Team VG-AI

Spencer Kruse, Jack Osterman, Craig Gorney, Nathan Jonas

Ali Minai

Project Overview

- Combine our passions for gaming and artificial intelligence to create an interesting and entertaining experience
- Use unreal engine to develop a new game from scratch
- Incorporate AI into the game
- **AI Components:** enemies, balancing, and level generation
- **Goal:**
    - Determine feasibility of self artificially intelligent systems in gaming
    - Learn about the development process required for video games
    - Research and development of an artificial intelligent program
    - Implement cloud computing, parallel training architecture, the unreal engine, and video game version control into a single project

The Team

Nathan Jonas

jonasnn@mail.uc.edu

Spencer Kruse

Krusesw@mail.uc.edu

Craig Gorney

gorneyca@mail.uc.edu

Jack Osterman

ostermjn@mail.uc.edu

Project Abstract

The goal of our project is to create a video game where gameplay can be altered based on the decisions from artificial intelligence to create entertaining gameplay. Our current plan is to create a small scale single player FPS game, where the in game enemy learns to defeat the player.

The game will record data tables for each encounter with an enemy. Upon defeating an enemy, the game will add the fight data to a training set. Between fights, the game creates new enemy patterns based on reinforcement learning or back propagation for historic player vs. enemy situations from the training set. Upon creating a new enemy pattern, the game will test it against a generated test set of predicted player actions. When the time comes to spawn the next enemy, the game will use the best AI it has come up with so far to fight the player.

We feel this concept can create fun and modular gameplay, where the engagement is generated from the game's systems and not it's content.

Back Propagation Neural Network

- Implementation of a feedforward neural network.
- We have 3 algorithms determining the movement of our AI, one will determine lateral movement and one will determine angular movement, such as which way the AI is facing. The final will determine whether to attack or not.
  - Inputs: Lateral Movement, Angular Movement, Player Health, Enemy Health, Line of Sight
  - Outputs: New Lateral Movement, New Angular Movement, Fire
- This is a basic implementation of the AI we want to create if our experiments with reinforcement learning do not go well.

Reinforcement Learning

- Familiarity with RL from classes and self research
- **Bot Controllers**: movement and individual contact
- Stored gameplay data is consistently gathered for gameplay decision analysis.
- **Rewards:** shooting close to the player, landing a shot, dodging the player's shots, and defeating the player
- **Penalties:** bot is shot, stalemate in movement, shooting not relatively close to the player
- **Goal:** maximize rewards and limit penalties

RL Elements

- **Agent:** Program that will track movement and combat decisions and makes decisions on the data accumulated
  - Decision based on reward and penalty policies
- **Environment:** Arena based game.
- **Action:** Methods for the agent to interact with the environment, therefore transferring between states
  - Rewards and penalties (back one slide)
  - User assessment to find which actions minimize loss and increase agent benefits.
- **Rewards:** The reward function describes agent behavior
  - Defined by how easily our agent will defeat a player
  - Use movement and combat data and are ever-changing
  - The rewards can then be considered continuous and differentiable.

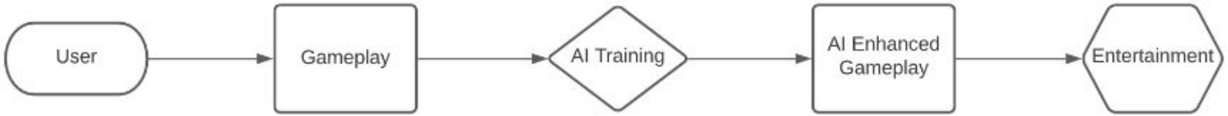
User Stories

As a team full of video game players, we want to make sure our game remains fun with the new A.I. elements. These are the acceptance criteria for our project to achieve that goal:

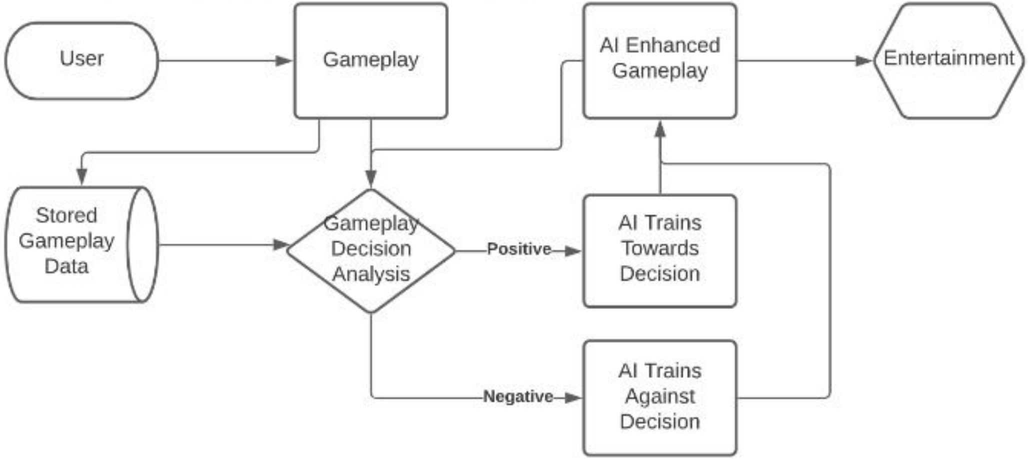
- As a Player, I want the game to reflect upon my decisions so that my gameplay experience is enhanced.
- As a Player, I want the AI to make fair alterations such that the gameplay remains fun.
- As a Player, I want the AI training to be as unobtrusive as possible such that the gameplay is unaffected.

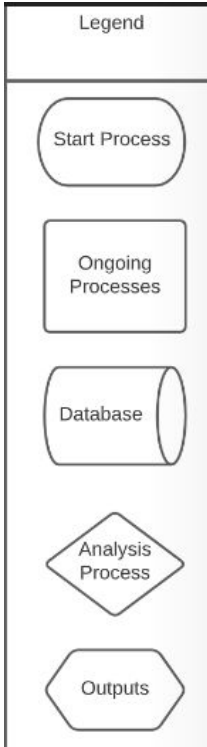
Design Diagrams Basic

D1:

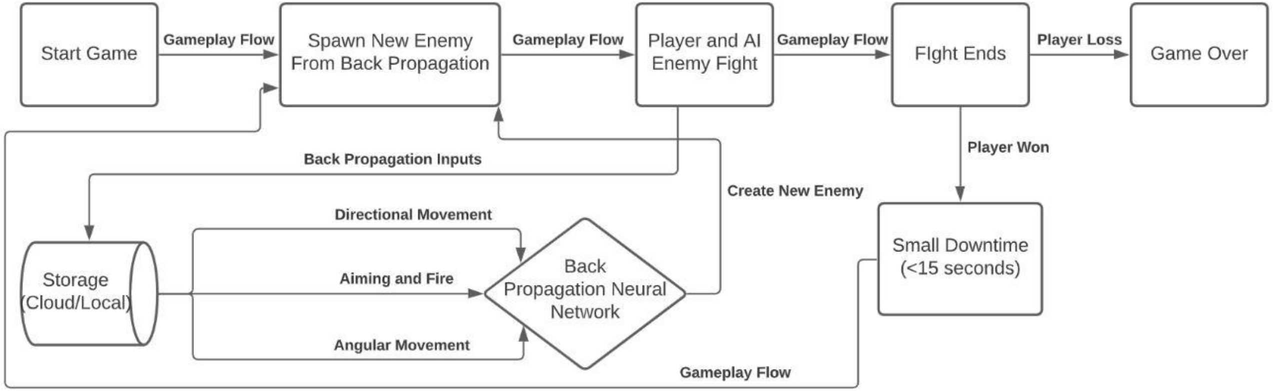


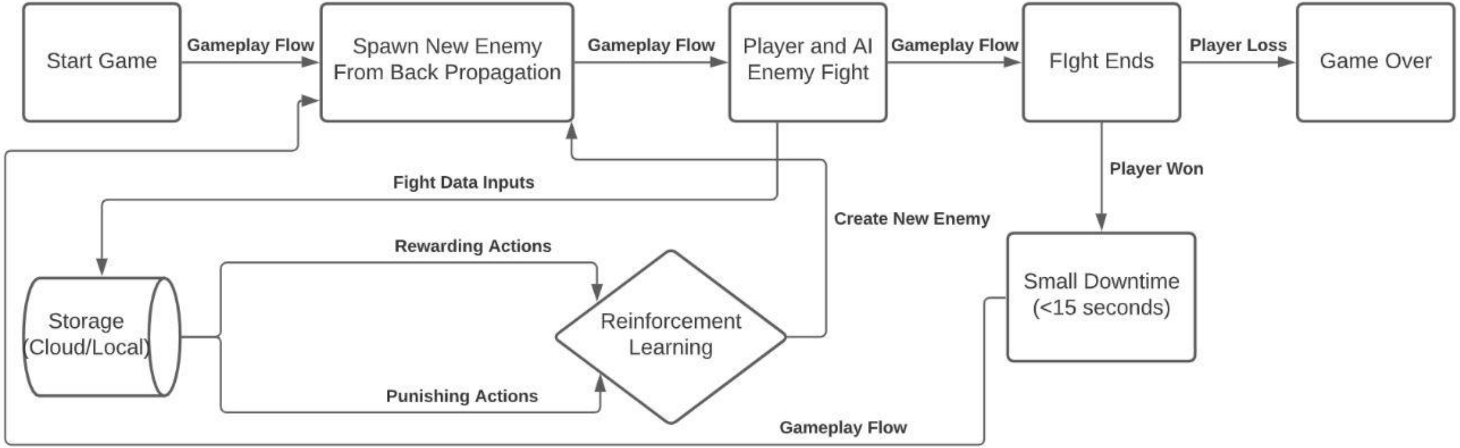
D2:





Design Diagrams Gameplay





Project Success Metrics

For the project to be considered a success, it needs to be entertaining and the AI to be successful in defeating the player.

If our game is entertaining, then players will return to the game to play again. This metric is called retention, and our exact measure of retention success is as follows:

- If 50% of players return to the game at least once and play at least 2 hours in the first week of having the game, then the game will be considered entertaining by our standards.

The way we will determine success of our algorithm is that for every x games the player plays, then the Algorithm won a percentage equal to:

$$\frac{60}{1 + e^{-0.1(x-50)}}$$

Major Projects Constraints

- **Time**
  - No worry of a game working with AI elements
    - Rather how impressive we can make the game given a time frame
  - **Goal:** Finish, expand, and improve a working product in a limited time frame
- **Scope**
  - We were sure to divide the project into goals agreeable by the team members
  - An AI game on the unreal engine has a large portion of this projects scope that we will need to research
- **Professional/ Technical Experience**
  - **Required Skills:** game development, artificial intelligence, model creation, software development.
  - Our lack of knowledge within the unreal platform may lead to issues along the way

Project Progress

- Methods of implementing Artificial Intelligence have begun to take shape
  - Exploring decision trees and blackboards using Unreal's own architecture
  - Looking at methods of implementing reinforcement learning in unreal
- Base project has been created and assets are being added to the game
  - The arena exists and users can go in and play around
  - Current working features:
    - Random pathfinding AI
    - Full alpha arena (still needs textured)
    - Collision detection for weapons

Expected Accomplishments for this Term

- Understanding of the Unreal Engine and dev environment
    - User Experience
    - Performance optimization/quality options (both for players and devs)
    - Game assets - usually large binary files
    - Collaboration tools
- Research AI techniques
    - Research Backpropagation
    - Research Reinforcement Learning

Division of Work

- We will all be active in research and development of the AI systems and game development.
- Specific game development tasks will be split as follows:
  - Spencer Kruse - Set up cloud / internet services for game
    - Leaderboards
    - User engagement metrics
  - Jack Osterman - Create any necessary game models
  - Craig Gorney - Using library to implement external AI code into unreal
  - Nathan Jonas - Creating textures for game map

Expected Demo

- The game will be playable without interruption or error
- The game will be fully textured and modeled with no placeholders
- The game will have at least one implementation of artificial intelligence that can learn player patterns
- Provide a demo download