Unity & Unreal Engine Sample Project

Game Design Document

Summary

1. About Project

This a sample project with some basic rules to understand the difference when developing for both engines Unity and Unreal. The idea is to achieve the same game feel and mechanics in both engines. But not exactly having the same code "line for line", it must use everything the engine has to offer in order to make the development process as smooth as possible.

Genre: Action
Perspective: 3D Top-down
Main platform: PC

2. Tools

We want to keep things as simple as possible, the main focus for this is not really graphics, so for this we are using the same asset pack, same UI images and same animations for both projects.

2.1. Assets:

Animations: [Adobe Mixamo](#)
Asset Pack: [Synty Studios: POLYGON - Prototype Pack](#)
UI Assets: [Kenny - UI Pack](#)

2.2. Engines

We want to keep the engines as vanilla as possible, so we are not using any third party assets for both of them. We want to use the most from each engine.

Unity: Version 2022.3.10f1 - C#
Unreal: Unreal Engine 5.3.2 - Blueprints

3. Game Mechanics

3.1. Entities

There must be a base entity class or object where the enemies and the player will derive from. An entity will have the components such as health, movement, stats and any other component needed. An entity will be created from a sample data file. So it can have different names, healths and abilities. The player must choose a character before joining the match.

Player: Will be controlled by the player
Enemy: Can be hurt by the player

3.2. Game Mode

The base game mode should be a domination like mode, where we will have bases around the map that are protected by enemies NPCs. Each base will initiate its defenders. After all the enemies are eliminated from a base, the base is free to capture. If a player enters a domination zone when all guardians are eliminated, he'll start capturing from 0-100% until it's owned by the player.

After the player has a base owned, he'll start earning points for each game tick (1 second). When he reaches a certain points target he wins the game.

Win Condition: Reach a domination point amount.
Lose Condition: Player health reaches zero.

3.3, Enemies

For now we want to keep things as simple as possible, so no need to make AI for enemies, they only need to take damage and be eliminated with some basic animations. We can leave the AI for a next iteration of the game.

3.4. Interactables

There must be an interactable object for players to interact with, we can have a chest that heals and a light that lights up. This must be opened to extend to anything we want. We must have an option to "Hold" and "Click" to interact.

3.5. Player Abilities

Player must be able to shoot fireballs and cast an AOE spell where he's aiming. We must also leave the system open for new abilities.

4. UI

On the UI we want to show the player health and name, bases captured and skills with cooldowns.

UI Template
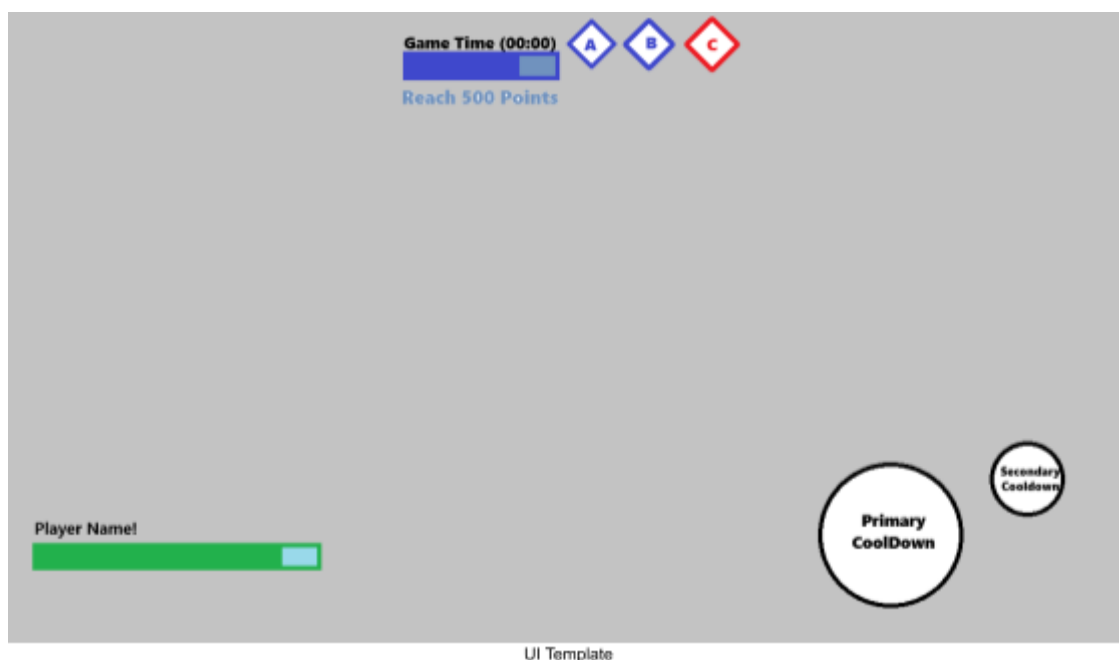
5. Controls

Movement: WASD
Aim: Mouse
Fire Primary Ability: Mouse 1
Fire Secondary Ability: Mouse 2
Interact: F