Derek Yang

Document (Design + Reflection)

Program Design

Problem: Create a fantasy combat game where there are separate characters each with their own individual special characteristics. These characters will inherit from the parent characters class and use polymorphism for their own special attacks. The following are necessary conditions of the game:

- Create the following characters: vampire, barbarian, blue men, medusa, and harry potter.
- Each will have it's own attack and defense rolls. The output depends on the number of sides to the die and the number of rolls.
- Each character will have it's own specific armor and strength points value. Strength points is like HP in regular games.
- Each character will have it's own characteristics or special skill. This skill will be created using polymorphism of existing or new classes. Usually this will mean that the parent class will have an abstract class.
- There will be a character base class
- The menu should display the attacker and defender's information
- Notes:
    - If Medusa uses "glare" on Harry Potter on his first life, then Harry Potter comes back to life after using "hogwarts".
    - If the Vampire's "charm" ability activates when Medusa uses "glare", the Vampire's charm trumps Medusa's glare.
    - The sample characters are unbalanced intentionally. This will help you in debugging your program! Some will win a lot, while others won't.

The program will loop and keep asking the user which characters to choose to fight against each other. Combat ends when a character dies. Each round consists of two attacks where the other character must defend. The damage is calculated as such: Damage = attacker's roll – defender's roll – defender's armor.

Character class – parent
Variables: attack, roll sides, roll turns, defense, strength

- Attack function
- Defense function
- Special ability abstract class
- Gets and sets for the variables

Vampire class – child

- This class will have the special ability charm
- Charm will activate during the defense phase and reduce incoming damage to 0

- This gives the effect of the skill activating

Barbarian – child

- This class will just inherit from the character class.
- Not much is changed in this class, it can be counted as our control class

Blue Men - child

- This class will activate mob during the defense phase
- Mob must take the factors of 4 in order to work correctly.
    - You can not use the incoming attack and divide it by 4 because the overall strength will not be used. You must take the overall strength and reduce the number of dice from there.

Medusa – child

- Medusa has a special attack called glare
- Glare activates during the attack phase
- The only extra consideration for glare is that it will activate Hogwarts and harry potter will survive if it is activated the first time

Harry Potter – child

- Harry potter has a special ability called Hogwarts where the user's strength points is set to 20 if the user is supposed to die
- This correlates with medusa's glare ability and will only activate once.
- Use a bool tracker for this special ability
- Activate during the defense phase of the character's turn

Game Class

- This will play the game
- Loop through each round and ask the user to see which character to pit against
- The round will end when a character dies
- Each round, the game class will take the attack and defense options of the characters and feed them into each other. The receiving end will be the defense character.
- The game class will keep track of the number of rounds.

Dice Class

- This class will implement a dice which will take in the number of sides and turns and return the result
- The dice class will also print each roll and random number in the class itself to keep the program clean

Main

- This should only have two or three functions to play the game

Input Validation

- The user will be asked mainly integers in the menu options. The cases are usually within a range of answers. Therefore, I will create a function that checks for an int, and then if that int is within a specified range.
    - Ex: choose 1 to 3 will ask for an int within range 1 to 3
    - Ex2: choose starting grid width will be min 1 and max int

Test Case	Input Values	Functions	Expected Outcomes	Observed Outcomes
Input too low	Input is less than the min	Main() Validation() If input < min	Loop back and ask the user to re-input	Continually asks user to enter a higher int
Input too high	Input is more than the max	Main() Validation() If input > max	Loop back and ask the user to re-input	Continually asks user to enter a lower int
Input in correct range	Input is within max and min	Main() Validation() If min>input>max	Accepts the correct answer and exits the while loop	Loop stops and accepts correct int
Input is not an int	Input is a string, bool, float or other variable	Main() Validation() If input != int	Loop back and ask the user to re-input an int	Loops until an int is entered within the designated range.
Medusa glare activates for harry potter	Medusa and harry potter chosen	Main() Harrypotter() Medusa() Game(0	Harry potter will not die the first time but rather the strength will be set to 20. If the skill glare activates a second time it will cause the user to die.	Harry potter dies the second time but not the first time. Strength points is set to 20. Medusa usually ends up winning if glare is activated.
Vampire uses charm as defense	50% of the time vampire will nullify any damage	Vampire() All other classes() Game()	Vampire will not take in any incoming damage or calculate it into the strength points	The damage value is not using during vampire's defense phase and the strength points are untouched when the skill is activated
Blue men uses mob	Mob activates and the number of dice is reduced	Bluemen() All other classes() Game()	Bluemen will activate mob if the dmg exceeds 4	The degree of factor of 4 is used for the total strength. Each time mob is activated, the blue men will reduce the number of defense die turns. This also only happens if the total strength is effected and not only the raw

				attack number. Otherwise there will be a lot of residual left over.
Attack	The character attacks	All child character classes() Game()	Character attacks and returns the attack to the game class. The game class will then use this and pass through the defender	The defender receives the results of the attacker's attack number from the game class. If the attacker is medusa and glare is activated, the attack value will be very large to ensure instant death
Defense	The character defense and receives the attack as input	All child character classes() Game()	The character defends and receives the attack information for the other character from the game class.	The defender receives the attack information from the game class and calculates the information accordingly. Bluemen, vampire, and harry potter will need to implement special functions in order to use their abilities.

Reflection

In this program, we had to pit two different characters up against each other in rounds and ask the user if they would like to continue playing the game. The hardest part about this lab was actually mixing the characters and having them react to each other. One thing I learned from this lab is that if I wanted to do this, I would have to have the characters be very independent from each other. This also allows them to be flexible because they are not affected by others. One really helpful assignment was the group project. The group project was very similar in implementation to this program and it helped me to keep the characters independent of each other. Medusa had to have a large output of attack for an instant kill whereas vampire had to ignore the incoming attack whenever charm activated. A really hard thing for me was keeping track of all the special abilities until I decided to design them independently of each other.

I also ran into a lot of errors with calculating the strength points. I found that it was hard to wrap my head around strength points, because this is usually used as a term for attacking strength. I would suggest using hp or health points instead, this may help others to wrap their head around the values of the strength points. There were often times where I would start using the strength points as

attack points in calculations and spend a lot of time debugging to see why my calculations where off. Otherwise after I had found this issue it was easy to use the defense points to calculate this. One other thing is the correlation between vampire and medusa. Vampire had to have a higher int number than medusa because of it's attack. The fact that medusa and vampire both had either instant kill or negating skills really made it challenging.

A few things which I found that I changed from the original design was how the characters interacted with one another. Originally for blue men, I had used the attack and divided it by four to reduce the dice. However, I quickly found that this led to errors because there would be a lot of remainders which would not be accounted for. In the end, I started to divide the overall strength points by 4 in order to calculate the mob effect of the blue men. I also changed how the characters behaved and started to use a game class to mediate the attacks. I found that this was easier to have a central hub where the character information could be exchanged. Much like an index, the game class was used to separate and distribute the points. Overall I found that this assignment was not too hard, however I can only imagine how hard the next one will be.

Class Hierarchy:

Character – parent class

- Int strength, defense, number of sides of die, number of rolls of die, special attack
- String description, skill description, name of character
- Getter's and setters for all variables
- Virtual functions for attack and defense

Vampire – child class

- Inherits from the character class
- Has special ability charm which negates all incoming attacks
    - This ability will polymorph the defense function

Medusa – child class

- Inherits from the character class
- Has special ability glare which maximizes attack output for an instant win
    - This ability will polymorph the attack function

Blue Men – child class

- Inherits from the character class
- Has the special ability mob which will reduce the number of defense rolls when the strength is modified by 4
- This ability will polymorph the defense function

Harry Potter – child class

- Inherits from the character class
- Has the special ability Hogwarts which revives the character and sets them to 20 strength after revival
- This skill will modify the defense function and have a bool flag to mark if the ability has been used.

Barbarian – child class

- This will be a direct inherit from the character class and will not be modified except for the base parameters.