Ethan Dunham

Project 3 design

2/14/17

Problem:

Create 5 derived classes of a Creature class, Vampire, Medusa, Harry Potter, BlueMen, and Barbarian. Each class has similar attributes, but several differences. Create a main that calls a menu to give the option to pick 2 fighters to fight each other.

Design:

Int Main():

Main function calls menu to start the game.

Seeds srand().

Void Menu():

Asks which of the 5 fighter types you want the first player to be.

Creates the player1 creature pointer to point to that type of fighter.

Asks which of the 5 fighter types you want the second player to be.

Creates the player2 creature pointer to point to that type of fighter.

Determines which player goes first with rand().

Calls the corresponding attack roll for the attacker.

Sends the attack roll to the defense roll.

Switches attacker and defender.

Checks to see if one has died.

Loops attack and defense until one dies.

Says which player won.

Creature and creature derived classes:

All 5 subclasses inherit from Creature.

The various Creature constructors fill in the appropriate number of die, att and def.

Fills in the number of sides on those dice.

Sets the armor and strength depending on which creature is called.

Int Roll attack:

Attack is rolled based on the number of attack dice and the number of sides. It will loop through, adding to the total until the number of dice is 0.

Medusa attack roll will override this and return 400 if they get a perfect roll. A perfect roll means the enemy gazed into their eyes and dies.

Void Roll Def(int):

This will work the exact same way as Roll Attack, except it will use the number of defensive dice and sides. It will also take in the attack roll and subtract the defense roll and armor from that to determine hit.

Vampire def will override this function to incorporate charm. If charm is true, it will not deduct the damage sent to it.

Bool Charm:

Charm will be a vampire only function that will use Rand to determine if the attack is calculated or ignored. There is a 50% chance that an attack will not be counted against the vampire.

Bool Death():

Death will be called after each round to see if the strength of the defender is at 0. If it is at 0, the game will end and the victor will be displayed by Menu()

Harry Potter Death():

Harry Potter will have a death value that is set to 0 by his constructor. Death will be overridden for Harry Potter and if his strength is at 0, he will be resurrected with 2x as much strength as before. If his death counter is at 1, he will die like normal.

Blue Men Death():

This will override the death function for Blue Men. Blue Men are a mob of little critters. For each 4 dmg they take, they lost one of their mob. This decreases the number of their defensive dice.

Test Plan:

Input	Expected results	Viewed results
HP vs vampire	HP wins more because of extra life.	HP 5 wins Vamp 1 win.

Vamp Vs Barbarian	Vamp wins more because of charm	Vamp 5, Barb 1.
Blue men vs Barb	Blue men win most because of mob	BM 6, Barb 0.
Medusa Vs Barb	Medusa wins most because barbarian is pain and boring.	Barbarian 5, Medusa 1. Damage of Barbarian lets it win unless Medusa uses glare.

Reflection:

It all went together quickly without any real issues. It worked as intended based on the design. I reused parts from the dice game and continue to reuse the menu function, which saved me a lot of time in making this program.