GAME 236 Final Game Proposal: A Danmaku Prototype Remake with Unity

Ryan Cai, 3/11/2024

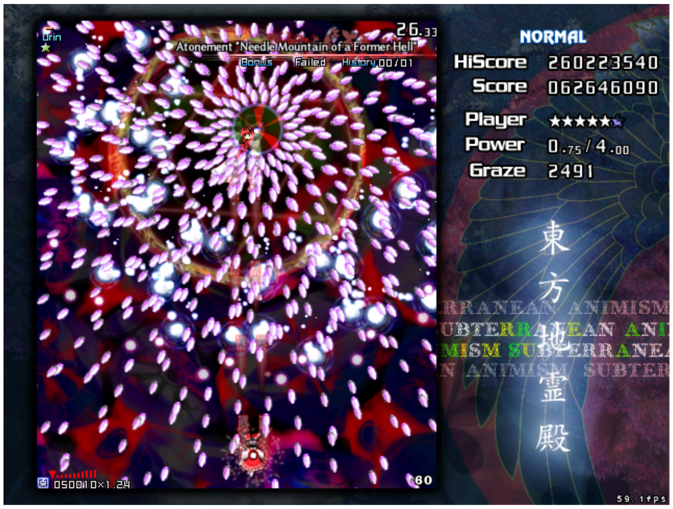
235 Final:

https://github.com/RyanChoiHuiYun/GAME235/tree/main/final_project

Overview

I am proposing to work from the concepts and prototypes that I created in GAME 235, which is a simple Danmaku game. Due to the limitation on Processing, there were several features that I couldn't get done last quarter. This time with Unity, hopefully I can get some of the features in. The most notable feature I want to get it done with Unity is creating levels, such as having patterned enemies spawning and patterned bullets/projectiles flying in the screen. At the same time, I am referring to the previous proposal here, and making a few changes for this final:

I am proposing to make a game in Unity 2D that resembles a Danmaku game, or a bullet hell game, where the player controls a character that needs to dodge multiple incoming projectiles on the screen and take down enemy objects that are shooting the projectiles. The main objective is to build a physics engine that can handle object collisions and calculate vector speed, and to include adding level design and sound/music.



Features

- ☑ ~~Player controller~~
  - ☑ ~~8 direction movements~~
  - ☐ "Evade" action that has i-frames with a cooldown
  - ☑ ~~"Shoot" action that shoots various projectiles~~
- ☑ ~~Game objects~~

    - ☑ ~~HP~~
    - ☑ ~~Collision~~
- □ Power-Ups
    - □ Health buff: replenish health by some amount
    - □ Attack buff: increase attack/projectile prowess in some form
    - □ Defense buff: gain a shield that mitigate some damage
- □ UI
    - □ HP
    - □ Scores
    - □ Menu
- ☑ ~~Enemy Objects~~
    - □ Set waypoints for movements
    - □ Set projectile shapes
- □ **EXTRA**:
    - □ Storytelling element
    - □ Background music

GAME 236 Final Game Postmortem: A Danmaku Prototype Remake with Unity

Ryan Cai, 03/22/2023

I am adding on the initial proposal to keep track of the original features and goals that I have achieved in the final game. Overall, I was able to implement all the core features except:
- Player can evade
- Player can collect buffs
- UI

And I have added some extra features that I did not include in my proposal, namely:
- EnemySpawner that spawns random enemies (2 types)
- Type 1 enemy shoots bullets, but only has 1 Hp.
- Type 2 enemy doesn’t shoot, but has 3 Hp.

Remaking the previous Processing game with Unity is more complex than I thought it would be. One thing that went well is it is easy to implement enemy shooting projectiles using Unity, which I couldn’t figure out in Unity. Using the *tag* functionality in Unity, it is also rather easy to keep track of the collision behavior among different objects, for example when enemy bullets colliding with enemy vs enemy bullets colliding with player. Unity 2D also has bulit-in rigidbody 2D that is easy to navigate. The most challenging part is probably tweaking the EnemySpawners and designing the enemy prefabs. I tweaked the spawn rate in a way to give incremental challenges to the player (hopefully), where the enemy will sync up at a certain time stamp to launch a more challenging and synced up attack. I left out the UI part that I think is very important to the project but due to time constraints I wasn’t able to finish implementing it.

For future development, the priority will be making sure a proper UI that consists of at least the hp of the player. Furthermore, I am looking to make bullet patterns shooting from the enemy. I have a basic idea on how to do that but did not try to implement it here.

To conclude, I learned a lot from doing this project, especially with using the Unity built-in functions to translate a lot of functions from Processing. I heard that it is challenging to change game engines midway through the project, now I understand why.