

Tatyana vlaskin
Final project

Design and Reflections:

Just to let you know that I wish I had at least one more month to work on this project.

Either I am not very smart or this is not a project that can be done in a week by a person who started learning C++ only last quarter. Maybe for the engineers and other computer savvy people, its easy, but for me it took really long time and its so far from being complete.

The major problem, I do not know how to put the character that stays in the battle at the end of the line. I worked mainly on implementation, so my design documents does not look good. Its better to follow the comments in the code to guide you whats going on.

1. First change to my code will be a new function CHARACTER eachPlayer. This is necessary because each team needs to choose multiple characters for their teams and we want to eliminate code redundancy. The function will look somewhat like this:

```
CHARECTOR eachPlayer(){
    Goblin g;
    Barbarian ba;
    ReptilePeople r;
    BlueMen bl;
    SaktiMan s;
    MegaMan m;
    CHARECTOR c;

    cout<< endl << "Choose your figher class. " << endl;
    cout<< "1. Golbin\n2. Barbarian\n3. Reprile\n4. Blue Man\n5. Sakti Man\n6. Mega Man" <<
endl;
    int result;
    string entered="";
    fflush(stdin);//
    getline(cin,entered);
    result = integerCheck(entered, '1', '6');
    //choose for player
    switch(result){
        case 1:
            c=g;
            break;
        case 2:
            c =ba;
            break;
        case 3:
            c = r;
            break;
        case 4:
            c = bl;
            break;
        case 5:
            c=s;
            break;
```

```

        case 6:
            c=m;
            break;
    }
    return(c);
}

```

2. Next, there will be a class, let's call it TEAM:

```

class TEAM{
private:
    string name;
    std::vector<CHARECTOR> v; // THIS WILL HOLD CHARACTERS SELECTED BY THE TEAM
public:
    TEAM(){}
    TEAM(string n,int no){name=n;}
    //THIS FUNCTION WILL SET A VECTOR WITH THE TEAM ONCE THE CHARACTERS ARE SELECTED
    void setVec(std::vector<CHARECTOR> x){v=x;}
    //THIS FUNCTION WILL DISPLAY THE NAME OF THE TEAM, A VS B FOR SIMPLICITY
    string getName(){return(name);}
    //THIS FUNCTION WILL DISPLAY THE MEMEBERS IN REACH TEAM
    void yourTeam(){
        cout<<"\nName : "<<name;
        cout<<"\nCombat Team : ";
        for(int i=0;i<v.size();i++){
            cout<< "["<<v.at(i).getname()<<" ";
        }
        cout<<endl;
    }
    //THIS FUNCTION WILL RETURN A VERCOR WITH THE MEMEBRS IN THE TEAM
    std::vector<CHARECTOR> getVec(){return(v);}
};

```

3. At the beginning of the game, the user will be asked how many characters they want to have on their team. The user needs to specify a number that is greater than 1. NOTE THIS IS IMPORTANT, YOU NEED TO HAVE AT LEAST 2 CHARACTERS IN EACH TEAM.
4. Next, a vector will be declared: `std::vector<CHARECTOR> v;`
5. Next each team will choose their characters. First a team object will be declared: let's say TEAM team1. And then there will be a for loop:

```

        for(int j=0;j<num;j++){ // where NUM IS THE NUMBER OF CHARACTERS
            v.push_back(eachPlayer()); //team a players
        }

```
6. Team1 vector will be set using the `team1.setVec(v)` function.
7. Next step will be to clear the `v` vector and repeat the process for the team 2
8. To make sure that everything worked as expected and all players were selected correctly, both teams will be displayed suing the `yourTeam()` function for team1 and team2

9. At this point, 2 vectors will be declared and initialized to team1 and team2:

```
std::vector<CHARECTOR> firstTeam,sceoendTeam;
firstTeam=team1.getVec();
sceoendTeam=team2.getVec();
```

10. Next, we need an iterator: std::vector<CHARECTOR>::iterator it. This will allow us to loop through the vector of characters

11. Next, I will display, the statistics of each character in the team, by doing something like this:
- ```
for(int i=0;i<num;i++){
```

```
 cout<<endl<<firstTeam.at(i).getname()<<":
 "<<int(firstTeam.at(i).getstrength()*100/firstTeam.at(i).getstnstore())<<"%\t\t\t"<<sceoendTe
 am.at(i).getname()<<":
 "<<int(sceoendTeam.at(i).getstrength()*100/sceoendTeam.at(i).getstnstore())<<"%\n";
 cout<<firstTeam.at(i).getname()<<": Strength:
 "<<firstTeam.at(i).getstrength()<<"/"<<firstTeam.at(i).getstnstore()<<"\t\t\t"<<sceoendTeam.at(
 i).getname()<<":Strength:
 "<<sceoendTeam.at(i).getstrength()<<"/"<<sceoendTeam.at(i).getstnstore();
}
```

12. A new variable will be declared. Lets call it each. This variable will be used to determine which team is attacking first. Odd turns – team A will attack first. When turns are even- team A will attack first.

13. Once the teams are set, we can start the battle. The players at the beginning of the line will start the battle: if(each==0){index1=0;index2=0;}

And the first player will be assigned as:

```
player=firstTeam.at(index1); // THIS WILL BE THE PLAYER FROM TEAM A
enemy=sceoendTeam.at(index2);//THIS WILL BE THE PLAYER FROM TEAM b
```

14. Next, 2 temporary variables will be introduced- int tmp1=index1,tmp2=index2; These two variables will be used, to alter who is attacking. The each variable described above, it used to decide who attacks first. However, then we need to reverse attack/defense actions of the characters. It's hard to explain. I'll better give an example. Lets say we have goblin on team a and barbarian on team b. the game starts, so each variable is odd. Thus, goblin will attack and barbarian will defend. Next, while VARIABLE EACH IS STILL ODD- barbarian need to attack and goblin need to defend. This will be done using 2 temp variables and the following else statement. I hope it makes sense:

There will be an else statement:

```
 else{
 int tmp1=index1,tmp2=index2;
 index1=tmp2;//index for altering enemy as player from recent fight
 index2=tmp1;//index for altering player as enemy from recent fight
 player=sceoendTeam.at(index1);
 enemy=firstTeam.at(index2);
 }
```

15. Nothing will be changed in the attack/defense functions; everything is kept from the previous games. The spinach will exist, the enchanted weapon will exist. However, I got rid of the poisons vials just to simplify the code. According to the professor the point of the assignment is to check out understanding of the containers, but not test our design abilities of the combat game.

16. Once the character was attacked and defended themselves, we need to change their statistics in the vector. This will be done with the following code. Please note, because I am reusing the code from the previous programs, ENEMY IS THE CHARACTER WHO IS DEFENDING.

```
//store in constant iteration to traverse from beginning
it=v.begin();
if(index2==0){
 v.erase(it);//delete old content from beginning
 v.insert(it,enemy);//add new content at beginning
}
```

17. So whoever was in front of the lines in team A and team B will fight until one of the characters is out. Whoever is out, will go to the new container. I decided that I will use list:  
std::list<CHARECTOR> lt; As for the characters that won, it will be placed back in the appropriate line. I should say that this part of the assignment is confusing, but I do not have time to check with professor or TA to get clarification.

According to the instructions:

You may not be able to just use the winner of the last combat and the top two fighters on the loser pile. AND THEN IT STATES: Basically, the last 3 creatures to exit are in first, second, and third place. I BELIEVE THAT THESE 2 SENTENCES CONTRADICT EACH OTHER.

IT ALSO STATES THAT: It's possible the last 3 or more fighters are from the same team. THE WAY MY GAME IS DESIGNED ITS NOT POSSIBLE FOR LAST 3 FIGHTERS TO BE ON THE SAME TEAM. I THINK THAT I DID SOMETHING WRONG, I hope I will not get 0 on this project. Unfortunately, I do not have time to check with TA or professor.

So, I really link this sentence in the assignment: Basically, the last 3 creatures to exit are in first, second, and third place and that's exactly what I will do with the following code:

```
// if this charector exist between top three add in list
if((firstTeam.size()+secondTeam.size())<=3){
 lt.push_back(v.at(index2));
 if(each%2==0){
 teamName.push_back('b');
 } // add corresponding team information
 else{
 teamName.push_back('a');
 }
}
```

Basically what happens is I sum the size of the team A and team B, if its equal to 3 or less they will go to my list and they will be assigned places 1-3. I hope that I will not fail this class because I decided to do it like that. The instructions are contradictory or maybe I just did not get instructions right.

18. The problem is that what if 2 characters are left on team lets say A and the winning team. Its straightforward that the 3<sup>rd</sup> place goes who whoever was added to the "loser container last". However, to decide which character on the team A get 1<sup>st</sup> place and which character will get 2<sup>nd</sup> place is tricky. I will use the strength points to determine who will get 1<sup>st</sup> and 2<sup>nd</sup> place. This will be done using the following code:

```
while(v.size())>0){
 it=v.begin();
 float max=0;//use for minimum strength charector in remaining winner players
 int index=0;
 // find top strength charector
 for(int i=0;i<v.size();i++){
 float tmp=v.at(i).getstrength()/float(v.at(i).getstrength());
```

```

 if(tmp>max){max=tmp;index=i;}
 }
 lt.push_back(v.at(index)); //add in list
 if(each%2==0){teamName.push_back('a');}
 else{teamName.push_back('b');}
 if(v.size()==1){v.erase(it);} // after add delete
 else{
 if(index==0){v.erase(it);}
 else{v.erase(it+index);}
 }
}
}

```

19. Finally, at the end of the game, result about the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> place will be displayed using the following code:

```

 cout<<"\nDisplaying Result : \n\n";
 lt.reverse();
 std::list<CHARECTOR>::iterator it;
 it=lt.begin();
 int size=lt.size();
 // top 3 players
 for(int i=0;i<size;i++){
 cout<<(i+1)<<" : position is "<<it->getname()<<" in team "<<teamName.at(size-i-
1)<<endl;
 it++;
 }

```

#### Testing:

I made sure that whoever is chosen for characters end up at the appropriate team.

Also I made sure that the character what loses is removed to the loser container

I tried to play different character combinations- see testing from the previous assignments.

Also, I the correct information about the winning team is displayed and that the correct information about the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> places are displayed.

I did not have that much time to test the code. It compiles and it works WHEN YOU CHOOSE 2 CHARACTERS.

WHEN YOU CHOOSE 3 CHARCTERS IT CRASHES TOWARDS THE END OF THE TOURNAMENT BECAUSE OF THE RUNTIME ERROR.

Please test it with 2 characters to check that it works and then test it with 3 or more characters to see how it crashes due to the runtime error.