

Assignment 3 Design

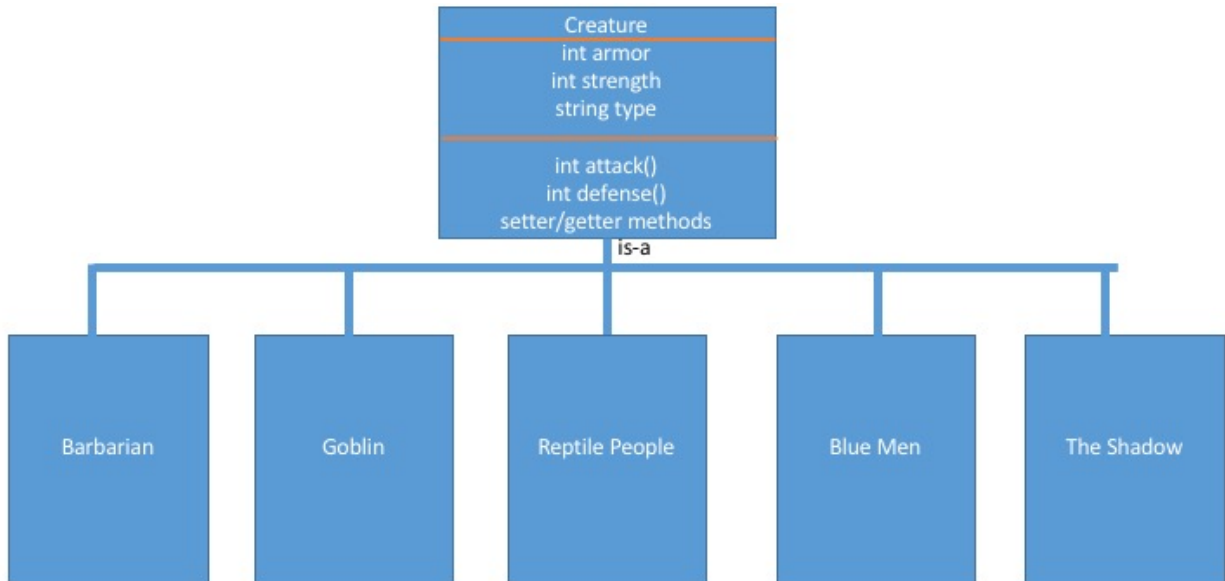
I am thinking that I will design this program into 4 separate parts. The main will contain the main menu for the fighting. This is where I will prompt the users for what type of characters they would like to use and I will then declare them. The second part will be the Die class. I am planning on using my design that I had made several labs ago. This will require little to no programming. The third part will be the Creature class. Inside of this class I will declare all of the different subclasses along with the parent Creature class. The final part of this project will be my battle class. Inside of this class I will write all of the programming for having two different creatures fight.

Will be creating a hierarchy for several different characters (Goblin, Barbarian, Reptile People, Blue Men, and The Shadows). Each character will have an attack, defense, armor, and strength points. These classes will be created from a parent create class, which will be abstract and will never be instantiated. Only one constructor is needed for each class since every new create will start out with the same stats (depending on the creature). The only value that can change is strength points.

The abstract class will have 4 protected variables: attack, defense, armor, and strength points. Besides these variables there will be 2 virtual functions attack and defense. The attack function will be different for each subclass and will several different die, based on the requirements outlined in the program description. The defense function will work basically the same way with dies rolling to give the total amount of defense. Both of these functions will return an int value that will be used in another class.

Once the base class is created I'll focus on making the Barbarian class first and making sure that I can get things to work correctly. This will involve testing the setter and getter methods to make sure that the data is being handled correctly.

Below you will find the class hierarchy that I made when designing this project:



I think after designing the Barbarian class I will make a Battle class, which will handle the fighting between two characters. This will allow me to get the battle oriented code in a closed environment. I am thinking that a do/while loop will be a good way to continue through the battle until one of the players win. Inside of this class I will also have checks in place to successfully process the specials that a few of the characters have.

I plan on making each class without any special features to begin with. Since they are all very similar in that regard I should be able to create them fairly quickly once the Barbarian class is complete. Once I have all 5 subclasses working correctly I will work on the specials for the Goblin and Shadow classes.

I will need some sort of check in place to see if a players Achilles has been cut by the Goblin special. I think the best way to do this would be to have a check inside the battle class but I may change it depending on how things work. The same goes for the Shadow special, as of now I will plan on placing this inside the battle class but if things don't

Design Changes:

Shortly after starting the design of the battle class I decided to add an alive Boolean to each class. This was to quickly check at the end of the do/while loop to see if both characters were still alive. I could have simply checked the strength number but I decided that I wanted to handle that elsewhere.

Another change that I made was adding a cutAchilles Boolean in the parent class. Each class uses it, except the goblin, to check and see if the Goblin cut the Achilles and if the amount of

Rob Navarro
CS162

attack points needs to be cut in half. For both of these changes I ended up adding setter/getter methods for each. This made it fairly simple to do comparisons inside the battle class.

Inside of my main I ended up creating a main that prompts the user for which creature they would like. I then place the object pointer inside of an array that will hold two object pointers for whatever creatures the users want. I made the main menu so that if you do not enter a valid selection you are re-prompted to enter a value.

I also came to a conclusion on how best to handle the special powers for a few of the creatures. For the Goblin, I used an if statement that checked if the total for the attack roll was 12 and if the object in question was labeled as a Goblin. If so, I proceeded into a nested if statement that made sure the player being attacked wasn't a Goblin and didn't already have a torn Achilles. If this check passed as well I printed a prompt to the screen and changed the cutAchilles parameter to true. Then, inside of the attack functions for each class I had an if statement that checked if the Achilles had cut. If it had I made sure to divide the return value by 2. I had written this out on some scratch paper so coding this into C++ was pretty straight forward.

For the Shadow special I ended up using a special return value within the Shadow attack function. I used a random number generator that would spit out a 1 or 2 and if the value was 2 I would return a value of -1. Then, inside of the Battle class I would check for the -1 return value. If it was found I would then set the attack for that turn to 0, so that no damage would occur.

For the most part I feel that the changes I listed above could have been better thought out in the beginning. The trouble that I ran into with this project was that after I got the Barbarian class created I dove right into the rest of it. The next time around I need to focus on the big picture just a bit more to try and get these finer details sorted out before I start programming.