Jason Lim

CS 162

Project 5: Sushi Adventures Reflection

Design

1. Space Class
   1. Create 4 pointers for the final project specs for right, left, top and bottom
   2. Create an integer value for the room number to store which room we are in
   3. Create a sushitype enum to determine which sushi type is in the room
   4. Create constructor which takes room number as a parameter
   5. Set the pointer based on pointer to space, left and right
      1. Linear structure for sushi adventures game
   6. Go inside virtual function to grab sushi and put into storage
   7. Get left and get right function to get the pointer for left or right and determining which room we are in
   8. Find sushi to get sushi or find it in room
   9. Space deconstructor
2. Sushi struct
   1. Enums for Tuna, Salmon, and Unagi sushi rolls
   2. Create sushi type based on enums so that the sushi can have a value of Tuna, Salmon, or Unagi
   3. Have sushi constructor which takes value of sushi type and room number to put sushi there
3. Derived classes for Space class
   1. Tuna Room
      1. The tuna room will have a constructor for the room number or which room the tuna is in
      2. Go inside function which will take the sushi in the room when the mini-game is complete
      3. Mini game to match rock paper scissors with Sushi master
   2. Salmon Room
      1. The Salmon room will have a constructor for the room number or which room the salmon is in
      2. Go inside function which will take the sushi in the room when the mini-game is complete
      3. Random mini game to jump to cliff y/n to continue
   3. Unagi Room
      1. Final room
      2. Has a Boolean value for if all tuna, all salmon, all unagi received
      3. Boolean value for if win game as well
      4. The Unagi room will have a constructor for the room number or which room the unagi is in
      5. Go inside function which will take the sushi in the room when the mini-game is complete
      6. Mini game with sushi wizard multiplication game
4. Game class
   1. Int value for total number of steps the player has taken
   2. Sushido space for main character or pointer to space
   3. Creating storage vector which is a pointer to a list of sushis
   4. Boolean for if the game has been won or not
   5. Game constructor to create game and set all attributes
   6. playGame to actually play the game and go through all the rooms and steps
      1. Provides left or right movement capability and # steps cannot exceed 18
   7. Game destructor
5. Menu class
   1. A display menu function to display first menu
   2. chooseFromMenu function to choose the option on menu
   3. DisplayObjective function to display the Sushi Adventures objectives
6. inputValidation class
   1. Yes Or No input for choosing yes or no
   2. Integer input and integer input for sushi (3 options vs 2 options)
   3. Choose size to let user choose a integer value

Sushi Adventures Game Test Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Input by User | Expected Output | Actual Output |
| Input Validation | Validate input for characters Y/N and integer values as well as menu | Incorrect values give error message | Incorrect values give error messages |
| Test ASCII art | Test ascii art to make sure it shows up in sushi adventures title and the actual sushi | Sushi adventures title and sushi ascii art show up | Sushi adventures title and sushi ascii art show up |
| Rock Paper Scissors Tuna Room | Use rock (1) and try to get a match | Matches with sushi master’s rock and my rock, ability to proceed | Matches with sushi master’s rock and my rock, ability to proceed |
| Can say No to a room and proceed | Type N for skipping room | Skips room | Skips room |
| Movement left and right of rooms | Moving right of rooms moves to the next room, moving left of the room moves to previous room | Moving right and moving left different rooms | Moving right and moving left different rooms |
| Jump cliff Salmon Room | Wins 50% of the time based on ready to jump | Wins 50% of the time after jump | Wins 50% of the time after jump |
| Math game sushi wizard Unagi Room | Math game with random values from 1-10 multiplying by the same random value | Correct value allows to proceed to get unagi | Correct value allows to proceed to get unagi |
| >= 18 steps | Do 18 steps or more and forces game to end | Game lost | Game lost |
| Able to obtain sushi | Take sushi allows us to put sushi in bag | Takes sushi and puts in bag | Takes sushi and puts in bag |
| Taking same sushi not allowed | Not allowed to take same sushi | Error message, sushi already taken | Error message, sushi already taken |
| Taking more than 3 sushi in storage | Not allowed to have more than 3 sushi in storage at same time | Error message, bag full | Error message, bag full |
| Win Game | Win game by collecting all tuna then putting on plate, collect all salmon put on plate, then unagi and put on plate | Wins game and gives story ending | Wins game and gives story ending |
| If not have the correct number of sushis | Putting sushi on plate but not all same | Message to go back and gather all same sushi | Message to go back and gather all same sushi |
| Valgrind test | Valgrind test through program (win game) | No mem leak |  |
| Valgrind test | Valgrind test through program (lose game) | No mem leak |  |

Reflection Sushi Adventures