Christian Roccanova

CS162 Lab 3

4/23/2017

Reflections

Originally, my program was designed such that the menu function would be part of the main file and ask the user to choose to play or not. It would have 3 classes, Game, Die and LoadedDie. LoadedDie would inherit from Die, but would alter the way it rolled dice by rolling 3 times and only returning the highest result. The main function would take the sides and types of the user's dice, which would then be used to create Die or LoadedDie objects by calling a Dice function or LoadedDice function, respectively. The play function of the Game class would then use these objects to call the roll function from the appropriate class (Die or LoadedDie) and compare the returns to determine a victor for that round. These results would then be placed into an array to be used to print the results in the Game class' results function.

Several organizational changes were made to more accurately reflect the lab instructions. The Menu function was moved to be inside of the Game class. Input validation was added to the program as its own class, its functions being called within the menu function. The originally planned Dice function was broken up into several parts setSides, setType, getSides and getType to better facilitate the creation of both loaded and normal dice. The planned LoadedRoll function was changed to be a virtual function overwriting the roll function of the Die class following the reading on polymorphism. For this to work, objects were created for both regular and normal dice and then assigned to pointer objects to represent the final dice based on whether the user chose to make them loaded or not. One issue that I ran into during testing was that my random number generator was outputting the same value every time it was run. This turned out to be because the srand() function was inside of a loop and was fixed after the srand() function was removed.

Testing

Input	Expected output	Output	Changes Made
Any dice, 0 rounds	Error, asks user to try again	Asks user to try again	n/a
Any dice, negative rounds	Error, asks user to try again	Asks user to try again	n/a
0 sides, any rounds	Error, asks user to try again	Asks user to try again	n/a
Negative sides, any rounds	Error, asks user to try again	Asks user to try again	n/a
2 1-sided die, any rounds, any load	draw	draw	n/a
2 6 sided die, any rounds, no load	50/50 distribution of wins	Initially the random number generator would only output a single value.	The srand() function was moved outside of the dice rolling loop. This fixed the issue.

		After being fixed, there was roughly a 50/50 distribution of wins	
2 6 sided die with 1 loaded, any rounds	Distribution of wins should be skewed towards loaded die	Wins were skewed towards the loaded die.	n/a
1 4 sided die, 1 20 sided die, any rounds, no load	Distribution of wins should be skewed towards the 20 sided die	Wins were skewed towards the larger die.	n/a
1 4 sided die, 1 20 sided die, any rounds, either die loaded	Distribution of wins should be skewed towards the 20 sided die	Wins were skewed towards the larger die.	n/a
1 4 sided die, 1 5 sided die, any rounds, no load	Distribution of wins should be skewed towards the 5 sided die	Wins were skewed towards the larger die, but the difference was significantly smaller.	n/a
1 loaded 4 sided die, 1 5 sided die, any rounds	Distribution of wins should be skewed towards the loaded die	Wins were skewed towards the loaded die as on average its rolls were higher despite having fewer sides.	n/a