Project 2

Title:  
Dungeons and Dragons Simplified

CIS-5

Kevin Mai

Riverside Community College

Table of Contents

1. Cover Page
2. Table of Contents
3. Rules and Explanations
4. Rules and Explanation Cont.
5. Approach and the program itself
6. Flow Charts
7. Flow Charts Cont.
8. Proof of Running the Program
9. Pseudo Code
10. Pseudo Code cont.
11. The Code itself

The GitHub repository:

<https://github.com/kmai3/Project-2>

1. Introduction

Explanation:

My game is a simplified version of the Dungeons and Dragons board game and instead of a dungeon master, you are fighting against an AI Dungeon master rather than a human player. The AI Dungeon master will use 5 enemies and you will have 3 different classes to choose from due to it being a simplified version of the board game.

Rules:

1. If your health drops to 0, it is game over
2. If you take too long in a fight, and run out of rounds it is game over as you are then surrounded by dungeon mobs and it is game over.
3. If you use an element that is super effective against the enemy, it will do critical damage.

Note: The only current playable class is mage and the only mob is the goblin due to the amount of time spent on the mid-term.

Clases: Mage, Ranger, Fighter

Enemies: goblin

Skeleton

Ogre

Skeleton Giant

Dark Mage

Dragon

Elements

Fire, Water, blunt, earth

Blunt equal damage to everything except for the dragon.

Goblin weak to fire, resistance to water

Skeleton weak to earth, resistance to fire

Ogre weak to fire, resistance to earth

Skeleton Giant weak to earth, resistance to fire

Dark Mage is resistant to everything but blunt force

Dragon weak to water, resistance everything else

Healing is based of damage value

Lvl 1-5 -> damage 2+2x; HP: 100:

Mage- fireball,hydro cannon, force smash, earthquake, barrier&Heal(100% damage nullification)

Lvl 1-5 -> damage 2+2^x(x=levels) HP:150:  
Archer - fire bolt, ice shot, charged shot, metal bolt, row&heal(50% damage nullification)

Lvl 1-5 damage 4+1xHP:300   
Fighter - Fire Blade, Water Slash, Heavy Strike, obsidian pillar, block&heal(100% damage nullification)

Goblin - 15hp, atk1

Turn to beat 9 before or game over.

Skeleton - 20Hp, atk2

Turns to beat 5 turns or game over

Ogre - 30hp, - attack 5

Turns to beat 5 or game over

Skeleton giant, - 56hp -atk10

Turns to beat 5 or game over

Dragon, 100 Hp, -atk 15

Turns to beat 10 or game over

My Approach on the game

**Translating Game Play to Programming Language**

With the knowledge given during class, the game is centered around chapter 4 making decisions, as the board game is relatively centered around making decisions to succeed in killing the boss, however because there is no other player, the game values and information are repeated through each new game, unlike the board game which has a player that is allowed to change certain aspects of the game known as the dungeon master. However, in this game, the boss minions are determined before the game starts as there is no player 2.

**Project 2 Enhancements**

Above I have wrote what finite data values for everything in this game, included with the if statements. After finishing project 1, project 2 was centered around me finishing the game with functions and arrays, this allowed me to minimize the size of my game while making the game more readable. Functions also allowed the game to be coded faster due to the code not needing to be copy and pasted every time I needed a similar line of code. This allows for easier troubleshooting and less time coding.

**Similarities to the card game**

The classes and combat are almost exactly the same as the board game.

Mobs(The Enemies) are created extremely similar to the mobs in the actual board game.

Moving down the dungeon after the death of each mob

The Program itself

Lines of Code: 198

26 Variables

Project 1:

The program took me around 6 hours of straight coding and roughly 2-3 hours of asking for help on the code itself. The program itself utilizes many of the things taught in the first half of the class, being based mainly around loops and conditional statements such as if and switch. Through the use of the conditional statements and loops, the creation of an unfinished RPG Game based on the Dungeons and Dragons board game, in which the dungeon master is the conditional statements, while the loops were the fight itself.

Lines of Code: 817

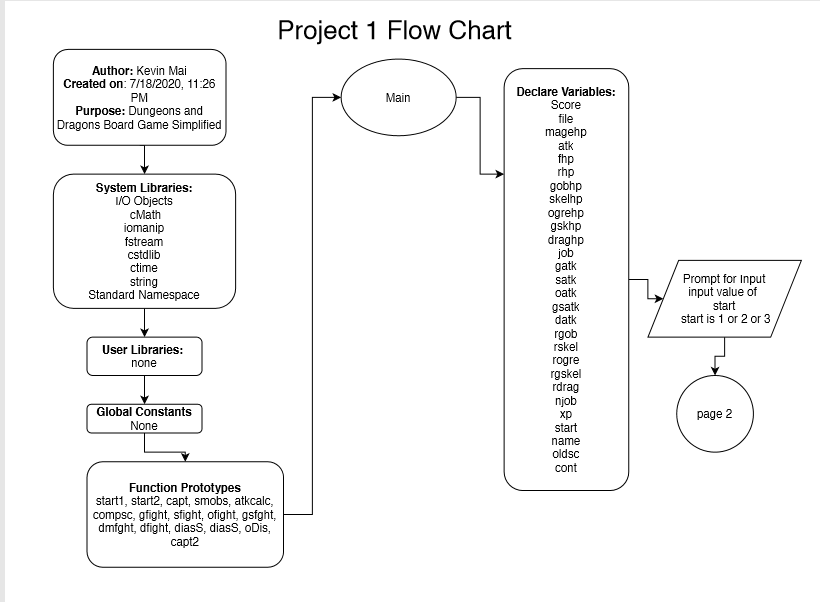
30 Variables(In main, there are much more variables in functions that are used temporarily)

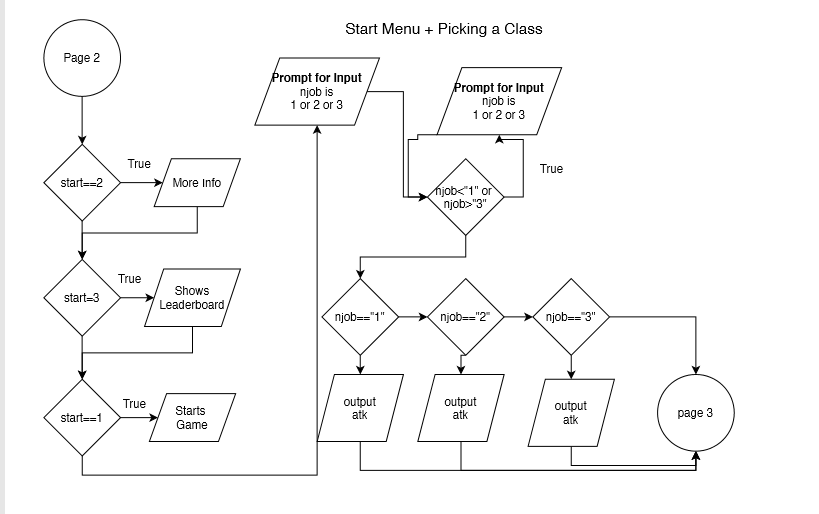
Project 2:

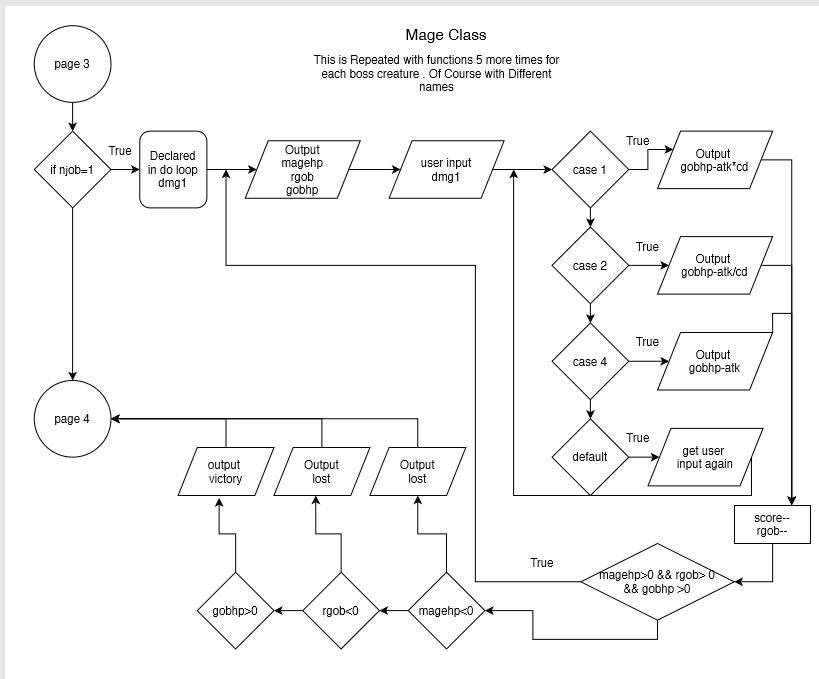
Project 2 is a continuation of project 1, in which I utilize functions, arrays, vectors, sorts, and search to finish up the game. It took me roughly 10 hours of straight coding and 3-4 hours of going to the syntax of vectors and functions to find out what my mistake was and what I could also do with functions and vectors, this also was combined with the knowledge from the class lectures which was used. The project was extremely fun because not only did I get to witness my growth in coding but also get to play something I created.

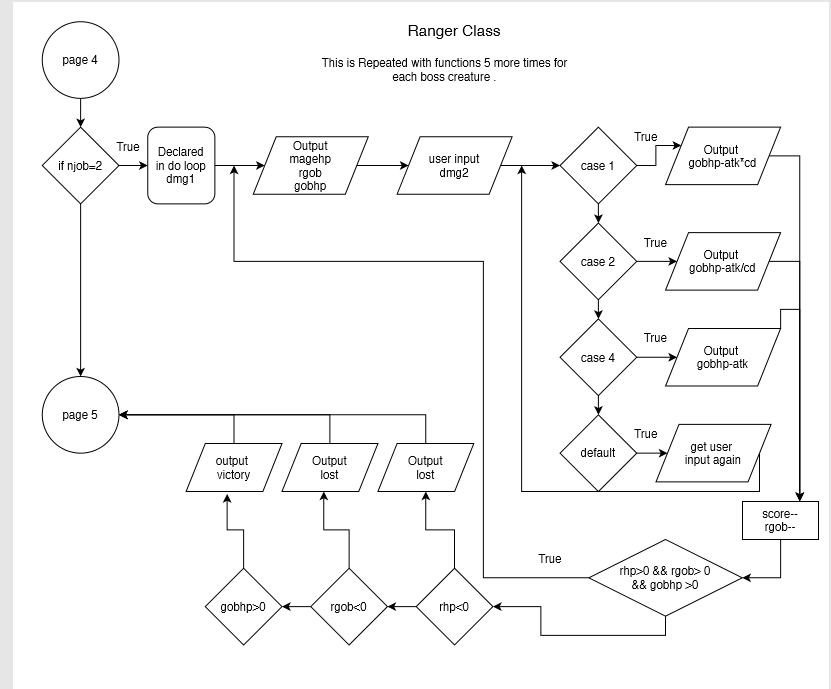
Flowchart is another file and here:

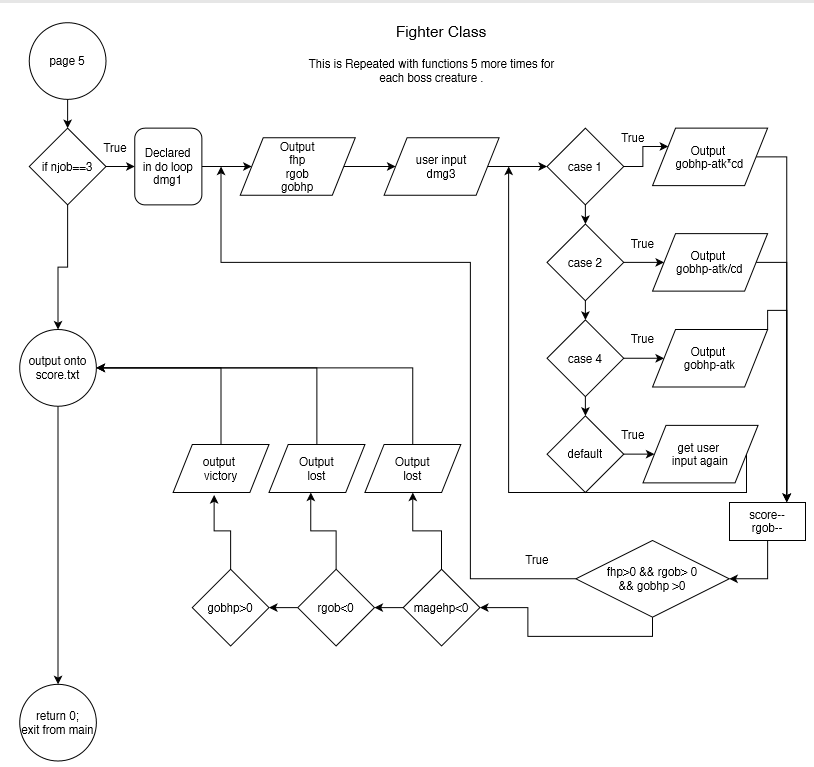
Flowchart only shows one mob because every other boss monster after goblin is essentially a copy paste with different stats. Which functions were used to copy and paste essentially all three classes. Each Class will be verified by an if statement running the function code inside which is shown through each of the flow charts. The User will be able to then pick 4 moves do damage and until either the player’s hp, the mob’s hp, or the rounds reach 0, it will repeat.









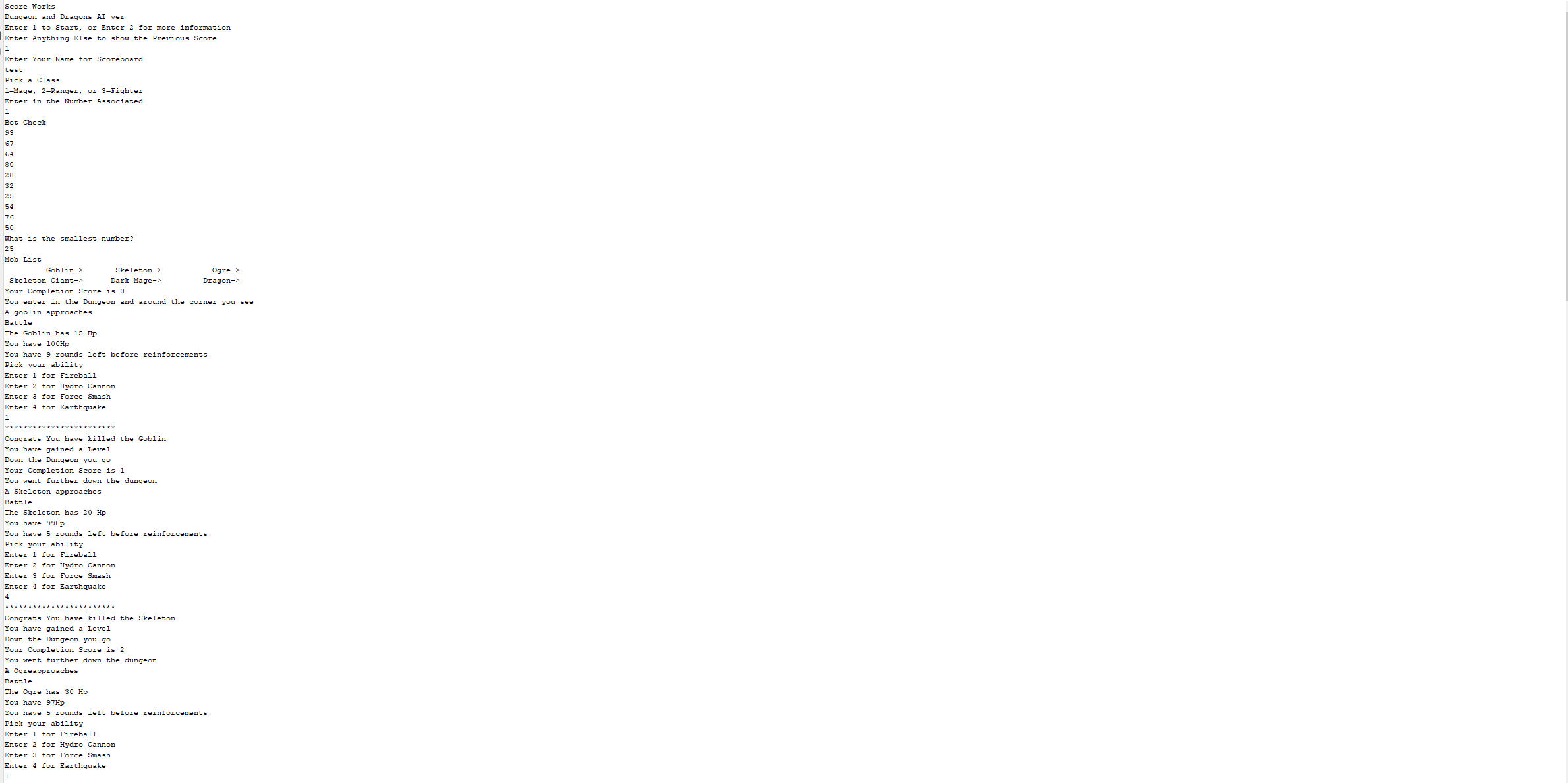


Check off is another file.

**My checkoff file does not every line that a concept is on due to it being ~800 lines big(it can’t fit on the check off sheet I tried) and that I am already running out of time, due to the final taking longer than expected.**

**However, it has every concept and 1 or more lines as an example, that I learned and used the concepts taught in class.**

Pictures of the game running and working is separated into 2 pictures due to the game being based around Dungeons and Dragons as each move is outputted by the computer. The Two files are Page1Proof, and Page2Proof and also here they are.





Pseudo Code

//Initialize the code

Declaring Values

Boss Values

Amount of Rounds allowed to beat the boss

Show Old Scores

Opens up Score.txt

Mapping

Starts up the game

Converts Menu options from char to int with type casting

Choice 2 gives more information about the game

After Starting with Choice one or after choice 2, player chooses a class.

Captcha starts

Player must confirm they are a human through picking the smallest number

Attack values will depend on xp and class

If class chosen is a mage

Do battle

Menu for picking abilities

Stops the loop when someone is dead or when reinforcements come

Visual separation of the battle being ended

If goblin is killed

If reinforcements came

If player dies

Continues if player killed the goblin

This is repeated through every boss

Except dragon in which another option is to shield and heal oneself

Player is allowed to see a numeric score, decimal score, or both in which they pick.

Score is shown in and saved score.txt

The Code itself

/\*

\* File: main.cpp

\* Author: Kevin Mai

\* Created on August 1, 2020, 5:00 PM

\* Purpose: Dungeons and Dragon Board Game Project

\*/

#include <iostream> //I/O Library

#include <string> // for strings

#include <iomanip> //setw

#include <cmath> // For sqrt and pow functions

#include <fstream> // To put in files

#include <cstdlib> //Random Seed Time, Exit Function

#include <ctime> //Random Seed

#include <vector> // Vectors

using namespace std;

//User Libraries

//Global Constants, no Global Variables are allowed

//Math/Physics/Conversions/Higher Dimensions - i.e. PI, e, etc...

//Function Prototypes

char start1(); // Basic Menu to Start the Game

void start2(char , char &, string &, string oldsc);

bool capt(vector <int> &, int); //Captcha to detect boting

void smobs(string[][4]); //Displays the Mobs able to be fought

int atkcalc(int, int); // Calculates Attack value

void compsc(); // Shows Completion Value

bool gfight(string [], int &, int , int, int ,int &, //Goblin Battle

int, int, float &, int, int, int, int);

bool sfight(string [], int &, int, int, int, int &, //Skeleton Battle

int, int, float &, int, int, int, int);

bool ofight(string [], int &, int, int, int, int &, //Ogre Battle

int, int, float &, int, int, int, int, string[][4]);

bool sGfght(string [], int &, int, int, int, int &,//Skeleton Giant Battle

int, int, float &, int, int, int, int, string[][4]);

bool dMfght(string [], int &, int, int, int, int &, //Dark Mage Battle

int, int, float &, int, int, int, int, string[][4]);

bool dfight(string [], int &, int, int, int, int &, //Dragon Battle

int, int, float &, int, int, int, int, string[][4]);

void disS(float); //Displays Score in Decimal Value

void disS(int); //Displays Score in Numeric Value

void oDis(float); //Displays Score Overall

bool capt2(); //Captcha 2

//Execution of Code Begins here

int main(int argc, char\*\* argv) {

//Random seed

srand(static\_cast<unsigned int>(time(0)));

unsigned int cd = rand() % 4 + 2; //Cd is the crit damage

//Declare all variables for this function

float score;

fstream file;

int magehp, //The hp value of the mage

atk ,//The atk value of the character

fhp, // The HP of the fighter

rhp, //The HP value of the ranger

gobhp, //HP of the goblin

skelhp, //HP of the Skeleton

ogrehp, //HP of the ogre

gskhp, //HP of the Skeleton Giant

dmhp, //Hp of the Darkmage

job, //job

gatk, //Attack of the Goblin

satk, //Attack of the Skeleton

oatk, //Attack of the Ogre

gsatk, //Attack of the Skeleton Giant

dMatk, //Attack of the Dark Mage

datk, //Attack of the Dragon

rgob, //Rounds to beat goblin

rskel, //Rounds to beat skeleton

rogre, //Rounds to beat the ogre

rgskel, //Rounds to beat Giant Skeleton

rDMage, //Rounds to beat the Dark Mage

rdrag, //Rounds to beat the dragon

n, //Size of the cpch vector

xp, // The amount of xp of the character, makes the character stronger

draghp; //HP of the Dragon

char njob, //Job represented in a number

start; // Start of the game Menu

string name, //Name of the Player

oldsc; // Old score from the previous game

string rclass[12]={"Fireball", "Hydro Cannon", "Force Smash", "Earthquake",

"Fire Arrow", "Ice Shot", "Charged Shot", "Metal Bolt", "Fire Sword",

"Hydro Slash", "Heavy Strike", "Obsidian Pillar"};

bool cont;

string bosses[3][4]={{"Goblin", "Skeleton", "Ogre"},

{"Skeleton Giant", "Dark Mage", "Dragon"}};

vector<int> cpch(n); //used in computer check

//Initialize the code

//Balancing Area Change Values to make it harder or easier

magehp=100;

fhp=300;

rhp=150;

xp=1;

//Boss Values

gobhp=15;

skelhp=20;

ogrehp=30;

gskhp=56;

dmhp=40;

draghp=100;

gatk=1;

satk=2;

oatk=5;

gsatk=10;

dMatk=20;

datk=20;

//Amount of Rounds to beat the bosses

rgob=9;

rskel=5;

rogre=5;

rgskel=5;

rDMage=10;

rdrag=20;

score=100;

n=10;

oldsc="";

string temp; //Temp String for scoreboard

while(file >> temp){ //Shows the Old Scores

oldsc+=temp+"/n";

}

//Scoreboard called score.txt

file.open("score.txt",ios::in|ios::out);

file >> oldsc;

if(file.is\_open()){cout<<"Score Works"<<endl;} //Checks if file is opened

//Start up of the Game

start=start1();

start2(start, njob, name, oldsc);

if(capt(cpch, n)){

cout<<"You Failed the Captcha"<<endl;

cout<<"Exiting....."<<endl;

exit(0);

}

//Attack values depend on xp and class

atk=atkcalc(njob,xp);

smobs(bosses);

if (njob==49) //If the class chosen is Mage

{

compsc();

cont=gfight(rclass, magehp, gobhp, atk, cd, xp,

gatk, rgob, score, 0, 1, 2, 3 ); //First Goblin Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=sfight(rclass, magehp, skelhp, atk, cd, xp,

satk, rskel, score, 0, 1, 2, 3);//Skeleton Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=ofight(rclass, magehp, ogrehp, atk, cd, xp,

oatk, rogre, score, 0, 1, 2, 3, bosses);//Ogre Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=sGfght(rclass, magehp, gskhp, atk, cd, xp,

gsatk, rgskel, score, 0, 1, 2, 3, bosses);

//^Skeleton Giant Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=dMfght(rclass, magehp, dmhp, atk, cd, xp,

dMatk, rDMage, score, 0, 1, 2, 3, bosses);

//^Dark Mage Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=dfight(rclass, magehp, draghp, atk, cd, xp,

datk, rdrag, score, 0, 1, 2, 3, bosses);

//^Dragon fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

}

// End of Mage Code

if (njob==50) //If the class chosen is Ranger

{

compsc();

cont=gfight(rclass, rhp, gobhp, atk, cd, //First Goblin Fight

xp, gatk, rgob, score, 4,5,6,7);

if (cont=false){

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and changes attack value

cont=sfight(rclass, rhp, skelhp, atk, cd, xp,

satk, rskel, score, 4, 5, 6, 7); //Skeleton Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=ofight(rclass, rhp, ogrehp, atk, cd, xp,

oatk, rogre, score, 4, 5, 6, 7, bosses); //Ogre Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=sGfght(rclass, rhp, gskhp, atk, cd, xp,

gsatk, rgskel, score, 4, 5, 6, 7, bosses);

//^Skeleton Giant Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=dMfght(rclass, rhp, dmhp, atk, cd, xp,

dMatk, rDMage, score, 4, 5, 6, 7, bosses);

//^Dark Mage Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=dfight(rclass, rhp, draghp, atk, cd, xp,

datk, rdrag, score, 4, 5, 6, 7, bosses);

//^Dragon fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

}

// End of Ranger Code

if (njob==51) //If the class chosen is Fighter

{

compsc();

cont=gfight(rclass, fhp, gobhp, atk, cd, xp, //First Goblin Fight

gatk, rgob, score, 8,9,10,11);

atk=atkcalc(njob,xp); //Levels up and changes attack value

if(cont=false){

exit(0);

}

else compsc();

cont=sfight(rclass, fhp, skelhp, atk, cd, xp,

satk, rskel, score, 8, 9, 10, 11); //Skeleton Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=ofight(rclass, fhp, ogrehp, atk, cd, xp,

oatk, rogre, score, 8, 9, 10, 11, bosses); //Ogre Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=sGfght(rclass, fhp, gskhp, atk, cd, xp,

gsatk, rgskel, score, 8, 9, 10, 11, bosses);

//^Skeleton Giant Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=dMfght(rclass, fhp, dmhp, atk, cd, xp,

dMatk, rDMage, score, 8, 9, 10, 11, bosses);

//^Dark Mage Fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

atk=atkcalc(njob,xp); //Levels up and Changes Attack Value

cont=dfight(rclass, fhp, draghp, atk, cd, xp,

datk, rdrag, score, 8, 9, 10, 11, bosses);

//^Dragon fight

if (cont=false){ //Checks for Game Over

exit(0);

}

else compsc();

}

//End of Fighter's Code

//Score Showing in the txt file

oDis(score);

file << oldsc;

file <<fixed << name <<setw(7) << setprecision(2) << score << endl;

file.close();

//Clean up the code, close files, deallocate memory, etc...

//Exit stage right

return 0;

}

char start1(){

char start;

cout<<"Dungeon and Dragons AI ver"<<endl;

cout<<"Enter 1 to Start, or Enter 2 for more information"<<endl;

cout<<"Enter Anything Else to show the Previous Score"<<endl;

cin>>start;

return start;

}

void start2(char start, char &njob, string &name, string oldsc){

int vstart=start-48; //Converts into menu options with Type Casting

if (vstart==2) //Choice 2 Gives more information about the game

{

cout<<"This is based off the board game Dungeons and Dragons"<<endl;

cout<<"The objective of the game is to clear the dungeon"<<endl;

cout<<"To clear the Dungeon you must kill 5 main monsters"<<endl;

cout<<"Each monster is weak to a certain attribute"<<endl;

cout<<"Read the game information to learn more about this"<<endl;

cout<<"You are to pick from 3 different classes"<<endl;

start=1;

}

else if (vstart==1) // After Starting, Player chooses a class

{

cout<<"Enter Your Name for Scoreboard"<<endl;

cin>>name;

cout<<"Pick a Class"<<endl;

cout<<"1=Mage, 2=Ranger, or 3=Fighter"<<endl;

cout<<"Enter in the Number Associated "<<endl;

cin>>njob;

while (njob>51 || njob<48)

{

cout<<"Please Enter your Class Again"<<endl;

cin>>njob;

}

}

else

{

cout<<oldsc<<endl;

}

}

bool capt(vector <int> &a, int n){

int s; //Smallest Number

cout<<"Bot Check"<<endl;

for(int i=0; i<n; i++){

a[i]=rand()%90+10; //[10,99]

}

a.shrink\_to\_fit();

for(int i=0; i<n; i++){

cout<<a[i]<<endl;

}

cout<<"What is the smallest number?"<<endl;

cin>>s;

for(int i=0; i<n-1;i++){

int minval=a[i];

int idx=i;

for(int j=i+1;j<n;j++){

if(minval>a[j]){

minval=a[j];

idx=j;

}

}

a[idx]=a[i];

a[i]=minval;

}

if(s==a[0]){

return false;

}

else{

return true;

}

}

void smobs(string bosses[][4]){

cout<<"Mob List"<<endl;

for(int i=0; i<2; i++){

for(int j=0; j<3; j++){

cout<<setw(15)<<bosses[i][j]<<"->";

}

cout<<endl;

}

}

int atkcalc(int njob, int xp){

int atk;

atk=(njob==49) ? 2+2\*xp:(njob==50)? 2+pow(2,xp):4+xp;

return atk;

}

void compsc(){

static int cscore=0; //Completion Score

cout<<"Your Completion Score is "<<cscore<<endl;

cscore++;

}

bool gfight(string rclass[], int &hp, int gobhp,int atk,int cd,int &xp,

int gatk, int rgob, float &score, int move1, int move2,

int move3, int move4){

bool cont;

do{ //Battle Begins!

cout<<"You enter in the Dungeon and around the corner you see"<<endl;

cout<<"A goblin approaches"<<endl;

cout<<"Battle"<<endl;

int dmg1;

cout<<fixed;

cout<<"The Goblin has "<<gobhp<<" Hp"<<endl;

cout<<"You have "<<hp<<"Hp"<<endl;

cout<<"You have "<<rgob;

cout<<" rounds left before reinforcements"<<endl;

cout<<"Pick your ability "<<endl;

cout<<"Enter 1 for "<<rclass[move1]<<endl;

cout<<"Enter 2 for "<<rclass[move2]<<endl;

cout<<"Enter 3 for "<<rclass[move3]<<endl;

cout<<"Enter 4 for "<<rclass[move4]<<endl;

cin>>dmg1;

switch(dmg1){ //Menu for picking abilities

case 1: gobhp=gobhp-(atk\*cd);break;

case 2: gobhp=gobhp-(atk/cd);break;

case 3: gobhp=gobhp-atk;break;

case 4: gobhp=gobhp-atk;break;

default: cout<<"Pick one of the numbers provided"<<endl;

cin>>dmg1;break;

}

score-=1;

rgob-=1;

hp-=gatk;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && rgob>0 && gobhp>0);

for(int i=1; i<25; i++) //For Visual Seperation of the End Game

{

cout<<"\*";

}

cout<<endl;

if (gobhp<=0){ //If Kills the Goblin

cout<<"Congrats You have killed the Goblin"<<endl;

cout<<"You have gained a Level"<<endl;

xp=xp+1;

cont=true;

}

if (rgob<=0){ //If Reinforcements came

cout<<"Goblin Reinforcements have came"<<endl;

cout<<"Game Over"<<endl;

score=score/2;

cont=false;

}

if (hp<=0){ //If Player Died

cout<<"You have Died"<<endl;

cout<<"Game Over"<<endl;

score=score/4;

cont=false;

}

if(cont==true) //continues down the dungeon if goblin was killed

{

cout<<"Down the Dungeon you go"<<endl;

}

return cont;

}

bool sfight(string rclass[], int &hp, int bosshp,int atk,int cd,int &xp,

int aboss, int rboss, float &score, int move1, int move2,

int move3, int move4){

bool cont;

do{ //Battle Begins!

cout<<"You went further down the dungeon"<<endl;

cout<<"A Skeleton approaches"<<endl;

cout<<"Battle"<<endl;

int dmg1;

cout<<fixed;

cout<<"The Skeleton has "<<bosshp<<" Hp"<<endl;

cout<<"You have "<<hp<<"Hp"<<endl;

cout<<"You have "<<rboss;

cout<<" rounds left before reinforcements"<<endl;

cout<<"Pick your ability "<<endl;

cout<<"Enter 1 for "<<rclass[move1]<<endl;

cout<<"Enter 2 for "<<rclass[move2]<<endl;

cout<<"Enter 3 for "<<rclass[move3]<<endl;

cout<<"Enter 4 for "<<rclass[move4]<<endl;

cin>>dmg1;

switch(dmg1){ //Menu for picking abilities

case 1: bosshp=bosshp-(atk/cd);break;

case 2: bosshp=bosshp-atk;break;

case 3: bosshp=bosshp-atk;break;

case 4: bosshp=bosshp-(atk\*cd);break;

default: cout<<"What ability did you use?, hurry you are dying"

<<endl;

cin>>dmg1;break;

}

score-=1;

rboss-=1;

hp-=aboss;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && rboss>0 && bosshp>0);

for(int i=1; i<25; i++) //For Visual Seperation of the End Game

{

cout<<"\*";

}

cout<<endl;

if (bosshp<=0){ //If Kills the Skeleton

cout<<"Congrats You have killed the Skeleton"<<endl;

cout<<"You have gained a Level"<<endl;

xp=xp+1;

cont=true;

}

if (rboss<=0){ //If Reinforcements came

cout<<"Skeleton Reinforcements have came"<<endl;

cout<<"Game Over"<<endl;

score=score/2;

cont=false;

}

if (hp<=0){ //If Player Died

cout<<"You have Died"<<endl;

cout<<"Game Over"<<endl;

score=score/4;

cont=false;

}

if(cont==true) //continues down the dungeon if skeleton was killed

{

cout<<"Down the Dungeon you go"<<endl;

}

return cont;

}

bool ofight(string rclass[], int &hp, int bosshp,int atk,int cd,int &xp,

int aboss, int rboss, float &score, int move1, int move2,

int move3, int move4, string bosses[][4]){

bool cont;

do{ //Battle Begins!

cout<<"You went further down the dungeon"<<endl;

cout<<"A "<<bosses[0][2]<<"approaches"<<endl;

cout<<"Battle"<<endl;

int dmg1;

cout<<fixed;

cout<<"The "<<bosses[0][2]<<" has "<<bosshp<<" Hp"<<endl;

cout<<"You have "<<hp<<"Hp"<<endl;

cout<<"You have "<<rboss;

cout<<" rounds left before reinforcements"<<endl;

cout<<"Pick your ability "<<endl;

cout<<"Enter 1 for "<<rclass[move1]<<endl;

cout<<"Enter 2 for "<<rclass[move2]<<endl;

cout<<"Enter 3 for "<<rclass[move3]<<endl;

cout<<"Enter 4 for "<<rclass[move4]<<endl;

cin>>dmg1;

switch(dmg1){ //Menu for picking abilities

case 1: bosshp=bosshp-(atk\*cd);break;

case 2: bosshp=bosshp-atk;break;

case 3: bosshp=bosshp-atk;break;

case 4: bosshp=bosshp-(atk/cd);break;

default: cout<<"What ability did you use?, hurry you are dying"

<<endl;

cin>>dmg1;break;

}

score-=1;

rboss-=1;

hp-=aboss;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && rboss>0 && bosshp>0);

for(int i=1; i<25; i++) //For Visual Seperation of the End Game

{

cout<<"\*";

}

cout<<endl;

if (bosshp<=0){ //If Kills the Ogre

cout<<"Congrats You have killed the "<<bosses[0][2]<<endl;

cout<<"You have gained a Level"<<endl;

xp=xp+1;

cont=true;

}

if (rboss<=0){ //If Reinforcements came

cout<<bosses[0][2]<<"Reinforcements have came"<<endl;

cout<<"Game Over"<<endl;

score=score/2;

cont=false;

}

if (hp<=0){ //If Player Died

cout<<"You have Died"<<endl;

cout<<"Game Over"<<endl;

score=score/4;

cont=false;

}

if(cont==true) //continues down the dungeon if ogre was killed

{

cout<<"Down the Dungeon you go"<<endl;

}

return cont;

}

bool sGfght(string rclass[], int &hp, int bosshp,int atk,int cd,int &xp,

int aboss, int rboss, float &score, int move1, int move2,

int move3, int move4, string bosses[][4]){

bool cont;

do{ //Battle Begins!

cout<<"You went further down the dungeon"<<endl;

cout<<"A "<<bosses[1][0]<<"approaches"<<endl;

cout<<"Battle"<<endl;

int dmg1;

cout<<fixed;

cout<<"The "<<bosses[1][0]<<" has "<<bosshp<<" Hp"<<endl;

cout<<"You have "<<hp<<"Hp"<<endl;

cout<<"You have "<<rboss;

cout<<" rounds left before reinforcements"<<endl;

cout<<"Pick your ability "<<endl;

cout<<"Enter 1 for "<<rclass[move1]<<endl;

cout<<"Enter 2 for "<<rclass[move2]<<endl;

cout<<"Enter 3 for "<<rclass[move3]<<endl;

cout<<"Enter 4 for "<<rclass[move4]<<endl;

cin>>dmg1;

switch(dmg1){ //Menu for picking abilities

case 1: bosshp=bosshp-(atk/cd);break;

case 2: bosshp=bosshp-atk;break;

case 3: bosshp=bosshp-atk;break;

case 4: bosshp=bosshp-(atk\*cd);break;

default: cout<<"What ability did you use?, hurry you are dying"

<<endl;

cin>>dmg1;break;

}

score-=1;

rboss-=1;

hp-=aboss;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && rboss>0 && bosshp>0);

for(int i=1; i<25; i++) //For Visual Seperation of the End Game

{

cout<<"\*";

}

cout<<endl;

if (bosshp<=0){ //If Kills the Skeleton Giant

cout<<"Congrats You have killed the "<<bosses[1][0]<<endl;

cout<<"You have gained a Level"<<endl;

xp=xp+1;

cont=true;

}

if (rboss<=0){ //If Reinforcements came

cout<<bosses[1][0]<<"Reinforcements have came"<<endl;

cout<<"Game Over"<<endl;

score=score/2;

cont=false;

}

if (hp<=0){ //If Player Died

cout<<"You have Died"<<endl;

cout<<"Game Over"<<endl;

score=score/4;

cont=false;

}

if(cont==true) //continues down the dungeon if skeleton giant was killed

{

cout<<"Down the Dungeon you go"<<endl;

}

return cont;

}

bool dMfght(string rclass[], int &hp, int bosshp,int atk,int cd,int &xp,

int aboss, int rboss, float &score, int move1, int move2,

int move3, int move4, string bosses[][4]){

bool cont;

do{ //Battle Begins!

cout<<"You went further down the dungeon"<<endl;

cout<<"A "<<bosses[1][1]<<"approaches"<<endl;

cout<<"Battle"<<endl;

int dmg1;

cout<<fixed;

cout<<"The "<<bosses[1][1]<<" has "<<bosshp<<" Hp"<<endl;

cout<<"You have "<<hp<<"Hp"<<endl;

cout<<"You have "<<rboss;

cout<<" rounds left before reinforcements"<<endl;

cout<<"Pick your ability "<<endl;

cout<<"Enter 1 for "<<rclass[move1]<<endl;

cout<<"Enter 2 for "<<rclass[move2]<<endl;

cout<<"Enter 3 for "<<rclass[move3]<<endl;

cout<<"Enter 4 for "<<rclass[move4]<<endl;

cin>>dmg1;

switch(dmg1){ //Menu for picking abilities

case 1: bosshp=bosshp-(atk/cd);break;

case 2: bosshp=bosshp-(atk/cd);break;

case 3: bosshp=bosshp-(atk\*cd);break;

case 4: bosshp=bosshp-(atk/cd);break;

default: cout<<"What ability did you use?, hurry you are dying"

<<endl;

cin>>dmg1;break;

}

score-=1;

rboss-=1;

hp-=aboss;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && rboss>0 && bosshp>0);

for(int i=1; i<25; i++) //For Visual Seperation of the End Game

{

cout<<"\*";

}

cout<<endl;

if (bosshp<=0){ //If Kills the dark mage

cout<<"Congrats You have killed the "<<bosses[1][1]<<endl;

cout<<"You have gained a Level"<<endl;

xp=xp+1;

cont=true;

}

if (rboss<=0){ //If Reinforcements came

cout<<bosses[1][1]<<"Reinforcements have came"<<endl;

cout<<"Game Over"<<endl;

score=score/2;

cont=false;

}

if (hp<=0){ //If Player Died

cout<<"You have Died"<<endl;

cout<<"Game Over"<<endl;

score=score/4;

cont=false;

}

if(cont==true) //continues down the dungeon if the dark mage was defeated

{

cout<<"Down the Dungeon you go"<<endl;

cout<<"Fighting the Dark Mage has taught you how to self heal"<<endl;

cout<<"You can now self heal and block at the same time"<<endl;

cout<<"Added on the next Update"<<endl;

}

return cont;

}

bool dfight(string rclass[], int &hp, int bosshp,int atk,int cd,int &xp,

int aboss, int rboss, float &score, int move1, int move2,

int move3, int move4, string bosses[][4]){

bool cont;

do{ //Battle Begins!

cout<<"You are nearing the end of the dungeon"<<endl;

cout<<"A "<<bosses[1][2]<<"approaches"<<endl;

cout<<"Battle"<<endl;

int dmg1;

cout<<fixed;

cout<<"The "<<bosses[1][2]<<" has "<<bosshp<<" Hp"<<endl;

cout<<"You have "<<hp<<"Hp"<<endl;

cout<<"You have "<<rboss;

cout<<" rounds left before reinforcements"<<endl;

cout<<"Pick your ability "<<endl;

cout<<"Enter 1 for "<<rclass[move1]<<endl;

cout<<"Enter 2 for "<<rclass[move2]<<endl;

cout<<"Enter 3 for "<<rclass[move3]<<endl;

cout<<"Enter 4 for "<<rclass[move4]<<endl;

cout<<"Enter 5 to Block and Heal"<<endl;

cin>>dmg1;

switch(dmg1){ //Menu for picking abilities

case 1: bosshp=bosshp-(atk/cd);break;

case 2: bosshp=bosshp-(atk\*cd);break;

case 3: bosshp=bosshp-(atk/cd);break;

case 4: bosshp=bosshp-(atk/cd);break;

case 5: hp+=aboss+atk;break;

default: cout<<"What ability did you use?, hurry you are dying"

<<endl;

cin>>dmg1;break;

}

score-=1;

rboss-=1;

hp-=aboss;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && rboss>0 && bosshp>0);

for(int i=1; i<25; i++) //For Visual Seperation of the End Game

{

cout<<"\*";

}

cout<<endl;

if (bosshp<=0){ //If Kills the dragon

cout<<"Congrats You have killed the "<<bosses[1][2]<<endl;

cout<<"You have gained a Level"<<endl;

xp=xp+1;

cont=true;

}

if (rboss<=0){ //If Reinforcements came

cout<<bosses[1][2]<<"Reinforcements have came"<<endl;

cout<<"Game Over"<<endl;

score=score/2;

cont=false;

}

if (hp<=0){ //If Player Died

cout<<"You have Died"<<endl;

cout<<"Game Over"<<endl;

score=score/4;

cont=false;

}

if(cont==true) //continues down the dungeon if dragon was killed

{

cout<<"You Returned Home with your Glory"<<endl;

}

return cont;

}

void disS(float score){

score=score/92;

cout<<"Your Decimal Score is:"<<endl;

cout<<score<<endl;

}

void disS(int score){

cout<<"Your Numeric Score is:"<<endl;

cout<<score<<endl;

}

void oDis(float score){

int score2=score+1-1;

int useri; //Userinput

cout<<"Do you want to display your score in numeric or by percentage"<<endl;

cout<<"Enter 1 for decimal, 2 for numeric, 3 for both"<<endl;

cin>>useri;

switch (useri){

case 1:disS(score);break;

case 2:disS(score2);break;

case 3:disS(score);disS(score2);break;

default:cout<<"Enter it in again"<<endl;cin>>useri;break;

}

}

bool capt2(){

cout<<"Bot Checker 2"<<endl;

int n=10;

int a[n];

int useri;

for(int i=0; i<n; i++){

a[i]=rand()%90+10; //[10,99]

}

for(int i=0; i<n; i++){

cout<<a[i]<<endl;

}

int i, j;

for (i = 0; i < n-1; i++)

for (j = 0; j < n-i-1; j++)

if (a[j] > a[j+1])

swap(a[j], a[j+1]);

for(int i=0; i<n; i++){

cout<<a[i]<<endl;

}

cout<<"Pick Any Number that Exist"<<endl;

cin>>useri;

int idx=-1;

for(int i=0;i<n;i++){

if(a[i]==useri)return true;

}

}