

1. Define what the program should do.
 - a. Purpose: Create a text based game where the player moves through various rooms to unlock the exit from a hidden temple.
 - i. Game will be based on the Escape the Curse of the Temple Game.
 - ii. While the actual game allows up to 5 “players” currently only one will be allowed. Additional computer player functionality might be added later.
 1. One player version changes rules by allowing the player 7 instead of 5 dice.
 - iii. The game has a 10-minute time limit.
 1. The player must escape the temple before the time limit runs out.
 - iv. Movement and interacting with the rooms are controlled by the player rolling dice and then “using” the rolled images to complete room tasks.
 - a. Black mask – causes a die to be temporarily removed from the players rolls.
 - b. Gold mask – each one rolled removes the effect of two black masks.
 - c. Red Torch – used to enter rooms, access treasure, or activate gems
 - d. Blue Key - used to enter rooms, access treasure, or activate gems
 - e. Green adventurer X 2 – used for moving between rooms and activating some gems.
 2. Rooms are generated randomly as players move from room to room.
 - a. To meet the program requirements the exit room will be locked until 6 rooms generated.
 - i. Central room will be the players starting position with rooms branching off from it.
 - b. Other than the exit and the central room other rooms have the possibility of being a treasure room, gem only, or a standard room.
 - i. Room characteristics randomly chosen from the path direction possibilities and icons that will allow interaction.
 - ii. Standard rooms will have a 50% chance of containing treasure chest.
 1. Treasure if unlocked results in “items” that player can use in other rooms to accomplish goals (e.g. treasure might contain torches that can be held and used later in a gem room to add to dice rolled)

3. Gems in rooms must be activated and collected in the players pack to escape the temple.
 - a. The game difficulty can be selected at the beginning changing how many gems are in the temples total pool of gems
 - i. 25 = hard, 15 = medium, 5 = easy
 - ii. To exit (win) player needs to roll number of keys on their dice greater than the number of gems still in the temple.
 - iii. This means gems in players pack will subtract from the number of keys required.
 - b. Input:
 - i. Human player will have to input the number of players for the game (default will be 1 for the assignment) and the difficulty level in the game setup.
 1. Main menu will have the option to setup the game, start the game, or exit.
 - ii. While the 10-minute timer counts down the player can roll dice as often as needed to escape the temple.
 1. Choosing to roll the dice and using the rolled "items" to interact with the rooms are the primary inputs
 2. Room interaction involves user being able to explore new rooms (if open paths are available, activate the rooms gems, move to adjacent rooms, or get the rooms treasure (if there is one)
 - iii. After the game is over there will be an option to ask the user if they would like to play again
 - c. Processing:
 - i. A linked list structure will hold the rooms. If I can make it work would like to make the rooms have the option of going more than 2 directions.
 - ii. There will be a timer loop/function running separate from the console input portion of the game to keep track of the 10 minutes of game play.
 1. Timer will only display an update every 30 seconds
 - iii. Dice rolls will be controlled using the dice class based on a 6-sided die. Faces will correspond to 1 of six images instead of numbers.
 - iv. Random number generation will control the properties of the generated rooms
 1. Assign each room type a number value
 - a. 1 – 4 is a gem room, 5 – 16 will be a standard room
 - b. Additionally a standard room will have a 50% chance of holding a treasure

- i. Treasure will either be 1 – 5 torches that can be held to use later, a magic key which can be held and used to teleport to any room, and a health kit which can be used to unlock all players black mask dice
- ii. Dice rolls act as items that are stored in users pack until used
- iii. After items are used they are removed from the pack
- iv. In the case of dice after using they must be rolled again
 - 1. Non-used dice can be kept in pack until used
- v. Treasure items are removed following use
- c. Each room will have two travel icons associated with it (torch, key, adventurer)
 - i. They can both be the same
 - ii. These are the required “items” to enter the room
- d. If the room has treasure the chest will also have 2 random icons associated with it to open it
- e. generating new room paths requires 2 adventurer tokens

d. Output:

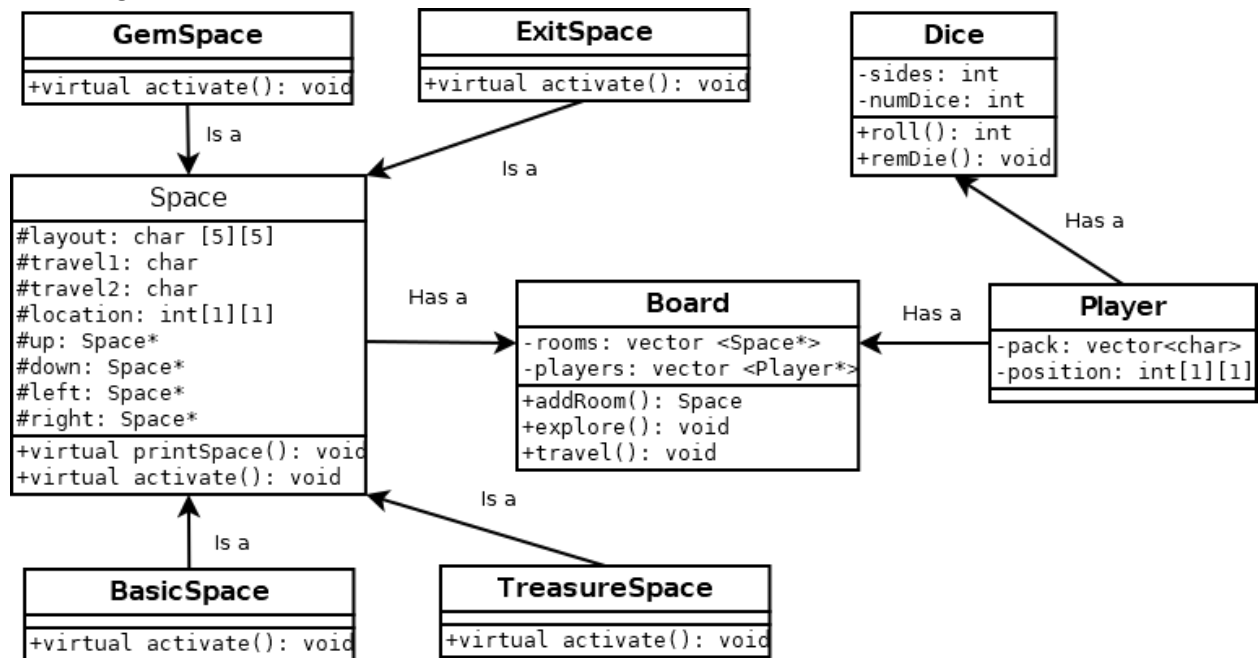
- i. Display countdown timer for the game.

2. Program design

- a. Main menu with the following options:
 - i. Setup game
 - ii. Start game
 - iii. Exit
- b. Setup game selected
 - i. Creates the game object
 - 1. Generates the 6 starting rooms
 - 2. Start room will have left and right path plus travel icons
 - 3. 4 other rooms generated branching off from start room
 - a. Each room must contain at least one non-null path connecting it to the previous room
 - b. Random probability for it to have 1-3 other paths branching off from it
 - i. New rooms can be attached to these paths
 - c. 50% chance that new room is standard, or gem only
 - i. Standard rooms further have 50% chance of having a treasure

- d. Gem room has three possible gem activations of increasing difficulty yielding 1, 2, or 3 gems
 - i. Only one of them can be used
 - ii. Increasing gems totals require 4, 5, or 6 of the associated icons to unlock (either key or torch)
- 4. Exit room not unlocked until 5 other rooms have been generated
 - a. Only has one entrance path and travel icons
- c. Start game will start the 10 minute timer and allow the player to start using dice
 - i. Main game loop exits when player can exit the temple, or the time runs out
 - ii. Player is positioned in the start room
 - 1. Start room only allows left and right movement
 - iii. Room types and generation
 - 1. Rooms layout is a 5x5 array
 - a. Travel icons are stored in [0][0] and [0][1]
 - b. Possible exits are [2][0], [0][2], [4][2], and [2][4]
 - 2. New rooms can have between 1 and 3 exits from them (determined randomly not including the entrance, so up to 4 paths allowed)
 - 3. Room types are standard room, treasure room, or a gem room
 - 4. Entrance requirements to room will be determined randomly by having two of the 3, 4, and 5 symbols from above being required to move between rooms
 - 5. Gem activation and treasure unlocking will also have random symbol
 - a. Gem activation only requires one type of symbol, but more based on the type of room
 - b. Standard room 3 symbols unlocks a single gem
 - c. In gym room 1, 2, or 3 gems may be unlocked requiring 3, 4, or 5 symbols respectively
 - iv. Player rolls dice
 - 1. Game returns and display the results of the roll
 - 2. Player will pick the dice they want to use for their turn
 - a. If black skull dice are rolled then the dice are temporarily removed from use, they will still show on display, but cannot be rolled
 - b. Used dice are returned to a ready to be rolled stack
 - v. In rooms with no exit rooms player can explore to add new rooms to the map.
 - 1. Exploring spends two adventurer icons

3. Class Diagram



4. Testing Plan

| TEST | INPUT VALUES | DRIVER FUNCTIONS | EXPECTED RESULTS | OBSERVED OUTCOMES |
|-------------------------------|------------------|---|---|--|
| PLAY GAME ON EASY | 1 player, mode 1 | timer loop, game loop | Try to make it to exit as quick as possible prompting game to end before time runs out. Exit should not be available until 5 rooms are available. | Timer functions as expected. Timer was set to twenty seconds to speed testing. If console left un-attended, or no input for longer than timer will cause the game to end as soon as the timer is polled. |
| INCREASE DIFFICULTY TO MEDIUM | 1 player mode 2 | increasing difficulty increases gem count | Should be harder to win, but not impossible. Gems placed in pack to aid escaping should decrease from the roll needed to escape the temple. | Test not performed due to incomplete project. |
| PLAY ON HARD | 1 player mode 3 | room creation | Travel to unexplored areas should prompt room generation. | Test not performed due to incomplete project. |

5. Reflection

A common theme through the class is that I tend to get fixated on things that are way outside the scope of the requirements for the program. In this case it was trying to figure out how to create a `std::map` to store all my room objects. I was able to figure out how to implement it using a `std::pair` for the **Space**'s key representing the coordinates that the space resided in on

the map. The only problem was I was neglecting to implement one of the key requirements, the Space direction pointers. Unfortunately, this was the day before the due date requiring a total re-work of my code that I was never able to finish.

While it took a while I am happy to say that despite the limitations of working in the console I could get a timer working for the game. Based on these limitations the console will only display the elapsed time when the render() method is called forcing the main game loop to reiterate. A main side effect of this is that if the user just sits at the console without inputting anything there is no indication that time has elapsed until they make a menu selection, but by this time a enough time may have passed to return the gameOver flag.

The biggest change I had to implement was dropping my Linked list structure to figure out some other method to have spaces stored. Ultimately, I was able to find some help by realizing that the pointers only had to function as doors to other rooms, not necessarily tying the rooms together as the nodes would in a linked list.

I did still need an array of pointers though, even though most of the other students did not go this route. Once again this is since I was probably forcing myself to do something more complicated than was required. I really wanted to have the array available to give me coordinates to keep track of where the spaces were in relation to each other. Then, I would be able to print out a full map of my game board for the player to use in the game. The way the game was supposed to work playing without a map would have made things difficult.

At this point the game does not fully function as a game. It can create the first type of space, will print the spaces that are stored in the array, and the timer works. I do not have any memory leaks in the end of this, but that is not saying much since the game doesn't do what it is supposed to. I also have menus with working input validation. I wish I had been able to get more working, but I ran out of time and had to submit what I had.