Jared Parkinson

Assignment 3 Reflection – CS 162

**Requirements:** The requirements of this assignment I understand as the following items:

Overall: Create a fight simulation program where there are 5 unique derived class fighters and a pure virtual base class of Creature. Each fighter except Barbarian has a Special that will modify Attack or Defend returns without using additional functions. In addition, each derived Creature can attack another copy of themselves. The fight will continue until one of the creatures is dead. Attack and Defend in Creature will be pure virtual. A Creature object will never be created or used, only derived Creature Classes. Fight 1 whole battle until death and end your program.

**Class Design:**

Creature: Creature has the default data members of strength, armor, and isDead bool. Everything else is either virtual or pure virtual. Creature contains the mersenne twister randomizer to generate random numbers. Additional Functions other than attack and defend are getStrength, setIsDead, getIsDead.

Barbarian: Barbarian is a generic cookie cutter class with nothing special and no additional data members or functions.

Baba Yaga: Baba Yaga just takes their attack and 1/3 of it goes to strength per Terry’s email to us. It does not take into account defense or armor as stated. I have created a printout for the user when it occurs so they can see it happen. No additional data members or functions.

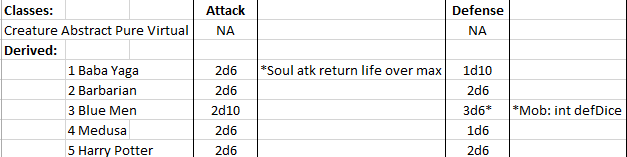
Blue Men: Blue Men have their own defense dice (int defDice) data member. This will be subtracted from when they use \*Mob after taking damage greater than 4. I have created a printout for Mob when it happens and have included the number of dice after it was applied. No additional data members or functions.

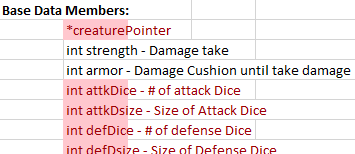
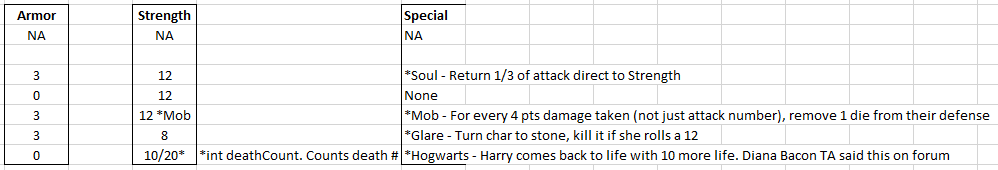
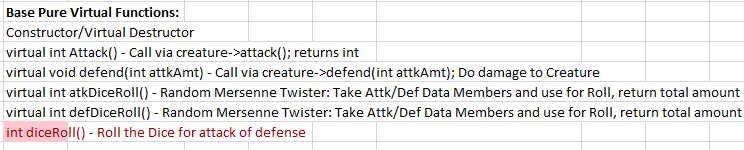
Medusa: As of right now, Medusa just does a bunch of damage (99) on a pure 12 dice roll to kill any Creature she is attacking. She has no additional Data Members or Functions. Medusa will tell you when she uses Glare with a printout and it will say roll of 12. To make this happen, generally fight Medusa vs Medusa a few times and get lucky. No additional data members or functions.

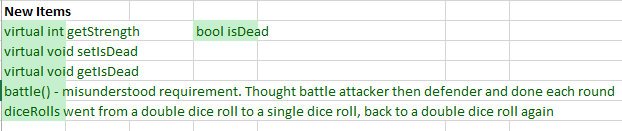
Harry Potter: The pdf states something to the effect of Harry having 20 life after death. However, Diana Bacon stated on the forum that Harry will die, recover and gain a full 10 strength again. I implemented a deathCount int for Harry that would tick over up to 1. When this is at 0, he gets an additional death per his unique defense function. This is how I chose to implement Hogwarts. No additional functions for Harry.

Menu: This was a total Menu system that would loop based one death originally and let you keep playing. For now, I dropped it back down to only running once. It has User Input validation and you must choose Menu items or it will complain to the User. Menu does all of the work with the objects based on User input. In the real world, you would probably have an additional Class do this. We already had enough on this project haha!

Below is a portion of my excel design sheet. Everything marked in Red is modified or taken out. Everything marked in Green is added:





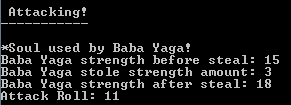
**Test Plan:**  Here is my test plan in table form:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Runs** | **Action** | **Expected** | **Pass**/**Fail** |
| Menu System | Press 1 - Begin | Go to Game Menu, display game menu | Initial: Pass! (previous saved code) |
|  | Press 2 - Quit | Quit the program | Initial: Pass! (previous saved code) |
| Fighter 1: 1st Attacker Screen | Press 1-5 and get desired Creature | When 1-5 is pressed, user will see Derived Class statement pertaining to that class Name | Init: FAIL: I had some of the names copied over from other Classes, this said a Barbarian was a Baba Yaga  2nd: Pass: Fixed Naming |
| Fighter 2: 1st Defender Screen | Press 1-5 and get desired Creature | When 1-5 is pressed, user will see Derived Class statement pertaining to that class Name | Init: FAIL: I had some of the names copied over from other Classes, this said a Barbarian was a Baba Yaga  2nd: Pass: Fixed Naming |
| Attack Dice | Baba Yaga | Roll greater than 6, max 12 (2d6) | Init: Pass! |
|  | Barbarian | Roll greater than 6, max 12 (2d6) | Init: Pass! |
|  | Blue Men | Roll greater than 10, max 20 (2d10) | Init: Pass! |
|  | Medusa | Roll greater than 6, max 12 (2d6) | Init: Pass! |
|  | Harry Potter | Roll greater than 6, max 12 (2d6) | Init: Pass! |
| Defend Dice | Baba Yaga | Roll less than 11 (1d10) | Init: Pass! |
|  | Barbarian | Roll greater than 6, max 12 (2d6) | Init: Pass! |
|  | Blue Men | Roll greater than 12, max 18 (3d6) | Init: FAIL! Was a copy from another class missed change  2nd: Pass |
|  | Medusa | Roll less than 6 (1d6) | Init: Pass! |
|  | Harry Potter | Roll greater than 6, max 12 (2d6) | Init: FAIL Potter was setup for 1d10 from a copy paste Class  2nd: Pass – Fixed Dice roll |
| Specials | \*\*Please see testing for Specials below (easier screenshots) |  |  |

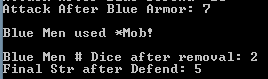
**Specials Testing**

**Barbarian: N/A**

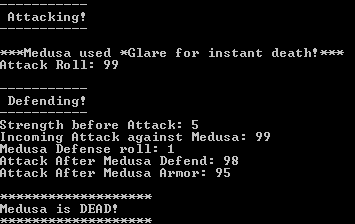
**Baba Yaga:**

****

**Blue Men:** \*Mob



**Medusa:** \*Glare took a bunch of Medusa vs Medusa runs but got it!

****

**Harry:** \*Hogwarts \*Per Diana Bacon Forum post, Harry Potter works like the above.



**Reflections:**

Oh boy where do I start? I spent massive amounts of time on this project due to some confusion. Let’s go through some things below!

Dynamic Issues: I wrote the entire Assignment and had it finished on Wednesday. After working on Mod E for a bit I went and checked the forum and found that Terry had posted and intended that we use dynamic creatures. This is not mentioned in the pdf. I had initially created the entire project using objects already created in my main. It worked great. So I went through and rewrote it all to fit with the way Terry had intended.

Design Issues: I initially thought that the attacker and defender were to be chosen each time. I didn’t realize that it was a fight to the finish where each would battle until one died. I thought each singular round was to be chosen 1 at a time with attacker and defender. I ended up creating a battle() function to deal with this where the attacker and defender will swap each run until death happens.

I spent a lot of time on a nice Menu system that would loop through and check for death and make sure you could not choose someone who had died. The battles would continue until 1 Champion remained. I read on the forum that it was supposed to be just a singular battle and have the program end. I kept a copy of what I had written saved locally and modified the program to remove all of my checkers for death and repeating menus to keep it the way it was requested. Yes this may have been a bit extreme on my part, but I thought I could see where the future was going with the project so I built it!

Deviating from my original design, I added 4 new functions to the Creatures, 1 new data member and messed around with different dice roll functions. I chose to go with two dice roll functions. One for defense and one for attack. This made it easy to differentiate between dice number and type for each derived class. The Menu system became the parent of all the working data. Menu has a choice int and a finishRun bool to mark the finish of the program to help with my Menus. attackCheck and defendCheck were used as functions to check for death originally, now they just call the next menu.

User Input issues: I spent a lot of time messing with those recurring menus before I realized none of that mattered. I had to fix a lot of issues with Death showing and recurring wrong menus. Now the only thing it does it run once and quits. It will check to make sure you enter a digit on the menu and give you another try.

Segmentation Fault: I had a segmentation fault on flip. This was because I failed to initialize pointers I was using to NULL. It took me a while but I used gdb and found what the problem was eventually. It was definitely a learning experience. Setting those in the Constructor resolved that issue.

Ultimately: It’s interesting. I started early this time and got a lot done, only to have to go back and rewrite a lot of it. I think that for the next assignment I am literally going to write down every single question I can think of in the first two days of the week that need clarification and post all of them in one fell swoop to the forum. Even though I spent a lot of time on this assignment, it was my favorite of the class by far. I really enjoyed this assignment and had a lot of fun on it.