

Tristan Santiago
January 28, 2018
CS162-400
Lab 3 Reflection

I worked on the original design and testing plan for this program with a classmate. In its infancy, it was rather rudimentary and unsophisticated. It simply consisted of main prompting the user to enter all of the necessary information needed to create the die object and play the game (number of sides, type, rounds, etc.). This information would, of course, be later implemented into the die class. It was my thought that I would then use this information to construct a die object that would later be sent to a function that would randomly select a number in range and simply return the result. The same would be done in the event of the user wanting to use a loaded die, only the result would be biased somehow.

I did my best to keep a narrow focus early on as to not overwhelm myself. I wanted to start with one small part of the program, get that working, and then add another small part to it, again making sure it worked. It was my plan to continue constructing smaller, working parts, before combining them with each other until eventually I had a functional, albeit, much larger program. This was my way of debugging while coding, as I would be able to easily identify and isolate bugs.

I knew the most difficult part of the program would be determining how to create a loaded die function that would return biased results. I considered many options early on, such as using a multiplier, or increasing the chances to roll higher with each roll, decreasing the likelihood of rolling lower possibilities, but eventually settled on producing a biased result every other roll, as this felt like a much more consistent way to produce a biased roll. In an earlier design of the program, I actually had a multiplier for the loaded die that applied to the higher end of the die. It was functional, but as I dug deeper into the development phase, I became more and more concerned with chance interfering with the average output. I thought perhaps I was leaving too much to chance and really wanted a better way to fix the results, which led me to fixing the results of every other roll, thereby definitely increasing the average output of the loaded die.

Much to my surprise, creating a loaded die function didn't prove to be as difficult as I originally imagined. I instead had more trouble with properly calling functions using objects and pointers. In the end, I think the logic I was using in my program was sound, I was just struggling with finding the proper syntax. After doing some additional reading from the assigned chapter and the additional resources provided, I was eventually able to figure out the solution. After properly calling the user-defined objects, completing the rest of the program was rather straightforward.

Overall, I found this program challenging, but not as challenging as last week's assignment. Inheritance was not a difficult concept to understand or implement, and I am pleased with how my program turned out when compared to my original design.