CS 162 – Assignment 3
Fantasy Battle

Daniel Beyer
Dan.beyer@gmail.com
05/05/2016

Requirements

Design, implement, and test a fantasy creature battle program using class hierarchy as the basis. Each creature will have individual characteristics for attack, defense, armor, and strength. Attack and defense values are determined by rolling 6 or 10-sided die. Some of the creatures will have special abilities.

Type	Attack	Defense	Armor	Strength
Medusa	2d6*Glare	1d6	3	8
Barbarian	2d6	2d6	0	12
Baba Yaga	2d6*Soul	1d10	3	12
Blue Men	2d10	3d6*	3	12*Mob
Harry Potter	2d6	2d6	0	10/20*Hogwarts

*Glare: If Medusa rolls a 12 in attack, then the target dies.
*Soul: When Baba Yaga attacks, she receives 1/3 of her attack value (rounded down) towards her strength points.
*Mob: For every 4 points of damage (rounded down), they lose one defense die.
*Hogwarts: If Harry dies, he immediately recovers and his strength becomes 20. If he dies again, he dies for good.

A base Creature class will act as an abstract class with pure virtual functions for attack and defend.

Program Input

 User enters:

- Initial Menu: Select 2 Creatures to fight
    o 1. Medusa
    o 2. Barbarian
    o 3. Baba Yaga
    o 4. Blue Men
    o 5. Harry Potter
    o 6. Exit

Program Output

The program displays to the screen the progress of the fight between the two creatures. For each round of attack and defend, the program outputs the attack value, defense value, and the amount of damage done to the defending creature. Any special abilities that are performed are

also displayed. Once the fight is over and one creature has died, the program returns to the main menu.

Design
Main method pseudocode
Initialize random number seed, set to time(0) and seed srand
Declare int exit, choice1, choice2

Initialize Creature pointer variables and set to NULL

While(exit!=6)

- Prompt user with Menu() function
- Prompt user to select Creature 1
    o If invalid selection, reprompt
- Prompt user to select Creature 2
    o If invalid selection, reprompt
- Using *if* statements, create new creatures based on menu selection
    o E.g. if (choice1 == 2)
        ▪ Creature1 = new Medusa;
- Display names of creature1 and creature2
- While(creature1->getStrength() > 0 && creature2->getStrength > 0)
    o Display creature2 remaining strength
    o Creature1 attacks creature2, returns attackValue
        ▪ If attackValue == 100, Medusa used Glare
            • Creature2->setStrength(0)
            • Creature2 died
        ▪ Else
            • Display attackValue
            • Creature2 defends against creature1, returns damage
            • Display defense roll and damage taken
        ▪ If creature2->getStrength() <= 0
            • Creature2 died
    o Display creature1 remaining strength
    o Creature2 attacks creature1, returns attackValue
        ▪ If attackValue == 100, Medusa used Glare
            • Creature1->setStrength(0)
            • Creature1 died
        ▪ Else
            • Display attackValue
            • Creature1 defends against creature2, returns damage
            • Display defense roll and damage taken
        ▪ If creature1->getStrength() <= 0
            • Creature1 died

Creature class methods

Protected Variables:

- Int armor, strength, attackRolls, defRolls, attackSides, defSides
- String name

Creature()
Default Constructor. Sets all int variables to 0 and the string variable to " ".

Creature(int armor, int strenth, string name, int attackRolls, int attackSides, int defRolls, int defSides)
Constructor for base class Creature. Sets variables to parameters specified.

string getName()
Getter function for name.

int getStrength()
Getter function for strength.

int getArmor()
Getter function for armor.

void setStrength(int)
Mutator function to set strength.

virtual int Attack()
initialize and set int attackValue to 0
for(int i =0; i<attackRolls; i++)
- Add result of random roll to attackValue
Return attackValue

virtual int Defend(int attackValue)
initialize and set int damage and defenseValue to 0
for(int i =0; i<defRolls; i++)
- Add result of random roll to defenseValue
If (attackValue < defenseValue + armor)
- Display "No damage done"
- Set damage = 0
Else
- Damage = attackValue – defenseValue – armor
If (damage >= strength)
- Strength = 0
Else

- Strength = strength – damage

Return damage

Barbarian class methods

Barbarian()
Constructor (use base class constructor)

Medusa class methods

Medusa()
Constructor (use base class constructor)

int Attack()
initialize and set int attackValue to 0
for(int i =0; i<attackRolls; i++)
- Add result of random roll to attackValue

If(attackValue == 12)
- Medusa has used glare
- attackValue = 100 (to act as indicator for Main that glare was used)

Return attackValue

BabaYaga class methods

BabaYaga()
Constructor (use base class constructor)

int Attack()
initialize and set int attackValue to 0
for(int i =0; i<attackRolls; i++)
- Add result of random roll to attackValue

Set strength = attackValue/3 for Soul ability
Display "Baba Yaga used Soul ability to add attackValue/3 to her strength"
Return attackValue

BlueMen class methods

BlueMen()
Constructor (use base class constructor)

Int Defend(int attackValue)
initialize and set int damage and defenseValue to 0
for(int i =0; i<defRolls; i++)
- Add result of random roll to defenseValue

If (attackValue < defenseValue + armor)
- Display "No damage done"
- Set damage = 0

Else

- Damage = attackValue – defenseValue – armor

If (damage >= strength)
- Strength = 0

Else
- Strength = strength – damage

If (strength is greater than 4 and less than or equal to 8)
- defRolls = 2
- Display "BlueMen have 2 dice rolls due to Mob ability"

If (strength is greater than 0 and less than or equal to 4)
- defRolls = 1
- Display "BlueMen have 1 dice roll due to Mob ability"

Return damage

HarryPotter class methods

Public Variable:
- Bool hasDied

HarryPotter()
Constructor (use base class constructor)
- Set hasDied to false

Int Defend(int attackValue)
initialize and set int damage and defenseValue to 0

for(int i =0; i<defRolls; i++)
- Add result of random roll to defenseValue

If (attackValue < defenseValue + armor)
- Display "No damage done"
- Set damage = 0

Else
- Damage = attackValue – defenseValue – armor

If (damage is greater than strength and hasDied is false)
- Set strength to 20
- Display "Harry Potter used the Hogwarts ability"
- Set hasDied to true

If (damage is greater than strength and hasDied is true)
- Strength = 0

Else
- Strength = strength – damage

Return damage

HelperFunctions class methods

Menu()
Menu function to select creature. Displays list of creatures

Sleep(int milliseconds)

Pauses the program for a specified number of milliseconds

Test Plan

Test Case	Input Values	Driver Functions	Expected Outcome	Observed Outcome
Input out or range or not an integer	Input > 6 or < 1 or input "a"	Main() while (choice1 < 1 || choice1 > 6 || !cin)	"Invalid selection, please select a number from 1 to 6"	"Invalid selection, please select a number from 1 to 6"
Barbarian vs Barbarian	Input = 1 Intput = 1	Main() Attack() Defend()	Barbarian does random attack damage 2-12 and random defense of 2-12. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Barbarian does random attack damage 2-12 and random defense of 2-12. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Barbarian vs Medusa	Input = 1 Input = 2	Main() Attack() Defend()	Barbarian does random attack damage 2-12 and random defense of 2-12. Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Barbarian does random attack damage 2-12 and random defense of 2-12. Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken. **Glare observed
Barbarian vs Baba Yaga	Input = 1 Input = 3	Main() Attack() Defend()	Barbarian does random attack damage 2-12 and	Barbarian does random attack damage 2-12 and

			random defense of 2-12. Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	random defense of 2-12. Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Barbarian vs Blue Men	Input = 1 Intput = 4	Main() Attack() Defend()	Barbarian does random attack damage 2-12 and random defense of 2-12. Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Barbarian does random attack damage 2-12 and random defense of 2-12. Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Barbarian vs Harry Potter	Input = 1 Intput = 5	Main() Attack() Defend()	Barbarian does random attack damage 2-12 and random defense of 2-12. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his	Barbarian does random attack damage 2-12 and random defense of 2-12. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his

			strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Medusa vs Medusa	Input = 2 Intput = 2	Main() Attack() Defend()	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken **Glare observed
Medusa vs Baba Yaga	Input = 2 Intput = 3	Main() Attack() Defend()	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Damage taken is attackValue – defenseValue – armor. If	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Damage taken is attackValue – defenseValue – armor. If

			attackValue is less than defenseValue+armor, no damage is taken.	attackValue is less than defenseValue+armor, no damage is taken. **Glare observed
Medusa vs Blue Men	Input = 2 Input = 4	Main() Attack() Defend()	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken. **Glare observed
Medusa vs Harry Potter	Input = 2 Input = 5	Main() Attack() Defend()	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he	Medusa does random attack damage 2-12 and random defense of 1-6. If she rolls 12 attack, Glare performed, attackValue = 100, opponent dies. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he

			dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken. **Glare observed
Baba Yaga vs Baba Yaga	Input = 3 Input = 3	Main() Attack() Defend()	Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Baba Yaga vs Blue Men	Input = 3 Input = 4	Main() Attack() Defend()	Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.

Baba Yaga vs Harry Potter	Input = 3 Input = 5	Main() Attack() Defend()	Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Baba Yaga does random damage 2-12 and random defense 1-10. For every attack, she adds 1/3 of attack value to her strength. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Blue Men vs Blue Men	Input = 4 Input = 4	Main() Attack() Defend()	Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12 while (4<strength<=8), and random defense 1-6 while strength <=4. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Blue Men vs Harry Potter	Input = 4 Input = 5	Main() Attack() Defend()	Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12	Blue Men does random attack damage 2-10 and random defense 3-18 while strength > 8, random defense 2-12

			while (4<strength<=8), and random defense 1-6 while strength <=4. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	while (4<strength<=8), and random defense 1-6 while strength <=4. Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Harry Potter vs Harry Potter	Input = 5 Input = 5	Main() Attack() Defend()	Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.	Harry Potter does random attack damage 2-12 and random defense of 2-12. When he dies the first time, his strength = 20 and he continues. When he dies the second time, he stays dead. Damage taken is attackValue – defenseValue – armor. If attackValue is less than defenseValue+armor, no damage is taken.
Exit	Input = 6	Main()	Program exits	Program exits

Reflection

My initial design worked well but there were some changes and issues I had to make in my final code.

My initial thought was to include a "Press enter to continue" option in between each attack and defend round. This proved to be difficult to implement in a simple and straightforward manner. I tried using cin.ignore(), but this would skip a few implementations depending on how many characters were saved in the buffer from previous inputs. Instead, I decided to utilize a sleep function to provide a pause between each attack and defend round. This proved to be a simple and elegant solution that I think gives the game a better flow.

I initially planned to use a different constructor for each creature variation, but this seemed cumbersome and defeated the purpose of working with class hierarchy. I eventually opted to use the base class constructors for each subclass constructor:

```
HarryPotter::HarryPotter() : Creature(0, 10, "Harry Potter", 2, 6, 2, 6)
```

This proved to be much easier and more streamlined.

One area that I spent probably too much time was organizing the display outflow flow so all the relevant information was displayed to the screen in an organized manner. I wanted to show the amount of strength remaining for each creature at the beginning of each round, the amount of attack damage rolled, the amount of defense points rolled, the total damage done, and any special abilities used. I ended up placing most of these outputs in Main(), except for the special abilities. Special ability output notifications I placed in their respective creature functions to be called when appropriate.

This was a great project for me to practice class hierarchy. This was a subject I had no experience with prior to this class, and this program really helped me understand how powerful a tool hierarchy and inheritance can be.