RL Shooter Game

Alexandru Andrei
andrei.alexandru2@s.unibuc.ro

Ţelicov Letiţia Ioana
letitia-ioana.telicov@s.unibuc.ro

Tîrîlă Patric Gabriel
patric-gabriel.tirila@s.unibuc.ro

Codarcea Alexandru Christian
alexandru-christian.codarcea@s.unibuc.ro

Abstract

This project presents a competitive team-based shooting game developed in Unreal Engine, in which two teams of AI-controlled agents engage in combat. Each agent is trained using reinforcement learning to make decisions such as movement, aiming, and shooting based on environmental inputs. The learning process is guided by a reward function that encourages strategic behavior, including cooperation, tactical positioning, and efficient targeting. The system is designed to explore how intelligent, adaptive behaviors can emerge in multi-agent adversarial settings without pre-programmed tactics. The results demonstrate the potential of reinforcement learning to drive complex decision-making in real-time environments.

1 Introduction

What is the problem you are trying to solve? Why did you choose it? What is your opinion about it?

The problem we are trying to solve is improving the behavior of bots in shooter games using reinforcement learning (RL). Many existing games feature bots with scripted or predictable actions, which limits the realism and challenge they offer. We chose this topic because we believe RL can enable more adaptive and intelligent behavior in virtual agents, creating a more engaging player experience. Games like *Counter-Strike* or *Call of Duty* include bots, but they often lack dynamic decision-making. By applying RL, we aim to create agents that learn from their environment and adjust their strategies over time. In our opinion, this approach has great potential to revolutionize AI in gaming and offer new ways to develop smarter, more human-like opponents. It also helps us explore how complex behaviors can emerge from trial-and-error learning in real-time environments.

2 Related Work

The integration of artificial intelligence (AI) in gaming has advanced notably with the advent of reinforcement learning (RL). Early game AI depended on scripted behaviors and finite state machines, limiting adaptability. The introduction of deep reinforcement learning, particularly the Deep Q-Network (DQN) by Mnih et al. (2015), enabled agents to learn optimal policies directly from high-dimensional inputs.

Further developments, such as the Proximal Policy Optimization (PPO) algorithm by Schulman et al. (2017), enhanced the stability and efficiency of training RL agents in complex environments. These advancements have paved the way for AI agents capable of learning intricate behaviors in dynamic and high-fidelity game settings.

Recent research has demonstrated the application of RL in various gaming scenarios. For instance, the ViZDoom platform has been instrumental in training agents for first-person shooter environments using visual inputs. Similarly, the WILD-SCAV benchmark provides an open-world FPS environment to evaluate RL algorithms' performance in complex tasks. These platforms underscore the potential of RL in enabling agents to learn navigation, combat, and strategic decision-making in real-time.

Several commercial games have explored large-scale battle simulations, offering insights into AI behavior and crowd dynamics:

- **Totally Accurate Battle Simulator (TABS):** Developed by Landfall Games, TABS is a physics-based strategy game that allows players to simulate battles with units exhibiting exaggerated physics and behaviors. The game's sandbox mode and unit creator facilitate experimentation with various combat scenarios, providing a platform to observe emergent behaviors.

- **Ultimate Epic Battle Simulator (UEBS):** Created by Brilliant Game Studios, UEBS enables players to orchestrate massive battles involving thousands of units. While the game emphasizes scale, it also highlights the challenges of AI pathfinding and decision-making in large-scale simulations. Community discussions have pointed out limitations in AI behavior, such as units taking repetitive paths or failing to respond dynamically to changing battle conditions.

Our project draws inspiration from these titles, aiming to enhance AI adaptability and decision-making in large-scale battle scenarios. By leveraging reinforcement learning techniques, we seek to develop agents capable of learning and adapting to complex environments, thereby addressing some of the limitations observed in existing simulations.

3 Method

This project presents a sci-fi shooter game developed in Unreal Engine 5, featuring two AI-controlled teams trained via reinforcement learning using the Learning Agents plugin. Red and Blue agents follow distinct behavioral strategies, with their actions refined through episodic in-game training. The game includes a custom map with interactive elements, a dynamic user interface, and real-time visual effects. Further technical details and AI integration are explained in the subsequent sections.

3.1 AI Model Architecture

To develop intelligent in-game behavior, we employed the Learning Agents plugin within Unreal Engine 5, which facilitates the training and deployment of reinforcement learning agents in real-time environments. The game features two distinct teams: Red Robots and Blue Robots, each guided by separate training logic.

Red Robots are designed to prioritize the closest visible enemy, rotate to align with their movement, fire accordingly, and perform dodging maneuvers to evade attacks. In contrast, Blue Robots are trained to consider the positions of all enemies, rotate and fire toward the target, and explore the map to maximize spatial awareness and area control.

Training parameters such as rewards and agent count can be modified via an in-game dynamic menu, enabling flexible experimentation and rapid iteration of learning configurations.

3.2 Animations

Animations are tightly integrated with the AI behavior to provide smooth and believable movement. Each robot is rigged with a skeletal mesh and controlled via Unreal Engine's animation blueprint system. State-based animations are triggered depending on the agent's action—such as walking, aiming, or dodging—and blended in real time for fluid transitions.

3.3 Game immersion

Game immersion is enhanced through dynamic AI behavior, responsive animations, and a visually rich environment. The map was custom-built in Unreal Engine and populated with a variety of static and interactive objects such as walls, cars, containers, cranes and natural terrain elements. These objects not only add visual depth but also influence the agents' behavior by providing cover, restricting movement, or enabling tactical advantages. The strategic differences between the Red and Blue teams, combined with a responsive environment, reinforce the sense of believability and engagement.

3.4 Game mechanics

The main gameplay loop centers around autonomous battles between two AI-controlled robot teams, each operating under distinct reinforcement learning strategies. A dynamic in-game menu allows real-time adjustments to training parameters such as reward functions, team configurations, and perception settings. Users can seamlessly switch between manual and AI control modes to evaluate behavior or test outcomes. The map supports both exploration and tactical positioning, with each match acting as a training episode from which agents continuously improve. This setup offers an interactive and flexible testing ground for emergent AI behaviors in a controlled, game-like environment.

4 Future Work

Yes, we definitely plan to continue working on this project. It has been both a pleasure and a valuable learning experience, allowing us to deepen our understanding of reinforcement learning (RL) and its application in dynamic gaming environments.

There are several interesting ideas we didn't manage to implement, but would like to explore in the future:

- **Curriculum Learning:** Training the agent using a progressive difficulty system to improve learning efficiency.
- **Behavior Cloning:** Mimicking human players using supervised learning to create more natural, lifelike bots.
- **Adaptive Shooting Mechanisms:** Making the AI dynamically adjust its shooting behavior based on past experiences.

The project can definitely be scaled up and improved. In its current form, it serves as a strong prototype, but with more polish and advanced features, it could be integrated into commercial games, research platforms, or AI training tools.

A more mature version of this project could even be sold to companies or made available to the public as a framework for intelligent game agents. We believe this is a promising direction for combining AI and gaming in innovative and impactful ways.

5 Conclusion

We learned new concepts throughout this project, particularly in the areas of game development, artificial intelligence, and the relationship between reinforcement learning and shooter based gameplay. It gave us hands-on experience with integrating AI agents into a dynamic 3D environment and understanding how learning models behave in fast-paced, adversarial scenarios. One thing we really liked was seeing the AI improve over time and adapt to complex game mechanics in a realistic setting. We found frustrating that the project required some computational resources, making training and testing both time-consuming and hardware-intensive.Additionally, developing the project effectively also demands resources to create custom maps, run the game at high graphics settings, and overall, a good laptop or PC is preferred to ensure smooth performance.

Limitations

One key limitation of our work is the high computational cost associated with training and testing the AI. The project requires substantial GPU resources, which can be a barrier for users without access to high-performance hardware. Running the environment at high graphics settings and generating custom maps also places additional demands on system resources, making a capable laptop or PC essential.

Currently, the project does not face a language barrier, as it operates primarily through visual and environmental cues rather than language-dependent input, making it broadly adaptable across different language settings. However, the system is scalable and can be further developed to handle more complex tasks, richer environments, and broader language support. Future work could also explore optimization techniques to improve efficiency and accessibility.

References

a. Learning agents 5.4. https://dev.epicgames.com/community/learning/courses/kRm/unreal-engine-learning-agents-5-4/MpeM/unreal-engine-what-s-new-in-learning-agents-5-4.

b. Learning agents intro. https://dev.epicgames.com/community/learning/courses/kRm/unreal-engine-learning-agents-5-4/4JPj/unreal-engine-learning-agents-intro-5-4.

c. Learning to drive. https://dev.epicgames.com/community/learning/tutorials/owjW/unreal-engine-learning-to-drive-5-4.

Youtube tutorial. https://www.youtube.com/watch?v=kSgs3JmUYpo.

2017. Ultimate epic battle simulator. https://store.steampowered.com/app/616560/Ultimate_Epic_Battle_Simulator/.

2019. Totally accurate battle simulator. https://store.steampowered.com/app/508440/Totally_Accurate_Battle_Simulator/.

2022. Ultimate epic battle simulator 2. https://store.steampowered.com/app/1468720/Ultimate_Epic_Battle_Simulator_2/.

(yt) (tab, 2019) (ueb, 2017) (ueb, 2022) (lea, b) (lea, a) (lea, c)