Robert Elsom

1/27/2019

Lab 3 Reflection

My original design was mostly based on the instructions, most of the actual game is done in a Game class, both Die and LoadedDie classes are just for creating objects with certain attributes, so they both had getter and setter functions. I also decided to add the round loop into the game loop instead of in my main file so that I could do everything with a simple object, and there was less chance of breaking the classes if using in another program.

During implementation, I decided to make function calls from other functions. For instance, once the initial menu is displayed, it automatically goes into user input so that I did not have to call every function from the main class. I am not sure if this is a good practice to be honest, but at my current knowledge level I did not see any reason not to, it made my class more modular where I was not trying to cram everything into a function in order to not have to call each step.

Surprisingly, I did not run into any major problems during this program. Everything went according to plan and there were not in issues besides minor errors like missing a closing bracket. This was definitely a first for me and put me in a really good mood until writing program 2, were I spent a whole day fixing errors.

Test Cases:

Starting Menu test cases

Test case	Input Values	Driver Function	Expected Outcome	Observed Outcome
Function a not an unsiged integer	A, 1.5, -4	ValidStr()	Display error, repeat options to user	Display error, repeat options to user
Integer not a valid option	5	ValidStr()	Display error, repeat options to user	Display error, repeat options to user
Integer a valid option	1	Menu() If choice == 1 || choice ==2	Print starting prompts and continue to user input	Print prompts and get inputs for functions
Integer second option	2	Exit()	Quit program	Quit program

Number of sides for input

Function a not an unsiged integer	A, 1.5, -4	ValidInt()	Display error, repeat options to user	Display error, repeat options to user
Integer not in range (<2 or >400)	-5, 0, 500, 1000	ValidInt()	Display error, repeat options to user	Display error, repeat options to user
Integer in valid range (1 = 400)	1, 244, 15	userInput()	Store in variable and in response array	Store in variable and in response array
Input at extreme low	1	userInput()	Store in variable and in response array	Store in variable and in response array
Input at extreme high	400	userInput()	Store in variable and in response array	Store in variable and in response array

Picking Type of Die input test

Function a not a menu choice	A, 1.5, -4	ValidStr()	Display error, repeat options to user	Display error, repeat options to user
Input is low menu choice	1	setDieType()	Creates normal die	Creates normal die
Input is second menu choice	2	setDieType()	Creates loaded die	Creates loaded die