Name: Daniel Maynard
Assignment: Project 3
Date: 8/4/17

Reflection

Working on this program gave me a better understanding of inheritance, in that creating a base class of the creature; I better understood that although each derived class would be inheriting the base function from the ‘base’ class, they would have to mostly have to be virtual function. The reason most of the functions needed to be virtual functions is because each of the derived class would have their own statistics, like strength, armor, attack and defense rolls. I initially did not have virtual functions for the getStrength and getArmor functions because I initially thought they would just inherit their values from creature. After working through this project I realized that there was a slicing issue happening, in that for the base function for were inheriting 0 (my return type in the base class) for strength and armor since I had not made them virtual in the base class.

One of the other issues I had in creating this program was thinking of how the vampire’s special defense move “charm” was to override medusa’s special attack “glare.” After working through the code, what I found that worked best for me was to add defense and attack values for the medusa’s stare and vampire’s charm. By making the vampire’s special defense move to a higher value than medusa’s stare I was able to successfully make the vampire’s defense override the medusa’s attack.

The last bug I was encountering was if the first player killed the 2nd player, the 2nd player was rolling their attack; even though they were dead. To fix this, I added an if statement under the first player to check their health, and if they were less than or equal to 0 it would break the while loop and end the game.

Originally my design was just to get the damage done in the defense variable, but as I worked through the program I found it might be easier to display with a damage return function. In the damage return function I had adjust it to take the opponent as an argument because otherwise it would not be able to get the armor type to deduct. Another area that had to change was that I have a while loop that checks the health of each player, if either went to 0 or below it would end the loop, but I found that the loop would continue even if one player would die. Which mean that even though player 1 might die, player 1 would still roll an attack die. To fix the while loop bug I inserted an if statement that checked the health of each player, and if one went to 0 or below it would break the while loop.

Some of my observations working through this project was that the blue men are the most overpowers, 2nd seems between Harry Potter and the Vampire, if the Vampire can roll his charm ability. 3rd seems between Medusa and the Barbarian.

Design

Test Case: inputValidation - Within menu

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
input too low	input < 1	getIntegerFromUser	Your input is invalid. Please enter an integer from 1 to 4.	Looped back to allow the user to enter integers between 1-4
input not a number	input %#	getIntegerFromUser	Your input is invalid. Please enter an integer from 1 to 4.	Looped back to allow the user to enter integers between 1-4
Input is a string	1234	getIntegerFromUser	Your input is invalid. Please enter an integer from 1 to 4.	Looped back to allow the user to enter integers between 1-4
Input too high	Input > 4	getIntegerFromUser	Your input is invalid. Please enter an integer from 1 to 4.	Looped back to allow the user to enter integers between 1-4

Test Case: play

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Player1 vs player2 = Player2's health <= 0		play	Player 2 died Player 1's score increases	Player 2 died Player 1's score increases
Player1 vs player2 = Player1's health <= 0		play	Player 1 died Player 2's score increases	Player 2 died Player 1's score increases

Test Case: Vampire - Charm

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Vampire's charm activated		getDefense	Vampire's defense = 1000	Vampire's defense = 500
Vampire's charm activated - Medusa's gaze activated		Vampire: getDefense Medusa: getAttack	Vampire defense = 1000 Medusa Attack = 500 Vampire defense	Vampire defend's Medusa's gaze

Test Case: Harry Potter - Hogwart's Ability

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Hogwart's ability activated		getDefense	Harry Potter's strength returns with 20	Harry Potter's strength returns with 20

Test Case: Blue Men - Mob “Ability”

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Blue Men lose 4 strength		getDefense	Blue men lose 1 defense die	Blue men lose 1 defense die

Test Case: Medusa’s Gaze

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Medusa Activates Gaze - Attack roll == 12		getAttack	Attack increases to 500	Attack increases to 500

Test Case: getRandomNumber

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Random integer min max		getRandomNumber	Random integer is generated between min-max	Random integer is generated between min-max

Test Case: Vampire

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Vampire fights Vampire		Play function	Vampire’s able to charm each other, player. who’s health <= 0 loses	Vampire’s able to charm each other, vampire who’s health <= 0 loses
Vampire fights Barbarian		Play function	Vampire able to charm each other. player who’s health <= 0 loses	Vampire able to charm each other. player who’s health <= 0 loses
Vampire fights Medusa		Play function	Vampire able to charm each medusa, even if medusa uses gaze. Player with str <= 0 loses	Vampire able to charm each medusa, even if medusa uses gaze. Player with str <= 0 loses
Vampire fights Harry Potter		Play function	Vampire able to charm each player. player who’s health <= 0 loses	Vampire able to charm each other. player who’s health <= 0 loses
Vampire fights Blue Men		Play function	Vampire able to charm each player. player who’s health <= 0 loses	Vampire able to charm each player. player who’s health <= 0 loses

Test Case: Barbarian

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Barbarian fights Barbarian		Play function	Player with health <=0 wins	Player with health <=0 wins
Vampire fights Barbarian		Play function	Player with health <=0 wins. Vampire able to use charm	Player with health <=0 wins. Vampire able to use charm
Barbarian fights Medusa		Play function	Player with health <=0 wins. Medusa able to use gaze	Player with health <=0 wins. Medusa able to use gaze
Barbarian fights Harry Potter		Play function	Player with health <=0 wins	Player with health <=0 wins
Barbarian fights Blue Men		Play function	Player with health <=0 wins	Player with health <=0 wins

Test Case: Medusa

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Medusa fights Medusa		Play function	Medusa able to use gaze on each other, player with str <= 0 loses	Medusa able to use gaze on each other, player with str <= 0 loses
Medusa fights Vampire		Play function	Medusa able to use gaze on each other, player with str <= 0 loses. Vampire able to charm medusa’s attack	Medusa able to use gaze on each other, player with str <= 0 loses. Vampire able to charm medusa’s attack
Barbarian fights Medusa		Play function	Medusa able to use gaze on each other, player with str <= 0 loses	Medusa able to use gaze on each other, player with str <= 0 loses
Medusa fights Harry Potter		Play function	Medusa able to use gaze on each other, player with str <= 0 loses	Medusa able to use gaze on each other, player with str <= 0 loses
Medusa fights Blue Men		Play function	Medusa able to use gaze on each other, player with str <= 0 loses	Medusa able to use gaze on each other, player with str <= 0 loses

Test Case: Harry Potter

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Harry Potter fights Harry Potter		Play function	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses
Harry Potter fights Vampire		Play function	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses
Barbarian fights Harry Potter		Play function	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses
Medusa fights Harry Potter		Play function	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses
Harry Potter fights Blue Men		Play function	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses	Harry Potter able to use hogwarts ability if he goes to 0 or below str. Player with str <= 0 loses

Test Case: Blue Men

Test Case	Input Value	Driver Function	Expected Outcome	Observed Outcome
Blue Men fights Blue Men		Play function	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses
Blue Men fights Vampire		Play function	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses
Blue Men fights Medusa		Play function	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses
Blue Men fights Harry Potter		Play function	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses
Barbarian fights Blue Men		Play function	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses	If blue men’s health goes down by 4, blue men lose 1 defense die. Player with str <=0 loses