Project 1

The Arena

Intro

=====================================================================================

The arena is a turn based style fighting game. The player must fight through five battles to win the game. As the player continues through the game, they will gain experience to level up and have access to a shop where they can receive healing items. The player will also have a bed that will heal them of any wounds and save the game.

The combat system is turn based and is fairly simple with three basic actions: attack, defend, or use an item. The attack action simply damages the enemy, reducing the damage if the enemy is blocking. Defending allows you to reduce the damage received by a foe, while potentially stunning them. The chance to stun an opponent is increased the more they attack your shield. But be wary, because the same can be done to you. Once the player unlocks the inventory, they can use an item to heal themselves during battles, prolonging their survival.

During the fighting sequences the player must fight the player until either one of them is dead. If the player dies, then the game is over. If the player succeeds and reduces the life of the enemy to zero, they continue forward to the next fight. Once the player defeats the final boss and wins five matches, they win the game.

Summary

=====================================================================================

Line count: 2606

Number of Variables: ~63 variables

The idea for the game came with the intent of making it object oriented. Many of the game’s functions are processed through classes, like the game’s loading and saving functions which reads and writes player and game information to the save folder under the player’s name. The player and the enemy’s actions are also run through their respective classes, calculating actions such as damage and healing.

A structure is utilized in both the main as the shop’s inventory and within the player class as the player’s inventory as well. The inventory uses a pointer of ints that allocates memory when created. When an item is stored into the inventory, a number is assigned to the pointer to designate what item was stored. Functions for the player also allow the inventory to be outputted as needed.

The program took two weeks to finish, but the program is not completely finished. Although the game works fine, there was no time to complete the dialogue of the story. I quickly wrote up an intro and outputted that to introduce the game, but no other story driven dialogue exists within the game. Other than the dialogue however, the game is mostly finished.

I had the most difficulty making the inventory working within the player class. Player’s stats were corrupted and were reassigned to random numbers. The issue was settled after I discovered that I had been declaring the structure wrong within the class allocating the inventory to a new int when it was supposed to be a new inventory.

A few minor bugs were also found with saving and loading a player’s inventory, but that was solved after modifying a few functions and adding an overloading function for setting items.

Description

=====================================================================================

The main goal of the program was to create an object oriented fighting ground with classes. It also allowed me to explore pointers in structures for inventory, which was a new concept for me.

Pseudo Code

=====================================================================================

Repeat until the user prompts to quit or if it is an invalid input

Prompt menu to user

Check user choice

If 1, new game

If 2, load game

If 3, quit

Else it is an invalid input

End of program

New Game

Prompt user for name

Create player

Create story

Begin chapter 1

Load Game

Prompt user for name

Create Player

Create story

Continue based on the story’s checkpoint

Load appropriate chapter

Chp1

Intro text

Arena function

If the player kills the enemy

Chp2

Chp2

Intro text

Arena function

If the player kills the enemy

Chp3

Chp3

Intro text

Arena function

If the player kills the enemy

Chp4

Chp4

Intro text

Arena function

If the player kills the enemy

Chp5

Chp5

Intro text

Arena function

If the player kills the enemy

Player wins!

Arena

Player chooses action

If 1, player attacks

If 2, player defends

If 3, player heals

Enemy chooses actions

If 1, enemy attacks

If 2, enemy defends

If 3, enemy heals

Repeat until one of them is dead

If the enemy dies

Player moves on

Else if the player dies

Game over

Flow Chart

=====================================================================================



