Elizabeth Poore
Final Project – Reflection

Project Design / Game Rules:

Growing up, I loved playing the game of Clue. Seeing that mentioned in the assignment specs, I thought it would be really fun to implement a game using a structure similar to the Clue mansion. Rather than solving a murder-mystery, I wanted to implement a more puzzle-like game where the user was to find similar items throughout the mansion with the ability to pick up items they would like, but to not be able to carry all the items with them at the same time, forcing them to possibly put down items in order to pick up something else.

The theme of the game is to be that a crazy old rich man has died and left behind his mansion for the first brave soul to hunt throughout the mansion to fulfill the requirements to get the vault open to win his vast treasures.

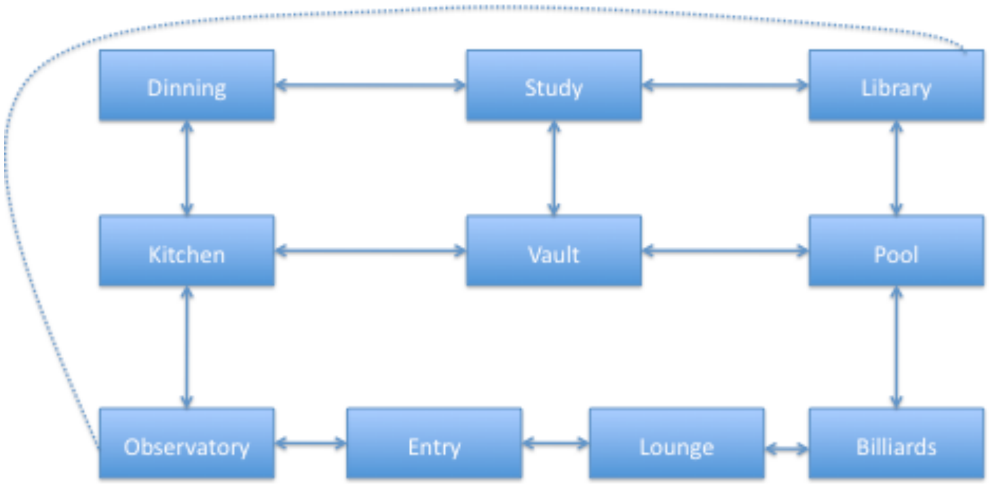
Room types will include:

- Object rooms
    - The object rooms are to have various objects in them that may help the player complete the objective of opening the vault. The player will have the option to pick up objects in the room or put down objects they already have with them to make room for the objects they would like to pick up now
- Boobie Trapped Rooms
    - As stated before… the old man was crazy! He boobie trapped several of the rooms in the mansion. The player will incur damage (lost turns) when entering these rooms. Damage will be determined by a roll of a dice.
        - Contemplating adding a shield object option where if the player picks up the shield to their object inventory, they will experience less damage in a boobie trapped room
- The Vault Room
    - Upon arriving to the vault, the user will find out if they've met the conditions necessary to open the vault and win the game.
    - The player must have 3 matching gems (of any of the types scattered throughout the house) in order to open the vault and win the game

The Room objects will all be derived from an abstract base class

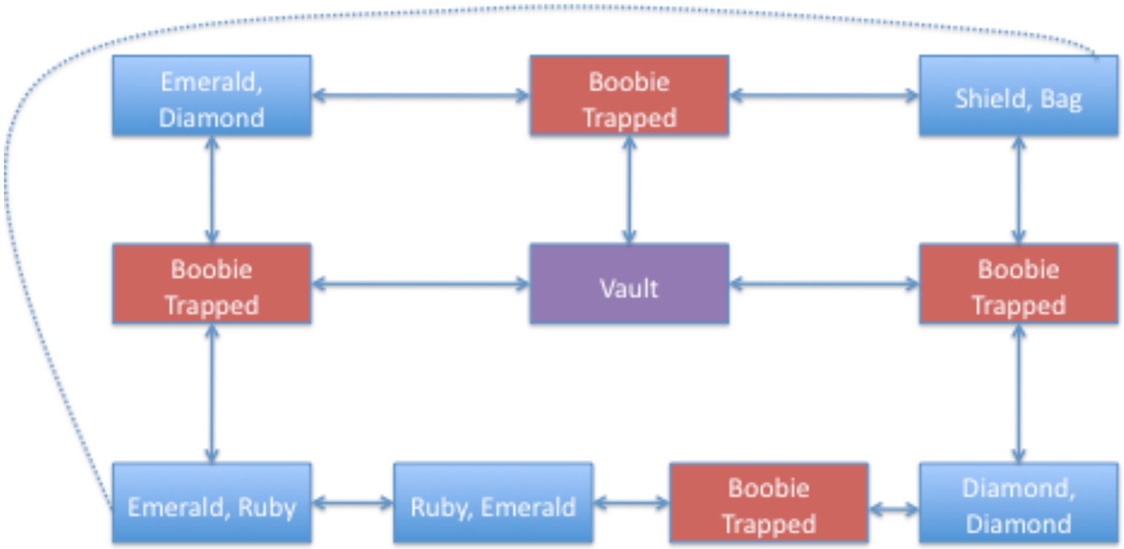
For the game to run and function, a GameWorld Class will be implemented where the Room objects will be initialized and pointers to the Rooms will be added to a vector of pointers to rooms. The GameWorld class will also keep track of the player information such as the current inventory of objects they have as well as the number of turns remaining, etc.

Map of the Mansion



Note: There is a secret passage from the Observatory to the Library. This can be reached going West from the Observatory or East from the Library

Map of the Mansion with Spoilers



Note: There is a secret passage from the Observatory to the Library. This can be reached going West from the Observatory or East from the Library

Changes / Struggles in Design:

- I initially was lost with how to connect the different rooms. At a point it occurred to me that I could make a vector of pointers to Rooms in a separate GameWorld class… from there things really started to fall into place.
- I found it extremely helpful to add as much input validation as possible for the sake of sanity during testing! I had to add functions that returned the size of the object vectors for the rooms and what the player currently had to allow for input validation of the add item and remove item functions. This helped to save a lot of time with just a small amount of work upfront!
- I ended up adding a second type of room which is essentially the same as the original object room, but this room type is an objects room where two items are initialized in the constructor
- I had to make changes to the implementation of the Boobie Trapped room implementation. More details in the testing section.

Testing / Incremental Implementation

Testing occurred throughout the development of this program. As pieces were incrementally put together, they were tested. After the full implementation was complete, testing of running entire game scenarios occurred with changing the number of turns available to either force the game to end early or have so many turns available that I could test and play for a long time before running out of turns.

- First implemented was the skeleton structure of all the rooms connected together. This was tested by using print statements to indicate which room the player was in. The tests performed as expected with a minor error where I had intended for a setEast() function and had placed a setWest() function… good thing for testing ☺
- Next for testing, I wanted to ensure that the player could pick up items and that they would be removed from the room's inventory correctly. I had to update the code for removing the object from the room appropriately, but after that, it worked as expected.
- Next for testing, I tested that if a player put down an object it would be removed from the player's inventory and added to the current room's inventory. This worked as expected.
- Also important for testing was to ensure that the player could only hold as many objects in their inventory as was allowed. This is 2 items at one time until the bag is retrieved… at that point, the player can hold 5 items including the bag. This performed as expected.
- I had the most amount of testing needing to be done while implementing the boobie trapped room concept. Every time I changed something, it would effect something else, but I think I finally got it working properly.
    - It made the most sense to have a flag thrown immediately if the player moves into a boobie trapped room. At that time, the number of lost turns is to be determined. Following determining if there are turns

remaining, the player is forced to choose a new room to go to and the end of turn flag is set to true.

    - Initially I had problems with having the turn proceed as usual even though the boobie trapped rooms don't have items to be picked up and every time an option other than change rooms was selected, the sequence to lose turns came up again.
    - I also had a problem where the room never changed again after going into a boobie trapped room, the lose points sequence just occurred over and over again until the player lost. Needed to declare the end of a turn from within the boobieTrapped room if statement.
- Testing that having the shield gave protection to the player. This worked as expected, but to make things more clear, I gave the player more prompts letting them know what was happening and explicitly letting them know how many turns if any were lost. To keep things simple, only integer division was used, so depending on the value rolled, there's a 50% chance that the player loses slightly less than half of the roll value or in the case of rolling a 1, there is no loss of turns.
- Testing for winning of the game gave results as expected. Upon entering the vault, if the player had 3 matching gems in their inventory, they won. If they entered the vault with turns remaining, but not 3 matching gems, they were informed that they needed 3 matching gems in order to open the vault.
- Testing for losing a game came up with some small errors
    - There was no trouble if a player ran out of turns in an object room because the player finishes their turn, the number of turns remaining reaches 0 and end of story, but problems arose when a player ran out of turns while in a boobie trapped room. In a boobie trapped room, it's possible for a player to technically go to less than 0 remaining turns. This caused the program to print out as many "you've lost" statements as there were turns at or less than 0. Ended up fixing this by adding an extra if/else statement in the decrementTurn function so that if the number of turns is less than 0 after assessing the number of turns lost, to set the number of turns remaining to 0. I also changed the print statement for determining if the player has won or lost the game to be outside of the do-while loop for the game rather than a print statement coming out of the inPlay function as I previously had.
        - I found an additional problem after fixing the problem where I wouldn't allow the number of turns to go below 0. Adding the inequality into the decrementTurn function made things look strange because the numLostTurns function from the boobieTrapped room class had print statements, so the function was initially being treated as if it were only a bool, but it wasn't. Ended up removing the print statements from within the class and adding them to the decrementTurn function making it so the only thing the numLostTurns of the boobieTrapped room class did was to return the explicit value of the number of turns lost by a roll of a 6-sided die.

- Fixing this led to another error which was fixed by calling the numLostTurns function only once and assigning the value to a variable rather than calling the function multiple times (which gives a different random value each time… oops!)

Reflection

- This assignment proved to be a worthwhile challenge for me. I first struggled with getting the idea of what we were supposed to be doing with having rooms connected by pointers, but after it clicked things really came together.
- My game design likely could have been a bit more creative admittedly, but I really wanted to have the chance to have the player have to pick up and put down items in a way that really utilized that requirement of the assignment specifications because that seemed like a good chance to be practicing with vectors and keeping track that everything is in the right place at the right time.
- If I had time and a chance to design things again, I think the overall design could have been more streamlined between my Room objects and the GameWorld, but overall I'm happy with how everything turned out (at least it's all working ☺)
- This was also good practice with using separate files to shorten my main function. I'm not certain if the main function is longer than it still should even be, but it really became condensed when making the gameSetUp and moveRooms their own functions.
- I started feeling the pain of having print statements in functions that return values. I had set up a lot of my functions to have print statements in them, but then if they were to be used in an inequality or anything where I wasn't explicitly meaning for anything to print, the statements would still show up. It was a valuable lesson about considering when and where to include print statements!!
- I'm looking forward to gaining more practice in this program, but overall am happy and surprised with how much I've learned in these courses… I guess it makes sense though sense these have taken nearly all my free time aside from work and family time!