Final Project

Space class   * ~~Each~~: 3 children needed

Space   * 6 spaces needed

```
         ┌──────┬──────┬──────┬──────┬──────┐
       Ryan    Alex   Jess  Taylor Morgan  Pat
```
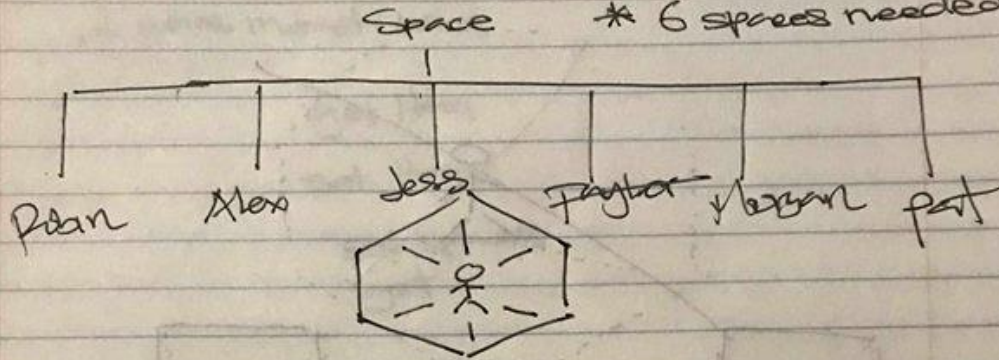
Container to hold items : vector?
- need to check if specific item is in container to allow certain interactions
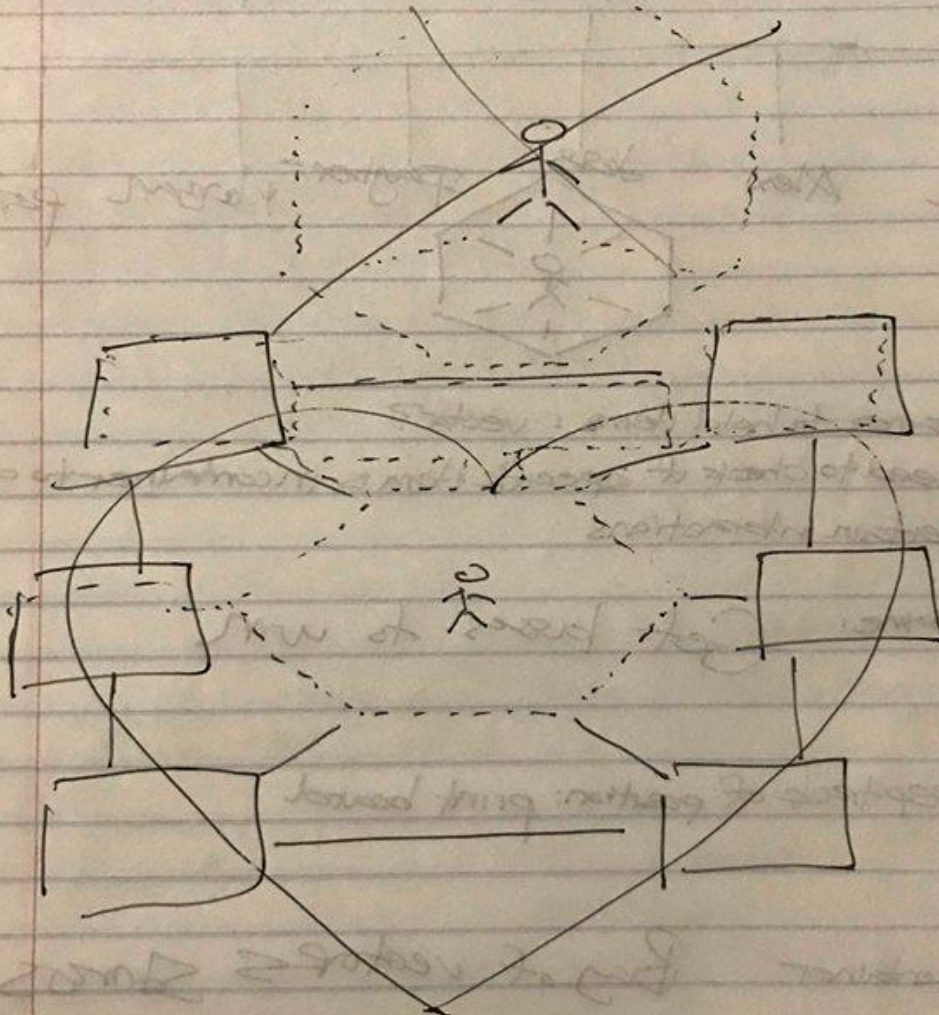
① Theme: Get kisses to win

② Keep track of position: print board

③ container   Bag of vectors strings

④ time limit...
        Health depletes

6 spaces: 6 characters
Alex, Jess, Morgan, Pat, Robin, Taylor

2 items in each room

get item

set item

set spaces
get

interests... 12 interests:

6		1
5	0	2	plone spaces?
4		3

Room 1 ⟶ Basketball Ⓙ   Alex   Anime
        ↘ Hiking Boots Ⓜ   Cast Iron Skillet

Room 2 ⟶ Anime Ⓐ   Jess   Basketball
        ↝ Camera Ⓟ   Tent

⟶ Room 3 ⟶ Cast Iron Skillet Ⓐ   Morgan   Shakespeare
          ↓ Tent Ⓙ   Hiking Boots

10 items

Room 4 ⟶ Ballet Shoes   Pat   Speedo
        ↝ Bear   Camera

Room 5 ⟶ Shakespeare Ⓜ   Robin   Ballet Shoes
        ↝ Speedo Ⓟ   Bear

Have 5 characters derived from Space class

Each character has their own action and interests

Interests are clues to what player must find in each characters room

Menu execution

       Shows users rules and how to win

       Shows the map I spent hours on

       Gives options to move about rooms depending on what the current room

            In a room:

                  Two items in each room that can be picked up

                  Must have bag size limit

                  If limit reached, must give the items to specific characters that like that hobby
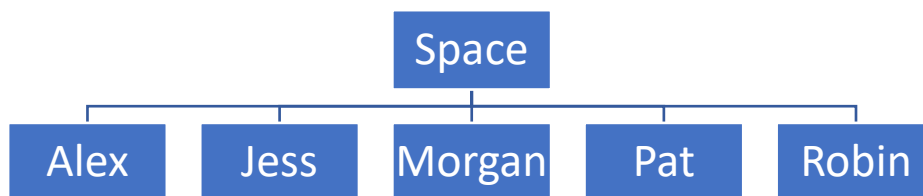

            Interact with players to give them the item

       Health = 20; depletes as enter each room

       Must received 3 kisses to win

       Receive a kiss by giving 2 items of interest to corresponding characters

```
                        Space
        ┌───────┬─────────┼─────────┬───────┐
      Alex     Jess    Morgan      Pat     Robin
```

Derived characters from space class

checkInt (input validation) with range

Test case	Input value	Expected outcome	Observed outcome
Input 1 to move to specific room	Input = 1	Moves character to room - Picks up item	Moves character to room - Picks up item
Input 2 to move to specific room	Input = 2	Moves character to room - Picks up item	Moves character to room - Picks up item
Input 3 to move to specific room	Input = 3	Moves character to room - Interacts with inhabitant	Moves character to room - Interacts with inhabitant
Input 4 to move to specific room	Input = 4	Moves character to room	Moves character to room
Input 5 to move to specific room	Input = 5	Moves character to room	Moves character to room
Input extremely high	Input = 10000000	Catches error and prints TOO HIGH Prompts for another input	Catches error and prints TOO HIGH Prompts for another input
Input nothing		Catches error and prints INVALID Prompts for another input	Catches error and prints INVALID prompts for another output
Input negative number	Input = -1	Prompts for another integer	Prompts for another integer
Input letters and numbers	Input = 10abc	Prompts for integer	Prompts for integer
Input numbers and letters	Input= abc10	Prompts for integer	Prompts for integer
Input float	Input = 100.1	Prompts for integer	Prompts for integer

***Segmentation fault occurred because I did not use the range overloaded version of my checkInt method.

Reflection

Changes in design:

The hardest thing about this project was settling on what kind of theme I wanted. The backbone of the project seemed simple, as it tested our knowledge of linked lists of pointers. Other than that, it seemed like an easy feat. A change in design was how I wanted the map to look. Initially, I wanted it to look like a heart to better simulate a Love Hotel. I had 6 points to connect around the edge of the heart, but found this to be hard, so I change it to only 5 points because the middle lobby room also counted as a Space. Miscellaneous changes included dialogue that I wanted inhabitants to say and what kinds of hobbies I wanted each one to have.

Problems:

The biggest problem I had was surprisingly not anything to do with managing memory, but how to display the hotel! The hardest thing was making sure that the backslash character was "\\" to avoid escaping the character. What made it worse was even though it looked good as I am writing the code, when I ran it, the console output did not look like how I hard coded it. Because of this, I had to constantly make little changes to the map, and then run it, and it was just tedious to say the least.

Other than coding the map display, everything else ran smoothly. Deallocating memory was not as difficult. Utilizing vectors made everything a breeze.

In conclusion, I came into this project knowing that I will spend a lot of time on the theme and eventually came up with something fun. Looking at the past final projects, it is amazing how the back bone of the code is essentially the same, but the theme is what kinds the spice and creativity to it all.