

# FightCLub: Adaptive AI for Dynamic Gameplay in 3D Fighting Games

Alex Chavez | Advised by Heather Guarnera | Department of Mathematical and Computational Sciences



## Abstract

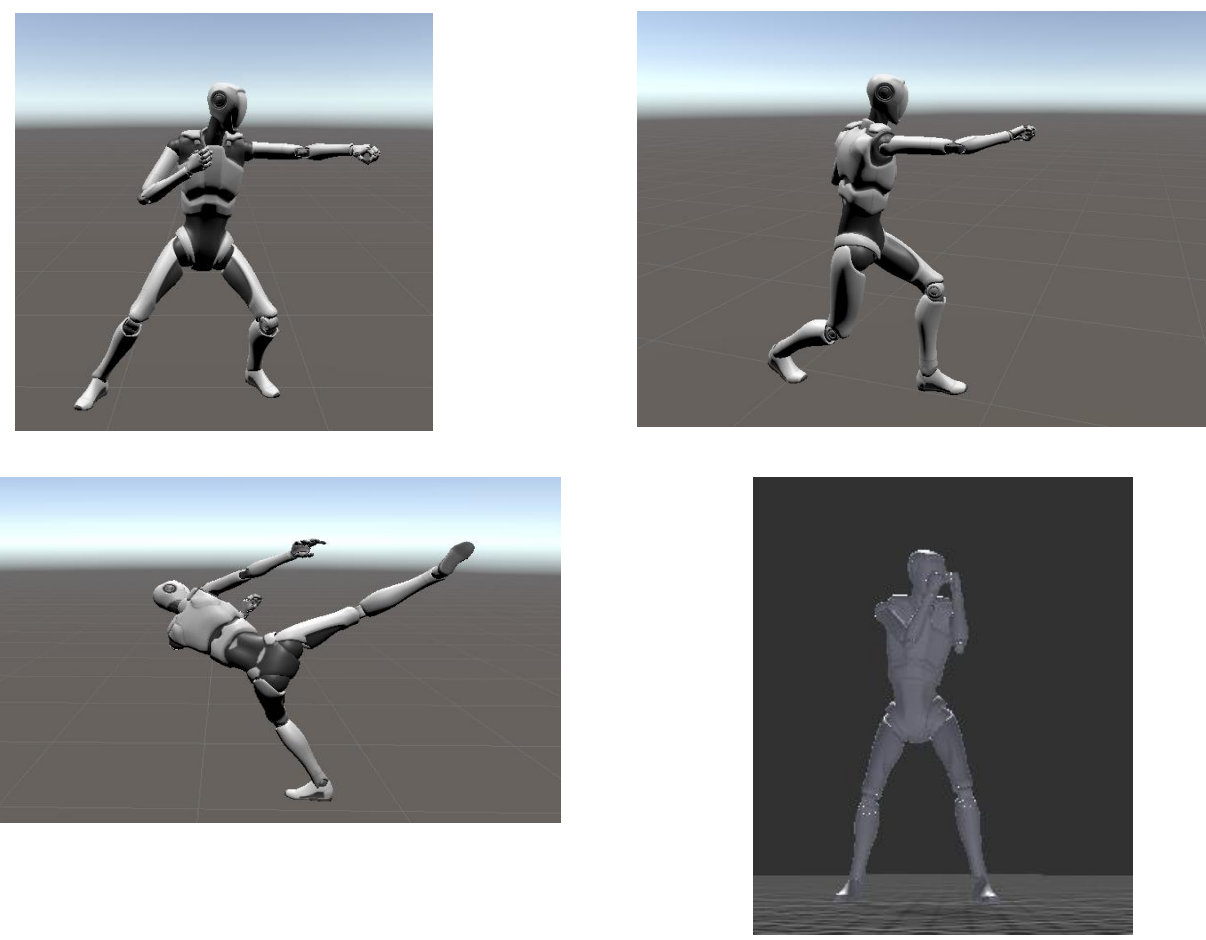
This Independent Study explored the development of a 3D fighting game incorporating Dynamic Difficulty Adjustment (DDA) to enhance player experience. The game was designed to analyze a player's skill level and playstyle in real time, adjusting difficulty accordingly to maintain an engaging and balanced gameplay experience for players of varying skill levels. The implementation focused on key aspects of game development, including game design theory, AI modeling, and the integration of DDA algorithms. The game was developed using Unity and featured a 3D movement system, and AI opponents that dynamically adapted their difficulty

## Game Mechanics

FightCLub features a 3D arena-style combat system with responsive controls. Core mechanics include light and heavy attacks, combos, and blocking. Accurate hit detection and real-feedback enhance immersion and player control

### Combat System

Players can perform punches and chain them into combos for fluid attacks. The system also includes blocking to reduce damage.



Attack System:

- Right Mouse Click: "Blocking"
- Left Mouse Click: "Attack"

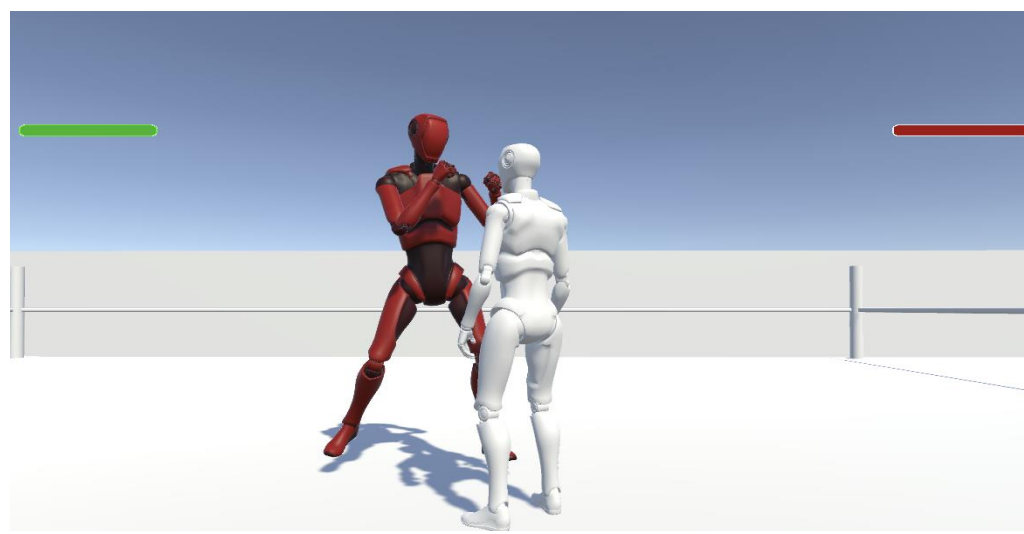
### Movement System :

Players move freely in the 3D arena using WASD:

- W: "Move Forward"
- A: "Move Left"
- S: "Move Backwards"
- D: 'Move Right"

### Health System :

Each Characters begins with 200 hp, displayed via a bar on top right and top left of the screen. Attacks inflict small amounts of damage, encouraging longer, more strategic battles. This designs ensure matches are engaging, allowing player to adapt and utilize the Dynamic Difficulty Adjustment system.



## Dynamic Difficulty Adjustment

**Dynamic Difficulty Adjustment (DDA)** is a game design technique that dynamically modifies the game's difficulty in real time based on the player's ongoing performance. The goal of DDA is to maintain an optimal level of challenge that keeps the player engaged ensuring the game is neither too easy to be boring nor too difficult to be frustrating. Rather than relying on static difficulty modes selected at the start (like "Easy" or "Hard"), DDA allows the game to continuously evaluate how well a player is doing and adjust its systems accordingly. These adjustments can involve altering enemy behavior, changing resource availability, scaling damage values, or modifying AI strategies. By doing so, DDA helps players stay in a state of "flow," where they feel consistently challenged but capable of overcoming obstacles.

### Different Types of DDA

#### Adaptive AI opponents:

- AI opponents change their strategies (aggression, defense, reaction time) in response to player performance.
- Dynamic Resource Management adjusts the availability of in game resources based on how the player is doing
- These systems offer feedback and subtle control to the player over difficulty adjustments. For example: If the player loses several rounds in a row, the system might reduce enemy aggression

#### Machine Learning:

- Uses algorithms (like supervised learning or reinforcement learning) to learn player behavior and adjust difficulty intelligently over time.

#### Rule Based AI Systems:

- Uses predefined "if-then" conditions to adjust game difficulty dynamically.
- For example: *If player wins 3 times in a row, increase enemy aggressiveness.*

### FightCLub AI Behavior & FSM

#### Rule-Based DDA Using Finite State Machines (FSM)

- A rule-based system that switches between predefined states (Easy, Normal, Hard).
- Transitions between these states are triggered by player performance metrics such as accuracy, health difference.

#### Adaptive AI Behavior

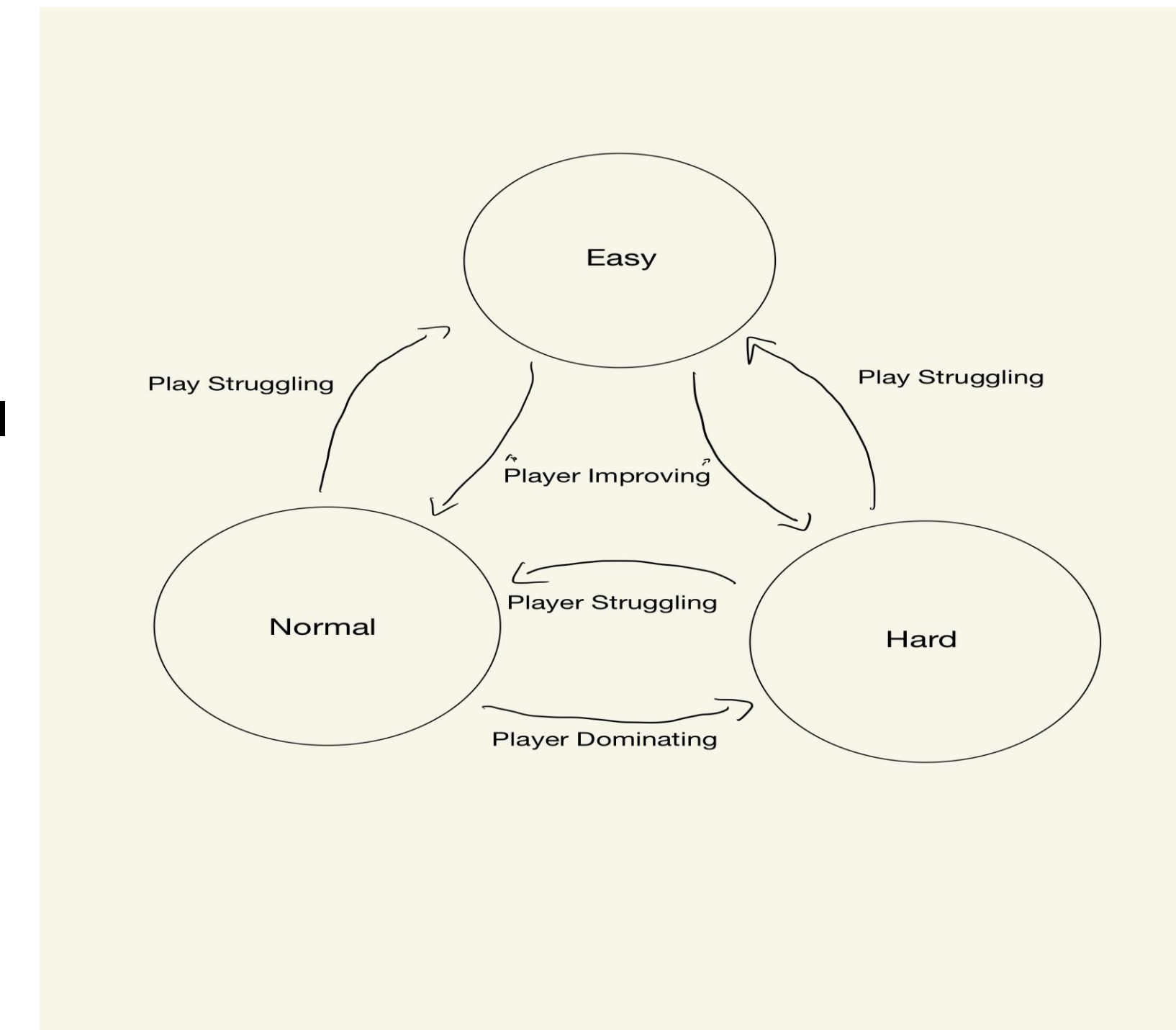
- The AI opponent dynamically changes combat behavior mid-match.
- For example, if a player lands too many attacks in a short time, the AI may switch to a more defensive style.

#### Real-Time Player Performance Tracking

- The system continuously monitors the player's in-game actions (e.g., success rate, combo efficiency, blocking rate).
- These metrics feed into the FSM to determine when to adjust difficulty.

#### Non-Intrusive Adjustments

- DDA in FightCLub is designed to be subtle and maintain immersion.
- Instead of giving direct assistance or showing difficulty changes, the AI adapts "naturally" as if it's learning from the player.



### DDA Logic

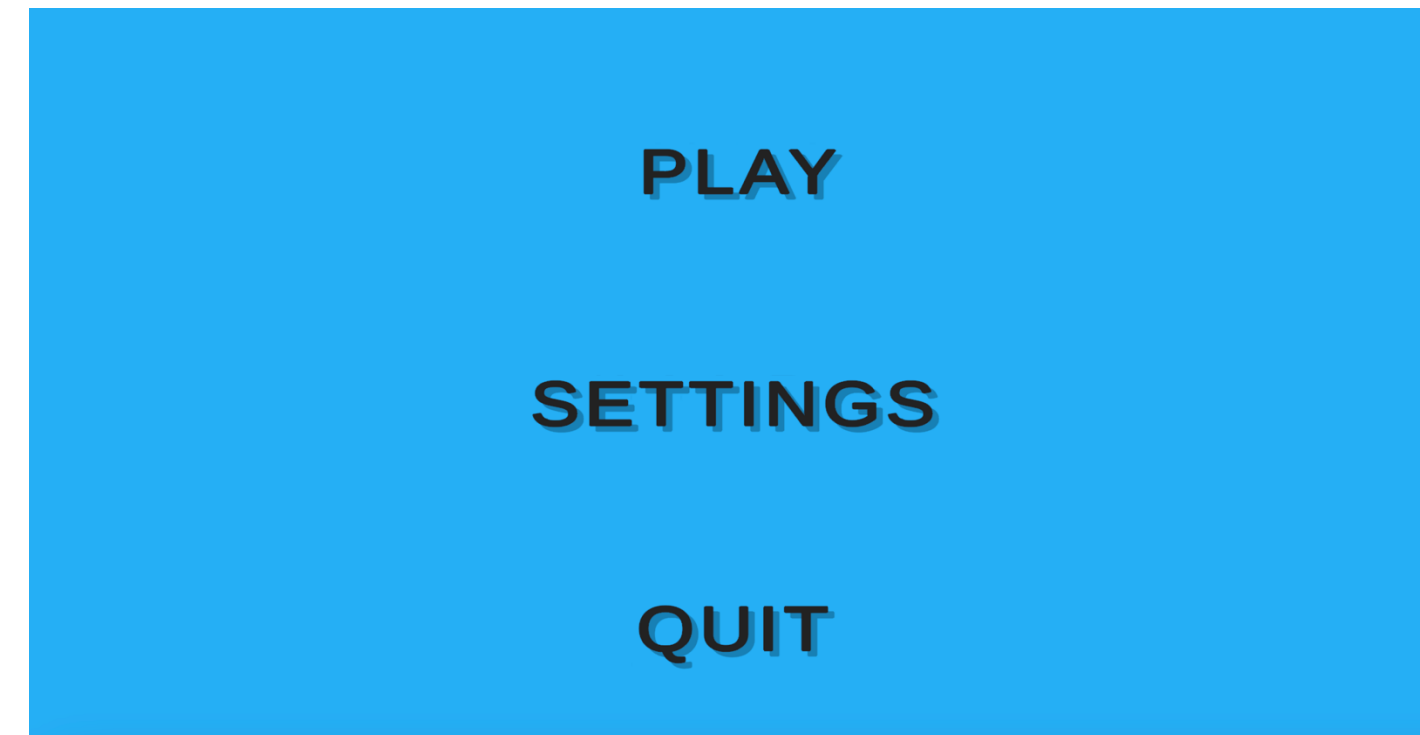
- If the **player is struggling** in any difficulty (e.g., taking too much damage), the system lowers the difficulty moving toward **Easy** to reduce pressure.
- If the **player is improving** in **Easy**, the game gradually increases difficulty to **Normal** to keep the challenge engaging.
- If the **player starts dominating** in **Normal** (e.g., winning easily, taking little damage), the game increases the challenge by transitioning to **Hard**.
- Conversely, if the **player begins to struggle** in **Hard**, the game drops difficulty back to **Normal** to keep things fair and fun.

## Bugs & Limitations

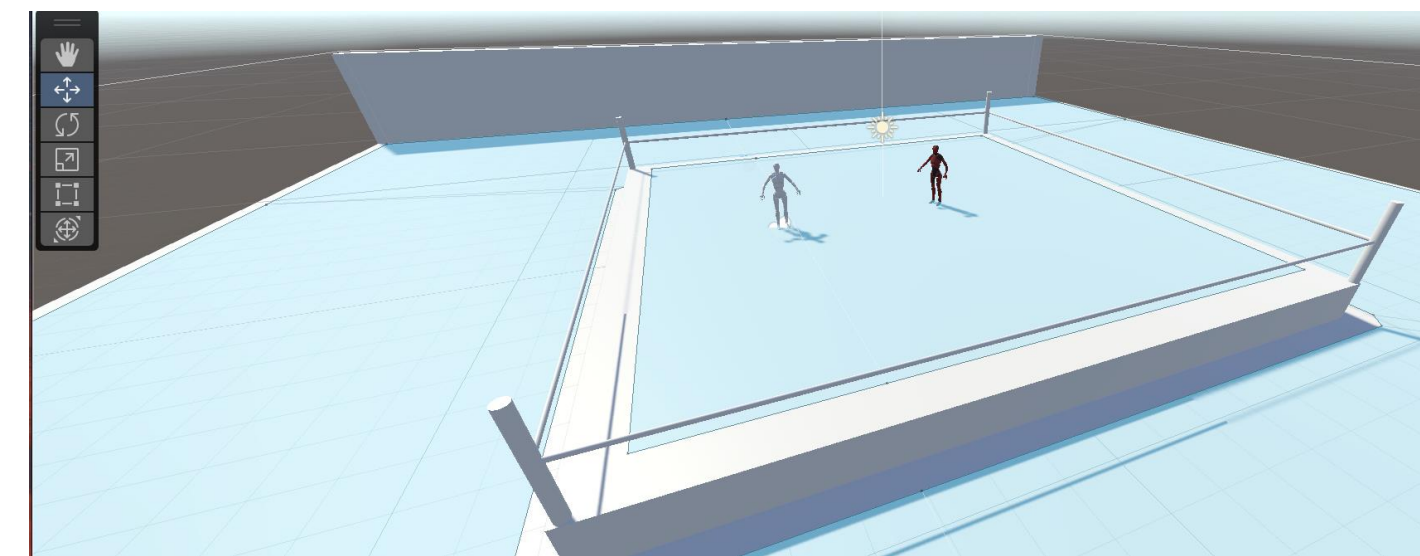
- Floating Character:** The character model appeared to float above the ground due to root motion issues. Resolved by implementing a Unity asset called Fantacode Studio
- Camera:** The camera would fail to render the scene. It was only resolved by deleting the camera object and recreating it.
- Character Model not Showing:** certain time the character wouldn't show nor wouldn't move. Resolve by deleting the character and reimplementing.
- Animations not Playing:** Some attacks wouldn't be triggered. Solved by implementing a unity asset called
- Level Restart:** After resetting the level, characters wouldn't reset and stayed stuck.
- End Screen button:** UI buttons became unclickable, hard to force the cursor to appear.

## Key Features

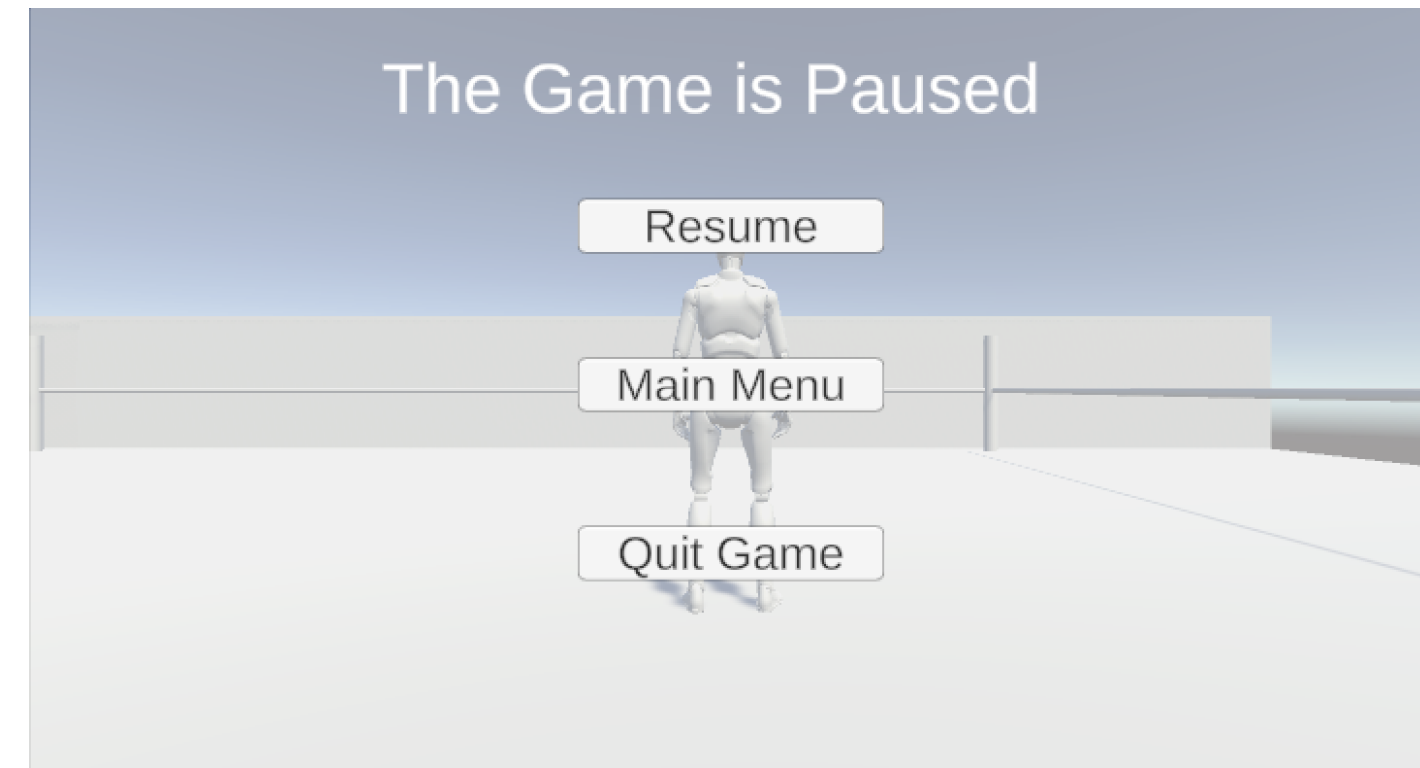
### Main Menu Screen



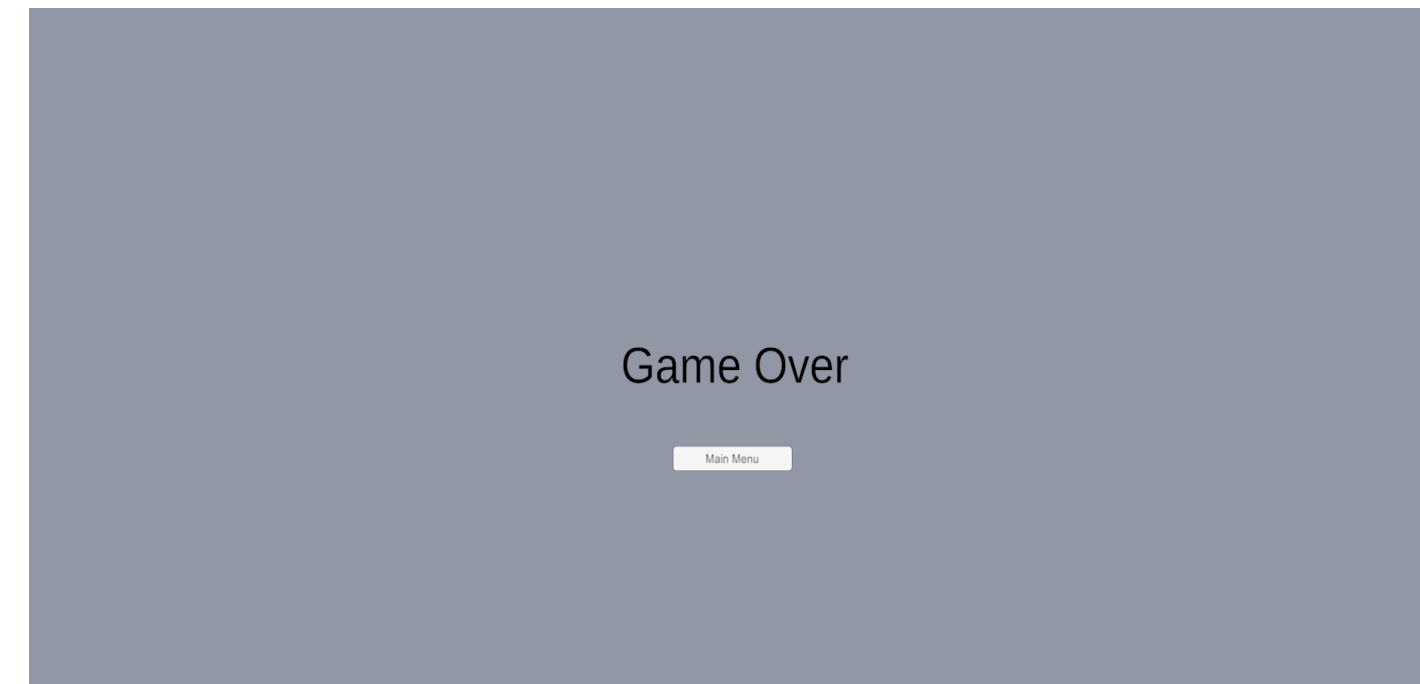
### Arena



### Pause Screen



### Game Over Screen



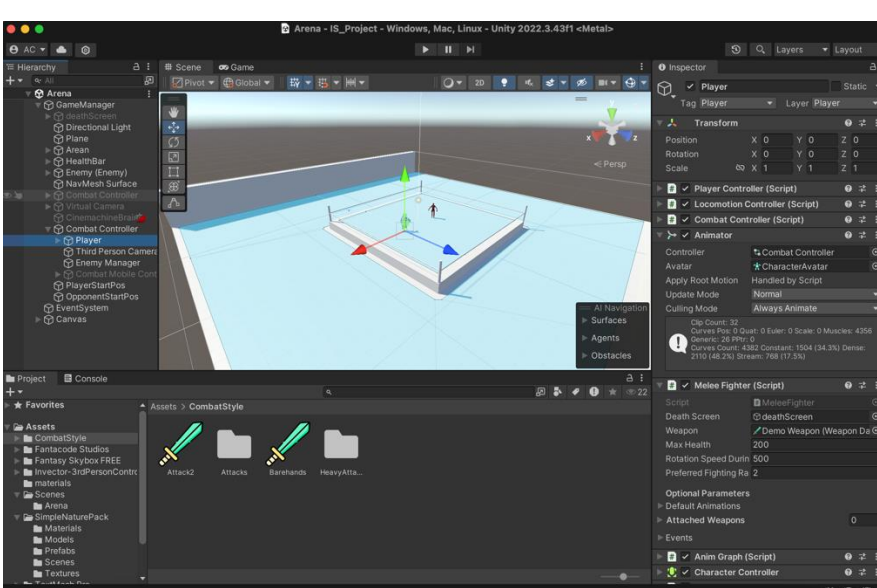
## Game Engine

### Unity

A powerful, cross-platform game engine used to create both 2D and 3D games. It offers a user-friendly interface, real-time editing, and a wide range of tools for physics, animation, and scripting. Unity supports C# programming and is widely used in both indie and professional game development for its flexibility, strong community, and compatibility with PC, consoles, mobile devices, and VR platforms.

Feature	Unity	Unreal
Programming Language	C#	C++ & Blueprints
Rendering Engine	URP/HDRP (Customizable)	Nanite & Lumen (Advanced)
Ease of Use	Beginner-friendly	Steeper Learning Curve
Best for	Mobile, 2D, Indie Games	AAA, High-Fidelity Graphics
Licensing Cost	Free & Paid Plans	Royalty-based Model

A comparison of the two leading game engines, and their differences



This shows Unity editor environment for FightCLub.

## Future Work

- Implement a special ability system
- Add visual effects
- Expand the character roster; include unique fighting styles
- Include Round based Systems
- Improve the arena graphics and features
- Refine animations and movement systems
- Refine the DDA, and explore reinforcement learning