International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 98-101, December-2024

https://doi.org/10.47001/IRJIET/2024.812014

Reinforcement Learning-based Snake Game

[1]Dr. Lokesh Jain, [2]Vasant Kumar

[1]Associate Professor, Department of IT, Jagan Institute of Management Studies, Rohini, Delhi, India
[2]Department of IT, Jagannath University, Haryana, India
Authors E-mail: [1]lokesh.jain@jimsindia.org, [2]vasantkumar6212@gmail.com

***Abstract* - This project reimagines the classic Snake Game with exciting twists, combining nostalgic gameplay, modern AI training techniques, and retro-style visuals. It features a vibrant and engaging experience where players control a snake to eat food, grow in size, and avoid obstacles. Unique elements like red food that increase your score and yellow food with penalties add strategic challenges. The game offers multiple modes, including a traditional play mode, an AI training mode where a Deep Q-Learning Agent learns to play through trial and error, and a challenge mode with added complexities. The AI leverages a neural network to make decisions, improving over time and showcasing its progress through real-time graphs. A retro-inspired menu enhances usability, allowing players to switch between modes seamlessly. Additionally, the game tracks and saves progress for both players and the AI, ensuring sessions can be resumed anytime. With its lively graphics, immersive sound effects, and innovative AI integration, this project blends fun, learning, and nostalgia into a single, user-friendly package.**

Keywords: Deep Q-Learning, AI-based Game, Reinforcement Learning.

I. INTRODUCTION

This project showcases the integration of artificial intelligence with the classic Snake game, featuring a reinforcement learning agent that uses a Deep Q-Network (DQN) to learn strategies for maximizing its score. Operating in a grid-like environment, the agent controls the snake to collect food, navigate efficiently, and avoid collisions with walls or its own body. The learning process involves observing the game state—such as the snake's position, direction, and food location—and deciding whether to move straight, turn left, or turn right based on action values predicted by the DQN. By leveraging replay memory to store past experiences and refining decisions using the Bellman equation, the agent improves its performance over multiple rounds of gameplay. Progress charts visually track the agent's growing competence, demonstrating its ability to adapt through rewards and penalties. This project highlights the synergy between classic gaming and advanced machine learning, showcasing how AI can master complex tasks through continuous learning and optimization.

II. LITERATURE REVIEW

The integration of artificial intelligence in classic games like Snake provides a compelling platform for exploring and applying underpinning learning ways. This design leverages a Deep Q-Network (DQN) to make upon foundational generalities and studies in underpinning literacy, neural networks, and game AI. Underpinning literacy, as introduced by Sutton and Barto, focuses on agents learning optimal actions through relations with their terrain to maximize accretive prices. The Snake game, with its simple grid-grounded terrain, aligns impeccably with the conditions of underpinning literacy's state- action- price cycles. The DQN armature, developed by Mnih et al. (2015), combines Q-literacy with deep neural networks to compare Q- values in high- dimensional spaces. By integrating experience renewal and target networks, it resolves stability issues in standard Q-literacy, enabling this design to use neural networks to prognosticate action values and renewal memory for effective training.

Classic games like Snake have long been marks for AI algorithms, furnishing structured yet grueling surroundings for experimenting with spatial navigation, handicap avoidance, and resource collection. The design incorporates an epsilon-greedy policy to balance disquisition(trying new conduct) and exploitation(choosing known good conduct), a well- proved system for optimizing dynamic literacy. The Bellman equation, a foundation of Q- literacy, serves as a frame for streamlining Q- values grounded on unborn prices, with temporal difference literacy styles enhancing real- time decision- timber. Operations of AI in games, from simple mystifications to complex strategies like Go and StarCraft, demonstrate how controlled surroundings like Snake give openings for applying and enriching AI algorithms.

By drawing from these established methodologies, the design implements a DQN- grounded Snake- playing agent that effectively learns and improves through iterative gameplay. It highlights the practical operation of underpinning learning generalities and contributes to the growing field of AI- driven gaming, incorporating ultramodern machine literacy ways with classic gaming challenges.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 98-101, December-2024

https://doi.org/10.47001/IRJIET/2024.812014

III. METHODOLOGY

This project's methodology revolves around training an AI agent to play the Snake game effectively using a Deep Q-Network (DQN). Here's a simplified and clear explanation of the process:

Setting up the Game Environment: The Snake game is built on a grid where the agent (the snake) interacts by observing its surroundings. The agent gets information about its position, direction, the location of the food, and obstacles like walls or its own body. It decides between three actions: move straight, turn left, or turn right.

Using a Deep Q-Network (DQN): The core of the AI is a neural network that helps the agent predicts the best action to take in any situation. It has:

- Input Layer: Processes the current game state.
- Hidden Layers: Learns patterns and strategies for better decision-making.
- Output Layer: Provides scores (Q-values) for each action, guiding the agent.

Training the AI:

- **Reward System:** The agent gets rewards for eating food and penalties for crashing or wasting time.
- **Experience Replay:** The agent remembers its past actions and outcomes in a replay memory. It learns by revisiting these experiences, which helps it improve steadily.
- **Target Network:** A secondary network calculates stable learning goals (target Q-values), updated periodically for consistency.
- **Learning Process:** The agent updates its understanding of rewards using a mathematical formula (the Bellman equation). It reduces errors between predicted and actual rewards using a loss function, optimized with tools like gradient descent.

Balancing Exploration and Strategy: The agent starts by experimenting with random actions (exploration) to learn the game. Over time, it shifts toward relying on its learned strategies (exploitation) to make smarter moves.

Tracking and Testing Progress: The agent's performance is measured by its scores during training. Visual graphs show how it improves over time. After training, the agent is tested in different scenarios to see how well it adapts to challenges, using metrics like average score and consistency.

Tools and Technology: The project uses Python, with libraries like NumPy for managing data, TensorFlow or PyTorch for building the neural network, and Matplotlib for creating graphs and visualizations. This methodology ensures the agent learns the game step-by-step, refining its strategies through trial and error. It's a great example of how AI can adapt to dynamic environments and develop intelligent behavior over time.

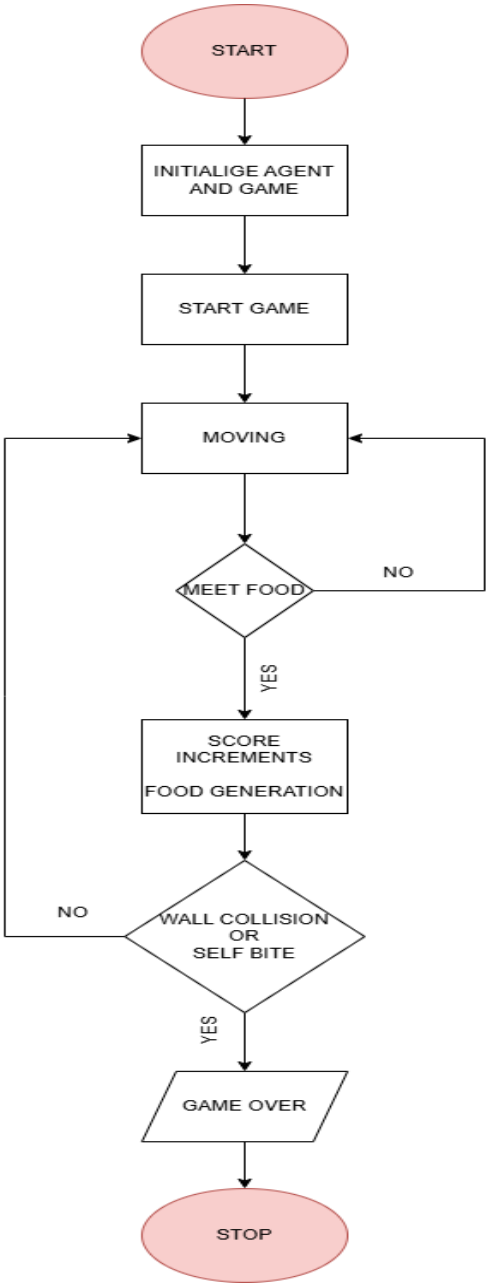

Figure 1: Flow Chart

IV. RESULTS (INPUT AND OUTPUT)

Main Menu: The main menu allows the user to select modes whether the user wants to play a classic snake game or wants to see a perfect snake game. In which the agent which is trained to control the snake will play the game, where the agent tries to increase its score to maximum by training itself with every iteration, learning from its previous experience.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 12, pp 98-101, December-2024

https://doi.org/10.47001/IRJIET/2024.812014

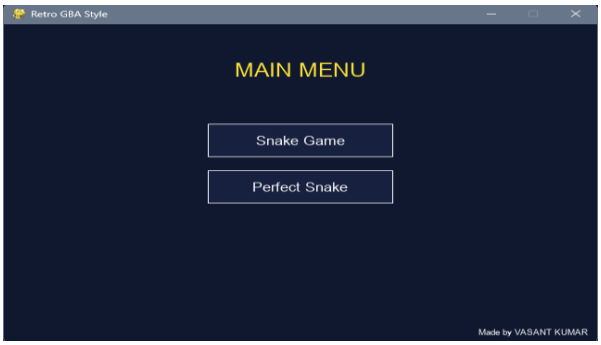

Figure 2: Main menu

Human Game Menu: In the human game menu the user can play the game which has two game modes, first that is the normal or classic snake game, and second is the snake game that involves the negative reward if the user reaches a score of 5 then the user starts getting food, one that increases the score and vice versa for the second food.

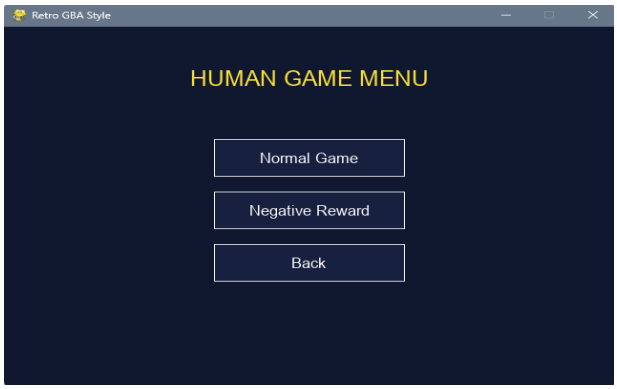

Figure 3: Human game menu

Game Screen: This is the main interface that represents the game which is our classic snake game, which displays its score at the top left corner which keeps on increasing when we eat the red food and vice versa when we eat the yellow food ,the score refreshes when the snake dies either by hitting the wall or when he bites himself(i.e when the head of the snake collides with the body of itself).

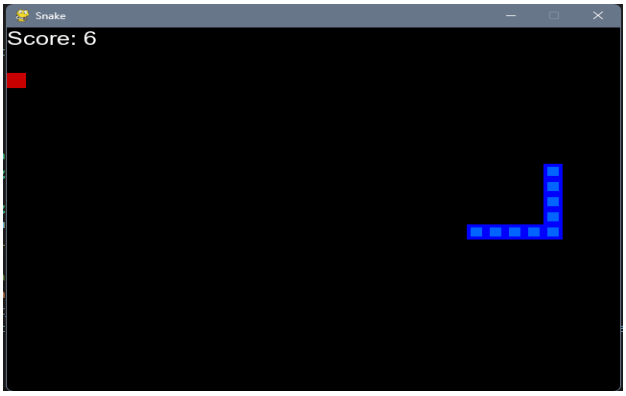

Figure 4: Game Screen

Plotting: The map continuously keeps track of the average score that is the score by the number of games it has played in total. The X axis represents the number of games it has trained for whereas the Y axis keeps the track of the score.

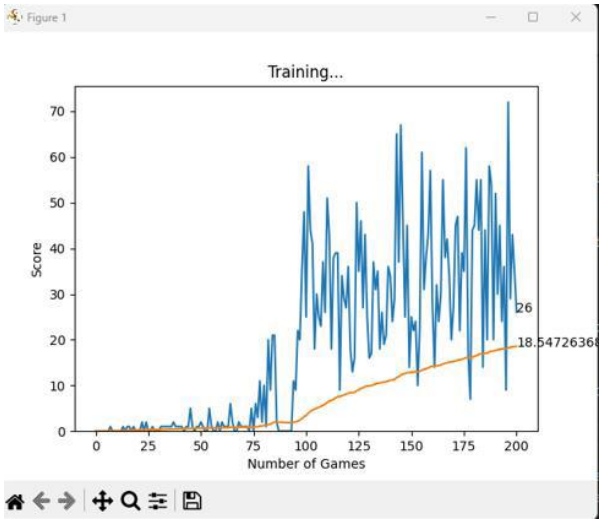

Figure 5: Plotting

V. LIMITATIONS

Limitation of the proposed game is as follows:

- **Overfitting:** The DQN model may overfit to the training data.
- **Non-stationary environments:** The current training approach may not be well-suited for non-stationary environments.
- **Scalability:** The agent may not scale well to more complex environments or large-scale applications.
- **Exploration-exploitation trade-off:** Balancing exploration and exploitation can be challenging.
- The game modes are not modularized or implemented as separate classes or modules, which could make future extensions more challenging.
- **Unoptimized Rendering:** The program redraws all menu elements on every frame, which could lead to performance issues on lower-end systems, especially if the menu becomes more complex.

VI. CONCLUSION

In summary, this project highlights the successful use of reinforcement learning to train an AI agent to play the classic Snake game with a Deep Q-Network (DQN). The agent learns to navigate the grid, eat food, and avoid obstacles by continuously refining its strategies through trial and error. By leveraging DQN, experience replay, and the Bellman equation, the agent is able to make smart decisions and adapt its behavior over time.

ISSN (online): 2581-3048
Volume 8, Issue 12, pp 98-101, December-2024
https://doi.org/10.47001/IRJIET/2024.812014

However, while the project demonstrates the capabilities of reinforcement learning in a simple game, there are some challenges that still need to be addressed. Issues like overfitting, finding the right balance between exploration and exploitation, and scaling to more complex environments need attention for future improvements. Additionally, there are areas for enhancement, such as game features, error handling, and performance optimization, to improve the overall user experience.

REFERENCES

[1] Reinforcement Learning Concepts: Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd Edition). *MIT Press*.

[2] Deep Q-Learning: Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). *Human-level control through deep reinforcement learning*.

[3] Game Development: Pygame: For creating the Snake game interface. *Pygame Documentation*: https://www.pygame.org/docs/

[4] Visualization Tools: Matplotlib: Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.

[5] Learning from GitHub Projects: Various GitHub repositories related to snake game AI or reinforcement learning served as references for structure and implementation patterns:

[6] OpenAI Gym: Brockman, G., Cheung, V., Pettersson, L., et al. (2016). OpenAI Gym. *arXiv preprint* arXiv:1606.01540.

Citation of this Article:

Dr. Lokesh Jain, & Vasant Kumar. (2024). Reinforcement Learning-based Snake Game. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 8(12), 98-101. Article DOI https://doi.org/10.47001/IRJIET/2024.812014

****

