Lab Report: Lab 9-Lets Make a Game

Jose Romero

04/29/2019

|  |  |
| --- | --- |
| Section | Pg. # |
| **Objective/Purpose of the program** | 2 |
| **Analysis/Design** | 2 |
| **Testing/Results** | 4 |
| **Conclusions / Lessons Learned** | 21 |
| Code: main | 22 |
| Code: Novice | 48 |
| Code: Ninja | 50 |
| Code: Knight | 53 |
| Code: Wizard | 56 |
| Code: Monster | 59 |
| Code: Dragon | 61 |

**Objective/Purpose of the program**

The purpose of this project is to make a game that implements different ideas about classes. We explore using inheritance mostly. In this project, I attempted to make an action/story game.

This story is mostly about the new Novice adventurer that goes to GRIEFTOWN to complete a small mission of battling some small monsters and different bandits to defend the town. But he gets to meet the bandits and they join his group. The then tell him the reason why they were attacking the town. The final boss will be defeating the groups leader’s dragon.

**Analysis/Design**

This game needs a base class (Character) and 6 branching classes based on Character.

* Character
  + Protected variables (6 of them)  
     std::string name;   
     int level;  
     int HP;  
     int XP; // everyone starts at 0  
     int NLP;  
     int maxHP; // same as starting HP
  + public:  
     Character(); //default constructor   
     Character(std::string, int, int, int, int);//parameterized constructor   
     virtual int getAttack() = 0; //pure virtual method to return damage character makes  
     virtual void print() = 0; //what should appear to show the character   
     virtual bool isDead(int) = 0; //reduces HP and returns bool true if HP <= 0  
     virtual void updateXP(int) = 0; //takes in XP and adds. IF XP >= NLP, level up.

virtual void heal(); //increase HP to a max of maxHP. Only differs with Dragon class, which it is why it is a virtual function.  
 void changeName(std::string); //change name  
 virtual bool lowHP(){} //if HP lower that a certain percentage(for forgiving)  
 virtual void collectWeapon(std::string) {} //make stronger by adding different points, depending on the character.  
 std::string getName() { return name;} //return name.

* Novice (one of the types of Characters)
  + Private
    - int luck; //base damage  
      int wpower; //extra damage  
      std::string weapon; //name of weapon
  + Public
    - No new functions besides constructors
* Ninja
  + Private
    - int lethality; //base damage  
      std::string weapon; //name of weapon  
      int wPower; //extra damage
  + Public
    - No new functions besides constructors
* Knight
  + Private
    - std::string wName; //name of weapon  
      int wPower; //base power/extra  
      int armor; //reduced damage
  + Public
    - No new functions besides constructors
* Wizard
  + Private
    - std::string mName; //name of weapon  
      int mPower; //extra damge
  + Public
    - No new functions besides constructors
* Monster
  + Private
    - std::string type; //what type of monster (name)
  + Public
    - No new functions besides constructors
* Dragon
  + Private
    - int scales; //different HP levels.  
      int maxScales; //maximum
  + Public
    - No new functions besides constructors

All functions are already in the character class. By making them virtual, it gave me a lot of flexibilities. Allowing me to make vertors of type Character pointers, which could store the address of any type of character, and call functions only named in the Character class. That is why Character class has all functions.

**Testing/Results**

This project required lots and lots and lots of testing. In order to test the classes, I would need to create objects and make sure each function worked. Something like this:

Wizard w1;  
Novice n1("jose");  
Knight k1("ires", 13);  
n1.getAttack();  
k1.getAttack();  
w1.getAttack();  
n1.print();  
n1.isDead(15);  
n1.updateXP(15);  
n1.print();  
k1.print();  
k1.isDead(15);  
k1.updateXP(15);  
k1.print();  
k1.getAttack();  
w1.print();  
w1.isDead(15);  
w1.updateXP(15);  
w1.print();

Output:

jose has dealt 11 damage!

ires has dealt 13 damage!

wizard has dealt 13 damage!

LVL 1 NOVICE: jose HP: 80/80

XP: 0/10

LUCK: 1

LVL 2 NOVICE: jose HP: 75/90

XP: 5/15

LUCK: 5

LVL 14 Knight: ires HP: 367/367

XP: 0/101

ARMOR: 14

LVL 14 Knight: ires HP: 366/367

XP: 15/101

ARMOR: 14

ires has dealt 12 damage!

LVL 1 Wizard: wizard HP: 100/100

XP: 0/10

MAGIC POWER: 0

LVL 2 Wizard: wizard HP: 93/108

XP: 5/15

MAGIC POWER: 3

But what I had to test the most, were the actual functions in main (and the use of functions with in main).

Main is just 1 big while loop that wont stop looping till you quit or beat the game. while (action != 5 && !finishedGame)

Inside it, there are options for each thing to do in town, and inside most (not all) are more menus with different things to do in each one.

action = inTownMenu();  
  
while (action != 5 && !finishedGame) {  
 if (action == 1) {  
 home(party);  
 } else if (action == 2) {  
 int exploringChoice;  
 exploringChoice = forest();  
 while (exploringChoice != 4 && !finishedGame) {  
 if (exploringChoice == 1)  
 jungle(party, ninjaPtr, ninjaDefeated);  
 else if (exploringChoice == 2)  
 cave(party, knightPtr, knightDefeated);  
 else if (exploringChoice == 3)  
 desert(party, wizardPtr, wizardDefeated);  
 else if (exploringChoice == 666)  
 fightFinalBoss(party, dragPtr);  
  
 cout << "Please enter a number between 1-4!\n";  
 if (!finishedGame) exploringChoice = forest();  
 }  
 } else if (action == 3)  
 shop(party);  
 else if (action == 4)  
 {  
 if (gold >= 10) {  
 cout << "You got a 10 g drink and left\n";  
 gold -=10;  
 } else  
 cout << "You dont even have enough money to get a drink;\n";  
 }  
  
 if (ninjaDefeated && knightDefeated && wizardDefeated && !ready)  
 {  
 /\* ending text \*/   
 ready = true;  
 }  
  
 if (!finishedGame)  
 action = inTownMenu();  
}

output:

-----GRIEFTOWN-----

1. Home (gold = 10 g)

2. DARK TERRITORY

3. Shop

4. Bar

5. Quit.

What would you like to do?(enter number): 1

Ruben: This is your house for now, it's not much but there is a:

1. Bed (to restore HP)

2. Door (to go back to town)

what did you want to do?(enter number): 1

You are all healed now

LVL 1 NOVICE: Jose Romero HP: 80/80

XP: 0/10

LUCK: 1

Come back when you are feeling tired!

-----GRIEFTOWN-----

1. Home (gold = 10 g)

2. DARK TERRITORY

3. Shop

4. Bar

5. Quit.

What would you like to do?(enter number): 2

-----DARK TERRITORY-----

1. Explore Jungle (level 1-5)

2. Explore Cave (level 6-10)

3. Explore Dessert (level 11-15)

4. GRIEFTOWN

What would you like to do?(enter number): 1

-----JUNGLE-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 1

HEROES:

LVL 1 NOVICE: Jose Romero HP: 80/80

XP: 0/10

LUCK: 1

ENEMY:

LVL 5 MONSTER: Bat

HP: 24/24

1. Fight

2. Escape

What would you like to do?(enter number): 1

Jose Romero has dealt 3 damage!

Bat has dealt 9 damage!

HEROES:

LVL 1 NOVICE: Jose Romero HP: 71/80

XP: 0/10

LUCK: 1

MONSTER:

LVL 5 MONSTER: Bat

HP: 21/24

would you like to continue? [Y/N]y

Jose Romero has dealt 10 damage!

Bat has dealt 10 damage!

HEROES:

LVL 1 NOVICE: Jose Romero HP: 61/80

XP: 0/10

LUCK: 1

MONSTER:

LVL 5 MONSTER: Bat

HP: 11/24

would you like to continue? [Y/N]y

Jose Romero has dealt 5 damage!

Bat has dealt 10 damage!

HEROES:

LVL 1 NOVICE: Jose Romero HP: 51/80

XP: 0/10

LUCK: 1

MONSTER:

LVL 5 MONSTER: Bat

HP: 6/24

would you like to continue? [Y/N]y

Jose Romero has dealt 6 damage!

YOU HAVE DEFEATED THE ENEMY!

You have earned 100 g

-----JUNGLE-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 2

ヽ(｀⌒´メ)ノ

HMMMMMMMM so you are the new Novice trying save the town, there's nothing you can do kid.

LVL 2 NOVICE: Jose Romero HP: 61/90

XP: 0/15

LUCK: 5

ENEMY:

LVL 6 Ninja: BANDIT HP: 160/160

XP: 0/45

LETHALITY: 11

1. Fight

2. Escape

What would you like to do?(enter number): 2

You have escaped!

-----JUNGLE-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 3

Please enter a number between 1-4!

-----DARK TERRITORY-----

1. Explore Jungle (level 1-5)

2. Explore Cave (level 6-10)

3. Explore Dessert (level 11-15)

4. GRIEFTOWN

What would you like to do?(enter number): 2

-----CAVE-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 1

HEROES:

LVL 2 NOVICE: Jose Romero HP: 61/90

XP: 0/15

LUCK: 5

ENEMY:

LVL 10 MONSTER: Goblin

HP: 71/71

1. Fight

2. Escape

What would you like to do?(enter number): 2

You have escaped!

-----CAVE-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 2

(๑•̀ㅂ•́)و

Hello . . . It looks like you are looking for trouble.

I guess I will just have to kick your ass since that ninja hasn't.

LVL 2 NOVICE: Jose Romero HP: 61/90

XP: 0/15

LUCK: 5

ENEMY:

LVL 13 Knight: BANDIT HP: 348/348

XP: 0/94

ARMOR: 13

1. Fight

2. Escape

What would you like to do?(enter number): 2

You have escaped!

-----CAVE-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 3

Please enter a number between 1-4!

-----DARK TERRITORY-----

1. Explore Jungle (level 1-5)

2. Explore Cave (level 6-10)

3. Explore Dessert (level 11-15)

4. GRIEFTOWN

What would you like to do?(enter number): 3

-----DESERT-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 1

HEROES:

LVL 2 NOVICE: Jose Romero HP: 61/90

XP: 0/15

LUCK: 5

ENEMY:

LVL 12 MONSTER: Golem

HP: 120/120

1. Fight

2. Escape

What would you like to do?(enter number): 2

You have escaped!

-----DESERT-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 2

｡;+\*(★`∪´☆)\*+;｡

Who has come to see my performance?

Today's show: kill the audience

LVL 2 NOVICE: Jose Romero HP: 61/90

XP: 0/15

LUCK: 5

ENEMY:

LVL 15 Wizard: BANDIT HP: 212/212

XP: 0/94

MAGIC POWER: 42

Fight

2. Escape

What would you like to do?(enter number): 2

You have escaped!

-----DESERT-----

1. Fight Monster

2. Fight Boss

3. Go Back

What would you like to do?(enter number): 3

Please enter a number between 1-4!

-----DARK TERRITORY-----

1. Explore Jungle (level 1-5)

2. Explore Cave (level 6-10)

3. Explore Dessert (level 11-15)

4. GRIEFTOWN

What would you like to do?(enter number): 4

-----GRIEFTOWN-----

1. Home (gold = 110 g)

2. DARK TERRITORY

3. Shop

4. Bar

5. Quit.

What would you like to do?(enter number): 3

-----SHOP----- (gold = 110 g)

1. 4-Leaf Clover (view)

2. Shurikens (view)

3. Long Sword (view)

4. Magic Staff (view)

5. GRIEFTOWN

What would you like to do?(enter number): 1

4-Leaf Clover:

This item grants the Novice an additional 15 luck points.

cost: 1000 g

Would you like to purchase?[Y/N]y

Please come back when you have enough gold to buy it.

You just need 890 g more

-----SHOP----- (gold = 110 g)

1. 4-Leaf Clover (view)

2. Shurikens (view)

3. Long Sword (view)

4. Magic Staff (view)

5. GRIEFTOWN

What would you like to do?(enter number): 1

4-Leaf Clover:

This item grants the Novice an additional 15 luck points.

cost: 1000 g

Would you like to purchase?[Y/N]n

-----SHOP----- (gold = 110 g)

1. 4-Leaf Clover (view)

2. Shurikens (view)

3. Long Sword (view)

4. Magic Staff (view)

5. GRIEFTOWN

What would you like to do?(enter number): 2

Shurikens:

This item grants the Ninja an additional 20 lethality points.

cost: 1500 g

Would you like to purchase?[Y/N]y

Please come back when you have enough gold to buy it.

You just need 1390 g more

-----SHOP----- (gold = 110 g)

1. 4-Leaf Clover (view)

2. Shurikens (view)

3. Long Sword (view)

4. Magic Staff (view)

5. GRIEFTOWN

What would you like to do?(enter number): 5

-----GRIEFTOWN-----

1. Home (gold = 110 g)

2. DARK TERRITORY

3. Shop

4. Bar

5. Quit.

What would you like to do?(enter number): 4

You got a 10 g drink and left

-----GRIEFTOWN-----

1. Home (gold = 100 g)

2. DARK TERRITORY

3. Shop

4. Bar

5. Quit.

What would you like to do?(enter number): 5

Thank you for playing!

To test the party system, I need to beat/forgive each boss for them to join my party.

CODE:

while(!meDead && !enDead && (toupper(keepGoing[0]) == 'Y') && !forgiven)  
{  
 for (Character \*p : heroes)  
 enDead = ninja->isDead(p->getAttack()); //exits  
 if (!enDead)  
 meDead = heroes[0]->isDead(ninja->getAttack());  
 if (!meDead)  
 {  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 ninja->print();  
 if (ninja->lowHP()){  
 cout << "┗|｀O´|┛ \nIT HAS BEEN A WHILE SINCE I HAD SUCH A GOOD MATCH.\nYOU ARE GETTING THE BEST OF ME!\n\n";  
 cout << "would you like to forgive her? [Y/N]";  
 cin >> forg;  
 if (toupper(forg[0]) == 'Y')  
 {  
 cout << "(╭ರ\_•́)\nWhy are you doing this? I cannot let you do that my boss will kill me\n"  
 "That means I have to join your party!ヾ(@^▽^@)ノ\n But i dont have a name ε٩( ºωº )۶з\n\n";  
 string tmpName;  
 cout << "So what do you want me to be called? ";  
 cin >> tmpName;  
 ninja->changeName(tmpName);  
 cout << "i love it ٩(♡ε♡ )۶\n";  
 forgiven = true;  
 d = true;  
 heroes.push\_back(ninja);  
 break;  
 }  
 }  
 else {  
 cout << "Would you like to continue? [Y/N]";  
 cin >> keepGoing;  
 }  
 }  
}

OUTPUT:

HEROES:

LVL 5 NOVICE: JOSE ROMERO HP: 50/120

XP: 22/30

LUCK: 17 + 15

ENEMY:

LVL 6 Ninja: BANDIT HP: 48/160

XP: 0/45

LETHALITY: 11

Would you like to continue? [Y/N]Y

JOSE ROMERO has dealt 36 damage!

BANDIT has dealt 19 damage!

HEROES:

LVL 5 NOVICE: JOSE ROMERO HP: 31/120

XP: 22/30

LUCK: 17 + 15

ENEMY:

LVL 6 Ninja: BANDIT HP: 12/160

XP: 0/45

LETHALITY: 11

┗|｀O´|┛

IT HAS BEEN A WHILE SINCE I HAD SUCH A GOOD MATCH.

YOU ARE GETTING THE BEST OF ME!

would you like to forgive her? [Y/N]Y

(╭ರ\_•́)

Why are you doing this? I cannot let you do that my boss will kill me

That means I have to join your party!ヾ(@^▽^@)ノ

But i dont have a name ε٩( ºωº )۶з

So what do you want me to be called? RONALD

i love it ٩(♡ε♡ )۶

-----JUNGLE-----

1. Fight Monster

3. Go Back

What would you like to do?(enter number): 3

Please enter a number between 1-4!

-----DARK TERRITORY-----

1. Explore Jungle (level 1-5)

2. Explore Cave (level 6-10)

3. Explore Dessert (level 11-15)

4. GRIEFTOWN

What would you like to do?(enter number): 4

-----GRIEFTOWN-----

1. Home (gold = 920 g)

2. DARK TERRITORY

3. Shop

4. Bar

5. Quit.

What would you like to do?(enter number): 1

Ruben: This is your house for now, it's not much but there is a:

1. Bed (to restore HP)

2. Door (to go back to town)

what did you want to do?(enter number): 1

You are all healed now

LVL 5 NOVICE: JOSE ROMERO HP: 120/120

XP: 22/30

LUCK: 17 + 15

LVL 6 Ninja: RONALD HP: 160/160

XP: 0/45

LETHALITY: 11

This shows the process in which a new Characte joins the party by forgiveness, and you type a name. However you can also kill them and have the zombie join your party:

if (enDead && !forgiven)  
{  
 for (Character \*p : heroes)  
 p->updateXP(30);  
 cout << "\n YOU HAVE DEFEATED THE ENEMY! \n";  
 cout << "\n\n NOW THE ZOMBIE NINJA IS IN YOUR TEAM";  
 ninja->changeName("ZOMBIE NINJA");  
 forgiven = true;  
 d = true;  
 heroes.push\_back(ninja);  
  
}

Output:

Would you like to continue? [Y/N]Y

JOSE ROMERO has dealt 38 damage!

HEROES:

LVL 6 NOVICE: JOSE ROMERO HP: 64/130

XP: 32/35

LUCK: 21 + 15

ENEMY:

LVL 6 Ninja: BANDIT HP: -1/160

XP: 0/45

LETHALITY: 11

┗|｀O´|┛

IT HAS BEEN A WHILE SINCE I HAD SUCH A GOOD MATCH.

YOU ARE GETTING THE BEST OF ME!

would you like to forgive her? [Y/N]N

YOU HAVE DEFEATED THE ENEMY!

NOW THE ZOMBIE NINJA IS IN YOUR TEAM

-----JUNGLE-----

1. Fight Monster

3. Go Back

What would you like to do?(enter number): 3

Please enter a number between 1-4!

-----DARK TERRITORY-----

1. Explore Jungle (level 1-5)

2. Explore Cave (level 6-10)

3. Explore Dessert (level 11-15)

4. GRIEFTOWN

What would you like to do?(enter number): 4

-----GRIEFTOWN-----

1. Home (gold = 1320 g)

2. DARK TERRITORY

3. Shop

4. Bar

5. Quit.

What would you like to do?(enter number): 1

Ruben: This is your house for now, it's not much but there is a:

1. Bed (to restore HP)

2. Door (to go back to town)

what did you want to do?(enter number): 1

You are all healed now

LVL 7 NOVICE: JOSE ROMERO HP: 140/140

XP: 27/40

LUCK: 25 + 15

LVL 6 Ninja: ZOMBIE NINJA HP: 160/160

XP: 0/45

LETHALITY: 11

I NEED THEM TO GET ALL THREE CHARACTERS BEFORE FIGHTING THE FINAL BOSS.

FINAL BOSS:

CODE:

void fightFinalBoss(vector<Character\*>& heroes, Character\* dragon)  
{  
 string ans;  
 string keepGoing = "Y";  
 string forg;  
 bool meDead = false;  
 bool enDead = false;  
 int counter = 0;  
 int focus = 0;  
  
 cout << "┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼███\n"  
 "┼┼┼┼┼┼┼┼┼┼┼┼┼███████\n"  
 "┼┼┼┼┼┼┼┼┼┼██████████████\n"  
 "┼┼┼┼┼┼┼████████████████████\n"  
 "┼┼┼┼██████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "██████████████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼████████████████████████████████\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼████████████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼████████████▒▒▒░░▒▒▒████████████\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼████████████▒▒▒▒▒▒▒▒████████████\n"  
 "┼████████████▒▒▒▒▒▒▒▒████████████\n"  
 "┼████████████▓▓▓▓▓▓▓▓████████████\n"  
 "████████████▓▓▓▓▓▓▓▓▓▓████████████\n"  
 "███████████▓▓▓▓▓▓▓▓▓▓▓▓███████████"  
 "\nPress 1 and Enter to continue . . . ";  
 cin >> ans;  
 cout << "\n\nWell, that a nice house! Let's go see who is this man!\n"  
 "1. Go knock on door.\n"  
 "2. Come back later.\n"  
 "What would you like to do? ";  
 cin >> ans;  
 while (ans != "2" && !finishedGame) {  
 if (ans == "1") {  
 cout << "\n(҂⌣̀\_⌣́)\nNicole II: It looks like someone has come to visit.\n"  
 "Nicole II: Perhaps they dare to challange me.\n";  
 cout << "\n|ʘ‿ʘ)╯\nRuben: Well you guys. Good luck! But I have to go do something important to do. \n";  
 cout << "\n╰༼=ಠਊಠ=༽╯\nNicole II: Where do you think you are going? \n"  
 "----------Giant Dragon is summoned!----------\n";  
 dragon->print();  
 cout << "Are you sure you would like to continue? [Y/N] ";  
 cin >> keepGoing;  
 if (toupper(keepGoing[0]) == 'Y')  
 cout << "Nicole II: I can't say I admire your stupidity, but okay. Time to fight!\n(⌐▨\_▨)\n";  
 else  
 cout << "(⌐▨\_▨)\nNicole II: Good try, but there is no going back now.\n";  
  
 while (!meDead && !enDead && !finishedGame) {  
 for (Character \*p : heroes)  
 enDead = dragon->isDead(p->getAttack()); //exits  
 if (!enDead) {  
 meDead = heroes[focus]->isDead(dragon->getAttack());  
 counter++;  
 if (counter % 3 == 0)  
 dragon->heal();  
 }  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 dragon->print();  
 if (!meDead) {  
 if (dragon->lowHP() && !enDead) {  
 cout << "(▼皿▼ﾒ)ﾉ\nDragon is almost dead!";  
 cout << "would you like to forgive her? [Y/N]";  
 cin >> forg;  
 if (toupper(forg[0]) == 'Y')  
 {  
 /\* Forgiving \*/  
 cout << "Щ(◣д◢)Щ\nNICOLE II: I dont need forgiveness.\n";  
 }  
 else  
 {  
 /\* escaping \*/  
 cout << "へ（>\_<へ)\nNICOLE II: What a cruel world.\n";  
 }  
 } else {  
 cout << "Would you like to continue? [Y/N]";  
 cin >> keepGoing;  
 if (toupper(keepGoing[0]) == 'Y')  
 {} else  
 cout << "\n(⌐▨\_▨)\nNicole II: Did you forget? you cannot escape!\n";  
 cout << "Who would you like the dragon to focus?\n(WHOEVER YOU CHOOSE, WILL GET HIT BY THE DRAGON NEXT, AND NO ONE CAN DIE!)\n"  
 "1. Novice\n"  
 "2. Ninja\n"  
 "3. Knight\n"  
 "4. Wizard\n"  
 "Choice (1, 2, 3, or 4): ";  
 cin >> focus;  
 while (focus < 1 || focus > 4) {  
 cout << "Please Choose 1, 2, 3, or 4 : ";  
 cin >> focus;  
 }  
 focus--;  
 }  
 }  
 if (enDead && !finishedGame)  
 {  
 for (Character \*p : heroes)  
 p->updateXP(30);  
 cout << "\n o(〃＾▽＾〃)o\nYOU HAVE DEFEATED THE ENEMY! \n";  
 cout << "\n\nRuben: Thank you so much now our town is called JOYTOWN again!!!\n";  
 finishedGame = true;  
 }  
 else if (meDead && !finishedGame)  
 {  
 cout << "c༽✖﹏✖༼ᓄ\nYou have been defeated! Go home and rest!\n";  
 cout << "PLEASE TRY AGAIN!";  
 finishedGame = true;  
 }  
 }  
 } else {  
 cout << "\nEnter '1' or '2'\nWell, that a nice house! Let's go see who is this man!\n"  
 "1. Go knock on door.\n"  
 "2. Come back later.\n"  
 "What would you like to do? ";  
 cin >> ans;  
 }  
 }  
}

SINCE I NEEDED TO TEST THIS ONE A LOT, I WOULD SIMPLY:

party.push\_back(new Ninja("NINJA", 15));  
party.push\_back(new Knight("BANDIT", 15));  
party.push\_back(new Wizard("BANDIT", 15));  
  
fightFinalBoss(party, new Dragon("Nicole's Pet", 1);

OVER AND OVER. TILL CODE WAS SMOOTH. USUALLY DRAGON WILL BE LVL 25 (NOT LVL 1).

OUTPUT:

┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼███

┼┼┼┼┼┼┼┼┼┼┼┼┼███████

┼┼┼┼┼┼┼┼┼┼██████████████

┼┼┼┼┼┼┼████████████████████

┼┼┼┼██████████████████████████

┼████████████████████████████████

██████████████████████████████████

┼████████████████████████████████

┼█░░░░██░░░░██████████░░░░██░░░░█

┼█░░░░██░░░░██████████░░░░██░░░░█

┼█░░░░██░░░░██████████░░░░██░░░░█

┼████████████████████████████████

┼█░░░░██░░░░██████████░░░░██░░░░█

┼█░░░░██░░░░██████████░░░░██░░░░█

┼█░░░░██░░░░██████████░░░░██░░░░█

┼████████████████████████████████

┼████████████████████████████████

┼████████████████████████████████

┼█░░░░██░░░░██████████░░░░██░░░░█

┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█

┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█

┼████████████▒▒▒░░▒▒▒████████████

┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█

┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█

┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█

┼████████████▒▒▒▒▒▒▒▒████████████

┼████████████▒▒▒▒▒▒▒▒████████████

┼████████████▓▓▓▓▓▓▓▓████████████

████████████▓▓▓▓▓▓▓▓▓▓████████████

███████████▓▓▓▓▓▓▓▓▓▓▓▓███████████

Press 1 and Enter to continue . . . 1

Well, that a nice house! Let's go see who is this man!

1. Go knock on door.

2. Come back later.

What would you like to do? 1

(҂⌣̀\_⌣́)

Nicole II: It looks like someone has come to visit.

Nicole II: Perhaps they dare to challange me.

|ʘ‿ʘ)╯

Ruben: Well you guys. Good luck! But I have to go do something important to do.

╰༼=ಠਊಠ=༽╯

Nicole II: Where do you think you are going?

----------Giant Dragon is summoned!----------

LVL 1 DRAGON: Nicole's Pet

HP: 180/180

SCALES: 25 / 25

Are you sure you would like to continue? [Y/N] N

(⌐▨\_▨)

Nicole II: Good try, but there is no going back now.

JOSE ROMERO has dealt 11 damage!

NINJA has dealt 45 damage!

BANDIT has dealt 11 damage!

BANDIT has dealt 58 damage!

Nicole's Pet has dealt 2 damage!

HEROES:

LVL 1 NOVICE: JOSE ROMERO HP: 78/80

XP: 0/10

LUCK: 1

LVL 16 Ninja: NINJA HP: 310/310

XP: 0/115

LETHALITY: 31

LVL 16 Knight: BANDIT HP: 405/405

XP: 0/115

ARMOR: 16

LVL 16 Wizard: BANDIT HP: 220/220

XP: 0/100

MAGIC POWER: 45

ENEMY:

LVL 1 DRAGON: Nicole's Pet

HP: 111/180

SCALES: 0 / 25

Would you like to continue? [Y/N]Y

Who would you like the dragon to focus?

(WHOEVER YOU CHOOSE, WILL GET HIT BY THE DRAGON NEXT, AND NO ONE CAN DIE!)

1. Novice

2. Ninja

3. Knight

4. Wizard

Choice (1, 2, 3, or 4): 3

JOSE ROMERO has dealt 11 damage!

NINJA has dealt 38 damage!

BANDIT has dealt 19 damage!

BANDIT has dealt 57 damage!

HEROES:

LVL 1 NOVICE: JOSE ROMERO HP: 78/80

XP: 0/10

LUCK: 1

LVL 16 Ninja: NINJA HP: 310/310

XP: 0/115

LETHALITY: 31

LVL 16 Knight: BANDIT HP: 405/405

XP: 0/115

ARMOR: 16

LVL 16 Wizard: BANDIT HP: 220/220

XP: 0/100

MAGIC POWER: 45

ENEMY:

LVL 1 DRAGON: Nicole's Pet

HP: -14/180

SCALES: 0 / 25

Would you like to continue? [Y/N]Y

Who would you like the dragon to focus?

(WHOEVER YOU CHOOSE, WILL GET HIT BY THE DRAGON NEXT, AND NO ONE CAN DIE!)

1. Novice

2. Ninja

3. Knight

4. Wizard

Choice (1, 2, 3, or 4): 3

o(〃＾▽＾〃)o

YOU HAVE DEFEATED THE ENEMY!

Ruben: Thank you so much now our town is called JOYTOWN again!!

**Conclusions / Lessons Learned**

1. Hardships:
   1. Balancing the enemies lvl, hp and everything. (it is still not very balance but I made it quicker to finish)
   2. Rand() function.
      1. I learned that I only have to set the seed once, to randomize all rand() functions in all the classes that are called
   3. Using global variables are useful
2. What I wished to improve
   1. I wanted to make a struct with the party, the gold, and everything at once but just having a global var gold was easier.
   2. Make an easier/quicker-to-read print() function bc it gets messy when you print 5 characters all at once (like in the final battle).
   3. collectWeapon function that also passes the value of the increase in abilities.
   4. A way to switch and store weapons
   5. A better battling system were the enemy attacks a member of the party in randomly (I tried to do this, but party.size() would bug out. So I changed the battling system)

Code:

Main.cpp:

#include <iostream>  
#include <stdlib.h> /\* srand, rand \*/  
#include <time.h> /\* time \*/  
#include <vector>  
#include <ctype.h> /\* toupper \*/  
#include "Character.h"  
#include "Novice.h"  
#include "Ninja.h"  
#include "Knight.h"  
#include "Wizard.h"  
#include "Monster.h"  
#include "Dragon.h"  
  
using namespace std;  
  
int gold = 10;  
bool ready = false;  
bool finishedGame = false;  
  
Character\* intro();  
int inTownMenu();  
void home(vector<Character\*>&);  
void shop(vector<Character\*>&);  
int forest();  
void jungle(vector<Character\*>&, Character\*, bool&);  
void cave(vector<Character\*>&, Character\* , bool&);  
void desert(vector<Character\*>&, Character\* , bool&);  
bool fightMonster(vector<Character\*>&, Monster, int);  
void fightJungleBoss(vector<Character\*>&, Character\*, bool&);  
void fightCaveBoss(vector<Character\*>&, Character\*, bool&);  
void fightDesertBoss(vector<Character\*>&, Character\*, bool&);  
void fightFinalBoss(vector<Character\*>&, Character\*);  
  
  
  
  
int main() {  
 srand (time(**NULL**)); //set up seed for random functions  
  
 vector<Character\*> party;  
 party.push\_back(intro());  
 int action;  
 string dummyVar;  
  
 Character\* ninjaPtr = new Ninja("BANDIT", 4);  
 bool ninjaDefeated = false;  
 Character\* knightPtr = new Knight("BANDIT", 9);  
 bool knightDefeated = false;  
 Character\* wizardPtr = new Wizard("BANDIT", 14);  
 bool wizardDefeated = false;  
 Character\* dragPtr = new Dragon("Nicole's Pet", 25);  
   
  
 action = inTownMenu();  
  
 while (action != 5 && !finishedGame) {  
 if (action == 1) {  
 home(party);  
 } else if (action == 2) {  
 int exploringChoice;  
 exploringChoice = forest();  
 while (exploringChoice != 4 && !finishedGame) {  
 if (exploringChoice == 1)  
 jungle(party, ninjaPtr, ninjaDefeated);  
 else if (exploringChoice == 2)  
 cave(party, knightPtr, knightDefeated);  
 else if (exploringChoice == 3)  
 desert(party, wizardPtr, wizardDefeated);  
 else if (exploringChoice == 666)  
 fightFinalBoss(party, dragPtr);  
  
 cout << "Please enter a number between 1-4!\n";  
 if (!finishedGame) exploringChoice = forest();  
 }  
 } else if (action == 3)  
 shop(party);  
 else if (action == 4)  
 {  
 if (gold >= 10) {  
 cout << "You got a 10 g drink and left\n";  
 gold -=10;  
 } else  
 cout << "You dont even have enough money to get a drink;\n";  
 }  
  
 if (ninjaDefeated && knightDefeated && wizardDefeated && !ready)  
 {  
 cout << "\n\n\n\n\n\n\n\n" << party[1]->getName() << ": We need to tell you something!\n"  
 << party[2]->getName() << ": The reason why there have been so many monsters around the area ...\n"  
 << party[3]->getName() << ": Even the reason why we have been terrorizing the city ...\n"  
 << "Ruben: It is because of that man! ヾ( ･`⌓´･)ﾉﾞ\n"  
 << party[1]->getName() << ": Her name is Nicole II and she wants to take over the city.\n"  
 << party[2]->getName() << ": We need to do something. \n"  
 << party[3]->getName() << ": I believe that if we work together, we can beat her up!\n";  
 cout << "\n\n\nWhat do you think? [Y/N]";  
 cin >> dummyVar;  
 cout << "Ruben: (•́⌄•́๑)૭✧ Whatever. I know you can defeat her. Go to the Dark Territory, turn left and you'll see her house\n";  
 ready = true;  
 }  
  
 if (!finishedGame)  
 action = inTownMenu();  
 }  
  
 cout << "\nThank you for playing!\n";  
 return 0;  
}  
  
  
//int main()  
//{  
// Wizard w1;  
// Novice n1("jose");  
// Knight k1("ires", 13);  
// n1.getAttack();  
// k1.getAttack();  
// w1.getAttack();  
// n1.print();  
// n1.isDead(15);  
// n1.updateXP(15);  
// n1.print();  
// k1.print();  
// k1.isDead(15);  
// k1.updateXP(15);  
// k1.print();  
// k1.getAttack();  
// w1.print();  
// w1.isDead(15);  
// w1.updateXP(15);  
// w1.print();  
//}  
  
Character\* intro()  
{  
 string name;  
 string dummyVar;  
 cout << "Welcome to Grieftown, ヽ( ´ー`)ノ\n\n"  
 "You must be the new novice from the capital, my name is Ruben and I am the "  
 "Mayor here. \nWhat was your name again? ";  
 getline(cin, name);  
 cout << "\n"  
 "Well, I am glad you came, " << name << ". As you may already know, Griefland is a small \n"  
 "town out in the country full of bandits and monsters. However, we were not \n"  
 "always like this. We used to be called Joyland, but everything changed when \n"  
 "she came! ヾ( ･`⌓´･)ﾉﾞ\n"  
 "\n"  
 "Well, don’t worry about it. Just try to fight the monsters in the forest, and collect the gold, \nso you "  
 "can go back home. That’s what you came here for, after all.\n\n"  
 "Do you have any questions?[Y/N] ";  
 cin >> dummyVar;  
 cout << "Well, I am sure you'll be fine\n\n";  
 return new Novice(name);  
}  
  
int inTownMenu()  
{  
 int ans;  
 cout << "\t-----GRIEFTOWN-----\n"  
 "\n"  
 "1. Home\t\t\t\t\t\t(gold = " << gold << " g)\n"  
 "2. DARK TERRITORY\n"  
 "3. Shop\n"  
 "4. Bar \n"  
 "5. Quit.\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 return ans;  
}  
  
void home(vector<Character\*> &people)  
{  
 int ans;  
 cout << "\nRuben: This is your house for now, it's not much but there is a:\n"  
 "1. Bed (to restore HP)\n"  
 "2. Door (to go back to town)\n"  
 "what did you want to do?(enter number): ";  
 cin >> ans;  
 while (ans != 3) {  
 if (ans == 1) {  
 for (Character \*p : people)  
 p->heal();  
 cout << "You are all healed now\n\n";  
 for (Character \*p : people)  
 p->print();  
 cout << endl;  
 break;  
 }  
 else if(ans == 2)  
 break;  
 cout << "please choose 1 or 2.";  
 cin >> ans;  
 }  
 cout << "Come back when you are feeling tired!\n";  
}  
  
void shop(vector<Character\*>& heroes)  
{  
 int ans;  
 string buy = "N";  
 cout << "\t-----SHOP----- \t\t(gold = " << gold << " g)\n"  
 "\n"  
 "1. 4-Leaf Clover (view)\n"  
 "2. Shurikens (view)\n"  
 "3. Long Sword (view)\n"  
 "4. Magic Staff (view)\n"  
 "5. GRIEFTOWN\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 while (ans != 5)  
 {  
 if (ans == 1){  
 cout << "4-Leaf Clover: \n"  
 "\n\tThis item grants the Novice an additional \e[1m 15 luck points.\e[0m"  
 "\n\n \t\t\t cost: 1000 g\n"  
 "Would you like to purchase?[Y/N]";  
 cin >> buy;  
 if (toupper(buy[0]) == 'Y')  
 {  
 if (gold >= 1000)  
 {  
 gold -= 1000;  
 heroes[0]->collectWeapon("4-Leaf Clover");  
 cout << "CONGRATS ON YOUR PURCHASE\n";  
 heroes[0]->print();  
 buy = "N";  
 }  
 else {  
 cout << "Please come back when you have enough gold to buy it.\nYou just need "<< (1000 - gold) <<" g more\n";  
 buy = "N";  
 }  
 }  
 }  
 else if (ans == 2){  
 cout << "Shurikens: \n"  
 "\n\tThis item grants the Ninja an additional \e[1m 20 lethality points.\e[0m"  
 "\n\n \t\t\t cost: 1500 g\n"  
 "Would you like to purchase?[Y/N]";  
 cin >> buy;  
 if (toupper(buy[0]) == 'Y')  
 {  
 if (gold >= 1500)  
 {  
 gold -= 1500;  
 heroes[1]->collectWeapon("Shurikens");  
 cout << "CONGRATS ON YOUR PURCHASE\n";  
 heroes[1]->print();  
 buy = "N";  
 }  
 else {  
 cout << "Please come back when you have enough gold to buy it.\nYou just need "<< (1500 - gold) <<" g more\n";  
 buy = "N";  
 }  
 }  
 }  
 else if (ans == 3){  
 cout << "Long Sword: \n"  
 "\n\tThis item grants the Knight an additional \e[1m 20 Weapon Power.\e[0m"  
 "\n\n \t\t\t cost: 2000 g\n"  
 "Would you like to purchase?[Y/N]";  
 cin >> buy;  
 if (toupper(buy[0]) == 'Y')  
 {  
 if (gold >= 2000)  
 {  
 gold -= 2000;  
 heroes[2]->collectWeapon("Long Sword");  
 cout << "CONGRATS ON YOUR PURCHASE\n";  
 heroes[2]->print();  
 buy = "N";  
 }  
 else {  
 cout << "Please come back when you have enough gold to buy it.\nYou just need " << (2000 - gold)  
 << " g more\n";  
 buy = "N";  
 }  
 }  
 } else if (ans == 4){  
 cout << "Magic Staff: \n"  
 "\n\tThis item grants the Wizard an additional \e[1m 25 Magic Power.\e[0m"  
 "\n\n \t\t\t cost: 2500 g\n"  
 "Would you like to purchase?[Y/N]";  
 cin >> buy;  
 if (toupper(buy[0]) == 'Y')  
 {  
 if (gold >= 2500)  
 {  
 gold -= 2500;  
 heroes[3]->collectWeapon("Magic Staff");  
 cout << "CONGRATS ON YOUR PURCHASE\n";  
 heroes[3]->print();  
 buy = "N";  
 }  
 else {  
 cout << "Please come back when you have enough gold to buy it.\nYou just need "<< (2500 - gold) <<" g more\n";  
 buy = "N";  
 }  
 }  
 }  
 cout << "\t-----SHOP----- \t\t(gold = " << gold << " g)\n"  
 "\n"  
 "1. 4-Leaf Clover (view)\n"  
 "2. Shurikens (view)\n"  
 "3. Long Sword (view)\n"  
 "4. Magic Staff (view)\n"  
 "5. GRIEFTOWN\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 }  
}  
  
int forest()  
{  
 int ans;  
 cout << "\t-----DARK TERRITORY-----\n"  
 "\n"  
 "1. Explore Jungle (level 1-5)\n"  
 "2. Explore Cave (level 6-10)\n"  
 "3. Explore Dessert (level 11-15)\n"  
 "4. GRIEFTOWN\n";  
 if (ready) cout << "666. NICOLE'S MASSION IN THE HILL (CAREFUL! ONCE YOU ENTER, YOU CANNOT LEAVE)\n";  
 cout<< "\nWhat would you like to do?(enter number): ";  
 cin >> ans;  
 return ans;  
}  
  
void jungle(vector<Character\*> &people, Character\* enPtr, bool &d)  
{  
 int ans;  
 string enemies[5] = {"Bat", "Slime", "Spider", "Fox", "Jungle Tortoise"};  
 int minL = 1;  
  
 cout << "\n\t-----JUNGLE-----\n"  
 "\n"  
 "1. Fight Monster\n";  
 if(!d) cout << "2. Fight Boss\n";  
 cout << "3. Go Back\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
  
 while (ans != 3)  
 {  
 if (ans == 1)  
 {  
 int numberForEnemy = rand() % 5;  
 int level = minL + rand() % 5;  
 fightMonster(people, Monster(enemies[numberForEnemy], level), level);  
 }  
 else if (ans == 2 && !d)  
 fightJungleBoss(people, enPtr, d);  
 cout << "\n\t-----JUNGLE-----\n"  
 "\n"  
 "1. Fight Monster\n";  
 if(!d) cout << "2. Fight Boss\n";  
 cout << "3. Go Back\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 }  
  
  
}  
  
void cave(vector<Character\*> &people, Character\* enPtr, bool &d)  
{  
 string enemies[5] = {"Big Bat", "Big Slime", "Wolf", "large Spider", "Goblin"};  
 int ans;  
 int minL = 6;  
  
 cout << "\n\t-----CAVE-----\n"  
 "\n"  
 "1. Fight Monster\n";  
 if(!d) cout << "2. Fight Boss\n";  
 cout << "3. Go Back\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
  
 while (ans != 3) {  
 if (ans == 1) {  
 int numberForEnemy = rand() % 5;  
 int level = minL + rand() % 5;  
 fightMonster(people, Monster(enemies[numberForEnemy], level), level);  
 }  
 else if (ans == 2 && !d)  
 fightCaveBoss(people, enPtr, d);  
 cout << "\n\t-----CAVE-----\n"  
 "\n"  
 "1. Fight Monster\n";  
 if(!d) cout << "2. Fight Boss\n";  
 cout << "3. Go Back\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 }  
}  
  
void desert(vector<Character\*> &people, Character\* enPtr, bool &d) {  
 string enemies[5] = {"Mummy", "Paladin", "Ghoul", "Werewolf", "Golem"};  
 int ans;  
 int minL = 11;  
  
 cout << "\n\t-----DESERT-----\n"  
 "\n"  
 "1. Fight Monster\n";  
 if (!d) cout << "2. Fight Boss\n";  
 cout << "3. Go Back\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
  
 while (ans != 3) {  
 if (ans == 1) {  
 int numberForEnemy = rand() % 5;  
 int level = minL + rand() % 5;  
 fightMonster(people, Monster(enemies[numberForEnemy], level), level);  
 }  
 else if (ans == 2 && !d)  
 fightDesertBoss(people, enPtr, d);  
 cout << "\n\t-----DESERT-----\n"  
 "\n"  
 "1. Fight Monster\n";  
 if (!d) cout << "2. Fight Boss\n";  
 cout << "3. Go Back\n\n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 }  
}  
  
bool fightMonster(vector<Character\*>& heroes, Monster m, int l)  
{  
 int ans;  
 string keepGoing = "y";  
 bool meDead = false;  
 bool enDead = false;  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 m.print();  
  
 cout << "\n1. Fight \n"  
 "2. Escape \n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
  
 if (ans == 1)  
 {  
 while(!meDead && !enDead && toupper(keepGoing[0]) == 'Y')  
 {  
 for (Character \*p : heroes)  
 enDead = m.isDead(p->getAttack()); //exits  
 if (!enDead)  
 meDead = heroes[0]->isDead(m.getAttack());  
  
 if (!meDead && !enDead)  
 {  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nMONSTER:\n";  
 m.print();  
 cout << "would you like to continue? [Y/N]";  
 cin >> keepGoing;  
 }  
 }  
 if (enDead)  
 {  
 for (Character \*p : heroes)  
 p->updateXP(2\*l);  
 cout << "\n YOU HAVE DEFEATED THE ENEMY! \n"  
 "You have earned " << 20\*l << " g\n";  
 gold += 20\*l;  
 }  
 else if (meDead)  
 cout << "c༽✖﹏✖༼ᓄ\nYou have been defeated! Go home and rest!\n";  
 }  
 else if (ans == 2)  
 cout << "\nYou have escaped!\n";  
}  
  
void fightJungleBoss(vector<Character\*>& heroes, Character\* ninja, bool &d)  
{  
 int ans;  
 string keepGoing = "Y";  
 string forg;  
 bool forgiven = false;  
 bool meDead = false;  
 bool enDead = false;  
 cout << "\nヽ(｀⌒´メ)ノ\nHMMMMMMMM so you are the new Novice trying save the town, there's nothing you can do kid.\n\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 ninja->print();  
  
 cout << "\n1. Fight \n"  
 "2. Escape \n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 if (ans == 1)  
 {  
 while(!meDead && !enDead && (toupper(keepGoing[0]) == 'Y') && !forgiven)  
 {  
 for (Character \*p : heroes)  
 enDead = ninja->isDead(p->getAttack()); //exits  
 if (!enDead)  
 meDead = heroes[0]->isDead(ninja->getAttack());  
 if (!meDead)  
 {  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 ninja->print();  
 if (ninja->lowHP()){  
 cout << "┗|｀O´|┛ \nIT HAS BEEN A WHILE SINCE I HAD SUCH A GOOD MATCH.\nYOU ARE GETTING THE BEST OF ME!\n\n";  
 cout << "would you like to forgive her? [Y/N]";  
 cin >> forg;  
 if (toupper(forg[0]) == 'Y')  
 {  
 cout << "(╭ರ\_•́)\nWhy are you doing this? I cannot let you do that my boss will kill me\n"  
 "That means I have to join your party!ヾ(@^▽^@)ノ\n But i dont have a name ε٩( ºωº )۶з\n\n";  
 string tmpName;  
 cout << "So what do you want me to be called? ";  
 cin >> tmpName;  
 ninja->changeName(tmpName);  
 cout << "i love it ٩(♡ε♡ )۶\n";  
 forgiven = true;  
 d = true;  
 heroes.push\_back(ninja);  
 break;  
 }  
 }  
 else {  
 cout << "Would you like to continue? [Y/N]";  
 cin >> keepGoing;  
 }  
 }  
 }  
 if (enDead && !forgiven)  
 {  
 for (Character \*p : heroes)  
 p->updateXP(30);  
 cout << "\n YOU HAVE DEFEATED THE ENEMY! \n";  
 cout << "\n\n NOW THE ZOMBIE NINJA IS IN YOUR TEAM";  
 ninja->changeName("ZOMBIE NINJA");  
 forgiven = true;  
 d = true;  
 heroes.push\_back(ninja);  
  
 }  
 else if (meDead && !forgiven)  
 cout << "c༽✖﹏✖༼ᓄ\nYou have been defeated! Go home and rest!\n";  
 }  
 else if (ans == 2)  
 cout << "\nYou have escaped!\n";  
 ninja->heal();  
}  
  
void fightCaveBoss(vector<Character\*>& heroes, Character\* knight, bool &d)  
{  
 int ans;  
 string keepGoing = "Y";  
 string forg;  
 bool forgiven = false;  
 bool meDead = false;  
 bool enDead = false;  
 cout << "\n(๑•̀ㅂ•́)و\nHello . . . It looks like you are looking for trouble. \n"  
 "I guess I will just have to kick your ass since that ninja hasn't.\n\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 knight->print();  
  
 cout << "\n1. Fight \n"  
 "2. Escape \n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 if (ans == 1)  
 {  
 while(!meDead && !enDead && (toupper(keepGoing[0]) == 'Y') && !forgiven)  
 {  
 for (Character \*p : heroes)  
 enDead = knight->isDead(p->getAttack()); //exits  
 if (!enDead)  
 meDead = heroes[0]->isDead(knight->getAttack());  
 if (!meDead)  
 {  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 knight->print();  
 if (knight->lowHP()){  
 cout << "(ꐦ°᷄д°᷅) \nHOW CAN THIS BE POSSIBLE. YOU ARE JUST A NOVICE.\n"  
 "YOU SHOULD NOT BE ABLE TO EVEN MAKE ME TICKLE\n\n";  
 cout << "would you like to forgive him? [Y/N]";  
 cin >> forg;  
 if (toupper(forg[0]) == 'Y')  
 {  
 cout << "༼ つ ͠° ͟ ͟ʖ ͡° ༽つ\nDONT GET COCKY KID. \n<(｀^´)>\nTHIS ONLY HAPPENNED BC I DONT HAVE MY WEAPON\n"  
 "That means I have to join your party and give me a name!\n\n";  
 string tmpName;  
 cout << "So what do you want me to be called? ";  
 cin >> tmpName;  
 knight->changeName(tmpName);  
 forgiven = true;  
 d = true;  
 heroes.push\_back(knight);  
 break;  
 }  
 }  
 else {  
 cout << "Would you like to continue? [Y/N]";  
 cin >> keepGoing;  
 }  
 }  
 }  
 if (enDead && !forgiven)  
 {  
 for (Character \*p : heroes)  
 p->updateXP(30);  
 cout << "\n YOU HAVE DEFEATED THE ENEMY! \n";  
 cout << "\n\n NOW THE ZOMBIE NINJA IS IN YOUR TEAM";  
 knight->changeName("ZOMBIE KNIGHT");  
 heroes.push\_back(knight);  
 d = true;  
 }  
 else if (meDead && !forgiven)  
 cout << "c༽✖﹏✖༼ᓄ\nYou have been defeated! Go home and rest!\n";  
 }  
 else if (ans == 2)  
 {  
 cout << "\nYou have escaped!\n";  
 }  
 knight->heal();  
}  
  
void fightDesertBoss(vector<Character\*>& heroes, Character\* wizard, bool &d)  
{  
 int ans;  
 string keepGoing = "Y";  
 string forg;  
 bool forgiven = false;  
 bool meDead = false;  
 bool enDead = false;  
 cout << "\n｡;+\*(★`∪´☆)\*+;｡\nWho has come to see my performance?\n"  
 "Today's show: kill the audience\n\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 wizard->print();  
  
 cout << "\nFight \n"  
 "2. Escape \n"  
 "What would you like to do?(enter number): ";  
 cin >> ans;  
 if (ans == 1)  
 {  
 while(!meDead && !enDead && (toupper(keepGoing[0]) == 'Y') && !forgiven)  
 {  
 for (Character \*p : heroes)  
 enDead = wizard->isDead(p->getAttack()); //exits  
 if (!enDead)  
 meDead = heroes[0]->isDead(wizard->getAttack());  
 if (!meDead)  
 {  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 wizard->print();  
 if (wizard->lowHP()){  
 cout << "(ฅ⁍̴̀◊⁍̴́)و ̑̑ \nDAMN. HOW COULD THIS BE?\n"  
 "YOU SHOULD HAVE ALL BEEN DEAD FROM MY SECOND ATTACK!\n\n";  
 cout << "would you like to forgive her? [Y/N]";  
 cin >> forg;  
 if (toupper(forg[0]) == 'Y')  
 {  
 cout << "୧( ˵ ° ~ ° ˵ )୨ ̑̑\nALRIGHT ALRIGHT. \n\nI DONT LIKE MY BOSS. \nI'LL JUST JOIN YOU TILL I GET A WEAPON SO I CAN KILL YOU.\n"  
 "FOR NOW ... give me a name! ♡〜٩( ˃́▿˂̀ )۶〜♡\n\n";  
 string tmpName;  
 cout << "So what do you want me to be called? ";  
 cin >> tmpName;  
 wizard->changeName(tmpName);  
 forgiven = true;  
 d = true;  
 heroes.push\_back(wizard);  
 break;  
 }  
 }  
 else {  
 cout << "Would you like to continue? [Y/N]";  
 cin >> keepGoing;  
 }  
 }  
 }  
 if (enDead && !forgiven)  
 {  
 for (Character \*p : heroes)  
 p->updateXP(30);  
 cout << "\n YOU HAVE DEFEATED THE ENEMY! \n";  
 cout << "\n\n NOW THE ZOMBIE WIZARD IS IN YOUR TEAM";  
 wizard->changeName("ZOMBIE WIZARD");  
 d = true;  
 heroes.push\_back(wizard);  
 }  
 else if (meDead && !forgiven)  
 cout << "c༽✖﹏✖༼ᓄ\nYou have been defeated! Go home and rest!\n";  
 }  
 else if (ans == 2)  
 {  
 cout << "\nYou have escaped!\n";  
 }  
 wizard->heal();  
}  
  
void fightFinalBoss(vector<Character\*>& heroes, Character\* dragon)  
{  
 string ans;  
 string keepGoing = "Y";  
 string forg;  
 bool meDead = false;  
 bool enDead = false;  
 int counter = 0;  
 int focus = 0;  
  
 cout << "┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼███\n"  
 "┼┼┼┼┼┼┼┼┼┼┼┼┼███████\n"  
 "┼┼┼┼┼┼┼┼┼┼██████████████\n"  
 "┼┼┼┼┼┼┼████████████████████\n"  
 "┼┼┼┼██████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "██████████████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼████████████████████████████████\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼████████████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "┼████████████████████████████████\n"  
 "┼█░░░░██░░░░██████████░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼████████████▒▒▒░░▒▒▒████████████\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼█░░░░██░░░░█▒▒▒▒▒▒▒▒█░░░░██░░░░█\n"  
 "┼████████████▒▒▒▒▒▒▒▒████████████\n"  
 "┼████████████▒▒▒▒▒▒▒▒████████████\n"  
 "┼████████████▓▓▓▓▓▓▓▓████████████\n"  
 "████████████▓▓▓▓▓▓▓▓▓▓████████████\n"  
 "███████████▓▓▓▓▓▓▓▓▓▓▓▓███████████"  
 "\nPress 1 and Enter to continue . . . ";  
 cin >> ans;  
 cout << "\n\nWell, that a nice house! Let's go see who is this man!\n"  
 "1. Go knock on door.\n"  
 "2. Come back later.\n"  
 "What would you like to do? ";  
 cin >> ans;  
 while (ans != "2" && !finishedGame) {  
 if (ans == "1") {  
 cout << "\n(҂⌣̀\_⌣́)\nNicole II: It looks like someone has come to visit.\n"  
 "Nicole II: Perhaps they dare to challange me.\n";  
 cout << "\n|ʘ‿ʘ)╯\nRuben: Well you guys. Good luck! But I have to go do something important to do. \n";  
 cout << "\n╰༼=ಠਊಠ=༽╯\nNicole II: Where do you think you are going? \n"  
 "----------Giant Dragon is summoned!----------\n";  
 dragon->print();  
 cout << "Are you sure you would like to continue? [Y/N] ";  
 cin >> keepGoing;  
 if (toupper(keepGoing[0]) == 'Y')  
 cout << "Nicole II: I can't say I admire your stupidity, but okay. Time to fight!\n(⌐▨\_▨)\n";  
 else  
 cout << "(⌐▨\_▨)\nNicole II: Good try, but there is no going back now.\n";  
  
 while (!meDead && !enDead && !finishedGame) {  
 for (Character \*p : heroes)  
 enDead = dragon->isDead(p->getAttack()); //exits  
 if (!enDead) {  
 meDead = heroes[focus]->isDead(dragon->getAttack());  
 counter++;  
 if (counter % 3 == 0)  
 dragon->heal();  
 }  
 cout << "\nHEROES:\n";  
 for (Character \*p : heroes)  
 p->print();  
 cout << "\nENEMY:\n";  
 dragon->print();  
 if (!meDead) {  
 if (dragon->lowHP() && !enDead) {  
 cout << "(▼皿▼ﾒ)ﾉ\nDragon is almost dead!";  
 cout << "would you like to forgive her? [Y/N]";  
 cin >> forg;  
 if (toupper(forg[0]) == 'Y')  
 {  
 /\* Forgiving \*/  
 cout << "Щ(◣д◢)Щ\nNICOLE II: I dont need forgiveness.\n";  
 }  
 else  
 {  
 /\* escaping \*/  
 cout << "へ（>\_<へ)\nNICOLE II: What a cruel world.\n";  
 }  
 } else {  
 cout << "Would you like to continue? [Y/N]";  
 cin >> keepGoing;  
 if (toupper(keepGoing[0]) == 'Y')  
 {} else  
 cout << "\n(⌐▨\_▨)\nNicole II: Did you forget? you cannot escape!\n";  
 cout << "Who would you like the dragon to focus?\n(WHOEVER YOU CHOOSE, WILL GET HIT BY THE DRAGON NEXT, AND NO ONE CAN DIE!)\n"  
 "1. Novice\n"  
 "2. Ninja\n"  
 "3. Knight\n"  
 "4. Wizard\n"  
 "Choice (1, 2, 3, or 4): ";  
 cin >> focus;  
 while (focus < 1 || focus > 4) {  
 cout << "Please Choose 1, 2, 3, or 4 : ";  
 cin >> focus;  
 }  
 focus--;  
 }  
 }  
 if (enDead && !finishedGame)  
 {  
 for (Character \*p : heroes)  
 p->updateXP(30);  
 cout << "\n o(〃＾▽＾〃)o\nYOU HAVE DEFEATED THE ENEMY! \n";  
 cout << "\n\nRuben: Thank you so much now our town is called JOYTOWN again!!!\n";  
 finishedGame = true;  
 }  
 else if (meDead && !finishedGame)  
 {  
 cout << "c༽✖﹏✖༼ᓄ\nYou have been defeated! Go home and rest!\n";  
 cout << "PLEASE TRY AGAIN!";  
 finishedGame = true;  
 }  
 }  
 } else {  
 cout << "\nEnter '1' or '2'\nWell, that a nice house! Let's go see who is this man!\n"  
 "1. Go knock on door.\n"  
 "2. Come back later.\n"  
 "What would you like to do? ";  
 cin >> ans;  
 }  
 }  
}

Character.h

#ifndef **FIGHTINGGAME\_CHARACTER\_H**#define **FIGHTINGGAME\_CHARACTER\_H**#include <string>  
  
class Character  
 {  
protected:  
 std::string name;  
 int level;  
 int HP;  
 int XP; // everyone starts at 0  
 int NLP;  
 int maxHP; // same as starting HP  
public:  
 Character();  
 Character(std::string, int, int, int, int);  
 virtual int getAttack() = 0;  
 virtual void print() = 0;  
 virtual bool isDead(int) = 0;  
 virtual void updateXP(int) = 0;  
 virtual void heal();  
 void changeName(std::string);  
 virtual bool lowHP(){}  
 virtual void collectWeapon(std::string) {}  
 std::string getName() { return name;}  
 };  
  
#endif //FIGHTINGGAME\_CHARACTER\_H

Character.cpp

#include "Character.h"  
  
Character::Character()  
{  
 name = "none";  
 level = 1;  
 HP = 0;  
 XP = 0;  
 NLP = 10;  
 maxHP = 0;  
}  
  
Character::Character(std::string n, int hp, int xp, int nextLevelPoints, int m)  
{  
 name = n;  
 level = 1;  
 HP = hp;  
 XP = xp;  
 NLP = nextLevelPoints;  
 maxHP = m;  
}  
  
void Character::heal()  
{  
 HP = maxHP;  
}  
  
void Character::changeName(std::string newName)  
{  
 name = newName;

Novice.h

#ifndef **FIGHTINGGAME\_NOVICE\_H**#define **FIGHTINGGAME\_NOVICE\_H**#include <iostream>  
#include "Character.h"  
  
class Novice : public Character  
 {  
private:  
 int luck;  
 int wpower;  
 std::string weapon;  
public:  
 Novice();  
 Novice(std::string);  
 int getAttack();  
 void print();  
 bool isDead(int);  
 void updateXP(int);  
 void collectWeapon(std::string);  
 };  
  
#endif //FIGHTINGGAME\_NOVICE\_H

Novice.cpp

#include "Novice.h"  
  
Novice::Novice() : Character("novice", 80, 0, 10, 80)  
{  
// std::srand (time(NULL));  
 luck = std::rand() % 2;  
 std::cout << "luck : " << luck << std::endl;  
}  
  
Novice::Novice(std::string n) : Character(n, 80, 0, 10, 80)  
{  
// std::srand (time(NULL));  
 luck = std::rand() % 2;  
}  
  
int Novice::getAttack()  
{  
// std::srand (time(NULL));  
 int attack = 1 + std::rand() % 10 + luck + wpower;  
 std::cout << name << " has dealt " << attack << " damage!\n";  
 return attack;  
}  
  
void Novice::print()  
{  
 std::cout << "LVL " << level << " NOVICE: " << name << "\t";  
 std::cout << "HP: " << HP << "/" << maxHP << std::endl;  
 std::cout << "XP: " << XP << "/" << NLP << std::endl;  
 std::cout << "LUCK: " << luck;  
 if (wpower > 0) std::cout << " + " << wpower;  
 std::cout << std::endl;  
}  
  
bool Novice::isDead(int damage)  
{  
 HP -= damage;  
 return HP <= 0;  
}  
  
void Novice::updateXP(int monsterLevel)  
{  
 XP += monsterLevel;  
 if (XP >= NLP)  
 {  
 level++;  
 XP -= NLP;  
 NLP += 5;  
 maxHP += 10;  
 HP += 10;  
 luck += 3 + std::rand() % 3;  
 }  
}  
  
void Novice::collectWeapon(std::string n)  
{  
 weapon = n;  
 wpower += 15;  
}

Ninja.h

#ifndef **FIGHTINGGAME\_NINJA\_H**#define **FIGHTINGGAME\_NINJA\_H**#include <iostream>  
#include "Character.h"  
  
class Ninja : public Character  
{  
private:  
 int lethality;  
 std::string weapon;  
 int wPower;  
  
public:  
 Ninja();  
 Ninja(std::string, int);  
 int getAttack();  
 void print();  
 bool isDead(int);  
 void updateXP(int);  
 void collectWeapon(std::string);  
 bool lowHP();  
};  
  
  
#endif //FIGHTINGGAME\_NINJA\_H

Ninja.cpp

#include "Ninja.h"  
  
Ninja::Ninja() : Character("ninja", 85, 0, 10, 85)  
{  
 lethality = 1;  
 weapon = "none";  
}  
  
Ninja::Ninja(std::string name, int lvl) : Character(name, 85, 0, 10, 85)  
{  
 lethality = 1;  
 weapon = "none";  
 for (int i = 0; i < lvl; i++)  
 {  
 level++;  
 NLP += 7;  
 maxHP += 15;  
 HP += 15;  
 lethality += 2;  
 }  
}  
  
int Ninja::getAttack()  
{  
 int attack = 7 + std::rand() % 10 + lethality + wPower;  
 std::cout << name << " has dealt " << attack << " damage!\n";  
 return attack;  
}  
  
void Ninja::print()  
{  
 std::cout << "LVL " << level << " Ninja: " << name << "\t" ;  
 std::cout << "HP: " << HP << "/" << maxHP << std::endl;  
 std::cout << "XP: " << XP << "/" << NLP << std::endl;  
 std::cout << "LETHALITY: " << lethality;  
 if (weapon != "none") std::cout << " + " << wPower <<"\nWEAPON NAME: " << weapon;  
 std::cout << std::endl;  
}  
  
bool Ninja::isDead(int damage)  
{  
 HP -= damage;  
 return HP <= 0;  
}  
  
void Ninja::updateXP(int monsterLevel)  
{  
 XP += monsterLevel;  
 if (XP >= NLP)  
 {  
 level++;  
 XP -= NLP;  
 NLP += 5;  
 maxHP += 15;  
 HP += 15;  
 lethality += 2;  
 }  
}  
  
void Ninja::collectWeapon(std::string nameOfWeapon)  
{  
 weapon = nameOfWeapon;  
 wPower = 15;  
}  
  
bool Ninja::lowHP()  
{  
 return ((double)HP/maxHP) <= (20.0/100);  
}

Knight.h

#ifndef **FIGHTINGGAME\_KNIGHT\_H**#define **FIGHTINGGAME\_KNIGHT\_H**#include <iostream>  
#include "Character.h"  
  
class Knight : public Character  
 {  
private:  
 std::string wName;  
 int wPower;  
 int armor;  
public:  
 Knight();  
 Knight(std::string, int);  
 int getAttack();  
 void print();  
 bool isDead(int);  
 void updateXP(int);  
 void collectWeapon(std::string);  
 bool lowHP();  
  
 };  
  
#endif //FIGHTINGGAME\_KNIGHT\_H

Knight.cpp

#include "Knight.h"  
  
Knight::Knight() : Character("knight", 120, 0, 10, 120)  
{  
 armor = 1;  
 wName = "none";  
 wPower = 0;  
}  
  
Knight::Knight(std::string n, int lvl) : Character(n, 120, 0, 10, 120)  
{  
 armor = 1;  
 wName = "none";  
 wPower = 0;  
 for (int i = 0; i < lvl; i++)  
 {  
 level++;  
 NLP += 7;  
 maxHP += 19;  
 armor += 1;  
 }  
 HP = maxHP;  
}  
  
int Knight::getAttack()  
{  
 int attack = 10 + std::rand() % 10 + wPower;  
 std::cout << name << " has dealt " << attack << " damage!\n";  
 return attack;  
}  
  
void Knight::print()  
{  
 std::cout << "LVL " << level << " Knight: " << name << "\t";  
 std::cout << "HP: " << HP << "/" << maxHP << std::endl;  
 std::cout << "XP: " << XP << "/" << NLP << std::endl;  
 std::cout << "ARMOR: " << armor << std::endl;  
 if (wName != "none") {  
 std::cout << "WEAPON NAME: " << wName << std::endl;  
 std::cout << "WEAPON POWER: " << wPower << std::endl;  
 }  
}  
  
bool Knight::isDead(int damage)  
{  
 if (damage > armor)  
 {  
 HP -= (damage - armor);  
 return HP <= 0;  
 }  
}  
  
void Knight::updateXP(int monsterLevel)  
{  
 XP += monsterLevel;  
 if (XP >= NLP)  
 {  
 level++;  
 XP -= NLP;  
 NLP += 5;  
 maxHP += 19;  
 HP += 19;  
 armor += 1;  
 }  
}  
  
void Knight::collectWeapon(std::string nameOfWeapon)  
{  
 wName = nameOfWeapon;  
 wPower += 20;  
}  
  
bool Knight::lowHP()  
{  
 return ((double)HP/maxHP) <= (20.0/100);  
}

Wizard.h

#ifndef **FIGHTINGGAME\_WIZARD\_H**#define **FIGHTINGGAME\_WIZARD\_H**#include <iostream>  
#include "Character.h"  
  
class Wizard : public Character  
 {  
private:  
 std::string mName;  
 int mPower;  
public:  
 Wizard();  
 Wizard(std::string, int);  
 int getAttack();  
 void print();  
 bool isDead(int);  
 void updateXP(int);  
 void collectWeapon(std::string);  
 bool lowHP();  
 };  
  
#endif //FIGHTINGGAME\_WIZARD\_H

Wizard.cpp

#include "Wizard.h"  
  
Wizard::Wizard() : Character("wizard", 100, 0, 10, 100)  
{  
 mName = "None";  
 mPower = 0;  
}  
  
Wizard::Wizard(std::string n, int lvl) : Character(n, 100, 0, 10, 100)  
{  
 mName = "None";  
 mPower = 0;  
 for (int i = 0; i < lvl; i++)  
 {  
 level++;  
 NLP += 6;  
 maxHP += 8;  
 HP += 8;  
 mPower += 3;  
 }  
}  
  
int Wizard::getAttack()  
{  
 int attack = 5 + std::rand() % 10 + mPower;  
 std::cout << name << " has dealt " << attack << " damage!\n";  
 return attack;  
}  
  
void Wizard::print()  
{  
 std::cout << "LVL " << level << " Wizard: " << name << "\t";  
 std::cout << "HP: " << HP << "/" << maxHP << std::endl;  
 std::cout << "XP: " << XP << "/" << NLP << std::endl;  
 std::cout << "MAGIC POWER: " << mPower;  
 if (mName != "None") std::cout << " + " << mPower << "\nMAGIC WEAPON: " << mName;  
 std::cout << std::endl;  
}  
  
bool Wizard::isDead(int damage)  
{  
 HP -= damage;  
 return HP <= 0;  
}  
  
void Wizard::updateXP(int monsterLevel)  
{  
 XP += monsterLevel;  
 if (XP >= NLP)  
 {  
 level++;  
 XP -= NLP;  
 NLP += 5;  
 maxHP += 8;  
 HP += 8;  
 mPower += 3;  
 }  
}  
  
void Wizard::collectWeapon(std::string nameOfWeapon)  
{  
 mName = nameOfWeapon;  
 mPower += 25;  
}  
  
bool Wizard::lowHP()  
{  
 return ((double)HP/maxHP) <= (20.0/100);  
}

Monster.h

#ifndef **FIGHTINGGAME\_MONSTER\_H**#define **FIGHTINGGAME\_MONSTER\_H**#include "Character.h"  
#include <iostream>  
#include <string>  
  
class Monster : public Character  
 {  
private:  
 std::string type;  
public:  
 Monster(std::string, int);  
 int getAttack();  
 void print();  
 bool isDead(int);  
 void updateXP(int);  
 };  
  
  
#endif //FIGHTINGGAME\_MONSTER\_H

Monster.cpp

#include "Monster.h"  
  
Monster::Monster(std::string t, int l)  
{  
 type = t;  
 double mult;  
 if (l < 6)  
 mult = 3;  
 else if (l < 11)  
 mult = 5.5;  
 else  
 mult = 8;  
 maxHP = (int)(mult\*(3+l));  
 HP = maxHP;  
 level = l;  
}  
  
int Monster::getAttack()  
{  
 int attack = level + (3 + std::rand() % level);  
 std::cout << type << " has dealt " << attack << " damage!\n";  
 return attack;  
}  
  
void Monster::print()  
{  
 std::cout << "LVL " << level << " MONSTER: " << type << std::endl;  
 std::cout << "HP: " << HP << "/" << maxHP << std::endl;  
}  
  
void Monster::updateXP(int) {}  
  
bool Monster::isDead(int damage)  
{  
 HP -= damage;  
 return HP <= 0;  
}

Dragon.h

#ifndef **FIGHTINGGAME\_DRAGON\_H**#define **FIGHTINGGAME\_DRAGON\_H**#include <iostream>  
#include "Character.h"  
  
class Dragon : public Character  
 {  
private:  
 int scales;  
 int maxScales;  
public:  
 Dragon();  
 Dragon(std::string, int);  
 int getAttack();  
 void print();  
 bool isDead(int);  
 void updateXP(int){}  
 void heal();  
 bool lowHP();  
 };  
  
#endif //FIGHTINGGAME\_DRAGON\_H

Dragon.cpp

#include "Dragon.h"  
  
Dragon::Dragon() : Character("Dragon", 180, 0, 15, 180)  
{  
 maxScales = 25;  
 scales = maxScales;  
}  
  
Dragon::Dragon(std::string n , int lvl) : Character(n, 180, 0, 15, 180)  
{  
 maxScales = 25;  
 scales = maxScales;  
 for (int i = 1; i < lvl; i++)  
 {  
 maxScales += 10;  
 level ++;  
 maxHP += 30;  
 }  
 scales = maxScales;  
 HP = maxHP;  
}  
  
int Dragon::getAttack()  
{  
 int attack = level + (std::rand() % 2\*level);  
 std::cout << name << " has dealt " << attack << " damage!\n";  
 return attack;  
}  
  
void Dragon::print()  
{  
 std::cout << "LVL " << level << " DRAGON: " << name << std::endl;  
 std::cout << "HP: " << HP << "/" << maxHP << std::endl;  
 std::cout << "SCALES: " << scales << " / " << maxScales << std::endl;  
}  
  
bool Dragon::isDead(int damage)  
{  
 if (scales > 0) {  
 scales -= (damage / 2);  
 if (scales < 0)  
 scales = 0;  
 } else  
 HP -= damage;  
 return HP <= 0;  
}  
  
void Dragon::heal() //will actually grow scales instead  
{  
 int newScales = level + (std::rand() % 2\*level);  
 std::cout << name << " has grown " << newScales << " new scales!\n";  
 scales += newScales;  
 if (scales > maxScales)  
 scales = maxScales;  
}  
  
bool Dragon::lowHP()  
{  
 return ((double)HP/maxHP) <= (10.0/100);  
}