Thomas Buteau
CS162-400
2-19-17
Project 3
Page 1 of 6

Program Overview
This project is an arena combat simulation. The user picks two fighters from a list of 5 different fighter types. The chosen fighters fight until one has been killed. The results are displayed and the user is asked if another round should be played.

Program Design
The program will consist of 7 main classes and 1 utility class. There is a pure virtual Fighter class which asks as a parent for the other creature classes, the 5 usable creature classes: Barbarian, Blue Men, Harry Potter, Medusa, and Vampire. The final main class is the Battle class which handles the game mechanics. Lastly, there is the Data Validation utility class.

Fighter Class
The fighter class is the template for the other creature classes. Fighter has the following variables: attDieNum, attDiePow, defDieNum, defDiePow, armor, health, name. the *DieNum and *DiePow variables combined to make dice rolls. For example, an attDieNum of 2 and attDiePow of 6 would result in an attack roll of 2 6 sided dice, or 2D6. Fighter has the following methods: attack, defense, isDead, rollDice, getHealth, and getName.

The attack method takes another Fighter object pointer as an argument and calls the rollDice method with the attDiePow attDieNum number of times, sums the results, prints the total attack, and sends the results as an argument to the other Fighter object's defense method.

The defense method takes as an argument an int that represents the opponents attack. The method then rolls defense in the same way as the attack method but using defDieNum and defDiePow. Then it prints out the results and subtracts the defense roll from the attack argument. If the attack is still greater than 0 and the defending Fighter has armor the armor is subtracted from the attack with this also printed. Finally, if the attack is still greater than 0 the defender's health is subtracted by the attack and the results are printed.

The isDead method sets the Fighter's health to 0.

The rollDice method takes an int diePow for the power of a die, or how many sides it has. It rolls a random number from 1 to the diePow and returns it.

The getHealth method returns the Fighter's health.

The getName method returns the Fighter's name.

Barbarian Class
The barbarian class is a basic creature with no variations from the Fighter class. The Barbarian has an attack of 2D6, a defense of 2D6, 0 armor, 12 health, and the name "Barbarian".

Blue Men Class

The BlueMen class operate the same way the Fighter class does except that their defDieNum is dependent on their current health and is calculated each time the defense method is called. BlueMen start with 2D10 for attack, 3D6 for defense, 3 armor, 12 health, and a name “Blue Men”.

The defense method works the same way as in the Fighter class except that at the start it sets defDieNum equal to health / 4. This means after the first point of damage the BlueMen have 2D6 defense and when their health drops to 7 they have 1D6 defense etc.

Harry Potter Class

The HarryPotter class operates the same way as the Fighter class for the most part. HarryPotter starts with 2D6 attack, 2D6 defense, 0 armor, 10 health, and a name “Harry Potter”. HarryPotter also has an additional variable which is a bool called hasDied which is initialized as false. The HarryPotter class has a modified version of the defense and isDead methods as well as a new method called resurrect.

The resurrect method sets the hasDied variable to true then sets health to 20.

The defense method operates the same as the Fighter method except that at the end if HarryPotter has less than 1 health and hasDied is false it calls the resurrect method.

The isDead method likewise will check if hasDied is false and if so will call resurrect.

Medusa Class

The Medusa class operates the same as the Fighter class except for a change to the attack method. Medusa starts with 2D6 for attack, 1D6 for defense, 3 armor, 8 health, and a name “Medusa”. The Medusa class has a special ability called glare. The way this works is that if Medusa rolls max damage on attack, 6 on both dice for a total of 12, then instead of calling the opponent’s defense method the isDead method is called instead.

Vampire Class

The Vampire class operates the same as the Fighter class except for a change to the defense method as well as an extra method called charm. Vampire starts with 1D12 for attack, 1D6 for defense, 1 armor, 18 health, and a name “Vampire”.

The charm method rolls randomly and returns either a 1 or a 0. The defense method checks at the beginning of the method and if charm returns 1, true, then no damage is taken.

Battle Class

The Battle class has 2 Fighter pointers as variables and has a menu, fight, and a setP method.

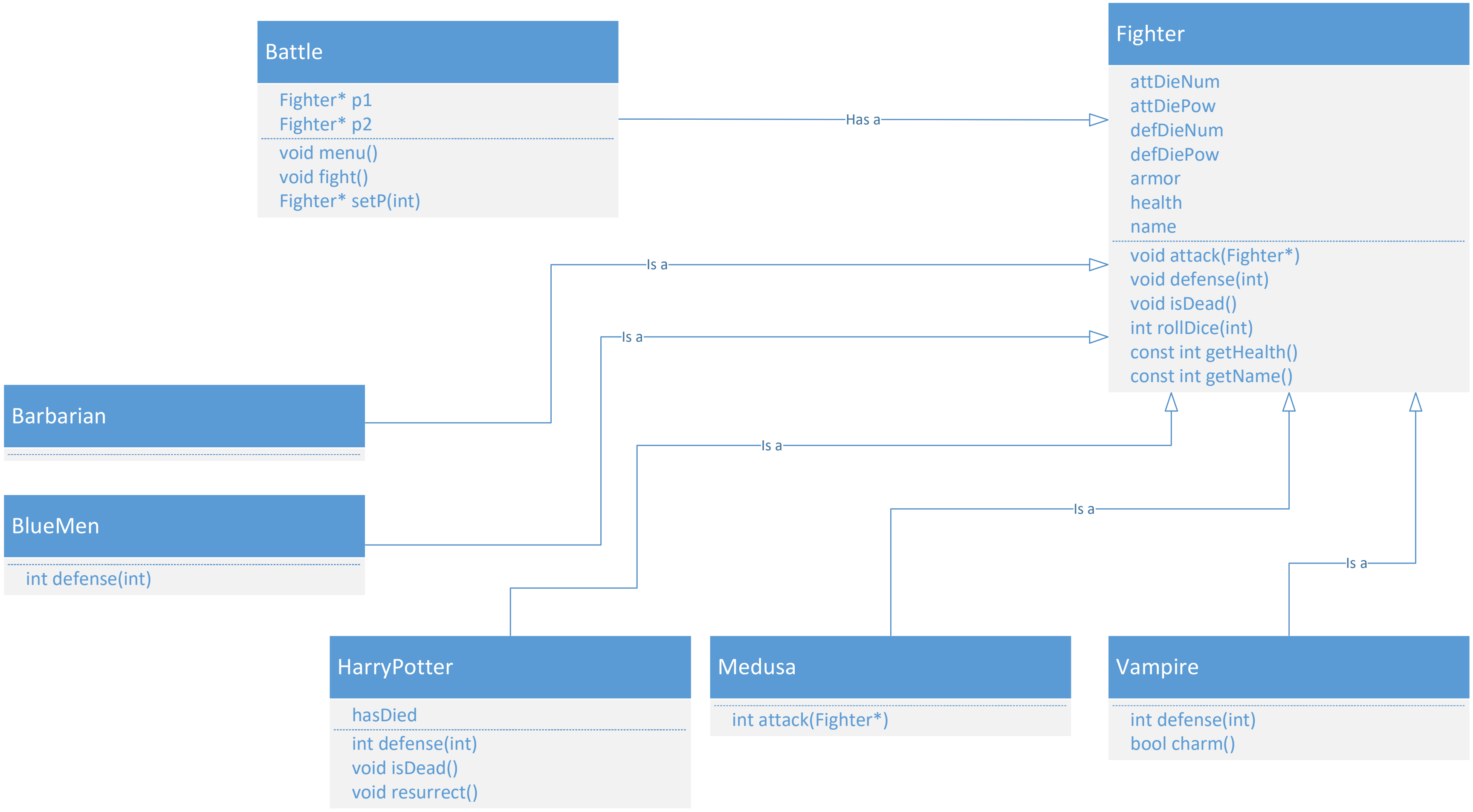
The menu method asks the user to select two fighters out of the list of fighter types and calls setP to generate the Fighter objects, then calls the fight method, deletes the Fighter objects, and asks the user if they would like to play again.

The fight method picks one of the Fighter objects at random to go first and calls its attack method on the other Fighter object. Then it calls the second fighters attack method on the first. This loops until one of the Fighter objects has died (health < 1) and displays the results.

The setP method creates and returns an object of one of the 5 playable creature classes based on the int argument it is provided.

Thomas Buteau
CS162-400
2-19-17
Project 3
Page 4 of 6

Class Hierarchy Diagram



Thomas Buteau
CS162-400
2-19-17
Project 3
Page 5 of 6

Tests

Iteration	Match-up	Result	Overall		Iteration	Match-up	Result	Overall		Iteration	Match-up	Result	Overall		Iteration	Match-up	Result	Overall		Iteration	Match-up	Result	Overall
1	Barb vs Barb	Barb			1	Blue vs Blue	Blue			1	HP vs HP	HP			1	Med vs Med	Med			1	Vamp vs Vamp	Vamp	
2	Barb vs Barb	Barb			2	Blue vs Blue	Blue			2	HP vs HP	HP			2	Med vs Med	Med			2	Vamp vs Vamp	Vamp	
3	Barb vs Barb	Barb			3	Blue vs Blue	Blue			3	HP vs HP	HP			3	Med vs Med	Med			3	Vamp vs Vamp	Vamp	
4	Barb vs Barb	Barb			4	Blue vs Blue	Blue			4	HP vs HP	HP			4	Med vs Med	Med			4	Vamp vs Vamp	Vamp	
5	Barb vs Barb	Barb	Barb		5	Blue vs Blue	Blue	Blue		5	HP vs HP	HP	HP		5	Med vs Med	Med	Med		5	Vamp vs Vamp	Vamp	Vamp
1	Barb vs Blue	Blue			1	Blue vs HP	Blue			1	HP vs Med	HP			1	Med vs Vamp	Vamp						
2	Barb vs Blue	Blue			2	Blue vs HP	Blue			2	HP vs Med	HP			2	Med vs Vamp	Vamp						
3	Barb vs Blue	Blue			3	Blue vs HP	Blue			3	HP vs Med	HP			3	Med vs Vamp	Vamp						
4	Barb vs Blue	Blue			4	Blue vs HP	Blue			4	HP vs Med	HP			4	Med vs Vamp	Vamp						
5	Barb vs Blue	Blue	Blue		5	Blue vs HP	Blue	Blue		5	HP vs Med	HP	HP		5	Med vs Vamp	Med	Vamp					
1	Barb vs HP	HP			1	Blue vs Med	Blue			1	HP vs Vamp	HP											
2	Barb vs HP	HP			2	Blue vs Med	Blue			2	HP vs Vamp	HP											
3	Barb vs HP	HP			3	Blue vs Med	Blue			3	HP vs Vamp	HP											
4	Barb vs HP	HP			4	Blue vs Med	Blue			4	HP vs Vamp	Vamp											
5	Barb vs HP	HP	HP		5	Blue vs Med	Blue	Blue		5	HP vs Vamp	HP	HP										
1	Barb vs Med	Barb			1	Blue vs Vamp	Blue																
2	Barb vs Med	Barb			2	Blue vs Vamp	Blue																
3	Barb vs Med	Barb			3	Blue vs Vamp	Blue																
4	Barb vs Med	Barb			4	Blue vs Vamp	Blue																
5	Barb vs Med	Barb	Barb		5	Blue vs Vamp	Blue	Blue															
1	Barb vs Vamp	Vamp																					
2	Barb vs Vamp	Vamp																					
3	Barb vs Vamp	Vamp																					
4	Barb vs Vamp	Barb																					
5	Barb vs Vamp	Vamp	Vamp																				

For the testing of this program I had each creature type fight each other creature type 5 times. As the results show, the Blue Men never lost a combat. Medusa was the likeliest to be able to defeat them with a 1/36 chance to glare them to death but it did not happen in 5 engagements. The shear amount of health the Harry Potter class has, combined with the ability to absorb one glare from Medusa made it the second-best fighter. Third place goes to the Vampire who's charm ability deflected large amounts of damage. The Barbarian and Medusa classes should tie for last place. The Barbarian lacks any abilities to help it stand out so it only wins when the dice rolls are lucky. The Medusa class has a very nice ability but it is rare to have it kick off and her low health is a liability.

Reflections

The hardest part of this project was deciding how to implement the attack and defense methods with the various special abilities available to the fighters. An early draft of the design had the defense class taking not only an int for the attack damage but also a string for any special abilities. This design was scrapped because it would also require the attack method to return multiple types of data and while a struct would have worked I felt it would make things more complex than they needed to be. Ultimately, I decided to add a new method to each class, the isDead method, to account for Medusa’s ability. This worked out nicely because it supersedes the Vampire’s charm ability like I wanted and the HarryPotter class can make a simple modification to the isDead method like it does for the defense method to account for its own resurrect ability. Depending on future requirements this design may have to be overhauled if new creature types are introduced that have abilities which are incompatible with the current design.