AI-Based Car Racing Game

CSE440.1

Group 6

Submission Date: 01/12/2024

Nowshin Nawal Mim	1911054042
Mehrab-E-Rashid Khan	1931071642
S.M Shamim Sarowar Pranto	1931514642

Introduction

Artificial Intelligence (AI) has revolutionized the gaming industry, introducing new ways to develop intelligent systems capable of learning and adapting autonomously. This project explores the application of AI in a car racing game by leveraging the NEAT (NeuroEvolution of Augmented Topologies) algorithm. NEAT is a genetic algorithm that evolves neural networks to optimize their performance over successive generations.

Objective

The objective of this project is to train virtual cars to navigate a predefined track autonomously without colliding with the boundaries. Using radars as sensors, the cars collect real-time data about their surroundings and make decisions based on neural network outputs. The NEAT algorithm facilitates the evolution of these networks, enabling cars to learn effective navigation strategies through iterative simulations.

This project demonstrates the integration of AI and game development, combining the Pygame library for visual representation and NEAT-Python for neural network training. It serves as a practical implementation of reinforcement learning concepts, showcasing how autonomous agents can be trained to perform complex tasks in dynamic environments.

Methodology

This project combines game mechanics with AI using Pygame for visualization and the NEAT algorithm for training. The steps include:

1. **Game Setup**:
   - A racing track is designed as the environment, and a car sprite is programmed to move, rotate, and detect boundaries.
2. **AI Integration**:
   - The car uses NEAT to evolve neural networks. Sensors (radars) detect obstacles, and their data is fed as inputs to the network. Outputs decide the car's movement (left, right, or straight).
3. **Training Process**:
   - The fitness function rewards cars for avoiding collisions and navigating effectively. Poor performers are removed, and successful ones evolve through mutation and reproduction.
4. **Simulation**:
   - Cars are tested over multiple generations. Neural networks improve over time, learning better navigation strategies.

This approach ensures an iterative improvement process, creating an autonomous car capable of navigating a predefined track.

Impact of the Project

This project demonstrates the practical application of Artificial Intelligence in a dynamic and interactive environment, showcasing how neural networks can learn and adapt to complex tasks. The key impacts include:

1. **Learning Reinforcement**:

- Highlights how AI agents can use evolutionary algorithms like NEAT to improve performance over time.

2. **Skill Development**:
    - Provides hands-on experience in game development, neural networks, and AI training processes, bridging the gap between theoretical concepts and real-world applications.
3. **Real-World Applications**:

    Serves as a foundation for autonomous systems, such as self-driving cars, where sensors and decision-making algorithms are crucial.

4. **Inspiration for Future Projects**:

- Encourages the development of more complex AI-driven applications in games and simulations, fostering innovation in AI and game development.

5. **Educational Use**:

- Offers a simplified, interactive way to learn about AI, making it an excellent teaching tool for students and enthusiasts.

Results

The project successfully demonstrates the application of the NEAT algorithm to train a virtual car to navigate a predefined track autonomously. Over successive generations, the neural networks evolve to improve navigation, avoid collisions, and maximize fitness scores. The cars effectively utilize radar-based input to make decisions and adapt their movements, showcasing the ability of AI to learn and perform complex tasks. This highlights the potential of combining game development with evolutionary algorithms for creating intelligent systems.

REFERENCE

- Stanley, K. O., & Miikkulainen, R. (2002). *Evolving Neural Networks through Augmenting Topologies*. Evolutionary Computation. This paper introduces the NEAT algorithm and explains how it can evolve neural networks dynamically to optimize learning.

- GitHub - NEAT Self-driving Car AI: Projects like this showcase the implementation of NEAT for game environments.

https://github.com/amalsam/Self-Driving-Car-Simulation

https://github.com/Uljibuh/AI_DRIVE_CAR_NEAT

- Zawidzki, M., & Zawidzki, M. (2021). "Using Neural Networks to Navigate Virtual Environments."

- CodeBullet's YouTube Channel: Offers tutorials and insights on AI-based games and NEAT.

- Pygame Documentation: Essential for understanding the game development environment and functionality used in your project.

- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*.

Provides foundational knowledge on AI strategies like NEAT.

Conclusion

This project successfully demonstrates the application of the NEAT algorithm to train an autonomous car in a simulated environment. By combining evolutionary neural networks with game mechanics, the system learns to navigate a track, adapt to obstacles, and improve performance over generations. The results highlight the potential of using AI to solve complex real-world problems, such as self-driving cars, through continuous learning and adaptation. This project not only reinforces the practical application of AI but also serves as a foundation for more advanced autonomous systems in the future.