Minh Nguyen

Assignment 3 Design paper

CS 162

Design

The first step for this program will be for me to create the parent class that holds the four races; Goblin, Barbarian, Reptile People, and Blue Men. The races are going to be manipulated using accessors and mutators that have corresponding variable names for each race. Such as under the goblin class, it will have GoblinStrength, GoblinArmor, GoblinDefense.. etc. Every race will its' own attributes according to the table is given in the instructions. Since only one mutator function is needed, I can set all the necessary variables for each in this function.

After creating the shells and appropriate variables, The next step is to work on the functions. The purpose of the game is to pit 2 races against each other and see who's strength points hits 0 first. With this in mind, The strength point variable is going to be the only one that changes. To start the game off, I will allow user input when prompted to see what two races will be fighting each other. After, user inputs in the race choices, then the game function begins. To simulate a turn-based system, the program will run in a while loop that will switch turns for the monsters at the end of each iteration. Something along the lines of changing the value from player = 1, to player = 2. So when player =1, it shall have the attributes of the goblin race, and player = 2 will have the attributes of the barbarian race.

When it is a monster's turn to attack, The attack will be simulated using a random number generator to decide what numbers are rolled and added together. This number will be stored to a variable. Then the attacker shall roll their appropriate defensive attributes and this number will also be stored to a variable. Then another variable called preArmorDamage will be the difference between the two attacks. This preArmorDamage will then be subtracted from the appropriate armor type of the race that is fighting into another variable which will be called realDamage. This realDamage variable will then be passed back into the mutator function for that class and subtract this number from the strength point variable that was pre determined when creating the sub classes. The strength point will be saved to that class at the end of the iteration, and it will be player = 2 's turn.

To determine when the fight ends, there will always be a conditional statement in the while loop that determines if the strength point attribute of the person being attacked is less than or equal to 0, then the game ends and the player that was attacking will be the victor.

The variable in the classes that will be constant is the armor value. Since this value is never suppose to change, it will be left as a const to allow me to catch any bugs with the math. At this point in my design, I'm not entirely sure how to save the strength points after each round.   But I'm guessing that saving a value for each iteration should not be too difficult for me to figure out.

I will create several of these fight functions depending on the matchup that was chosen. Since there are only 4 classes, there can be a total of 6 different match ups. Each fight function will be modified depending on what 2 creatures are fighting since each creature has a different attack roll dice, and defense.. etc.