

GameBrain: The Evolving Unity Experience

Purposes:

- Create a unique and adapting player and game experience
- Understand the synergy between game development and machine learning

Goals:

1. Build a Unity game
2. Integrate reinforcement learning
3. Adapt Unity game features based on RL-agent's playing

Project Team

Team Members:

- Grant Fullenkamp: Computer Science
fullengm@mail.uc.edu
- Noah Heinen: Computer Science
heinennnc@mail.uc.edu
- Roshan Krishnan: Computer Science
krishnr2@mail.uc.edu

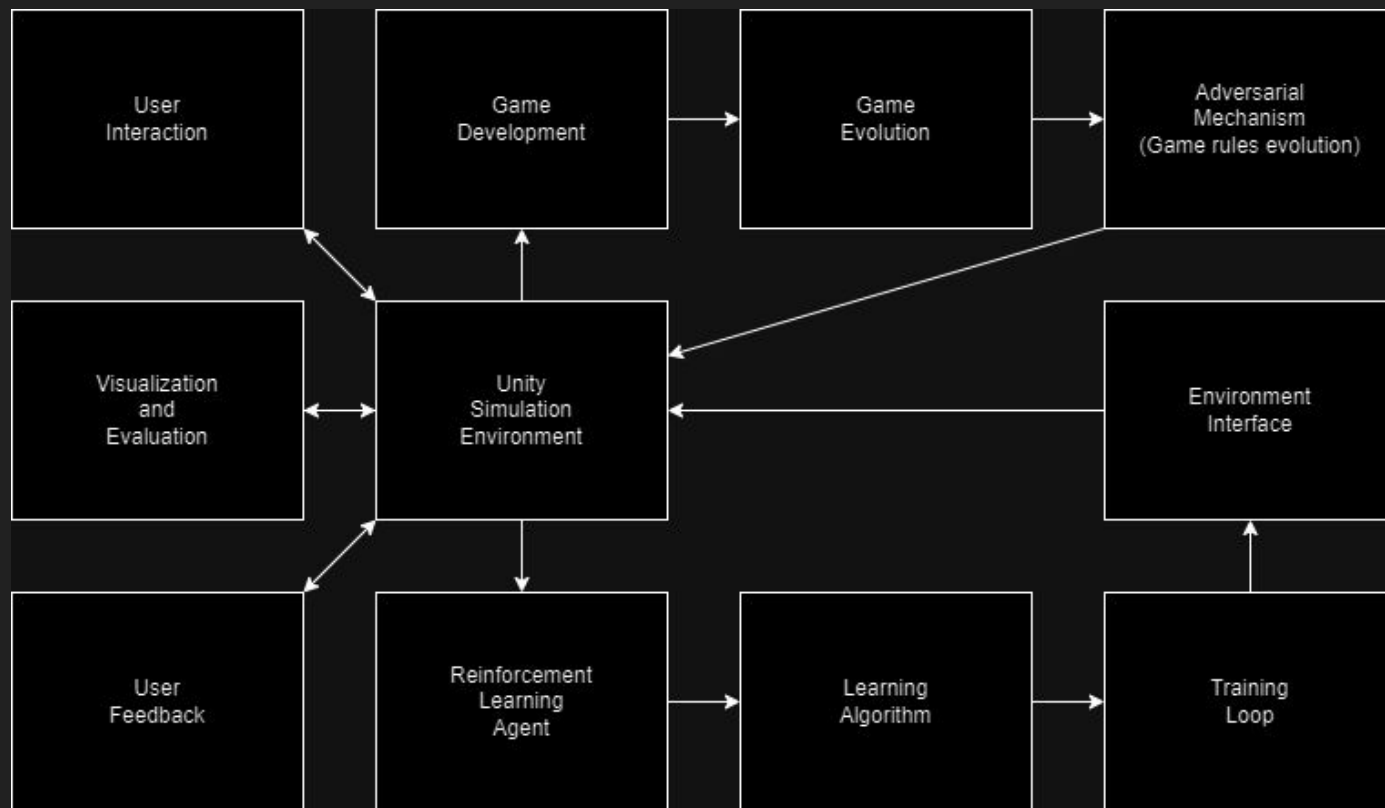
Advisor:

- Bayley King PhD: Riverside Research
bking2@udayton.edu

Project Abstract

Our project revolves around the iterative development of an engaging Unity game powered by reinforcement learning. We embark on a creative journey, starting with crafting a Unity game and introducing a reinforcement learning agent. The agent learns to excel at the game as we continuously add complexity and new features, thereby enhancing the gameplay challenge. This dynamic loop of game developer-driven augmentation and reinforcement learning-driven adaptation not only creates an evolving gaming experience but also represents a novel approach to game design. Our project promotes a deep understanding of the synergy between game development and machine learning, fostering an exciting and intellectually stimulating journey for players.

Diagram



User Stories

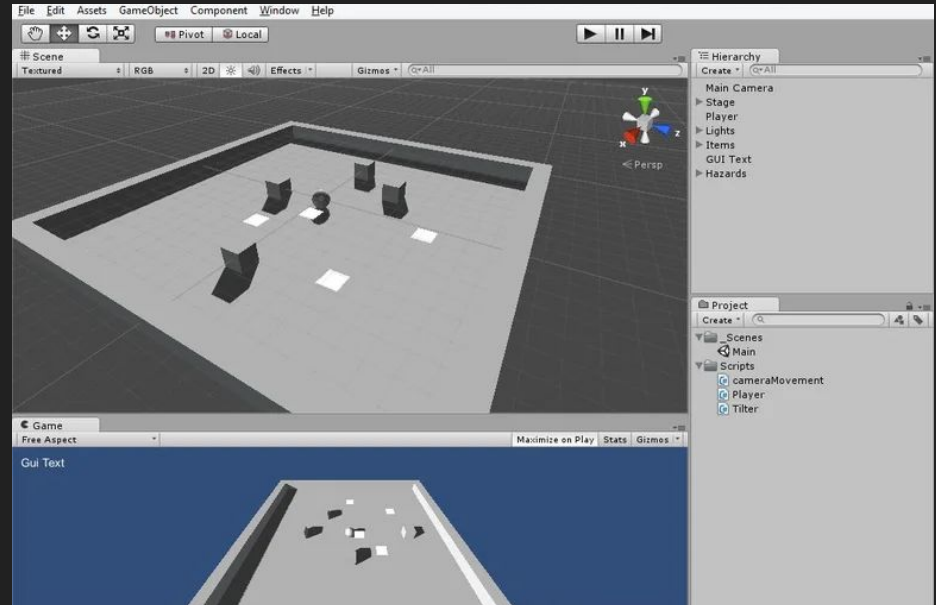
1. As a game developer, I want to design and implement new game features, so that I can continually challenge the reinforcement learning agent and observe its adaptability in evolving game scenarios.
2. As a player, I want to watch the RL agent play the game, so that I can learn new strategies and tactics from its gameplay.
3. As a project supervisor, I want to track the RL agent's learning progress, so that I can assess the project's overall success and make informed decisions about adjustments.

Major Project Constraints

- **Development Time:** The iterative process of creating an evolving Unity game powered by reinforcement learning demands a significant amount of development hours. Balancing game development, integration of reinforcement learning, and continual adaptation is a time-intensive endeavor.
- **Compute Resources:** The computing resources required to run both the game and reinforcement learning processes simultaneously can be demanding. Ensuring the efficiency of these resource-intensive tasks is crucial to meet project milestones effectively.

Current State of Project

1. Continuous development of the initial Unity game
2. Unity Python reinforcement learning integration



Expected Accomplishments by the end of this term

1. Specify Game Mechanics
2. Develop Game Prototype
3. Integrate Unity and RL Framework
4. Design RL Agent Architecture
5. Implement RL Agent
6. Create Training Environment
7. Define Training Scenarios
8. Train RL Agent

Division of Work

- | | |
|--|--|
| 1. Specify Game Mechanics | (Roshan Krishnan, Game Developer) |
| 1. Develop Game Prototype | (Roshan Krishnan, Game Developer) |
| 2. Integrate Unity and RL Framework | (Grant Fullenkamp, Project Supervisor) |
| 3. Design RL Agent Architecture | (Grant Fullenkamp, Project Supervisor) |
| 4. Implement RL Agent | (Noah Heinen, Machine Learning Engineer) |
| 5. Create Training Environment | (Roshan Krishnan, Game Developer) |
| 6. Define Training Scenarios | (Roshan Krishnan, Game Developer) |
| 7. Train RL Agent | (Noah Heinen, Machine Learning Engineer) |
| 8. Evaluate RL Agent | (Noah Heinen, Machine Learning Engineer) |
| 9. Refine Game Mechanics | (Roshan Krishnan, Game Developer) |
| 10. Enhance Game Features | (Roshan Krishnan, Game Developer) |
| 11. Update RL Agent | (Noah Heinen, Machine Learning Engineer) |
| 12. Test Gameplay Experience | (Grant Fullenkamp, Project Supervisor) |
| 13. Document Project Progress | (Grant Fullenkamp, Project Supervisor) |
| 14. Project Presentation and Reporting | (Grant Fullenkamp, Project Supervisor) |

Expected Demo at Expo

- We plan on presenting a video showing our initial Unity game implementation and showing our RL-agent continuously learning until it reaches some 'good' state in the game.
- Then we plan on presenting this RL-agent again in the updated and more challenging game to show the continuous game development and how the agent adapts.