Phungia

Nguyễn Quốc Việt - 165019

Introduction:

For this midterm mini project, I made a Darksouls-esque 2D platformer game using Unity3D engine. Darksouls-esque are typically used to group extremely hard game where there are a lot of emphasis on the difficulty. Phungia belongs to that category. It requires a lot of determination, patience and precision. I was inspired from the likes of Celeste, Hollow Knight and Ori And The Blind Forest, where I died countless times during my runs but found them all extremely pleasurable. Phungia might not be enjoyable for many, but I sincerely hope there are those that like to fail and try and try again until they succeed who would appreciate this tiny game.

About the game:

Story:

Our nameless character is a boy who had woken up in Phungia, not knowing who he was or how he got there. The only thing he can do is survive and thrive. Signs are scattered in the world which helps guide our hero. These instructions where left behind by ancient beings who had seen it all in the land of Phungia.

Concept:

The signs are what introduces the player to world, what challenges to expect from it, and the mentally that would help overcome those obstacles. Each platforming section places emphasis on a particular concept such as deception, timings, and adaptability. They are metaphors for challenges we face everyday in the real world but sometimes too caught up in them to find a solution.

Features:

The main character can run and jump. Has animation for running, jumping, falling, and dying.
Sound effects for running, jumping, dying.
Sound effects indicator for moving to the next level.

Each level has it’s own tiles textures, background and music.
There 3 main types of objects:

- Non-collidable platforms
- Collidable platforms
- Collidable-fatal objects
- Collidable-fatal platforms

There are also moving platforms and stationary platforms.
Enemies include traps on the ground, traps that fall from the sky, and moving objects that chase our hero.
The levels are integrated seemlessly into one. Players will be notified of moving on to level 2 but never have to face a cut or transition.

Techniques:

Map making:

To easily manage platforms, I made a Grid with 5 tilemaps, each with it’s tag, name and collider 2D. I made 5 tilemaps:

- Spring platforms: for normal platforms of level 1
- Winter platforms: for normal platforms of level 2
- Traps: for fatal platforms of level 1
- Traps2: for fatal platforms of level 2
- BaitPath: for non-collidable tiles only used in level 1

Level integration:

In order to make the level transition seemless, the Scene for level 1 has to contain all tiles and objects of both levels. Everything from level 2 is intially disabled in Scene Level1. In Scene Level2, everything is set as normal. When the player steps into transition box, all tiles and background music from level 1 are disabled and everything from level 2 is set up. Each level has it’s own respawn area so a global static variable is used in the Respawn script to manage which level the player will respawn into.

Reusable scripts:

I tried to make resuable scripts that could be applied to different objects wanting the same behavior. ActivateTarget and DeactiveTarget are the most used scripts. They basically do SetActive() for a referenced object. Another one is the MovingPlats script which makes an object move horizontally, taking in "range", "direction" and "speed" as parameter. I used that script for every single moving platform.

Resource references:

Sprites:

- https://www.gameart2d.com/freebies.html
- https://www.spriters-resource.com/pc_computer/buster/

Sound:

- https://opengameart.org

Scripting:

- https://answers.unity.com/index.html

Tutorials:

- https://www.youtube.com/playlist?list=PLCqjfeAzeHipgsRPjgzsGLZVIBe0sTkN9
- Various other youtube tutorials

Documentation:

- https://docs.unity3d.com/Manual/index.html