

## FINAL PROJECT - REFLECTIONS

### DESIGN DESCRIPTION

#### Purpose:

In this project we are tasked with writing a program that will play a game involving carried items and rooms to navigate that are linked together using linked structures. We must gather items as part of the solution for completing the game.

#### Requirements:

- There must be 6 rooms total where at least 3 are of a different type.
- Each space will have 4 pointers to the surrounding rooms.
- The parent space class must be abstract.
- Provide a goal for the player using a theme.
- Keep track of what space the player is in.
- Items must be included as part of the solution and will be held in a container that stays with the player.
- There should be a time limit in some form.
- The player must interact with part of the space structure such as opening doors.

#### Design:

##### Game Class:

Where the room objects are allocated and linked. Also where the game progression functions are located including: the timer, state changes, room switches, inventory management, and movement.

##### Room Class:

The base class for the multiple rooms. Consists of setter, getter, and display functions meant to be grabbed by the game class. Also determines next space for player based on inputs, and bool functions for checking if the player is at doorways. Acts as an abstract class for room descriptions and items.

##### Room Child Classes:

The constructors build on the basic constructor from the room class to create unique rooms. Using abstract functions from the base class, each room can have its own descriptor and item. The items are "picked up" and a function returns the string value for the game class to add to the inventory.

##### Main function:

Simply creates a game object and calls its play function.

## TEST PLAN / RESULTS

Case	Input	Function	Expected Outcome	Actual Outcome
Switching between rooms	User move player to doorway	changeRoomRight() changeRoomLeft() changeRoomUp() changeRoomDown()	New room should display with the player spawning in the corresponding doorway	New room displays with the player spawning in the corresponding doorway
Adding items to backpack	User walks over item	inventory.push_back() ( )	The item should be displayed in the inventory	The item is displayed in the inventory
North Room door opening	User collects the three jewels	checkInv()	The player should receive a message that a passage opened and the north room should change states	The player receives a message that a passage opened and the north room changed states
Leaving the ruins	User moves toward entrance staircase in central room	Bool leave()	Should output some message to the user that they left the ruins and end the program	A message was outputted to the user and the program ended.
Health drain increase	User takes the Idol	checkInv()	User should receive a message that their health is draining faster and each movement should drain quadruple health	User received a message that their health is draining faster and each movement drained quadruple health
Input validation	User inputs incorrect menu option		Should tell user that the input was invalid and ask for new input	User was told that the input was invalid and asked for new input
Attempting to leave room boundaries	User try to walk through #		User stays in place / overwrite wall space	User stayed in place / did not overwrite wall space

## REFLECTIONS

This assignment in particular forced me to take things step by step. Initially I was being overly creative on theme ideas when what I needed to do was set up the basic functions of the game, such as player movement between rooms, before coming up with a theme. I figured that the biggest hurdle was going to be properly linking the rooms, which it was. Using what we had learned from linked lists, at first I tried creating a struct that would link the rooms together as they were created, but I wasn't able to get that idea to a functioning state. Instead I decided to allocate the spaces and link them all together in the game constructor which ended up working best. My next issue was moving between rooms properly. I didn't realize that as I left a room it kept the player symbol where it last was, so when I'd try to return to a room the game would basically crash. Once I made that leap, I was able to fix the issue, set up the layout of the game, and begin figuring out a theme. I ended up going with an indiana jones type of setup where the player must collect three jewels before a door to the final chamber would open. I had wanted to do a boulder at the end, but that was somewhat ambitious, so I went with speeding up the health drain instead, so that players would have to make a mad dash back to the entrance, which I thought would have the same effect. Overall, as intimidating as the project was at the beginning this ended up being my most favorite and helped me to see what potential side projects I could do in the future.