Tatyana Vlaskin
Lab 3 report

My last week design was too complex, I started implementing it, but it is taking me too long so I decided to make the game shorter. I decided its not worth implementing everything that I've wrote for just a lab and made slight revisions. According to the instruction for this lab: "Keep it simple. Implement the base class from your design."
When I wrote design for this game, last week, I had to idea that it would be implemented, thus I made it long and complex.

So for this implementation, I will have 3 classes: monsters, character and combat. I decided to leave out classes weapon and armor.
The user will be presented with 3 choices of fighter class: fighter, rogue and adventure and the user will chose who they want to be. Appropriate constructor will be called.

```
switch(result){
   case 1:
     return Character("Fighter", 100,25,10);
     break;
   case 2:
      return Character("Rogue", 120,38,6);
      break;
   case 3:
      return Character("Adventurer", 12,5,3);
   break;
}
```

As for the monsters, I'll create a vector of monsters and randomly chose a monsters from the vector:

```
std::vector<Monster*> monsters;
monsters.push_back(new Monster("Dragon", 50,40));// we can do arbitrary number of monsters
monsters.push_back(new Monster("Hobbib", 30,10));
monsters.push_back(new Monster("Goblin", 10,2));
srand(time(0)); // each time a game is played a different monster will be generated.
// Select a random monster from the array
// http://gamedev.stackexchange.com/questions/33705/how-can-i-randomly-choose-a-monster-from-an-array-of-monsters
int random_index = rand() % (monsters.size());// randomly chooses an element from the arrray
Monster* selected_monster = monsters[random_index];// player will finght againts that raendom monster
```

As for the combat class, I'll have 2 main functions for the fighter:

```
cout << endl<< "What king of weapon do you want to use to attack the monster?" <<endl
<< "1. Use baseball bat - SWUNG WEAPON" << endl
        Fighter.attackSwung(ActiveMonster);
        //user will use baseball bat for this function, this function will NOT increase the strength of the hit
<< "2. Use spears - THRUST WAPON" << endl
        Fighter.attackThrust(ActiveMonster);
        // user will use spears for this function. There is a limited number of spears, thus once the user runs out of spears they will not be able to use spears in battle any more.
```

As for the monster, it will just hit with his strength.
If the player runs out of health first, there is a notification that the player lost. If the monster runs out of health first there is a notification that the player won.

Testing:

According to the instructions: For testing just implement 3-4 different types of creatures to demonstrate that your class is working properly.

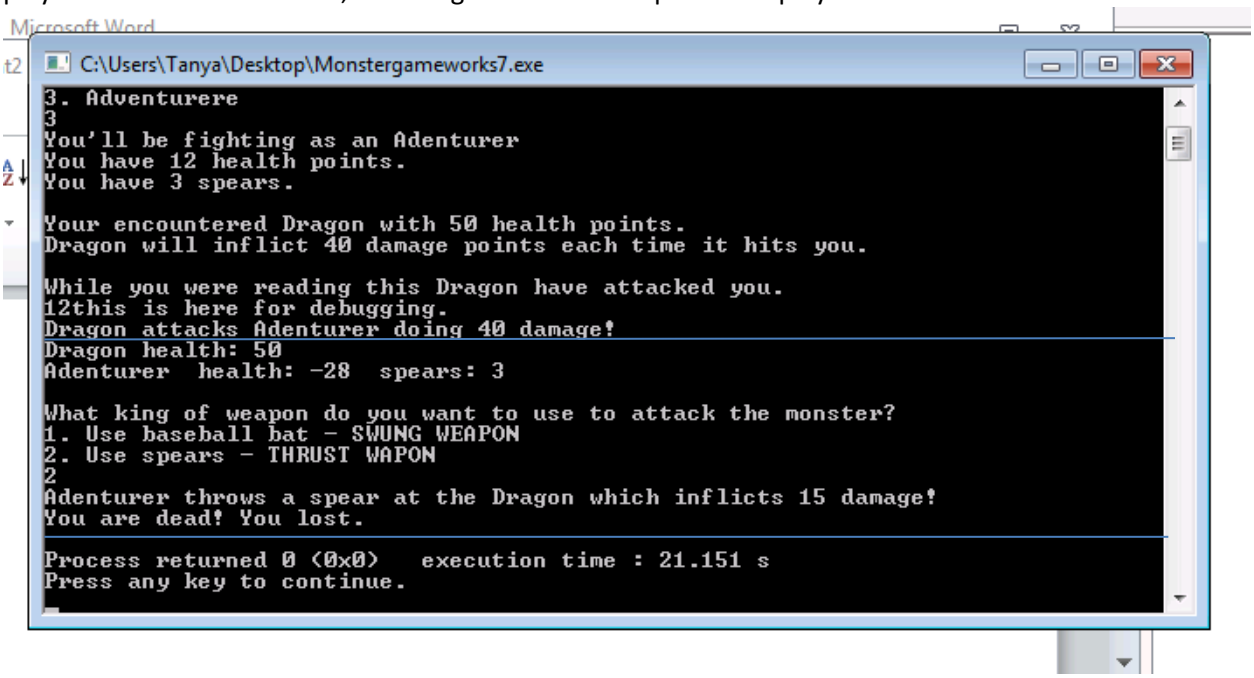
Test	How tested	What is expected	Outcome
a user chose a class fighter and the appropriate constructor is called	When the user is asked what fighter class they want , test option 1, 2 and 3 Depending on the selection, information about the fighters class, health, damage and spears are displayed.	case 1: Character("Fighter", 100,25,10); case 2: Character("Rogue", 120,38,6); case 3: Character("Adenturer", 12,5,3); (CLASS, HEALTH, DAMAGE, SPEARS)	Pass
Each time a game is started a different monster if selected	Start game several times and make sure that different monster is selected. There is a piece of code that displays with monster was selected	Different monster is selected	Pass
Check that its possible for a Rogue to win ROGUE/HOBBIB/THRUST COMBIATION/thrust	choose Character("Rogue", 120,38,6); Lets say that the hobbib is randomly generated and you decided to user thrust weapons, you should win the monster after the first attack. Hobbit looses	Message that the hobbit lost after player choses swung weapon	Pass
Check that its possible for a Rogue to **loose** ROGUE/DRAGON/THRUST COMBIATION	choose Character("Rogue", 120,38,6); because it has the strongest health. Lets say that the dragon is randomly generated and you decided to use thrust weapons, you should lose the monster after the first attack.	Message that the player lost after player choses swung weapon	pass
Check that its possible for a Rogue to win ROGUE/GOBLIN/THRUST COMBIATION	choose Character("Rogue",	Message that the goblin lost after player	Pass

	120,38,6); Lets say that the goblin is randomly generated and you decided to user swung weapons, you should win the monster after the first attack. Hobbit looses	choses swung weapon, Player won	
Check that its possible for a Rogue to win ROGUE/DRAGON/SWUNG COMBIATION	choose Character("Rogue", 120,38,6); because it has the strongest health. Lets say that the dragon is randomly generated and you decided to use swung weapons, you should win the monster after the first attack.	Message that the player won after player choses swung weapon	Pass
Check that its possible for a Rogue to WIN ROGUE/GOBLIN/SWUNG COMBIATION	choose Character("Rogue", 120,38,6); because it has the strongest health. Lets say that the goblin is randomly generated and you decided to use swung weapons, you should win the monster after the first attack.	Message that the player won after player choses swung weapon	Pass
Check that its possible for a Rogue to WIN ROGUE/HOBBIB/SWUNG COMBIATION	choose Character("Rogue", 120,38,6); because it has the strongest health. Lets say that the hobbib is randomly generated and you decided to use swung weapons, you should win the monster after the first attack.	Message that the player won after player choses swung weapon	Pass
FIGHGHER/GOBLIN/SWUNG COMBINATION	CHOSE FIGHTER, HOPE THAT GOBLIN IS RANDOMLY SELECTED, CHOSE SWUNG	USER WINS	PASS
FIGHGHER/GOBLIN/THRUSTCOMBINATION	CHOSE FIGHTER, HOPE THAT GOBLIN IS RANDOMLY SELECTED,	USER WINS	PASS

	CHOSE THRUST		
ROGUE/DRAGON/THRUST	ROGUE/DRAGON/THRUST	USER LOOSES	PASS

OK, THERE IS SO MANY COMBINATIONS, that it might take me a while to test all of them.
We need to do the above mentioned tests for each character time
Also we need to mix thrust and swung weapons and test those outcomes as well.

Reflection:
The main problem I had, the player was asked to make an attack when health was <0. Only after the player attacked the monster, a message would come up that the player is dead.



The problem was that I was entering battle with the following code.

```
while  (ActiveMonster.health>0 || Fighter.health>0){
                ActiveMonster.attack(Fighter);
                combatChoice(Fighter);
        }
```

As you can see this loop was entered with either fighters or monsters health was >0. When I replaced or with and, the problem was solved.
During the testing stage, I found out that when monsters or players health get to exactly 0, the game stops and no message is displayed. The problem was with the following code:

```
        if (ActiveMonster.health<0){
                cout <<endl<< "Congratulations! You killed the monster!" << endl;}
        if (Fighter.health<0){
                cout << "You are dead! You lost." << endl;
  }
```

Basically, there was a message of how the game resolved only when health <0, but when health was 0, there was no indication what to do. At the same time, the battle functions were entered only when health was >0:

```
while  (ActiveMonster.health>0 || Fighter.health>0){
                ActiveMonster.attack(Fighter);
```

```
            combatChoice(Fighter);
        }
```

Thus, when health was 0, there were no instructions what to do and program was ending.
Finally, most of my variables are public; I heard that it's a bad practice. At this point I have no idea why it's a bad practice, I hope Ill figure it out sometime. For the final step of this lab, I was planning to make my variables private and make setter/ getter function, but its already 11:14 pm and I do not have enough time to do that.
It just came to my mind that I never tested a combination at which both monster and fighter will run out of health. At this point I do not have any code that will resolve this situation.
As always, I've used information from different blogs online.
List of websites are provided in the code.