Lab 3 Reflection

In my original design I had separate files for each class each with its own header file. I found this didn't work when using a derived class, that they needed to be in the same file. I also found using header files unnecessary for this assignment and felt like they made it harder to navigate my code since I had so many files. I finally decided to just combine all three classes into one file which made it the easiest on me with sharing and passing variables and it also was easier to read it all on just one page instead of flipping through different files.

During the implementation of the game I kept changing how I loaded the die. I had a lot of trouble seeing the results I wanted to see. I thought the loaded die I would win a lot more than it currently does. I also changed the amount of files I was using. I found less files was much easier to manage.

During implementation I had trouble creating a dynamically allocated two-dimensional array. I just couldn't get the non-static variable to compile right so I decided to just create two dynamically allocated one-dimensional arrays and this made the assignment much easier for me to manage. I also had trouble creating a child class and getting it to properly work since it was my first time creating one. It took me a while to realize the child class had to be in the same file as the parent class.

Test Case	Input Values	Driver Functions	Expected Outcomes	Observed Outcomes
Loaded Die vs Unloaded Die	Loaded Die = 50 sides Unloaded Die = 50 sides	getRoll()	Loaded Die wins	Loaded Die wins
Loaded Die vs Unloaded Die	Loaded Die = 40 sides Unloaded Die = 50 sides	getRoll()	Loaded Die barely wins or tie	Unloaded Die wins
Loaded Die vs Unloaded Die	Loaded Die = 50 sides Unloaded Die = 40 sides	getRoll()	Loaded Die wins by a lot	Loaded die won by a lot
Loaded Die vs Unloaded Die	Loaded Die = 25 sides Unloaded Die = 50 sides	getRoll()	Unloaded die wins	Unloaded die won