Program name : Lab3_Design_Reflection_Schaefer_Kristin.pdf
Author : Kristin Schaefer
Date : 04-21-2019

Design Reflection for Lab3

Design

In terms of design, the most challenging aspect was thinking about where to declare objects, and when and where to use pointers. It was a difficult process to sketch out the order of the program, and particularly to determine how to handle the die objects. Ultimately, I declared the die objects in the main function, and then passed the addresses of the objects to die pointers in the Game class as anticipated in my design plan. The tricky part was getting all of the syntax correct! Compared to the design plan, I had many more getter and setter functions to help pass information to different functions. It was helpful to assign an integer variable to each die to have it’s type available to determine if a pointer to a dynamically created object needed to be declared and deleted.

Testing

Recycling my input validation from Project 1 proved challenging. After receiving the results of the grading from Project 1 I realized that my testing plan was not robust enough. Specifically, the input validation failed for floats and for input with spaces. I attempted to reconfigure my input validation, using mainly cin.good() to handle floats and spaces, to no success. I tested out several other methods based on suggestions from Piazza and Stack Overflow. The final solution I found to work was to treat the input as a string and then test the string for non-digit input. After testing the string, I then cast it to an integer and checked the range to correspond with the variable’s requirements. For the final round of my input testing, it seemed that all validity checks were working as planned!

In terms of testing the loadedDie average, I had to modify my original roll die function to give a higher value to the loaded die. It took me awhile to realize that the problem was within the roll die function and not in the allocation of pointer!

Improvement

Moving forward, I’m happy that I have a better understanding of how to deal with input validation for integers. I think for the next program I will spend much less time on input validation! In terms of handling dynamic memory allocation and pointers, I can always improve. I think my main challenge is speed. Often, I struggle with syntax and have to look things up. Hopefully by the end of the term I will be a bit faster!