Derek Yang

Document (Design + Reflection)

Program Design

Problem: Create a themed game where the user will go through rooms and interact with the room in order to complete the objectives of the game. Each room will need to have four pointers. Each of the pointers will either need to point to other rooms, point to an interaction, or point to null. You will need to create at least 6 spaces, 3 of which must be derived.

Theme: You are a student in CS162 who is guiding their friends Heart and Brain through the challenge of the course. The student will need to control the level of serotonin gained from Brain so that Brain can prevent Heart from exploding. Think of Brain and Heart as individuals with their own personalities. Brain is the more logical fellow, whereas Heart is more spontaneous. Throughout the game, each room is represented as a topic in CS162. The rooms have a TA or temper analyzer in them to help guide our heroes. The TA will analyze the temper and master of the student. This is done by rolling a die and comparing the results to the student's roll. Once the TA has deemed that the student has passed, they will award a mastery. There are a total of 6 rooms to acquire masteries. The student can only obtain 5 masteries and no more. Once the student has obtained 5 masteries, they will turn in the masteries to the instructor in the final room. The instructor will then inspect to make sure that the student has collected 5 masteries and award a large supply of serotonin for Brain.

Classes:
Space: Parent Class
- Returns information about the spaces
- Sets information about the spaces
- Has four pointers to different rooms or NULL

Topics 1 – 6: Child Classes
- These classes inherit from the Space class and point to each other
- The classes have differing levels of difficulty between them

Finals: Child Class
- The final room which contains the instructor
- If the student does not have enough masteries, the instructor will turn them away

Course:
- This class is much like a gameplay class
- The rooms will be linked together in this class

Input Validation

- The user will be asked mainly integers in the menu options. The cases are usually within a range of answers. Therefore, I will create a function that checks for an int, and then if that int is within a specified range.
    - Ex: choose 1 to 3 will ask for an int within range 1 to 3
    - Ex2: choose starting grid width will be min 1 and max int

Test Case	**Input Values**	**Functions**	**Expected Outcomes**	**Observed Outcomes**
Input too low	Input is less than the min	Main() Validation() If input < min	Loop back and ask the user to re-input	Continually asks user to enter a higher int
Input too high	Input is more than the max	Main() Validation() If input > max	Loop back and ask the user to re-input	Continually asks user to enter a lower int
Input in correct range	Input is within max and min	Main() Validation() If min>input>max	Accepts the correct answer and exits the while loop	Loop stops and accepts correct int
Input is not an int	Input is a string, bool, float or other variable	Main() Validation() If input != int	Loop back and ask the user to re-input an int	Loops until an int is entered within the designated range
User tries to enter a non-existent room	User enters out of range number	Main() Course() Topics() Validation()	The validation for max numbers will kick in and the user will be asked to input a number within range	The vector size was reported correctly and the user had to enter in an integer of correct size in order to proceed
User tries to enter final room without all masteries	User collects less than 5 masteries and enters final room	Main() Finals() Course() Validation()	The user is turned away by the professor	The user is indeed turned away by the professor with an if statement that checks if the size of masteries vector is at the requirement
User tries to gain too many masteries	User attempts to collect 5+ masteries	Main() Finals() Course() Validation(0	The user will not be able to collect more masteries as each room will keep track if it	The rooms use an if statement to execute the interaction and give a mastery IF the user does not already have one OR has not reached max capacity

			handed out a mastery	

Reflection

This program seemed to be a lot easier than the last. The only reason why it was much easier than the other projects was because I felt more comfortable coding in C++ than before. I found that I looked less at old projects in order to code this one. Other projects I would spend hours looking up references and past projects to see how to code certain pointers, data structures, or even to do basic syntax. Now it seems to be all second nature (well mostly). I found that the planning phase of this project went a lot smoother than others as well. Almost everything went according to plan. The only thing that didn't go according to plan was how I planned the room pointers. I found that I ran into a slight hiccup when I tried to make a pointer to a pointer instead of just directly pointing to my object. I had to look back at my project 4 to solve this problem.

The hardest part of this project was the rooms as stated previously. I found that even though it was hard, it was not complicated. It was actually pretty straight forward compared to making doubly linked lists and queues. I just decided to straight have the pointers point to other rooms because my room layout was not dynamic. As in, the rooms did not change locations every round. I ended up choosing this specific theme because I felt like the TA's were a HUGE help this quarter. Temper analyzer really is a keyword because they would be super patient with a lot of students, including myself. I found that with their guidance, I was able to do most of the work. Even when the TA did not have the answer, they always had a direction or material to point to the answer.