REALMQUEST – Project Report

Group 12 – David Kochanski, Habeeb Fadak, Devesh Aggarwal

DEMO LINK (unlisted): https://youtu.be/CQViQ_DSMws

Introduction

For our final CSC207 project, REALMQUEST, we decided to expand on A2 by offering a fully interactive world that the player can walk around and explore in, rather than moving between rooms with typing in commands. The game consists of a series of rooms, each of which has a defined tile set and interactive objects. Objects range from equipable items, like potions or swords, to enemies like slimes and bosses. Doors in rooms take you from room to room.

On booting up the game, there will be a menu screen with a button to start the game, as well as an option to play with light mode enabled. There will also be a short description on how to play the game.

Gameplay

The player can move around freely in these rooms by using the WASD keys. (W for UP, D for RIGHT, S for DOWN, and A for LEFT). Some tiles, like STONE and LOG, are solid, meaning the player cannot walk through them. To collect an item, the player walks over the tile where the item is. This will place the item in the player’s inventory, located in the upper right corner.

When the player moves onto a tile, then that tile is checked to see if it’s solid. If not, then there is a check to see if there is an item on that tile.

To use an item in your inventory, click on its button. Some items, like potions, are consumable and disappear on use. Some are persistent, like your sword, and can be used indefinitely.

Combat

On the adventure, the player will encounter enemies like slimes, and the boss. To attack an enemy, if you have a sword in your inventory, then walk into the enemy. It might take multiple hits to fully kill an enemy. When you attack an enemy, the enemy will attack you back, damaging you.

In the top left is the player’s current health. The player has a max health of 10, and can regain health by using health potions. Enemies damage the player, and if the players health reaches 0, then the game is over.

Progressing through the Game

You spawn in your house with a few items in reach. To progress through the game, then find the door that leads to the next room. Some doors may require a certain item present in the player's inventory to go through.

The game ends as a loss if the player's health reaches zero.

The game wins if one of two conditions is met:

- The player defeats the boss enemy,
- The player finds the locked treasure room.

When the game ends, then a menu with a message will be shown. This menu will give the option to restart from the main menu, or to quit the application.

Accessibility

For users who are visually impaired, REALMQUEST offers a light mode which brightens all the tiles in a room. This helps to see walls and tiles more clearly, giving better contrast between them.

Design Patterns

Factory Pattern

Every room's geometry (tile grid layout), and the initial object locations, are defined separately in each of their own files. Thus, when the file is read at runtime, then the game has a factory that converts game objects (like swords, keys, etc.) into InventoryItem buttons that the player can use.

Strategy Pattern

Each item can be used, and all have different (or no) effects. All InventoryItems implement the same use() method, that is called when the user presses the button. For example, while a health potion may heal the user as it's effect, the sword may slash the tile in front of the player.

Observer Pattern

The AdventureGameView is subscribed to the player object so that whenever the player's health changes, then the View can update the health bar of the player.

Singleton Pattern

Although there are many enemies in the game, there is one main Boss. There may only exist one instance of this boss at once, since killing the boss makes the player win the game.

Accomplishments

In this project, we accomplished the goal we set in making A2 feel more like an adventure game rather than just inputting commands. We created a fully functional map that the player can explore, all of which is editable from modular files. We created a system for finding and using items. We formulated a combat system with different types of enemies. We made an accessible GUI with toggleable light/dark modes. We created a game that has a start, progression between different rooms, and an ending.