VG-AI

TEAM MEMBERS: NATHAN JONAS, SPENCER KRUSE, CRAIG GORNEY

PROJECT ADVISOR: ALI MINAI

Project Goals

1. Create a video game where gameplay is enhanced by decisions made through artificial intelligence. This will create entertaining gameplay. Our plan was to create a small-scale single player FPS game, where the in-game enemy learns to defeat the player.

2. Utilize Mindmaker to implement reinforcement learning AI in our project. There are in game assets that track data for the RL algorithms. Data is in the form of observations, actions, and rewards. The RL algorithms are fed observations to create an action, and the action creates rewards based on its success against the player. Rewards are used to train towards or against the selected action, and this process will create an intelligent system. While the game persists or the user continues through the rounds, the enemy will continue using reinforcement learning to learn how to defeat the player.

3. Create a fun and modular gameplay, where the engagement is generated from the game's systems and not it's content.

Intellectual Merits

1. **Reinforcement Learning in a Video Game:** This video game focuses on enhancing gameplay through reinforcement learning where the enemy continues to gain intelligence as the game persists. Reinforcement learning, one of the basic machine learning paradigms, enhances our video game in a unique way. This is not common in many modern-day video games, and we hope to promote this unique gameplay experience.

2. **Utilization of Unreal Engine's Various, Amazing Libraries:** The game map has been codesigned by the team members for a realistic, intense game experience. We have created a spooky, abandoned building located in a valley of large mountains. Most of the components used to build this map utilize Unreal Engine's many free, or low-priced, libraries. This promotes fast, easy design of a custom game map.

3. **Additional Game Logic Creates an Enhanced User Experience:** Both the player and AI enemy have a functional health system, ranged weapon, and 3-dimensional movement. Game menus have been created to direct the user within the game. These additional enhancements provide a more superior user experience.

4. **Multiple Reinforcement Learning Algorithms in a Single Video Game:** Reinforcement learning has been created for "Offensive Movement" and "Weapon Aim & Fire". The observation space for both learning algorithms has been simplified to increase the training rate.

Broader Impacts

Artificial intelligence (AI) and reinforcement learning are growing fields in today's world. They both impact our everyday lives. Companies are using AI to enhance workplaces, provide personalized recommendations, and replicating human behavior.

We are attempting to bring AI into the video game world. Companies spend millions of dollars on video game content, and this could be lowered using AI to make decisions for them. It could also improve the quality of gameplay in the long-term, as it would avoid monotony in gameplay without manually changing how the game functions (which could require hundreds of hours of development).

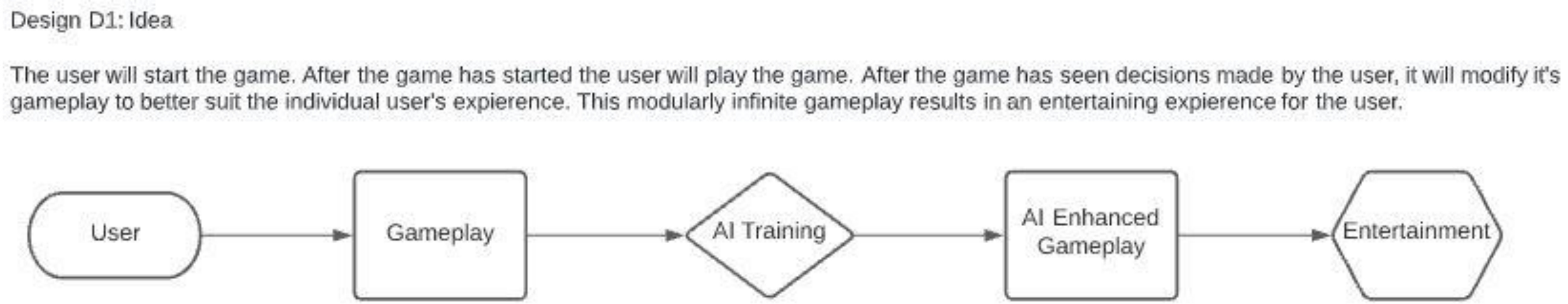
By considering our game to be an environment for training autonomous machine learning agents. We can collect data and real-time update the learning agent to behave in a structured manner. This game development strategy will be widely used and continue to grow in complexity in the future.

Design Specifications

Our project is written with in combination of C++ and Unreal Engine blueprints.

- C++ is used by default in UE4. Our project uses it mostly for the UE4 physics engine.
- The rest code is written in UE4 blueprints, which controls the artificial intelligence and game flow.

The design of the game is simple. As a video game, our end goal is to create an entertaining experience for the player through modifications to the enemy AI behavior.



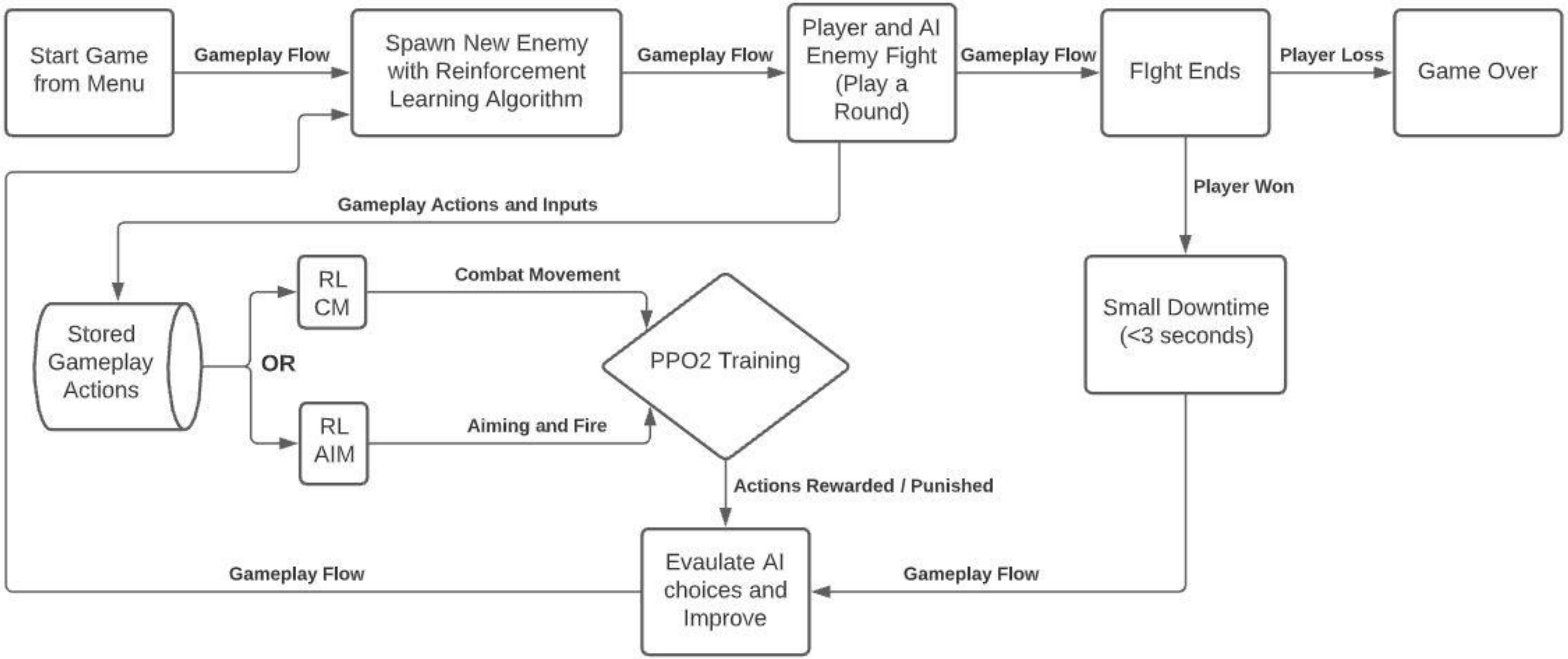
Design D1: Idea

The user will start the game. After the game has started the user will play the game. After the game has seen decisions made by the user, it will modify it's gameplay to better suit the individual user's expierence. This modularly infinite gameplay results in an entertaining expierence for the user.

Design D3: Gameplay Loop and AI

After starting the game, a new randomized AI will be spawn with no data for training. Then the during the course of the game, data will be collected on the AI characters actions, observations, and successes. This data will be constantly fed to a reinforcement learning algorithm, of which choice of algorithm is togglable from the menu. One for control of the AI character's aim and firing and one for control of the AI character's combat movement. Each algorithm requires different game data inputs, and outputs a selected premade action or state for the situation. The premade actions and states were made as the original design of learning everything from scratch was too heavy to realistically become intelligent in short amount of time for the game to be fun.

After defeating an enemy, the game will have a small downtime to evaluate it's actions against the player and create an improved neural network. Then it will spawn a new enemy and repeat with the improved algorithms. The game ends when the player has been defeated by the AI.


Start Game from Menu
Gameplay Flow
Spawn New Enemy with Reinforcement Learning Algorithm
Gameplay Flow
Player and AI Enemy Fight (Play a Round)
Gameplay Flow
FIght Ends
Player Loss
Game Over
Player Won
Gameplay Actions and Inputs
Stored Gameplay Actions
OR
RL CM
Combat Movement
RL AIM
Aiming and Fire
PPO2 Training
Small Downtime (<3 seconds)
Actions Rewarded / Punished
Gameplay Flow
Evaulate AI choices and Improve
Gameplay Flow

Reinforcement Learning Design

Aim and Fire

- Observes the player dodge pattern by marking how the shot hit or missed
- Chooses an appropriate shot angle from predetermined choices
- After firing, receives a reward based on how close the shot was to hitting the player

Combat Movement

- Observes the distance to the player character and the distance to the nearest cover point
- Chooses an appropriate action between strafing, approaching the player, and taking cover
- After moving, receives a reward based on being at an appropriate firing distance and being close to cover

Technologies

We have developed this game using Unreal Engine 4 and made the AI with a program called Mindmaker.

Unreal Engine 4

- Unreal Engine 4 is a free 3D game engine that has prebuilt physics and many other useful tools to build our game. It also comes with editing software with built in simulator for testing. Unreal was designed to be accessible to new game developers and has plenty of great accessories.
  - We chose UE4 over Unity for our 3D engine as UE4 had better tools for developing and working with AI.

Mindmaker

- Mindmaker is an Unreal Engine 4 plugin and python application that allows Unreal Engine 4 to act as an environment for machine learning. It is linked to the engine through Socket IO (local webapp on port 3000).
- We used the built in PPO2 algorithm for our Reinforcement Learning algorithms with customized rewards, observation spaces, and action spaces.

Milestones

1st milestone – determine project specifications, including goals, methods, tools, etc. 9/20/20

2nd milestone – Unreal engine working and installed on all team members devices as well as working knowledge of Unreal. 10/24/20

3rd milestone – Basic opponent created and able to move and perform actions / beta map created and functional. 11/21/20

4th milestone – Ensure Mindmaker is installed and working all team members devices as well as a working knowledge of reinforcement learning. 1/16/21

5th milestone – Reinforcement learning algorithms are developed to control the AI character. 2/14/21

6th milestone – Full demo completed with a textured map and fightable opponent with reinforcement learning algorithms controlling its actions. 3/23/21

Results

Complete:

- Map completed and explorable, fully textured with lighting
- Opponent has systems in place for movement, firing, and decision making with certain states blocked
- FPS game system implemented with round counter, health bar, weapon charge.
- A reinforcement learning algorithm determines where to aim opponents fire.
- A reinforcement learning algorithm determines where and how to move in combat situations.

To be done:

- 3rd reinforcement learning algorithm to determine where and how to move in non-combat situations.
- Ability to run all the RL algorithms at the same time (currently blocked by Mindmaker)
- Small bug fixes for opponent behavior tree and collision physics.

Challenges

Time

- The problem was not creating a video game that implements Artificial Intelligence (AI), but rather,
- **How impressive can we make this game given a time frame?**
- **Challenge Goal:** Finish, expand, and improve an AI enhanced, functional video game in a limited time frame

Scope

- **How can we divide the project's work items in a fair, productive way?**
- **Challenge Goal:** Group the work items into sprint-like time periods, then fairly assign the work items to each team member
- **How can we best understand the scope of building an AI video game on Unreal Engine?**
- **Challenge Goal:** Research examples of AI video games and reinforcement learning within Unreal Engine

Professional/ Technical Experience

- Our lack of knowledge within the unreal platform led to many issues along the way.
- **How can we overcome our lack of knowledge for video game design and development, artificial intelligence, and unreal engine to create a video game?**
- **Challenge Goal:** Study the unreal engine documentation and use the first semester (of two) to attempt various tutorials related to our project. This will build our understanding of the technologies required to accomplish the project goals.

We Hope You Enjoy!