Kraft, Riley
Final Project Design + Reflection
05 December 2017

Design & Test Plan

*changes to design

Theme: Underwater Treasure Hunt
Goal: recover the emerald and make it back to the surface
Step limit: diver's supply of air reduces with each action and cannot be replenished

Print description of current space
Main Menu - validate
- Go to linked space: up, down, left, or right
  - If linked space is NULL, print message and stay in current space
  - Else, move current linked space to user's selection and reprint menu
- Explore the current space – menu - validate
  - Pick up items – each space has its own item container
  - Space specific interactions – solve puzzle, find clue, battle enemy, etc.
- View Stats
  - Print diver's current supply of air, health points, and list of items in container
  - Prompt user to remove an item from diver's container
- Hints/Spoilers
  - Print space-specific list of hints, answers, and tips for interacting with the space

Main Deck
- (Start) with knife and flashlight in container
- Link to: Captain's Quarters, Gun Deck, and the Surface (ends the game)
- (End) with "boss" battle

Captain's Quarters
- Solve riddle, collect item, find clue
- Link to: Main Deck, Crew Quarters

Crew Quarters
- Find clue, find item, battle enemy – enemy knocks flashlight into an adjacent space
- Link to: Infirmary, Gun Deck, Captain's Quarters

Infirmary
- Endless supply of bandages to replenish health points – costs air each time, cannot carry bandages
- Link to: Crew Quarters

Gun Deck
- Find weapons, battle enemy – enemy knocks flashlight into an adjacent space
- Link to: Main Deck, Crew Quarters, Cargo Hold

Cargo Hold
- Find clue, find item, battle enemy – enemy knocks flashlight into an adjacent space
- Link to: Brig, Gun Deck

Brig
- Need crowbar and all clues to access emerald
- Find emerald and trigger trap; need any metal tool to escape

*Being in an interior space without having the flashlight in diver's container increases air consumption

Test Case	Input Values	Expected Outcome	Observed Outcome
Enter name of user	Anything	Print introduction to the game, description of the Main Deck, and a main menu	As expected
Main Menu (invalid inputs)	Negative number, float, string of alphanumeric characters, number less than 1, number greater than 7, 0	Error message Choose again	As expected
Main Menu (valid inputs)	1 - 7	1 – 4: print error message if linked space is null, or description of new space and another main menu 5: print new menu for space specific interactions 6: print air, health, and list of items in diver's container, then prompt user to remove and item from their container 7: print space specific hints, then main menu	As expected
Remove item from user container (invalid inputs)	Negative number, float, string of alphanumeric characters, number less than 1, number greater than 2, 0	Error message Choose again	As expected
Remove item from user container (valid inputs)	1, 2	1: Print list of items and prompt user to choose item 2: Reprint description of current space, and main menu	As expected
Space Specific Menu (invalid inputs)	Negative number, float, string of alphanumeric characters, number less than 1, number greater than up-to 4, 0	Error message Choose again	As expected
Space Specific Menu (valid inputs)	1 – up to 4	1: Print list of items in the space and prompt user to pick up an item, or print a message indicating no items in the space 2 – up to 4: Prompt user to choose riddle answer, or run simulation of specified action and/or make new items available	As expected
Pick up an item (invalid inputs)	Negative number, float, string of alphanumeric characters, number less than 1, number greater than 2, 0	Error message Choose again	As expected

Pick up an item (valid inputs)	1, 2	1: Print list of available items and prompt player to choose 2: Reprint description of current space and a main menu	As expected
Choose item to pick up or remove from diver container (invalid inputs)	Negative number, float, string of alphanumeric characters, number less than 1, number greater than printed list, 0	Error message Choose again	As expected
Choose item to pick up or remove from diver container (valid inputs)	Any integer in the printed list	Reprint description of current space and a main menu. Item has been transferred from diver's container to current space container, or vice versa	As expected
Memory leaks	valgrind	No memory leaks	As expected

Sequence to win and exit in the least amount of steps
Go left to Captain's Quarters -> Examine box -> answer: East -> Pick up pin -> Open desk to get clue -> Go down to Crew Quarters -> Read journal to get clue -> Go right to Gun Deck -> Reach into canon -> Pick up crowbar -> Go down to Cargo Hold -> Read manifest to get clue -> Go left to Brig -> Pry open floor boards -> Search your bag for something to open the door -> Pick up the emerald -> Go right to Cargo Hold -> Go up to Gun Deck -> Go up to Main Deck

Reflection

For the last course project I worked on I had a significantly hard time with memory leaks and errors in valgrind. This time around I made sure to test every step of new code with valgrind, and the method worked out very well for me. I had my memory allocation and deallocation under control from start to finish and did not lose any of my pointers.

With this assignment I mostly struggled with how complex to make the spaces. As in: Should there only be one action in the space? Does each space need to have an item to collect? What is meant by "structure" of the space? Ultimately I understood the requirements as meaning I need to have more menu options per space other than "pick up this item". Thus, I decided to use clues, items, puzzles, traps, and battles throughout my entire game environment. Some are choices, some are automatic. Each space has several interactions to choose from, but I did also include a "spoilers" option to guide the user in the right direction if they aren't understanding my design and purpose.

One of the major challenges in creating the spaces was my Brig space. I needed to devise a way to make only certain interactions available until the user finds all the clues in the game, then close the door to the room so the player can't get out without using a tool, then opening the room back up again for the user to move freely in an out of it again. I started off with a "solved"

variable for all spaces because each space has at least one interaction that changes based on the choices and actions the player takes, thus this worked well for making certain actions in the Brig available when I wanted them. Then I created an “open” variable to signify if the room was open or not. This feature ultimately just restricted the use of the main menu until the user opened the Brig door again. Then I needed another setting to essentially make the room harmless and without interaction, thus I had to create a third variable called “done” to signify that all major interactions were done in that space. This was a lot of work for only one of eight spaces, and given more time I would take the opportunity to find a more elegant way of solving this issue.

My biggest overall challenge in creating this project was working with the item containers. I didn’t want the player to just walk into a room and automatically acquire and item, then lose the item when they used it for a singular purpose. I wanted the player to be able to pick up and drop any items they wanted in any space they wanted, thus they could drop an item in one space and come back for it later if they decided they needed it. Therefore, both my Character class and my Space class needed to have item containers that could be accessed by the Game class. Originally, I designed the containers as arrays, but I continually had segfault issues when exchanging the item pointers between the character’s container and the current space’s container. Thus, I decided to give vectors a try, and it worked like magic! I spent nearly a full day trying to make this feature work and was relieved when I finally got all the kinks worked out. Plus, with the conditional statements I could create with vectors I was able to take the use of the items to a whole new level that I wasn’t even planning on. I’m very satisfied with the flow and flexibility of the game play I designed.