Final Project Design and Reflection

Eric Riemer

12/3/17

Design

- Space Class (Base Class)
  - Protected:
    - Space Pointer: Top
    - Space Pointer: Right
    - Space Pointer: Left
    - Space Pointer: Bottom
    - String: Password to win game

    Public:
    - Pure Virtual Destructor
    - setTop
    - setRight
    - setLeft
    - setBottom
    - virtual interact with room
- Coutyard (Derived Class)
  - Public:
    - Default Constructor
    - Default Destructor
    - Interact with Room
      - Explore Fountain
      - Leave Room
      - Exit Game
- Stables (Derived Class)
  - Public:
    - Default Constructor
    - Default Destructor
    - Interact with Room
      - Brush Horse (Find Letter)
      - Talk to Stable Master
      - Leave Room
      - Exit Game
- Library (Derived Class)
  - Public:
    - Default Constructor
    - Default Destructor
    - Interact with Room

- Read Book (Clue)
- Peruse Books in Library (Find Letter)
- Leave Room
- Exit Game

❖ King's Chambers (Derived Class)
  - ➢ Public:
    - Default Constructor
    - Default Destructor
    - Interact with Room
      - Look in Desk Drawers (Find Letter)
      - Look in King's Closet (Clue)
      - Leave Room
      - Exit Game

❖ Dungeon (Derived Class)
  - ➢ Public:
    - Default Constructor
    - Default Destructor
    - Interact with Room
      - Talk to Prisoner (Find Letter)
      - Investigate the Torture Room (Find Passage to Tower)
      - Leave Room
      - Exit Game

❖ Barracks (Derived Class)
  - ➢ Public:
    - Default Constructor
    - Default Destructor
    - Interact with Room
      - Look under soldiers' bed
      - Look through soldiers' chests
      - Leave Room
      - Exit Game

❖ Kitchen (Derived Class)
  - ➢ Public:
    - Default Constructor
    - Default Destructor
    - Interact with Room
      - Look in Pantry
      - Talk to Chef (Find Letter)
      - Leave Room
      - Quit Game

❖ Armory (Derived Class)
  - ➢ Public:
    - Default Constructor

        - Default Destructor
        - Interact with Room
            - Investigate Rack of Spears (Find Letter)
            - Look at King's Sword (Clue)
            - Leave Room
            - Quit Game
- Main Hall (Derived Class)
    - Public:
        - Default Constructor
        - Default Destructor
        - Interact with Room
            - Sit in Kings Chair (Find Letter)
            - Look around Table (Transported back to Courtyard)
            - Leave Room
            - Exit Game
- Player Class
    - Private:
        - Vector (letters): holds letter collected from the rooms
        - Space Pointer (location): holds current location
    - Public:
        - Constructor
        - Destructor
        - setLocation
        - getLocation
        - move
        - add letters
        - print letters
- Game Class
    - Private:
        - Player Pointer
        - Space Pointer: Courtyard
        - Space Pointer: Library
        - Space Pointer: Main Hall
        - Space Pointer: King's Chamber
        - Space Pointer: Kitchen
        - Space Pointer: Armory
        - Space Pointer: Dungeon
        - Space Pointer: Stables
        - Space Pointer: Barracks
        - Max Number of Turns Allowed
        - Number of Turns Taken by User
    - Public:
        - Constructor
            - courtyard = new Courtyard

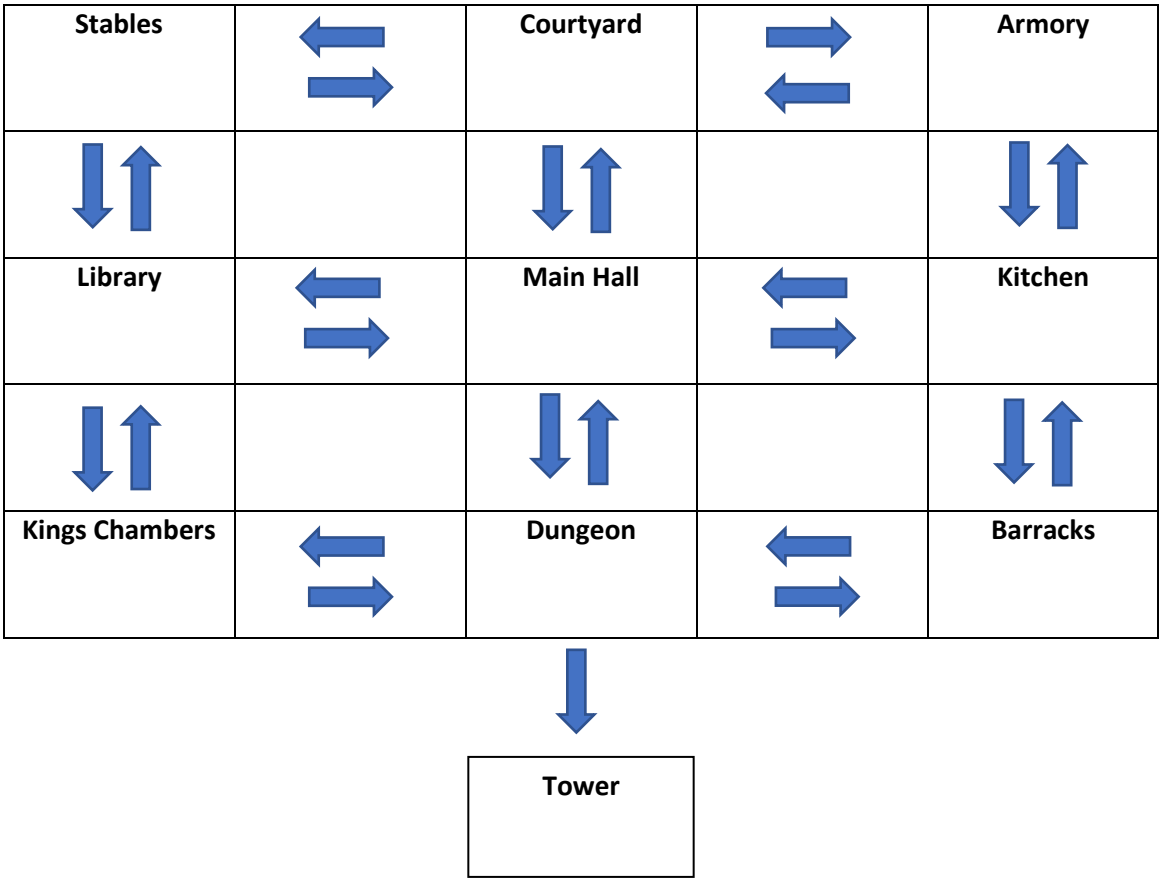
- library = new Library
- mainHall = new Main Hall
- kingsChamber = new King's Chamber
- kitchen = new Kitchen
- armory = new Armory
- dungeon = new Dungeon
- stables = new Stables
- barracks = new Barracks
- player = new Player
- SET EACH ROOM'S TOP, BOTTOM, RIGHT AND LEFT POINTERS HERE INSTEAD OF IN EACH ROOM'S CONSTRUCTOR
- Set starting location of player to courtyard
- Set Max number of turns
- Set the count of turns to zero

❖ Menu Class
- ➢ Default Constructor
- ➢ Default Destructor
- ➢ Input validation for 2 options
- ➢ Input validation for 3 options
- ➢ Input validation for 4 options

Game Objective:

- Collect 6 letters
- Find the Entrance to the tower
- Guess the password to the tower by rearranging the letters to save the princess
- Interact Function (different return options for testing):
  - Return 1: A letter was found
  - Return 2: Move to a different room
  - Return 3: Nothing was found
  - Return 4: Game over, you win

Game Map



Test Plan

Test Case	Input Values	Expected Outcome	Actual Outcome
Courtyard	1	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns
Courtyard	2	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Courtyard	3	Program Quits	Program Quits
Courtyard: Change Room Menu	1	Go to Main Hall; Subtract 1 from Turns	Go to Main Hall; Subtract 1 from Turns
Courtyard: Change Room Menu	2	Go to Stables; Subtract 1 from Turns	Go to Stables; Subtract 1 from Turns
Courtyard: Change Room Menu	3	Go to Armory; Subtract 1 from Turns	Go to Armory; Subtract 1 from Turns
Main Hall	1	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns
Main Hall	2	Go to Courtyard;	Go to Courtyard;

		Subtract 1 from Turns	Subtract 1 from Turns
Main Hall	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Main Hall	4	Program Quits	Program Quits
Main Hall: Change Room Menu	1	Go to Courtyard; Subtract 1 from Turns	Go to Courtyard; Subtract 1 from Turns
Main Hall: Change Room Menu	2	Go to Kitchen; Subtract 1 from Turns	Go to Kitchen; Subtract 1 from Turns
Main Hall: Change Room Menu	3	Go to Dungeon; Subtract 1 from Turns	Go to Dungeon; Subtract 1 from Turns
Main Hall: Change Room Menu	4	Go to Library; Subtract 1 from Turns	Go to Library; Subtract 1 from Turns
Stables	1	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns
Stables	2	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns
Stables	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Stables	4	Program Quits	Program Quits
Stables: Change Room Menu	1	Go to Courtyard; Subtract 1 from Turns	Go to Courtyard; Subtract 1 from Turns
Stables: Change Room Menu	2	Go to Library; Subtract 1 from Turns	Go to Library; Subtract 1 from Turns
Library	1	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns
Library	2	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns
Library	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Library	4	Program Quits	Program Quits
Library: Change Room Menu	1	Go to Stables; Subtract 1 from Turns	Go to Stables; Subtract 1 from Turns
Library: Change Room Menu	2	Go to Main Hall; Subtract 1 from Turns	Go to Main Hall; Subtract 1 from Turns
Library: Change Room Menu	3	Go to King's Chambers; Subtract 1 from Turns	Go to King's Chambers; Subtract 1 from Turns
Kings Chambers	1	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns

Kings Chambers	2	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns
Kings Chambers	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Kings Chambers	4	Program Quits	Program Quits
Kings Chambers: Change Room Menu	1	Go to Dungeon; Subtract 1 from Turns	Go to Dungeon; Subtract 1 from Turns
Kings Chambers: Change Room Menu	2	Go to Library; Subtract 1 from Turns	Go to Library; Subtract 1 from Turns
Dungeon	1	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns
Dungeon	2	Display menu with options to either attempt to enter password or keep exploring the castle	Display menu with options to either attempt to enter password or keep exploring the castle
Dungeon	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Dungeon	4	Program Quits	Program Quits
Dungeon: Torture Room Menu	1	Prompts user to enter password; If user enters "BINARY", output the winning message; If user doesn't enter the correct password, display Dungeon menu and subtract 1 from turns	Prompts user to enter password; If user enters "BINARY", output the winning message; If user doesn't enter the correct password, display Dungeon menu and subtract 1 from turns
Dungeon: Torture Room Menu	2	Display Dungeon menu and subtract 1 from turns	Display Dungeon menu and subtract 1 from turns
Dungeon: Change Room Menu	1	Go to Barracks; Subtract 1 from Turns	Go to Barracks; Subtract 1 from Turns
Dungeon: Change Room Menu	2	Go to King's Chambers; Subtract 1 from Turns	Go to King's Chambers; Subtract 1 from Turns
Dungeon: Change Room Menu	3	Go to Main Hall; Subtract 1 from Turns	Go to Main Hall; Subtract 1 from Turns
Barracks	1	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns
Barracks	2	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns

Barracks	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Barracks	4	Program Quits	Program Quits
Barracks: Change Room Menu	1	Go to Kitchen; Subtract 1 from Turns	Go to Kitchen; Subtract 1 from Turns
Barracks: Change Room Menu	2	Go to Dungeon; Subtract 1 from Turns	Go to Dungeon; Subtract 1 from Turns
Kitchen	1	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns
Kitchen	2	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns
Kitchen	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Kitchen	4	Program Quits	Program Quits
Kitchen: Change Room Menu	1	Go to Armory; Subtract 1 from Turns	Go to Armory; Subtract 1 from Turns
Kitchen: Change Room Menu	2	Go to Main Hall; Subtract 1 from Turns	Go to Main Hall; Subtract 1 from Turns
Kitchen: Change Room Menu	3	Go to Barracks; Subtract 1 from Turns	Go to Barracks; Subtract 1 from Turns
Armory	1	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns	Randomly Generate and Add a Letter to the Player's Vector; Subtract 1 from Turns
Armory	2	Output Message; Subtract 1 from Turns	Output Message; Subtract 1 from Turns
Armory	3	Display menu with options for which room to go to next	Display menu with options for which room to go to next
Armory	4	Program Quits	Program Quits
Armory: Change Room Menu	1	Go to Courtyard; Subtract 1 from Turns	Go to Courtyard; Subtract 1 from Turns
Armory: Change Room Menu	2	Go to Kitchen; Subtract 1 from Turns	Go to Kitchen; Subtract 1 from Turns

Reflection:

I found this project to be relatively easy. I created a base class called space and created 9 derived classes from it. The base class includes 4 space pointers that are used by the derived classes to point to the rooms they are connected to. Each derived class represents a different room in the castle. In order to figure out the configuration of room, I created a table/grid and filled it with 9 rooms. This heled me a lot when trying to program the way in which the rooms were linked. I designed the layout of

the castle so that corner or side rooms would have some of their space pointers set to null since they pointed to less than 4 other rooms. When the user enters a room, they are prompted with a menu with options for interacting with the room. Some of the options will lead to finding letters that the user will later need to refer to in order to guess the password and save the princess. Other options in the room menus will give the user clues to the whereabouts of the secret passage leading to the tower where the princess is being kept. And other menu options will be useless and will set the user back a turn. If the user fails to find the secret passage and correctly guess the password within 25 turns, they lose the game.

I decided to use the game class' constructor to create each room object and set each room object's space pointers. I decided to set each room object's space pointers in the game constructor instead of in each room constructor because I needed the room objects to be created first before I could link them with the space pointers. If I had tried and set the space pointers in each room's constructor, it wouldn't have had an object to be set to, since that room object had not been created yet. Because all of the room objects were created in the game class constructor, I used the game class destructor do delete each room.