Spring Interactive Reflection

Time Management

This project was worked on in sessions of 1 to 3 hours every available day. No time was spent on artwork as all the in-game assets were obtained either from Google Images or were SFML text and rectangle shape objects. The entirety of the time spent on the project was coding. In total, there were 12 sessions of varying lengths, with 21 hours spent on the project coding.

Prior to beginning the project, I had estimated around 14 – 16 hours, so I misjudged just how complicated the project would turn out to be. This certainly was expected given this being my first full coding project, and I will be using this experience to inform my future estimates.

Workflow

1st March – Added basic UI elements, stats display, Enemy and Player classes, alternating basic and FIGHT menu options (2 sessions, 2 hours each)

4th March – Added damage calculations, icons for enemy type, started turn order (1 session, 2 hour)

6th March – Added enemy turn text, issues with “okay” button (1 session, 1.5 hours)

7th March – Started Hp Bar (1 session, 2.5 hours)

8th March – Finished Hp Bar, fixed “okay”, finished basic animation, started damaged animation (2 sessions, 2 hours each)

11th March – Damage and Game Over graphics (1 session, 2 hours)

14th March – Finished damage animation and started button highlights (1 session, 2 hours)

15th March – Finished button highlights, added attack images for Player, started critical hit and miss mechanics (2 sessions, 2 hours and 1 hour)

18th March – Finished hit and miss mechanics, final QA additions (1 session, 1 hour)

New Technologies

The first new technology I used was classes and objects. This was a new technology as I hadn't used them in a full-scale project before, which made them incredibly fun to work with. I also used a method for menu buttons, something I had used once prior as part of a labsheet but had otherwise not used, especially not to this extent. Due to how often this project used this technology, it proved vital to make good use of these mechanics. Building off this menu navigation was a mechanic for highlighting text boxes where the mouse is hovering. I also introduced mechanics for a turn-based system of combat. Finally, I used proper damage calculations for the first time, which may not count as a new technology, but it was a tricky mechanic to solve and proved to be a challenge to make use of.

Problems Encountered

The primary problem I encountered while working on this project was with my damage calculation. There was a strange bug where upon pressing the "okay" button to progress the enemy's turn, damage would be dealt upon both the Player and the Enemy. After many hours of work and help from the lecturer, I was able to fix this problem by only registering a mouse click when the mouse is released, which prevented a mouse click being processed multiple times.

I also struggled with implementing a turn-based system. There was a multitude of issues that arose because of my inelegant implementation. The solution I ended up with was certainly not an ideal scenario, but it was so engrained in the project that I feared changing it could risk breaking things, so I left it be for now and took a note to improve this concept if I were ever to return to it in a new project.

Lessons Learned

The primary lesson I learned from this project was to not underestimate the complexity of a game project. While I was able to achieve the game as initially planned, it took far longer than I anticipated. I need to keep this in mind, as if I don't I risk designing a project I won't have time to complete.

Another big lesson to be learned with this project is to be more sparing with Booleans. I ended up with what would likely be double the number of Booleans that were necessary for a project like this. I anticipate that investing more in Enums could help alleviate this problem in future.

Link to Screen Cast

https://setuo365-my.sharepoint.com/:v:/g/personal/c00295678_setu_ie/ERZ7iyaP2wlPvI5QRGhA5wUBEEiaRCkuCk5Mxg77X2_9ag?e=J9Ehfp