham.pdf .