Tucker Lavell
CS162 Fall 2017
Project 3: Fantasy Combat Game
Documentation

Problem

Create the first part of a Fantasy Combat Game. For this part of the project, we are creating a game that allows two characters to fight each other. When the program starts, it will show a list of the playable characters, and ask the user to choose 2 to have fight. Some of the characters have special abilities that change the way their combat works. When a fight commences, the user should be able to see the attacker type and defender type, as well as their dice roll, and calculated damage. Then it should ask if the player would like to play again or exit.

Design

```
class Die {
	protected:
		int sides;
		int rolls;
	public:
		Die();
		Die(int sides, int rolls);
		~Die();
		void setSides(int s);
		virtual int getSides();
		void setRolls(int r);
		virtual int getRolls();
		virtual int roll(int rolls) {
			if (rolls < 1) {
				return 0;
			}
			return (rand() % sides + 1) + roll(rolls - 1);
		}
};
```

```
class Character {
        protected:
                string name;
                int attack;
                int defense;
                int armor;
                int strength_points;
        public:
                Character();
                Character(string name, int attack, int defense, int armor, int strength_points);
                ~Character();
                void setName(string name);
                virtual string getName();
                virtual void attackRoll() = 0;
                virtual int getAttack();
                virtual void defenseRoll() = 0;
                virtual int getDefense();
                void setArmor(int ar);
                virtual int getArmor();
                void setStrength_Points(int armor);
                virtual int getStrength_Points();
                virtual void combat(Character& opponent);
};
class Barbarian : public Character {
        public:
                Barbarian();
                ~Barbarian();
                void setName(string n) {name = n;}
                string getName() {return name;}
                void attackRoll() {attack = new Die(6, 2).roll();}
                int getAttack() {return attack;}
                void defenseRoll() {defense = new Die(6, 2).roll();}
                int getDefense() {return defense;}
                void setArmor(int ar) {armor = ar;}
                int getArmor() {return armor;}
                void setStrength_Points(int str_pts) {strength_points = str_pts;}
                int getStrength_Points() {return strength_points;}
                void battle(Character& opponent);
};
```

Test Table

Test Case	Input Values	Expected Outcome	Observed Outcome
Barbarian1 vs Barbarian2	Barb1: attack = 5; Defense = 2; Armor = 0; Str_pts = 12; Barb2: attack = 4; Defense = 6; Armor = 0; Str_pts = 12;	Barb1 str_pts = 10; Barb2 str_pts = 12;	Barb1 str_pts = 10; Barb2 str_pts = 13;
Barb1 kills Barb2	Barb1: attack = 5; Defense = 2; Armor = 0; Str_pts = 5; Barb2: attack = 4; Defense = 3; Armor = 0; Str_pts = 3;	Print winner Ask to play again	Printed winner Asked to play again
Medusa uses gaze vs Harry Potter's revive	Medusa: attack = 12; Harry Potter: defence = 12; Hogwarts = 1;	Medusa uses glare Harry Potter dies once, comes back with 20 str_pts	Medusa used glare Harry Potter died Harry Potter came back with 20 str_pts
Vampire uses charm vs Medusa's gaze	Vampire: charm = 3; Medusa: attack = 12;	Medusa does no damage because of charm	Medusa did no damage because of charm

Injured Blue Men mob	Blue Men:
	defense = 11;
	armor = 3;
	str_pts = 8;
		Blue Men should only roll 2 dice instead of 3	Blue Men rolled 2 dice

Reflection

This was definitely my best project so far. I gave myself a week to work on it, but it wasn't until the deadline, so if I want to fix something I still have time. I don't think I have an interest in making games for a living, but this one was definitely more fun to make than the Zoo Tycoon. Last project my biggest takeaway was trying to make my code neater on larger projects. I think I succeeded this time, but I am not fully happy with the way it is currently. I have repeated code for attack and defense logic, I will try to go back and fix it, but I have other homework at the moment.

This project seemed easier than the other two, maybe because I did a better job on my design step than normal. This time I actually thought the program out before I started, which I think was a big contributor to neater code. I did like the suggestion of making the game work with a barbarian versus a barbarian. When I went to code, it was much easier to get combat to work with a reduced scope.

While trying to make combat work, I decided that it would be better to have combat receive both characters, rather than this vs opponent. I also came upon something I forgot to consider in my design, but thankfully I had my test table. The design for combat (at this stage) did not account for the fact that a defenseRoll could be greater than an attackRoll. At first it was adding strength points to the defender if the attacker rolled less, but I used a ternary operator to set damage properly if it would be below 0.

Making my second class, Vampire, is when I changed up my combat code. I had all the couts related to rolling, attack, and defense in combat, then I read the rubric and it said about overriding Vampire and Medusa. That is when I moved everything related to rolling attack, including couts, to each classes attackRoll, and everything related to rolling defense, including couts, was moved to defenseRoll. I think this is where I messed up, the roll functions have a lot of repeated code. Though my code works as expected, special abilities are a little funky, especially Medusa's glare. After making sure attackRoll and defenseRoll were printing exactly how I wanted them, with barbarian and vampire, I made the rest of my classes.

The classes were pretty straightforward after that, and didn't take too much time. I feel like my project might not be as good as I think, because it seemed too easy once I got started. I will try to go back and make things better, especially if we have to use this on another project later.