Name: Daniel Maynard
Assignment: Final Project Design and Reflection
Date: 8/15/17

Reflection and Design

Working on this project was probably the most difficult since the beginning of this course. Designing this program I realized the two base classes were going to the Spaces and Tiles classes. The Space class would basically have the up, down, right and left pointers which will point to the corresponding spaces. From the base class I created derived classes, each with it's own array, items and walls. For the Tiles, I created a derived classes of player and tilesWithItems. The player derived class represent the player, which has a tile type of @, which the derived class tilesWithItem contains an item, and has a tile type of 'o'. The space will be created with tiles, and will start off with NULL tiles, which will allow the player to move across the board.

Some of the challenges I had in creating this program was detecting an item at all sides, I was having difficulty detecting it on all sides so I had to resort to detecting it, if it was north of the player. Another difficulty I had was with the bag creation, which was not deallocating correctly through its constructor. Eventually I had to delete the bag through the game class, which seems to free up memory correctly.

Originally I had a lot of ideas for the game, for example I wanted to make a Harry Potter game that would make the user fight various monsters across the spaces and each monster would drop a horcrux. After collecting all the horcruxes Harry would have to fight Voldemort, and if the player did not have all the horcruxes then there would be no way Harry could win. After realizing I just needed to do the basics of this game before making it more interesting I realized the time constraint on this project wouldn't allow me to create the game how intended. So I resorted to following the instructor's advice in just doing the basics for the game, and if time allowed to make it more interesting.

Test case #1
Input Validation Functions

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
input too low	input < 1	inputValidation	Your input is invalid. Please enter an integer	Looped back to allow the user to enter integers between 1-7

			from 1 to 7.	
input not a number	input %#	inputValidation	Your input is invalid. Please enter an integer from 1 to 7.	Looped back to allow the user to enter integers between 1-7
Input is a string	1234	inputValidation	Your input is invalid. Please enter an integer from 1 to 7.	Looped back to allow the user to enter integers between 1-7
Input too high	Input > 7	inputValidation	Your input is invalid. Please enter an integer from 1 to 7.	Looped back to allow the user to enter integers between 1-7

Test case #2
Move Player Functions

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Moves into a NULL array		movePlayer	Player moves into NULL location	Player moves into NULL location
Player moves into a wall ‘|’		movePlayer	“The way is blocked”.	Game lets user know the door is locked
Player moves		movePlayer	Game lets	Game lets user

into a locked door			user know the door is locked	know the door is locked
Player moves into unlocked door		movePlayer	Player moves through to next space	Player moves through to next space

Test case #3
WinCheck Function

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
If player does not collect all items	5	WinCheck	Player unable to move through the final door	Player unable to move through the final door
Player does collect all 5 items	5	WinCheck	Player able to move through the final door	Player able to move through the final door

Test case #4
Move Player Functions (moving across an item)

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Player moves north, onto an item ‘o’	1	movePlayer	“The way is blocked”.	“The way is blocked”.

Player moves south, onto an item ‘o’	2	movePlayer	“The way is blocked”.	“The way is blocked”.
Player moves east , onto an item ‘o’	4	movePlayer	“The way is blocked”.	“The way is blocked”.
Player moves west, onto an item ‘o’	3	movePlayer	“The way is blocked”.	“The way is blocked”.

Test case #5
spellCheck - Verifies user casted spell

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Player casts unlock door spell	7	spellCheck	Player is able to move to the next door	Player is able to move to the next door
Player does not cast unlock door spell	7	spellCheck	Outputs to user the door is locked.	Outputs to user the door is locked.