Sooreechine, Tida
CS162 | Week 10
December 5, 2015

Final Project

REQUIREMENTS

- Space
  - Create a series of spaces for the player to move through.
  - Each space will be a class with at least four pointer variables that link to other spaces.
  - There must be at least 5 spaces with 3 different types.
  - The space abstract class will have a Special pure virtual function for special action.
  - There will be at least 3 derived classes representing different types of spaces.
- Game
  - The player must work through the spaces toward a solution. An additional menu option will be made for revealing the goal to the user.
  - The player will be collecting items in a type of container. The container must have a maximum capacity. One or more of these items should be required for the solution.
  - The player must interact with the spaces and not just collect items.
  - There must be a mechanism that limits the game from going on indefinitely.

DESIGN & ANALYSIS

Design
To complete the requirements, I will be designing a short game based on J.K. Rowling's Harry Potter books. The player will play as Harry, who is tasked by Professor Dumbledore at the beginning of the game, to find and destroy Lord Voldemort, who had broken in and taken up residence inside the school. Voldemort had taken 3 captives somewhere in the castle, and Harry will need to find 3 socks to free each of the captured elves: Kreacher, Winky, and Harry's friend, Dobby. Harry will start the game by himself, but finding his two friends, Ron and Hermione, will increase his chance of success against Voldemort. The game will be timed, and Harry is expected to defeat the Dark Lord and free Dobby within one hour.

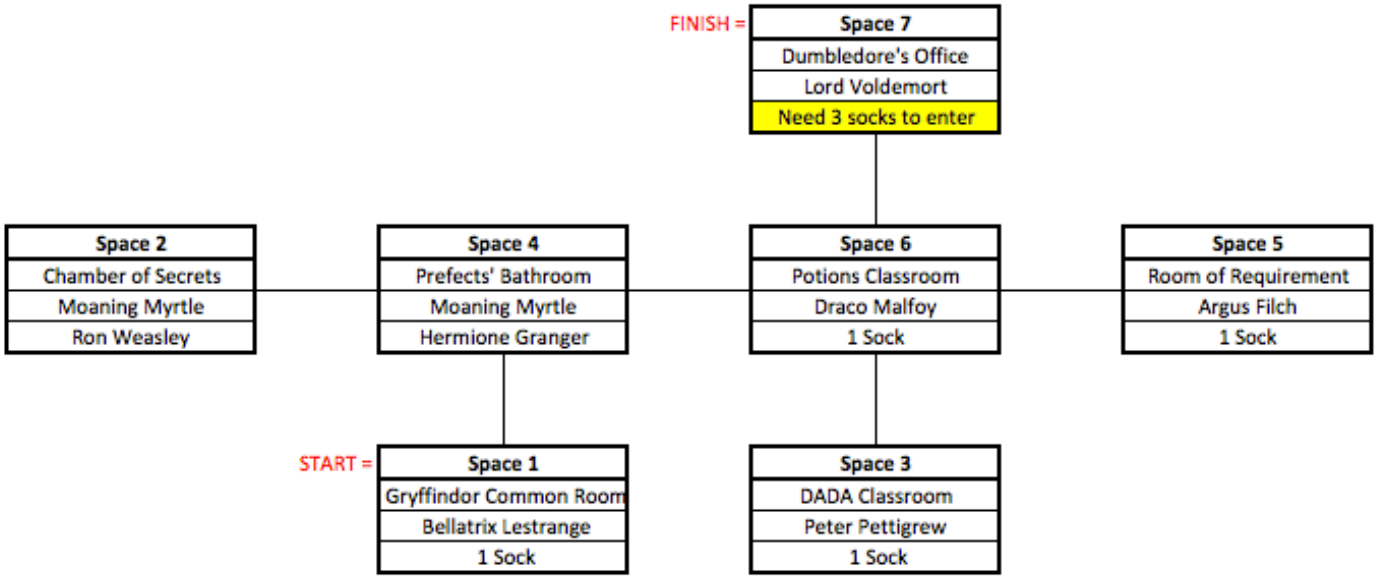
There will be a total of seven spaces of three different types:

Rooms:
1. Gryffindor Common Room
2. Chamber of Secrets
3. Defense Against the Dark Arts classroom
4. Prefects' bathroom
5. Room of Requirements
6. Potions classroom
7. Dumbledore's Office

Types:
- Type 1 – Sock Rooms
  - Spaces: 1, 3, 5, and 6
  - Prerequisite: None
  - Enemy: Death Eaters and Argus Filch
  - Gains: Enemy's sock

- Task: Harry will duel against the enemy via a random number generator. The side that produces a larger sum of 3 rounds wins. Ron and Hermione can help if they are with him.

- Type 2 – Bathrooms
  - Spaces: 2 and 4
  - Prerequisite: None
  - Enemy: Moaning Myrtle (not really an enemy)
  - Gains: Ron Weasley and Hermione Granger
  - Task: Harry will be playing a number guessing game with Myrtle. The prisoners will be freed upon the mentioning of the correct answer. After three wrong answers, however, Harry will be forced to move on without his friends.
- Type 3 – Voldemort Room
  - Room: 7
  - Prerequisite: 3 socks
  - Enemy: Lord Voldemort
  - Gains: Free elves and Hogwarts' safety
  - Task: Harry will duel with Lord Voldemort using Creature classes created in Assignments 3 and 4. Again, Ron and Hermione can add strength if they are present with him.



Analysis

To create a linked structure, a Space abstract base class will be designed similarly to the Creature class from Assignments 3 and 4. Each of these spaces will be linked to one another via pointer variables representative of the four cardinal directions. The three derived classes will be based off of the three space types.

Space classes (base and derived)

- Member variables
  - Number
  - Name
  - Enemy
  - Task description
  - Reward for completing task
  - *north, *east, *west, *south pointers linking adjacent spaces
- Member Functions

- Constructor
  - The base class will not have a constructor
  - The derived class constructors will contain all the space definitions
- Special function (pure virtual) performing the task unique to the space

The player's status will be documented through a Potter class.

Potter class

- Member variables
  - Alive/dead Boolean variable
  - Ron/Hermione Boolean variables indicating who Harry had saved
  - Number of socks acquired
- Member functions
  - Add socks function
  - Related get and set functions

The main file will function as a client program, used for holding the interactions between the Space and Potter classes necessary to move the game forward.

Main Program

Harry is set to start at space 1 (Gryffindor Common Room)
While Harry is still alive and elapsed time is under limit
Harry enters space
Space's task is displayed
Player can choose to either complete the task or explore adjacent rooms
Player will have the ability to re-enter the room, whether or not the task was completed
If player chooses to attempt task
Update Harry's dead/alive status
If player chooses to relocate room, adjust location
Ask player to play again

TESTING

Test Plan & Test Results

Test Case	**Input Values**	**Expected Outcomes**	**Observed Outcomes**
Space pointer variables work as intended	Choice of direction: 1-4	All the spaces are true to the map design, regardless of where the player is located at the moment	Worked as expected after having made space pointers public variables
Player can keep track of what space they are currently in	N/A	- All the spaces are true to the map design - A title bar displaying room information is used for validation	Worked as expected
Player must be able to interact with the rooms and not just collect items	For spaces 2 and 4, player can enter integer values for the guessing game	For spaces 2 and 4, player can enter integer values for the guessing game	Worked as expected.
Player's container is used to store collected items	N/A	- The container has a maximum capacity of 3.	Worked as expected.

and has a physical limit		Additional socks will not be stored.	
Mechanism that limits the game from going on indefinitely	N/A	Game keeps track of elapsed time and player is forced to forfeit once game exceeds the limit.	Worked as expected

REFLECTION

I had a lot of fun working on this project. I think the prompt was well chosen, and the project was a really good opportunity to review and test all the material we have covered this term. I really enjoyed that this was a good accumulative test of our knowledge and that it gave us a lot of freedom to do whatever we want after having completed many assignments that had a lot of structural limits. While all the assignments and labs had a certain expected outcome, this final project is completely free-style, which is both liberating and scary at the same time.

The most difficult aspect for me was the interpretation of the project requirements. I had a hard time understanding the scope, specifically: 1) how to create linked rooms that the player can traverse through and 2) what the puzzle or text-game part of the program is supposed to entail. It was not until after having attended the Google help session at the end of last week that I started understanding and formulating ideas on how to proceed with this project. It sounds strange, but I just had difficulty visualizing a space object (which seems to be more abstract) when we have only worked with concrete objects previously.

After I had gotten the conceptual analysis down, my next two roadblocks were how to set the spaces so that they would link or be adjacent to one another and how to control the flow of the game in the main program. The first question I was able to figure out after getting some tips off of Q&A forum, but the second question wasn't answered until I was almost done with my program. I approached this assignment with incremental development, starting with creating space1, creating space3 (both are socks room), and then linking them so that the player can get and in and out of both without a problem. At this point, I realized that I wanted to introduce another class, Hogwarts, to handle both Potter and Space classes before sending them to the main program to reduce redundant codes. I had to go back and review Lab 2 and 3 for ideas, as I had not been doing this on any of the recent assignments. This was also the primary deviation from my initial outline.

On the other hand, my favorite aspect of the project, besides making it Harry Potter themed, was that there were no major surprises. Everything is straightforward, and I was able to borrow a lot of ideas from my previous assignments, and fine-tuning them in the process. I did not have to do a lot of researching this round, besides the time limit implementation, which, once again, really allowed me to test how much I have learned this quarter.