[Wayward Warlock]

Technical Design Documentation

## Technical Requirements

Engine: Unity **2022.3.36f1**

Version Control: **Git**

Itch Page:

Player Mechanics

# Player Controls

The player controller will handle all the input needs for the player. Attacking, picking up items/power-ups, and movement.

# Enemy Behavior

The enemies will seek or pursue towards the player's position. Then attack once they are colliding with the player.

# Player/Enemy Health

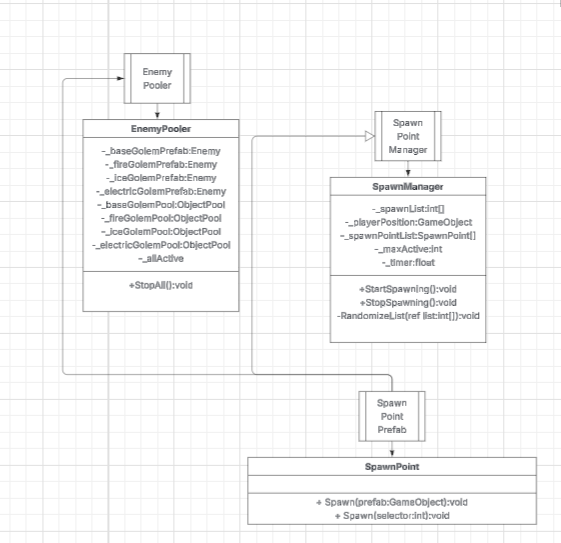
Both the player and enemies have health.

Each enemy's health is dependent upon their movement speed.

If the enemy is a fast moving lightning element, it will move fast, but have low health.

# Sound Manager

The sound will be handled via the sound manager. There will be a music track playing whilst the player is actively fighting/taking their last stand.



# Enemy Pooler

This will hold all the object pools for each enemy type. It will manage each pool. It will have a function to check all active enemies. There can only be one instance of the Enemy Pooler.

# Spawn Point

This will spawn enemies. It will have a function to Instantiate a prefab. It will also have a function to get from the Enemy Pooler.

# Spawn Point Manager

The spawn manager will delegate the first few spawners closest to the player to spawn enemies

To use the Spawn Point Manager, place the script on an empty object.

Put All the Spawn Points that are in the scene into the Spawn Point list.

Give the spawn manager the amount of enemies to spawn.

# Game Manager

Deals with game state, track kill count, and if the player is dead or alive.

Will tell the spawn manager when to start. This will also manage the amount of enemies that need to spawn.

# Input Management

Input Management is handled via scripting.  
Will use Unity’s new input to allow cross platform usage and avoid any possible bugs.

# Weapon Attacks

The main attack will be a magic missile.

Three special projectiles: Fire, Ice and Lightning.

One Special Screen wipe: Forbidden Spell.

Each attack gives visual effects relative to what type of projectile it is.

# User Interface Manager

The user interface manager handles the interface. Displaying the players stats, which will be health, mana, and their current score/killcount.

# Coding Conventions

\_myVariable camel casing is used for private variables.

No underscore for public variables.

# Commit Conventions

Title

Put the main thing you worked on

Body

Put bullet points of all the changes that you have made.