Ken Hall
CS 162 400
3/22/18

Design and Reflection

DESIGN DESCRIPTION

Exit: a child of the Space class. The player may enter this space at any time, but the effect (ending the game with victory) only triggers if the player brings the correct number of items when they enter the Exit. Exit has a string member variable "name" (assigned in the constructor as "EXIT"), a mutator function for the four inherited Space pointers, a getter function for name, and a Space*-returning function moveTo, which returns the pointer to the connected space when passed an appropriate integer argument (1 returns the Space* top, 2 returns the Space* right, and continues like so clockwise). winTheGame checks to see if the player has deposited enough items at the exit to win. depositItem adds a tick to the itemsDeposited counter, which is used in winTheGame. All of the children of spaces are similar, but different in important ways. For subsequent children, a brief description will be provided as well as a list of member variables/methods as they are different from Exit.

Room: a child of the Space class. Rooms contain an item, and a question, and require the player to supply the correct answer to the question before surrendering the item to the player.
member variables:

- bool hasItem: keeps track whether or not the player has taken the item from the room
- string name: the name of the room (assigned by constructor)
- int answer: the correct option from menu for the user to select for the right answer
- string question: the question posed to the player to answer
- string itemName: the name of the item the player can obtain from answering correctly

methods:

- getters and setters for member variables;
- bool guessAnswer provides the interactive element. The user's guess is passed to the object and the object returns whether or not the guess was correct
- moveTo...same behavior as Exit

Start: a child of the Space class. Start contains a backpack which the player must pick up in order to advance. The backpack is used to collect items needed to complete the game. Start's interactive component is that it will not allow the player to leave until the backpack has been collected and the rest of the game can continue.
member variables:

- string name: the name of the area (constructor assigns "START")
- bool pickedUpBackpack: tracks whether or not the backpack has been picked up

methods:

getters and setters
moveTo...see Exit for behavior of this function
bool backpackTaken: a method of particular note...a getter for pickedUpBackpack, which dictates whether the rest of the game can progress.

Space: Parent of Exit, Room, and Start. Space is an abstract class. Space contains four pointers to Space, which are used to link Space objects to other Space objects, as in a 2D plane. All of Space's functions are either virtual or pure virtual

Gameplay rules
Player starts with 10 health, 25 moves. If either reaches 0, the player loses the game. Special exception for the player winning on with the 25th step. Player is given "bonus step" if in the process of winning on the 25th step.

Attempting to move out of bounds, where there is no indicated space, will cause the player to lose one health. Answering a question incorrectly will lose the player varying amounts of points (more details below).

Every action costs the player 1 step.

The player must pick up the backpack in the start area in order to leave the start area.

Once out of the start area, the player navigates through the Spaces using a menu. The menu also allows the player to interact with the Space they are in, or to check the contents of their backpack (once the backpack has been obtained).

Most spaces will prompt the player with a question related to one of the assignments in CS 162. The player can attempt to answer the question by choosing the interact option from the menu. Doing so will present the player with four options to choose from, one of which is the correct answer. If the player chooses the correct answer, an item is added to their backpack. If the player chooses incorrectly, the player will lose between 1-10 health (predetermined).

After collecting an item or two, the player may go to the exit area to begin depositing items. Interacting with the Exit space will cause the player to remove an item from their backpack and place it in the exit area. Once two items have been placed in the exit area, the game is won.
TEST PLAN

Condition Observed	Expected Outcome	Actual Outcome	Notes
'1' entered at main menu	game begins	game begins	checked for both initial main menu and play again main menu

‘2’ entered at main menu	game ends	game ends	checked for both initial main menu and play again main menu
player attempts to walk “out of bounds”	health -= 1	health -=1	
player takes any action	time left -= 1	time left -= 1	
‘5’ pressed at starting area before backpack has been picked up	player picks up backpack	player picks up backpack	
player attempts to leave start before picking up backpack	player remains in starting area	player remains in starting area	
player attempts to check contents of backpack before having backpack	nothing happens	nothing happens	
player checks contents of empty backpack	nothing happens	nothing happens	
player checks contents of backpack with one or more items inside	contents of backpack are listed	contents of backpack are listed	
player selects a valid move option	player is moved to the room they indicated	player is moved to the room they indicated	
player answers a question incorrectly	player suffers “bad consequence”	player suffers “bad consequence”	Checked for each of 5 Rooms
player answers a question correctly with sufficient space in backpack	player receives item	player receives item	Checked for each of 5 Rooms
player attempts to answer a question they already solved	player suffers “bad consequence”	player suffers “bad consequence”	Checked for each of 5 Rooms

player solves a question after already solving two	player is killed, ending the game	player is killed, ending the game	Checked for 3 different combinations of Rooms
player enters the exit with 0 items	nothing happens	nothing happens	
player enters the exit with 1 item	nothing happens	nothing happens	Checked for each of 5 items
player enters the exit with 2 items	game ends with victory	game ends with victory	checked for 3 different combinations of 2 items

PROBLEMS/SOLUTIONS

In my first design, the player was allowed to leave the starting area without the backpack in spite of being warned that they needed it to complete the game. Weird things would happen if the player is allowed to leave the starting area without the backpack and answer questions correctly. I fixed it by forcing the player to take the backpack before leaving the starting area.