SPRING REFLECTION

Issues I faced

Buttons are hard. I didn't realize that I'd have to make an individual rect and boolean for EVERY SINGLE BUTTON, because it was all a singleton. This meant that the main menu was way too bloated than I wanted it to be, but it worked!

I also faced issues with making the choice in the battle menu scroll through using the arrow keys. I settled on using integers to represent each choice. When it went over, it reset back to the original choice, and vice versa.

My third and final REAL issue was sprite sheets, I completely forgot how texture rects worked. *Now,* I do know how they work, and sans can visibly take damage!!

The Structure

At the beginning of my project, I had no *real* structure. But as it went on, I decided to tackle it screen by screen. Starting with the main menu, then the movement screen, and finally the battle screen. Here is a detailed list of my commits:

Feb 14: Added the ability to interact with the buttons using the mouse, you can click, and when you hover over it the color of the button darkens. Also added movement to the preGame section, meaning the player could move around the screen before the battle. Also added the NPC prompt which lets you PROPERLY start the battle.

Feb 17: Added and removed the "cutscene code". This was using text files to read off texting using ifstream and ofstream, and it just seemed to be WAY too complicated for what was necessary.

Feb 24: Added the ability to change choice in the battle screen with the arrow keys.

Feb 28: Added the Sans sprite, an options menu, and fixed the scaling for almost every screen. The library computers are awful, and aren't standard resolution, so I had to account for this in my code, because it was . . . *bugging* me.

Mar 2: Added an exit button to the options menu, so you could actually close it.

Mar 3: Most of the work was done on this day, in my opinion. This added the actual battle functionality, using Enums and Bools for attacks, and if the player CAN attack. This also added the ability to kill Sans and added a good health bar to him.

Mar 4: Added music and added dialogue for when Sans talks. This uses the sf::clock, basically meaning the player can't do anything until the timer runs down (three seconds).

Mar 5: Added music to every game state, and changed the color scheme. I changed it to the colors associated with fights in Undertale, the game series Sans is from.

Why choose this project?

I *love* games of this genre, especially when they go in depth with it. To me, these kinds of games are very pure, and are an interactive experience that you can't really go wrong with.

What would I do differently?

I would most definitely NOT spend as much time as I did getting the ifstream and ofstream to work. That being said, for a larger scale game, it'd DEFINITELY be a much better idea. It might be a good idea for our next project, if we need dialogue. I could've probably used the SF::Clock so he says random things in battle, too.