April Castañeda
Professor: Luyao Zhang
CS 162 400 S2019
June 11, 2019

Design + Reflection + Testing

Design

“Welcome to Hell” is a Horror Escape game wherein the user has to figure out how to leave hell within 24 hours - or face staying there for eternity.

Theme: Horror Escape game

Goal: To escape hell within the allotted 24 hours

Space Types:
Dungeon
Hallway
Theater
Apt
Hospital

Character Types
<u>User:</u>
Player
<u>Monsters:</u>
Audience
Ex Boyfriend
Cerberus
Grim Reaper

Structure:
Game Engine class runs a Map class that has a 2D array of Space pointers as a member.

Player Inventory:
Item Class
Vector of Item pointers is a member of Map class

Things player can do:
Move: Right, Down, Left, Up
Examine: Room
Inventory: Look at Inventory, Use Items
Quit

Win Conditions:

Player finds exit within allotted time and is able to exit.

- User must defeat all monsters in rooms to gain final item required to exit

Lose Conditions:

Player doesn't exit within 24 hours.

Pseudocode:

```
Build the world in Game Engine - Map Class constructor
        1. Create 2D array of Space pointers
        2. Create Dungeons in board - 4 different dungeons
        3. Create Hallways in board - default hallway and 4 specific hallways
        4. Create Theater
        5. Create Apt
        6. Create Hospital
        7. Set up space pointers to connect to adjacent spaces
                - where player is allowed to move
        8. Set up obstructions - closed doors
        9. Create player from Character class
        10. Set time - 24 hours max that decrements in the game
        11. Set if user has a minimap - alters display
        12. Create Menus
        13. Place Space pointer of player at starting area

Create Menus for Game Engine

Run first script Player sees

Show the main menu to user - (Options above of things "player can do")

while(user doesn't quit || within allotted time || hasn't won)
        Display minimap if player has it equipped
        Display time
        Determine if user is within allotted time (less than 24 hours, more than 0 hours)
                - losing condition if not
        Display Available exits
        Display main menu

Menu choices:
        User chooses to move Right, Down, Left, Up
                moves within grid - guided by Space pointer for player connections
        User chooses to examine room
```

```
    Sees virtual implementation of room
        Room may have items to use or discover
            determine if correct item used
            determine if there is room in inventory for discovered item
                if no room, choose to drop an item or not
                    if dropped item gave attributes, lose attributes
        Room may have monster to defeat
User chooses to look at inventory
    Display inventory - determine if empty/not
User chooses to use item in inventory
    Determine if inventory is empty
    Determine which item user chooses to use
    Determine if item user chooses exists
    Use item
        Determine if item is Quest item
            if Quest item, see if it matches space quest parameters
        Determine if item is Equipment
            if Equipment, see if user has equipped already
                if not, equip and gain attributes
```

Changes

There were plenty of challenges with this project. At first, I couldn't even figure out what idea to go with. I spent days choosing between ideas, then starting a plan for one, and then changing my mind. In the end, I actually didn't go with any of the first ideas that I started with, and wrote the game pretty much as I coded it. So this was a huge change, because I kind of didn't even know what game I was creating until it was in front of me.

Also, I didn't know how to implement the game either. I started out with a GameEngine class that I had used for our other game projects. I thought at first that I was going to just put a 2d Array of Space pointers within this class. But upon reading entries on piazza and slack, I became convinced that I actually needed to create another class - to act as my linked list class. So, the Space class became a node, within a Map class - which was my linked list.

After just figuring out this organization, most everything else fell into place.

Problems/Solutions:

Like I had said previously, my main problem was even figuring out what game to create. In Writing, there are two kinds of writers, a plotter who plans everything in advance (like Tolkien) or a pantser, who let characters take a life of their own. For projects in this class, I'm used to "plotting" what my program was going to do with involved pseudocode and writing out my classes before coding them.

But for some reason, with this project, this didn't work with me. And I spent days wasted planning an idea, then scrapping it. It was when I decided to just start coding that the game wrote itself for me. Even today, I didn't know how the game would end until I wrote it. It isn't what is recommended or what I've been taught in this class and I'm sorry to say that this is how I did it. But it was the only thing that worked. So that's how I solved that problem, was just by creating the game as I coded.

Another main problem I had was trying to have a map for the game that wasn't just printed, but actually a bit interactive. I wanted the user to see themselves within the confines of the map and see how and where they move. I tried to learn how to use unicode to do this but ultimately decided that I did not have the time to learn. So instead I used the characters readily available on the keyboard and hard coded how each node/index on the array should look like.

Within the Derived spaces themselves, I wanted certain objects to perform unique things. Like if I had a Dungeon space, then I wanted the different dungeons within the map to have unique items or characters or uses. So in order to do this, I included a member variable in the derived spaces to delineate choice. So their constructor would create that object based on the if statement within that choice.

Test Cases

#	TEST SUBJECT	INPUT VALUES	EXPECTED OUTCOMES	OBSERVED OUTCOMES
1	Script 1	None	First script runs and user is shown the main menu	First script runs and user is shown the main menu
2	Main Menu	Input == w || a || s || d	User moves to ascribed destination	User moves if there is no obstruction and exit exists at direction
3	Main Menu	Input == r	User sees virtual examinRoom function for each derived Space class	User sees virtual examinRoom function for each derived Space class
5	Main Menu	Input == I	User sees Inventory contents and Status	User sees Inventory contents and Status
6	Main Menu	Input == u	Displays if inventory is empty - if not empty, user is asked which item to use	Displays if inventory is empty - if not empty, user is asked which item to use
7	Main Menu	Input == q	Program quits	Program quits
8	Dungeon 1	Examine Room	Room opens door once user talks to Grim Reaper	Room opens door once user talks to Grim Reaper
9	Dungeon 1	Examine Room	User gets mini map if they choose to	User gets mini map if they choose to

#	TEST SUBJECT	INPUT VALUES	EXPECTED OUTCOMES	OBSERVED OUTCOMES
10	Display	Minimap	If user chooses to use minimap, minimap will display every turn	If user chooses to use minimap, minimap will display every turn
11	Hallway 1	User picks up key	If user examines room, user can choose to pick up a key	If user examines room, user can choose to pick up a key
12	Hallway 1	Moving to the Right	If user moves to the right and door is locked, user cannot go through	If user moves to the right and door is locked, user cannot go through
13	Hallway 1	User uses steel key	If user uses steel key, door is opened, and item is destroyed	If user uses steel key, door is opened, and item is destroyed
14	Theater	Examine Room	If user examines room, user fights Audience. User can choose to pick up an item - bronze medal	If user examines room, user fights Audience. User can choose to pick up an item - bronze medal
15	Item	Bronze Medal	If user chooses to use/ equip Bronze Medal, user's charisma increases	If user chooses to use/ equip Bronze Medal, user's charisma increases
16	Dungeon 2	Examine Room	If user examines room, user finds an item - user can choose to pick up an item - silver key	If user examines room, user finds an item - user can choose to pick up an item - silver key
17	Hallway 2	User uses silver key	If user uses silver key, door is opened, and item is destroyed	If user uses silver key, door is opened, and item is destroyed
18	Apt	Examine Room	If user examines room, user fights ExBoyfriend. User can choose to pick up an item - watch	If user examines room, user fights ExBoyfriend. User can choose to pick up an item - watch
19	Item	Watch	If user chooses to use/ equip watch, user's time left is increased by 5	If user chooses to use/ equip watch, user's time left is increased by 5
20	Dungeon 3	Examine Room	If user examines room, user can choose to fight Cerberus	If user examines room, user can choose to fight Cerberus
21	Dungeon 3	Player battles Monster	If user chose to fight monster - user can pick up an item - gold key	If user chose to fight monster - user can pick up an item - gold key

#	TEST SUBJECT	INPUT VALUES	EXPECTED OUTCOMES	OBSERVED OUTCOMES
22	Dungeon 3	Player battles Monster in Dungeon 3	If user battled monster, door to Dungeon 4 opens	If user battled monster, door to Dungeon 4 opens
23	Dungeon 4	Examine Room	If user examines room, player can choose to pick up an item - fur coat	If user examines room, player can choose to pick up an item - fur coat
24	Item	Fur Coat	If user uses/equips item, User increases armor +2	If user uses/equips item, User increases armor +2
25	Inventory	Full	If inventory is full, user is asked if they'd like to drop an item, and which item	If inventory is full, user is asked if they'd like to drop an item
26	Inventory	Drop Item	If an item is dropped, user loses attributes granted by that item.	If an item is dropped, user loses attributes granted by that item.
27	Hallway 3	User uses gold key	If user uses gold key, door is opened, and item is destroyed	If user uses gold key, door is opened, and item is destroyed
28	Hospital	Examine Room	If user examines room, user fights Grim Reaper, and can choose to pick up an item - a Scythe	If user examines room, user fights Grim Reaper, and can choose to pick up an item - a Scythe
29	Item	Scythe	If user uses/equips item, User increases strength +3	If user uses/equips item, User increases strength +3
30	Hallway 4	Examine Room	If user's strength is below 21, door cannot be opened	If user's strength is below 21, door cannot be opened
31	Hallway 4	Examine Room	If user strength == 21, user can exit - if they choose	If user strength == 21, user can exit - if they choose