#include <iostream>

#include <cstdlib>

#include <ctime>

#include <stdio.h>

#include <conio.h>

#include <fstream>

using namespace std;

int experience = 0;

int coins = 0;

int level = 1;

int health = 75;

int maxHealth = 0;

int sPotion = 1;

int nPotion = 1;

int lPotion = 1;

int saveGameSlot = 0;

string name;

void mainScreen();

void mainOptions();

void inventory();

void shop();

void fight();

void levelUp();

void saveGame();

void loadGame();

int main()///FIRST OPENING SCREEN

{

int selectNum = 0;

cout << "RPG Game\n--------\n" << endl;

cout << "1} New Game" << endl;

cout << "2} Continue" << endl;

cout << "3} Options" << endl;

cout << "4} Quit\n" << endl;

while ( selectNum != 1 && selectNum != 2 && selectNum != 3 && selectNum != 4)

{

cin >> selectNum;

}

system("CLS");

if ( selectNum == 1 ){ mainScreen(); }

if ( selectNum == 2 ){ loadGame(); }

if ( selectNum == 3 ){ mainOptions(); }

}

void mainOptions()///MAINSCREEN OPTIONS

{

int randNum = 0;

bool saveUsed;

cout << "RPG Game\n--------\n" << endl;

cout << "1} Delete Save" << endl;

cout << "2} Back" << endl;

while ( randNum != 1 && randNum != 2 ){

cin >> randNum;

}

system("CLS");

if ( randNum == 1 )

{

cout << "RPG Game\n---------\n" << endl;

cout << "Wich slot would you like to delete?\n" << endl;

cout << " 1} Slot 1" << endl;

cout << " 2} Slot 2" << endl;

cout << " 3} Slot 3" << endl;

cout << " 4} Back" << endl;

randNum = 0;

while ( randNum != 1 && randNum != 2 && randNum != 3 && randNum != 4 ){

cin >> randNum;

}

system("CLS");

ifstream takeData;

if ( randNum == 1 )

{

takeData.open("rpgSave1.txt");

saveGameSlot = 1;

}

if ( randNum == 2 )

{

takeData.open("rpgSave2.txt");

saveGameSlot = 2;

}

if ( randNum == 3 )

{

takeData.open("rpgSave3.txt");

saveGameSlot = 3;

}

if ( randNum == 4 ){ mainOptions(); }

takeData >> saveUsed;

takeData.close();

if ( saveUsed )

{

cout << " Are you sure you want to delete this save?" << endl;

cout << " 1} Yes\n 2} No\n" << endl;

randNum = 0;

while ( randNum != 1 && randNum != 2 )

{

cin >> randNum;

}

if ( randNum == 1 )

{

if ( saveGameSlot == 1 )

{

remove ("rpgSave1.txt");

}

if ( saveGameSlot == 2 )

{

remove ("rpgSave2.txt");

}

if ( saveGameSlot == 3 )

{

remove ("rpgSave3.txt");

}

cout << "\n\n \*\*\*DELETED\*\*\*\n" << endl;

system("PAUSE");

}

system("CLS");

mainOptions();

}

}

main();

}

void mainScreen()///MAIN SCREEN (NEW GAME/LOADED CONTINUED GAME SCREEN)

{

int selectNum = 0;

maxHealth = level \* 75 ;

if ( health <= 0 ){ health = 75; }

if ( coins <= 0 ){ coins = 0; }

cout << "RPG Game\n--------" << endl;

cout << "Health: " << health;

cout << " Level: " << level << endl;

cout << "Coin: " << coins;

cout << " Experience: " << experience << endl;

cout << "--------------------------\n" << endl;

cout << "1} Fight" << endl;

cout << "2} Inventory" << endl;

cout << "3} Shop" << endl;

cout << "4} Save" << endl;

cout << "5} Homescreen\n" << endl;

while ( selectNum != 1 && selectNum != 2 && selectNum !=3 && selectNum != 4 && selectNum != 5){

cin >> selectNum;

}

system("CLS");

if ( selectNum == 1 ){ fight(); }

if ( selectNum == 2 ){ inventory(); }

if ( selectNum == 3 ){ shop(); }

if ( selectNum == 4 ){ saveGame(); }

if ( selectNum == 5 )

{

cout << "Are you sure you want to do that?\n" << endl;

cout << " \*\*\* ANY UNSAVED PROGRESS WILL BE LOST \*\*\*\n\n" << endl;

cout << " 1} Yes" << endl;

cout << " 2} No\n" << endl;

selectNum = 0;

while ( selectNum != 1 && selectNum != 2 )

{

cin >> selectNum;

}

if ( selectNum == 1 )

{

system("CLS");

main();

}

if ( selectNum == 2 ){ system("CLS"); mainScreen(); }

}

}

void inventory()

{

int selectNum = 0;

int leftOver = 0;

while ( selectNum != 4 ){

selectNum = 0;

cout << "RPG Game\n--------" << endl;

cout << "Health: " << health;

cout << " Level: " << level << endl;

cout << "Coin: " << coins;

cout << " Experience: " << experience << endl;

cout << "--------------------------\n" << endl;

cout << "1} Small Potions ---- " << sPotion << endl;

cout << "2} Normal Potions --- " << nPotion << endl;

cout << "3} Large Potions ---- " << lPotion << endl;

cout << "4} Back\n" << endl;

while ( selectNum != 1 && selectNum != 2 && selectNum != 3 && selectNum != 4 ){

cin >> selectNum;

}

system("CLS");

if ( selectNum == 1 )

{

if ( sPotion > 0 )

{

sPotion--;

health = health + 15;

if ( health > maxHealth )

{

cout << "You forfeited ";

leftOver = maxHealth - health;

cout << leftOver << " health when you used that Small potion.\n" << endl;

system("PAUSE");

system("CLS");

}

}else

{

cout << "You don't have any more Small Potions.\n" << endl;

system("PAUSE");

system("CLS");

}

}

if ( selectNum == 2 )

{

if ( nPotion > 0 )

{

nPotion--;

health = health + 50;

if ( health > maxHealth )

{

cout << "You forfeited ";

leftOver = maxHealth - health;

cout << leftOver << " health when you used that Normal potion.\n" << endl;

system("PAUSE");

system("CLS");

}

}else

{

cout << "You don't have any more Normal Potions.\n" << endl;

system("PAUSE");

system("CLS");

}

}

if ( selectNum == 3 )

{

if ( lPotion > 0 )

{

lPotion--;

health = health + 150;

if ( health > maxHealth )

{

cout << "You forfeited ";

leftOver = maxHealth - health;

cout << leftOver << " health when you used that Large potion.\n" << endl;

system("PAUSE");

system("CLS");

}

}else

{

cout << "You don't have any more Large Potions.\n" << endl;

system("PAUSE");

system("CLS");

}

}

if ( health > maxHealth ){ health = maxHealth; }

}

mainScreen();

}

void shop()

{

int selectNum = 0;

while ( selectNum != 4 ){

selectNum = 0;

cout << "RPG Game\n--------" << endl;

cout << "Health: " << health;

cout << " Level: " << level << endl;

cout << "Coin: " << coins;

cout << " Experience: " << experience << endl;

cout << "--------------------------\n" << endl;

cout << "1} Small Potion ---- 50 --- " << sPotion << endl;

cout << "2} Normal Potion --- 100 -- " << nPotion << endl;

cout << "3} Large Potion ---- 200 -- " << lPotion << endl;

cout << "4} Back\n" << endl;

while ( selectNum != 1 && selectNum != 2 && selectNum!= 3 && selectNum !=4 ){

cin >> selectNum;

}

system("CLS");

if ( selectNum == 1 )

{

if ( coins > 49 )

{

coins = coins - 50;

sPotion ++;

}else

{

cout << "Not Enough Coins.\n" << endl;

system("PAUSE");

system("CLS");

}

}

if ( selectNum == 2 )

{

if ( coins > 99 )

{

coins = coins - 100;

nPotion ++;

}else

{

cout << "Not Enough Coins.\n" << endl;

system("PAUSE");

system("CLS");

}

}

if ( selectNum == 3 )

{

if ( coins > 199 )

{

coins = coins - 200;

lPotion ++;

}else

{

cout << "Not Enough Coins.\n" << endl;

system("PAUSE");

system("CLS");

}

}

}

mainScreen();

}

void saveGame()

{

int randNum = 0;

bool saveUsed;

cout << "What slot would you like to save in?\n" << endl;

cout << " 1} Slot 1" << endl;

cout << " 2} Slot 2" << endl;

cout << " 3} Slot 3" << endl;

cout << " 4} Back\n" << endl;

randNum = 0;

saveGameSlot = 0;

while ( randNum != 1 && randNum != 2 && randNum != 3 )

{

cin >> randNum;

}

cout << "\n" << endl;

ofstream saveData;

ifstream takeData;

if ( randNum == 1 )

{

takeData.open("rpgSave1.txt");

saveGameSlot = 1;

takeData >> saveUsed;

takeData.close();

}

if ( randNum == 2 )

{

takeData.open("rpgSave2.txt");

saveGameSlot = 2;

takeData >> saveUsed;

takeData.close();

}

if ( randNum == 3 )

{

takeData.open("rpgSave3.txt");

saveGameSlot = 3;

takeData >> saveUsed;

takeData.close();

}

if ( randNum == 4 ){ mainScreen(); }

if ( saveUsed )

{

cout << " This save is already in use...\n" << endl;

cout << " Save over it?" << endl;

cout << " 1} Yes\n 2} No\n" << endl;

randNum = 0;

while ( randNum != 1 && randNum != 2 )

{

cin >> randNum;

}

if ( randNum == 1 )

{

saveUsed = true;

if ( saveGameSlot == 1 )

{

remove ("rpgSave1.txt");

saveData.open("rpgSave1.txt");

}

if ( saveGameSlot == 2 )

{

remove ("rpgSave.txt");

saveData.open("rpgSave2.txt");

}

if ( saveGameSlot == 3 )

{

remove ("rpgSave.txt");

saveData.open("rpgSave3.txt");

}

saveData << saveUsed << ' ' << experience << ' ' << coins << ' ' << level << ' ' << health << ' ' << sPotion << ' ' << nPotion << ' ' << lPotion;

saveData.close();

cout << "\n\n \*\*\*SAVED\*\*\*\n" << endl;

system("PAUSE");

}

}

if ( saveUsed == false )

{

saveUsed = true;

if ( saveGameSlot == 1 )

{

remove ("rpgSave1.txt");

saveData.open("rpgSave1.txt");

}

if ( saveGameSlot == 2 )

{

remove ("rpgSave2.txt");

saveData.open("rpgSave2.txt");

}

if ( saveGameSlot == 3 )

{

remove ("rpgSave.txt");

saveData.open("rpgSave3.txt");

}

saveData << ' ' << saveUsed << ' ' << experience << ' ' << coins << ' ' << level << ' ' << health << ' ' << sPotion << ' ' << nPotion << ' ' << lPotion;

saveData.close();

cout << "\n\n \*\*\*SAVED\*\*\*\n" << endl;

system("PAUSE");

}

system("CLS");

mainScreen();

}

void loadGame()

{

int randNum = 0;

bool saveUsed;

ifstream takeData;

cout << "Which slot would you like to load?\n" << endl;

cout << " 1} Slot 1" << endl;

cout << " 2} Slot 2" << endl;

cout << " 3} Slot 3" << endl;

cout << " 4} Back\n" << endl;

randNum = 0;

while ( randNum != 1 && randNum != 2 && randNum != 3 && randNum != 4 )

{

cin >> randNum;

}

cout << "\n" << endl;

if ( randNum == 1 )

{

takeData.open("rpgSave1.txt");

}

if ( randNum == 2 )

{

takeData.open("rpgSave2.txt");

}

if ( randNum == 3 )

{

takeData.open("rpgSave3.txt");

}

if ( randNum == 4 ){ system("CLS"); main(); }

takeData >> saveUsed;

takeData.close();

if ( saveUsed )

{

if ( randNum == 1 )

{

takeData.open("rpgSave1.txt");

}

if ( randNum == 2 )

{

takeData.open("rpgSave2.txt");

}

if ( randNum == 3 )

{

takeData.open("rpgSave3.dat");

}

takeData >> saveUsed >> experience >> coins >> level >> health >> experience >> sPotion >> nPotion >> lPotion;

takeData.close();

cout << " Game Loaded.\n" << endl;

system("PAUSE");

}

if ( saveUsed == false )

{

cout << " No save file found.\n" << endl;

system("PAUSE");

system("CLS");

loadGame();

}

system("CLS");

mainScreen();

}

void levelUpCheck()

{

bool levelingUp = false;

if ( level == 1 && experience > 49 ){ level ++; levelingUp = true; }

if ( level == 2 && experience > 149 ){ level ++; levelingUp = true; }

if ( level == 3 && experience > 299 ){ level ++; levelingUp = true; }

if ( level == 4 && experience > 499 ){ level ++; levelingUp = true; }

if ( level == 5 && experience > 749 ){ level ++; levelingUp = true; }

if ( level == 6 && experience > 999 ){ level ++; levelingUp = true; }

if ( level > 6 && experience > ( level\* 300 ) ){ level ++; levelingUp = true; }

if ( levelingUp )

{

///LEVELUP SCREEN

cout << "YOU JUST LEVELED UP!!!!!\n" << endl;

cout << " You leveled from level " << level - 1 << " to level " << level << " !!\n" << endl;

system("PAUSE");

system("CLS");

}

}

void fight()

{

int selectNum = 0;

int randNum = 0;

int weaponNum = 0;

int mobHealth = 0;

int mobLevel = 0;

int mobLevelVal = 0;

int damage = 0;

int damageVal = 0;

int winnings = 0;

int coinWinnings = 0;

int experienceWinnings = 0;

int skip = 0;

cout << "RPG Game\n--------" << endl;

cout << "Health: " << health;

cout << " Level: " << level << endl;

cout << "Coin: " << coins;

cout << " Experience: " << experience << endl;

cout << "--------------------------\n" << endl;

cout << "You are fighting a ";

srand( time (0) );

if ( level > 0 && level <= 5 ){ mobLevel = ( ( rand () % 79 ) + 1 ); }

if ( level > 7 ){ mobLevel = ( ( rand () % 100 ) + 1 ); }

if ( mobLevel > 0 && mobLevel < 50 ){ cout << "Basic "; mobLevelVal = 1; }

if ( mobLevel > 49 && mobLevel < 80 ){ cout << "Stronger "; mobLevelVal = 2; }

if ( mobLevel > 79 && mobLevel < 95 ){ cout << "Powerful "; mobLevelVal = 3; }

if ( mobLevel > 94 && mobLevel < 100 ){ cout << "Leet "; mobLevelVal = 4; }

if ( mobLevel == 100 ){ cout << "Super-Ultra-Beast "; mobLevelVal = 5; }

if ( mobLevel == 100 ){ mobLevel = 1000; }

int mob = ( ( rand () % 4 ) +1 );

if ( mob == 1 ){ cout << "Zombie"; }

if ( mob == 2 ){ cout << "Skeleton"; }

if ( mob == 3 ){ cout << "Spider"; }

if ( mob == 4 ){ cout << "Bandit"; }

cout << "!!" << endl << endl;

randNum = ( ( rand () % 20 ) + 41 );

mobHealth = mobLevelVal \* randNum ;///FINDING THE MOB'S HEALTH

system("PAUSE");

system("CLS");

int turn = ( ( rand () % 2 ) +1 );

while (mobHealth > 0 && health > 0 ){

cout << "RPG Game\n--------" << endl;///STATS AND STUFF

cout << "Health: " << health;

cout << " Level: " << level << endl;

cout << "Coin: " << coins;

cout << " Experience: " << experience << endl;

cout << "--------------------------" << endl;

cout << "Mob\n---" << endl;

cout << "Health: " << mobHealth;

cout << " Level: " << mobLevel << endl;

cout << "--------------------------\n" << endl;

if ( turn == 1 )///PLAYER'S TURN

{

cout << " Your Turn!" << endl << endl;

cout << " 1} Punch" << endl;

if ( level >= 2 )

{

cout << " 2} Kick" << endl;

}

if ( level >= 3 )

{

cout << " 3} Dagger" << endl;

}

if ( level >= 4 )

{

cout << " 4} Shuriken" << endl;

}

if ( level >= 5 )

{

cout << " 5} Sword" << endl;

}

if ( level >= 7 )

{

cout << " 6} Flail" << endl;

}

if ( level >= 10 )

{

cout << " 7} Mace" << endl;

}

if ( level >= 15 )

{

cout << " 8} GreatSword" << endl;

}

cout << endl << endl;

selectNum = 0;

weaponNum = 0;

while ( selectNum == 0 ){

while ( selectNum != 1 && selectNum != 2 && selectNum != 3 && selectNum !=4 && selectNum != 5 && selectNum != 6 && selectNum != 7 && selectNum != 8)

{

cin >> selectNum;

}

if ( selectNum == 1 ){ weaponNum = 1; }

if ( selectNum == 2 && level >= 2 ){ weaponNum = 2; }

if ( selectNum == 2 && level < 2 ){ selectNum = 0; }

if ( selectNum == 3 && level >= 3 ){ weaponNum = 3; }

if ( selectNum == 3 && level < 3 ){ selectNum = 0; }

if ( selectNum == 4 && level >= 4 ){ weaponNum = 4; }

if ( selectNum == 4 && level < 4 ){ selectNum = 0; }

if ( selectNum == 5 && level >= 5 ){ weaponNum = 5; }

if ( selectNum == 5 && level < 5 ){ selectNum = 0; }

if ( selectNum == 6 && level >= 7 ){ weaponNum = 6; }

if ( selectNum == 6 && level < 7 ){ selectNum = 0; }

if ( selectNum == 7 && level >= 10 ){ weaponNum = 7; }

if ( selectNum == 7 && level < 10 ){ selectNum = 0; }

if ( selectNum == 8 && level >= 15 ){ weaponNum = 8; }

if ( selectNum == 8 && level < 15 ){ selectNum = 0; }

}

damage = 0;

damageVal = 0;

damageVal = ( ( rand () % 15 ) + 1 );

if ( weaponNum == 1 )

{

///PUNCH

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 5 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 5 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 5 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 5 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 2 )

{

///KICK

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 7 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 7 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 7 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 7 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 3 )

{

///DAGGER

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 10 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 10 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 10 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 40 ) + 1 ) \* 2 );

damage = ( ( ( rand () % 10 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 4 )

{

///SHURIKEN

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 12 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 12 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 12 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 12 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 5 )

{

///SWORD

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 15 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 15 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 15 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 15 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 6 )

{

///FLAIL

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 20 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 20 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 20 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 20 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 7 )

{

///MACE

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 25 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 25 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 25 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 25 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 8 )

{

///GREATSWORD

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 40 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 40 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///SOMEWHAT-HIGH STRIKE

damage = ( ( ( rand () % 40 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///VERY-HIGH STRIKE

damage = ( ( ( rand () % 40 ) + 1 ) \* 5 );

}

}

mobHealth = mobHealth - damage;

turn = 2;

skip = 1;

}

if ( turn == 2 && skip != 1) ///COMPUTER'S TURN

{

cout << " Mob's Turn\n" << endl << endl;

system("PAUSE");

damage = 0;

damageVal = ( ( rand () % 15 ) + 1 );

if ( level < 8 ){ weaponNum = ( ( rand () % 5 ) + 1 ); }

if ( level > 7 ){ weaponNum = ( ( rand () % 9 ) + 1 ); }

if ( weaponNum > 0 && weaponNum <= 5 )

{

///SCRATCH

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 5 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 5 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///HIGH-POWER STRIKE

damage = ( ( ( rand () % 5 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///CRITICAL STRIKE

damage = ( ( ( rand () % 5 ) + 1 ) \* 5 );

}

}

if ( weaponNum > 5 && weaponNum <= 8 )

{

///BITE

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 12 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 12 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///HIGH-POWER STRIKE

damage = ( ( ( rand () % 12 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///CRITICAL STRIKE

damage = ( ( ( rand () % 12 ) + 1 ) \* 5 );

}

}

if ( weaponNum == 9 )

{

///WALLOP

if ( damageVal >0 && damageVal <= 7 )

{

///LOW-POWER STRIKE

damage = ( ( rand () % 25 ) + 1 );

}

if ( damageVal > 7 && damageVal <= 12 )

{

///MEDIUM-POWER STRIKE

damage = ( ( ( rand () % 25 ) + 1 ) \* 2 );

}

if ( damageVal > 12 && damageVal <= 14 )

{

///HIGH-POWER STRIKE

damage = ( ( ( rand () % 25 ) + 1 ) \* 3 );

}

if ( damageVal == 15 )

{

///CRITICAL STRIKE

damage = ( ( ( rand () % 40 ) + 1 ) \* 2 );

}

}

health = health - damage;

turn = 1;

}

system("CLS");

if ( health < 0 ){ health = 0; }

if ( mobHealth < 0 ){ mobHealth = 0; }

cout << "RPG Game\n--------" << endl;///STATS AND STUFF

cout << "Health: " << health;

cout << " Level: " << level << endl;

cout << "Coin: " << coins;

cout << " Experience: " << experience << endl;

cout << "--------------------------" << endl;

cout << "Mob\n---" << endl;

cout << "Health: " << mobHealth;

cout << " Level: " << mobLevel << endl;

cout << "--------------------------\n" << endl;

if ( damageVal > 0 && damageVal <= 7 ){ cout << " Low-Power Strike!\n" << endl; }

if ( damageVal > 7 && damageVal <= 12 ){ cout << " Medium-Power Strike!\n" << endl; }

if ( damageVal > 12 && damageVal <= 14 ){ cout << " High-Power Strike!\n" << endl; }

if ( damageVal == 15 ){ cout << " Critical Strike!\n" << endl; }

cout << " Damage dealt to ";

if ( turn == 1 ){ cout << "you"; }

if ( turn == 2 ){ cout << "the mob"; }

cout << ": " << damage << " !!!\n" << endl;

system("PAUSE");

system("CLS");

skip = 0;

}

winnings = 0;

coinWinnings = 0;

experienceWinnings = 0;

cout << "RPG Game\n--------" << endl;///STATS AND STUFF

cout << "Health: " << health;

cout << " Level: " << level << endl;

cout << "Coin: " << coins;

cout << " Experience: " << experience << endl;

cout << "--------------------------" << endl;

cout << "Mob\n---" << endl;

cout << "Health: " << mobHealth;

cout << " Level: " << mobLevel << endl;

cout << "--------------------------\n" << endl;

randNum = ( ( rand () % ( mobLevel / 2 ) ) + 1 );

winnings = ( mobLevel + randNum );

randNum = ( ( rand () % 5 ) + 6 );

coinWinnings = winnings + ( winnings / randNum );

if ( mobHealth <= 0 )

{

randNum = ( ( rand () % 5 ) + 6 );

experienceWinnings = coinWinnings + ( coinWinnings / randNum );

}

if ( health <= 0 )

{

///LOSING SCREEN

coins = coins - coinWinnings;

cout << " You have lost " << coinWinnings << " coins by losing this battle.\n" << endl;

}

if ( mobHealth <= 0 )

{

///WINNING SCREEN

coins = coins + coinWinnings;

experience = experience + experienceWinnings;

cout << "You have won " << coinWinnings << " coins by winning this battle!!!\n" << endl;

cout << "You have won " << experienceWinnings << " experience by winning this battle!!!\n" << endl << endl;

}

system("PAUSE");

system("CLS");

levelUpCheck();

mainScreen();

}