Alazae’ Davis

April 08, 2020

Object Oriented Design Final Project

Program Description

You are an Entertainment trainee in CSU Entertainment. In this game, you will work on increasing your skills and group relationship, which are all detrimental to your success. You will have an array of commands such as: train, buy, debut, name, say, drop, open, pickup, inventory, and back, to help you navigate the world and progress. The main commands I want to cover is: train, buy, eat, and debut.

With train, you have 4 skills/levels: dancing, singing, modeling, and your group relationship. With groups, practices are typically spent making sure the group performances are cohesive and synchronized. So, there is not any time for individual work. On the other hand, it helps create a closer relationship with your group members. Opposite of training in a group, training individually helps increase your skills faster but not so much relationships.

With buy, you can purchase items from the bodega and the mall to help with energy as it decreases when you train. The items to help with regaining energy is food and drinks. You can also buy nonedible items, but they will have no energy. Some, however, can help increase your skill level. Items have weights and if the weight is more than 5.0, you will not be able to obtain anymore items.

As you train, your energy will be affected. I created the eat command so that the player can replenish their energy therefore, they will be able to continue to train.

With the debut option, you will have to open the door to the stage. You can open the door at any time, but your skills will determine whether your debut will be successful. If your skills and group relationship is at least 5 and up(some fluctuating) in levels, your debut will be successful. If not, the debut will not be, thus meaning you have lost the game.

For these specific commands, I utilized the State Design. I also used it for Character creation as well. Depending on the command, the state can change, and they will produce their own individual unique commands. The Delegate Design pattern was utilized for classes such as Item, ItemContainer, Door, Room, Bodega, and BodegaContainer. The delegate design pattern allows the reusage of code functions under inheritance so, I was able to reuse functions for their containers to hold their items or skill levels or to simply create a door. In GameWorld, the Singleton pattern was utilized to create instance so that the game will start in the Character Creation room. The Decorator design patter was used to help with the incrementation of functions inside the class it was utilized in as well as other classes that call the class. I utilized the decorator design pattern in the Item, ItemContainer, Skill, and SkillContainer classes. The Façade pattern was utilized in the Gameworld. Façade patterns are usually used to essentially hide all the complex coding into the form of a method call. It was utilized to create Rooms and Doors in between the rooms. The Observer pattern helped keep track of the player’s current location and would send a notification to the console.

The bugs I have encountered thus far is pertaining to the buy command inside the Bodega. For some reason, the player is still able to buy the item and the item would be in their inventory but, there is no deduction from their wallet. Another bug that I have encountered occasionally is with the locked room called Stage. Sometimes, it will say that there is no door to open and I would have to go back to my previous room then try again and it will work. There is also an occasional glitch if I am exiting the Bodega back into the hallway. The program would switch between the two before going into the lobby.