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/Fall-2020-CSC17-A/tree/master/Projects/Project%201>

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. In a sense technically I am the dungeon master because I coded everything for the computer, and you are to pass the dungeon.

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

My entire main approach was to make the project from CIS 5 more efficient, and better with the knowledge of CIS 17A. I already had a pretty well functioning game with some minor bugs per say in files and how big my project was it originally was more than 800 lines of code. However due to the power of structures, pointers, and more knowledge on fstream, I was able to make the project 659 Lines of code. In addition I wanted to revamp the entire file system I had orginally, to something much more efficient and smooth in a sense of the user can check the score.txt file and it will always show the last score. This was vs the old version in which it would have a chunk of information rather than just one smooth old score.

The Program itself

New Lines of Code: Aprox 550

I say aprox because I spent a lot of time playing around with my old code using some of the new knowledge, such as putting strings in structures seeing the limits of files, in fact ver 1.5 is completely me playing around and me messing with the game figuring out new ways to use code that I haven’t think up, essentially I would call it limit testing. But in terms of the new code you can visually see, I would say around 200 new lines of code compared to my original CIS 5 Project, of course I deleted a lot of code but that was because structures allowed me to declare multiple variables for the minions and their stats. For example I didn’t have to keep declaring a minion’s attack, hp, rounds multiple times rather I could just make a nested structure or even an array of structures to trim up all that space of declaration and change.

Lines of Code: 659

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 checkout sheet is in another file + in this file to save both time and in case the file somehow gets corrupted.**

Check-Off

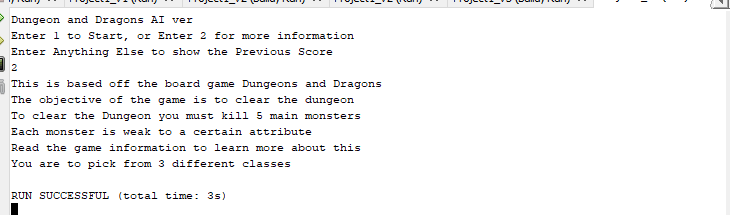
**CSC/CIS 17A Project 1 Check-Off Sheet**

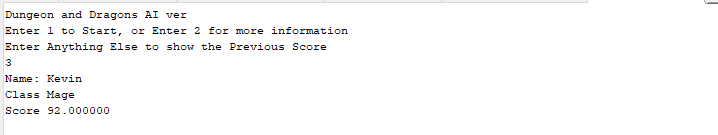
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Chapter** | **Section** | **Concept** | **Points for** | **Location in** | **Comments** |
|  |  |  | **Inclusion** | **Code** |  |
|  |  |  |  |  |  |
| **9** |  | **Pointers/Memory Allocation** |  |  |  |
|  | 1 | Memory Addresses |  |  |  |
|  | 2 | Pointer Variables | 5 | 570 |  |
|  | 3 | Arrays/Pointers | 5 | 535-562 |  |
|  | 4 | Pointer Arithmetic |  |  |  |
|  | 5 | Pointer Initialization |  |  |  |
|  | 6 | Comparing |  |  |  |
|  | 7 | Function Parameters | 5 | 43 |  |
|  | 8 | Memory Allocation | 5 | 535-562+644+647 |  |
|  | 9 | Return Parameters | 5 | 563 |  |
|  | 10 | Smart Pointers |  |  |  |
|  |  |  |  |  |  |
| **10** |  | **Char Arrays and Strings** |  |  |  |
|  | 1 | Testing |  |  |  |
|  | 2 | Case Conversion |  |  |  |
|  | 3 | C-Strings | 10 | 659 |  |
|  | 4 | Library Functions |  |  |  |
|  | 5 | Conversion |  |  |  |
|  | 6 | Your own functions |  |  |  |
|  | 7 | Strings | 10 | 571, 574, 57 |  |
|  |  |  |  |  |  |
| **11** |  | **Structured Data** |  |  |  |
|  | 1 | Abstract Data Types |  |  |  |
|  | 2 | Data |  |  |  |
|  | 3 | Access |  |  |  |
|  | 4 | Initialize |  |  |  |
|  | 5 | Arrays | 5 | 123-126 |  |
|  | 6 | Nested | 5 | 536-562 | Bosses is nested inside Mobs |
|  | 7 | Function Arguments | 5 | 41 |  |
|  | 8 | Function Return | 5 | 563 |  |
|  | 9 | Pointers | 5 | 536-562 |  |
|  | 10 | Unions \*\*\*\* |  |  |  |
|  | 11 | Enumeration | 5 | 580-596 |  |
|  |  |  |  |  |  |
| **12** |  | **Binary Files** |  |  |  |
|  | 1 | File Operations |  |  |  |
|  | 2 | Formatting | 2 | 61-62 |  |
|  | 3 | Function Parameters | 2 | 44-45 |  |
|  | 4 | Error Testing |  |  |  |
|  | 5 | Member Functions | 2 | 123-126 |  |
|  | 6 | Multiple Files | 2 | 61-62 | The Display function displays their viability in the program |
|  | 7 | Binary Files | 5 | 62 |  |
|  | 8 | Records with Structures | 5 | 123-126 |  |
|  | 9 | Random Access Files | 5 | 651 |  |
|  | 10 | Input/Output Simultaneous | 2 | 61-62 |  |

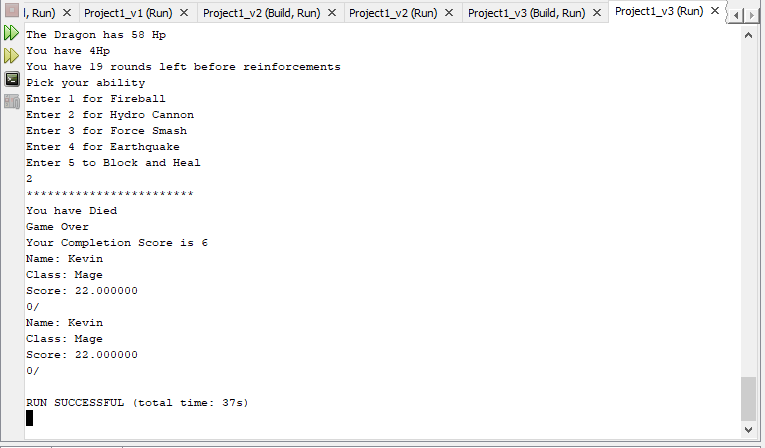
Total 100

Page 1

Pictures of the game functioning(Essentially all menu options and then some.) However full gameplay will not be shown as that is a lot of pictures of basically the same product. Plus I don’t want to spoil any of the answers now do I 😊.







FlowChart both individual file and in here  
it looks better as a pdf.

**Author:**

Kevin Mai

**Finished on:**

10

/25/2020, 5:00 PM

**Purpose:**

Simplified Dungeons and

Dragons

**System Libraries:**

I/O Objects

cMath

iomanip

fstream

cstdlib

ctime

string

Using Standard Namespace

**User Libraries:**

mobs.h

heroes.h

info.h

**Global Constants:**

None

**Function Prototypes:**

start1, start2, smobs,

atkcalc, compsc, gfight,

sfight, ofight, sGfight,

dmfight,

dfight, mobvalues, game.

destroy, display, inBin

Main

Declare Variables:

score,

fhp,

rhp, job

njob, start, name,

cont, board, binsc

function start1;

function start2;

function game

function inbin

function display

close all files

return 0;

# Start1

start1

Display Info

user input start

variable

return start

variable

The Beginning Information shown to players

start2

start variable

start==2

start==1

start=3

More Info

Starts

Game

Shows

past score

in txt file

True

**Prompt for Input**

njob is

or 2 or

1

3

njob<"1" or

njob>"3"

**Prompt for Input**

njob is

3

or 2 or

1

True

True

True

njob=="1"

njob=="2"

njob=="3"

output

atk

output

atk

output

atk

Start Menu + Picking a Class

return void;

game

**Declared Variables:**

hp, xp, cont, atk, \*easy,

rclass, bosses,

cd, moves1, moves2,

moves3, moves4

njob=mage

njob=range

false

false

moves=fight

moves=mage

moves=range

atk calc

boss1

boss2

boss3

boss 4

boss 5

boss 6

exit(0)

death message

display

Text

Check generally

bossfight for the

bossfight code,

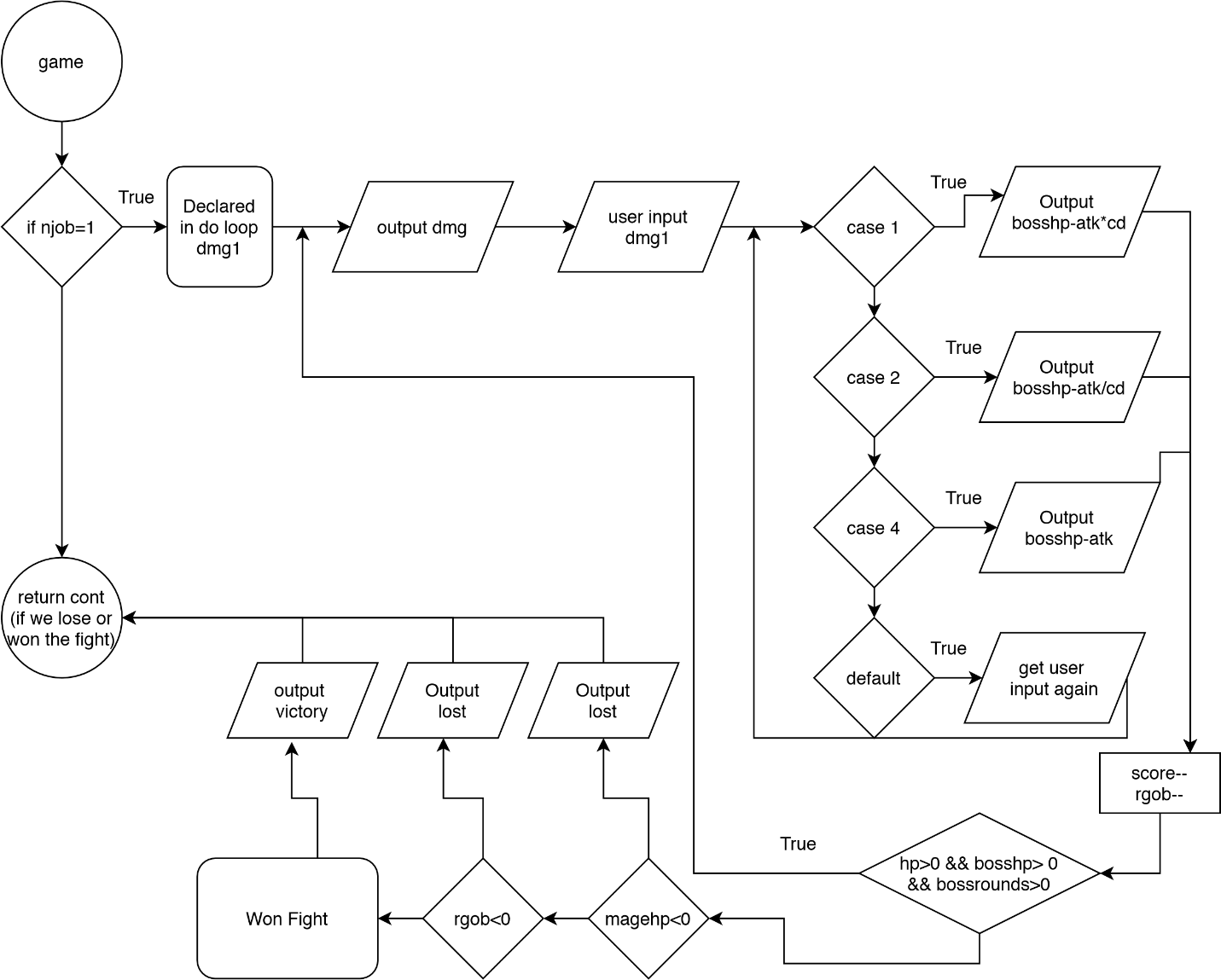
each boss is a

function

**Gameplay**

This function holds the gameplay

of the game



# Display

The Function puts stuff in the txt and bin files while cout the things they would have put in the txt and bin file.

Display

shows name, class, score

puts data in

score.bin, score.txt

page 1

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.

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 and score.bin

The Code itself

/\*

\* File: main.cpp

\* Author: Kevin Mai

\* Created on October 14, 2020, 11:00AM

\* 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

using namespace std;

//User Libraries

#include "mobs.h"

#include "heroes.h"

#include "info.h"

//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 &, fstream &);

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 &,

float &, int , int , int , int , string [][4], Mobs \*);

bool sfight (string [], int &, int ,int ,int &,

float &, int , int , int , int , string [][4], Mobs \*);

bool ofight (string [], int &, int ,int ,int &,

float &, int , int , int , int , string [][4], Mobs \*);

bool sGfght (string [], int &, int ,int ,int &,

float &, int , int , int , int , string [][4], Mobs \*);

bool dMfght (string [], int &, int ,int ,int &,

float &, int , int , int , int , string [][4], Mobs \*);

bool dfight(string [], int &, int ,int ,int &,

float &, int , int , int , int , string [][4], Mobs \*);

Mobs \*mobvalues();

void game(float &, char );

void destroy(Mobs \*);

void display(fstream &, string, char, float);

void inBin(fstream &, string, char, float);

//Execution of Code Begins here

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

//Declare all variables for this function

float score;

int magehp, //The hp value of the mage

fhp, // The HP of the fighter

rhp, //The HP value of the ranger

job; //job

char njob, //Job represented in a number

start; // Start of the game Menu

string name; //Name of the Player

bool cont;

fstream board;

fstream binsc;

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

binsc.open("score.bin", ios::in | ios::out | ios::binary);

//Initialize the code

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

magehp=100;

fhp=300;

rhp=150;

score=100;

//Scoreboard called score.txt

//Start up of the Game

start=start1();

start2(start, njob, name, board);

//Attack values depend on xp and class

game(score, njob);

//Score Showing in the txt file

display(board, name, njob, score);

display(binsc, name, njob, score);

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

board.close();

binsc.close();

//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,fstream &file){

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

do{

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;

exit(0);

}

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

{

Info info[3];

for(int i=0; i<4; i++){

getline(file, info[i].data);

cout<<info[i].data<<endl;

}

exit(0);

}

}while(vstart!=1);

}

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 atk,int cd,int &xp,

float &score, int move1, int move2,

int move3, int move4, string bosses[][4], Mobs \*x){

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 "<<x->gob->hp<<" Hp"<<endl;

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

cout<<"You have "<<x->gob->rounds;

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: x->gob->hp=x->gob->hp-(atk\*cd);break;

case 2: x->gob->hp=x->gob->hp-(atk/cd);break;

case 3: x->gob->hp=x->gob->hp-atk;break;

case 4: x->gob->hp=x->gob->hp-atk;break;

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

cin>>dmg1;break;

}

score-=1;

x->gob->rounds-=1;

hp-=x->gob->atk;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && x->gob->rounds>0 && x->gob->hp>0);

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

{

cout<<"\*";

}

cout<<endl;

if (x->gob->hp<=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 (x->gob->rounds<=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 atk,int cd,int &xp,

float &score, int move1, int move2,

int move3, int move4, string bosses[][4], Mobs \*x){

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 "<<x->skele->hp<<" Hp"<<endl;

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

cout<<"You have "<<x->skele->rounds;

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: x->skele->hp=x->skele->hp-(atk/cd);break;

case 2: x->skele->hp=x->skele->hp-atk;break;

case 3: x->skele->hp=x->skele->hp-atk;break;

case 4: x->skele->hp=x->skele->hp-(atk\*cd);break;

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

<<endl;

cin>>dmg1;break;

}

score-=1;

x->skele->rounds-=1;

hp-=x->skele->atk;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && x->skele->rounds>0 && x->skele->hp>0);

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

{

cout<<"\*";

}

cout<<endl;

if (x->skele->hp<=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 (x->skele->rounds<=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 atk,int cd,int &xp,

float &score, int move1, int move2,

int move3, int move4, string bosses[][4], Mobs \*x){

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 "<<x->ogre->hp<<" Hp"<<endl;

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

cout<<"You have "<<x->ogre->rounds;

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: x->ogre->hp=x->ogre->hp-(atk\*cd);break;

case 2: x->ogre->hp=x->ogre->hp-atk;break;

case 3: x->ogre->hp=x->ogre->hp-atk;break;

case 4: x->ogre->hp=x->ogre->hp-(atk/cd);break;

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

<<endl;

cin>>dmg1;break;

}

score-=1;

x->ogre->rounds-=1;

hp-=x->ogre->atk;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && x->ogre->rounds>0 && x->ogre->hp>0);

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

{

cout<<"\*";

}

cout<<endl;

if (x->ogre->hp<=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 (x->ogre->rounds<=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 atk,int cd,int &xp,

float &score, int move1, int move2,

int move3, int move4, string bosses[][4], Mobs \*x){

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 "<<x->gskele->hp<<" Hp"<<endl;

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

cout<<"You have "<<x->gskele->rounds;

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: x->gskele->hp=x->gskele->hp-(atk/cd);break;

case 2: x->gskele->hp=x->gskele->hp-atk;break;

case 3: x->gskele->hp=x->gskele->hp-atk;break;

case 4: x->gskele->hp=x->gskele->hp-(atk\*cd);break;

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

<<endl;

cin>>dmg1;break;

}

score-=1;

x->gskele->rounds-=1;

hp-=x->gskele->atk;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && x->gskele->rounds>0 && x->gskele->hp>0);

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

{

cout<<"\*";

}

cout<<endl;

if (x->gskele->hp<=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 (x->gskele->rounds<=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 atk,int cd,int &xp,

float &score, int move1, int move2,

int move3, int move4, string bosses[][4], Mobs \*x){

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 "<<x->dmage->hp<<" Hp"<<endl;

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

cout<<"You have "<<x->dmage->rounds;

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: x->dmage->hp=x->dmage->hp-(atk/cd);break;

case 2: x->dmage->hp=x->dmage->hp-(atk/cd);break;

case 3: x->dmage->hp=x->dmage->hp-(atk\*cd);break;

case 4: x->dmage->hp=x->dmage->hp-(atk/cd);break;

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

<<endl;

cin>>dmg1;break;

}

score-=1;

x->dmage->rounds-=1;

hp-=x->dmage->atk;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && x->dmage->rounds>0 && x->dmage->hp>0);

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

{

cout<<"\*";

}

cout<<endl;

if (x->dmage->hp<=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 (x->dmage->rounds<=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 atk,int cd,int &xp,

float &score, int move1, int move2,

int move3, int move4, string bosses[][4], Mobs \*x){

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 "<<x->dragon->hp<<" Hp"<<endl;

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

cout<<"You have "<<x->dragon->rounds;

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: x->dragon->hp=x->dragon->hp-(atk/cd);break;

case 2: x->dragon->hp=x->dragon->hp-(atk\*cd);break;

case 3: x->dragon->hp=x->dragon->hp-(atk/cd);break;

case 4: x->dragon->hp=x->dragon->hp-(atk/cd);break;

case 5: hp+=x->dragon->atk+atk;break;

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

<<endl;

cin>>dmg1;break;

}

score-=1;

x->dragon->rounds-=1;

hp-=x->dragon->atk;

//Stops the loops when someone dies or reinforcements come

}while(hp>0 && x->dragon->rounds>0 && x->dragon->hp>0);

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

{

cout<<"\*";

}

cout<<endl;

if (x->dragon->hp<=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 (x->dragon->rounds<=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;

}

Mobs \*mobvalues(){

//Boss Values

Mobs \*x=new Mobs;

x->gob=new Boss;

x->skele=new Boss;

x->ogre=new Boss;

x->gskele=new Boss;

x->dmage=new Boss;

x->dragon=new Boss;

x->gob->hp=15;

x->skele->hp=20;

x->ogre->hp=30;

x->gskele->hp=56;

x->dmage->hp=40;

x->dragon->hp=100;

//Attack Values

x->gob->atk=1;

x->skele->atk=2;

x->ogre->atk=5;

x->gskele->atk=10;

x->dmage->atk=20;

x->dragon->atk=20;

//Amount of Rounds to beat the bosses

x->gob->rounds=9;

x->skele->rounds=5;

x->ogre->rounds=5;

x->gskele->rounds=5;

x->dmage->rounds=10;

x->dragon->rounds=20;

return x;

}

void game(float &score, char njob){

int hp=(njob==49)?100:(njob==50)?150:300; //Declared Hp based on class

int xp=1;

bool cont;

int atk;

Mobs \*easy=mobvalues();

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"};

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

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

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

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

int moves1, moves2, moves3, moves4; //Used to Determine rclass index

smobs(bosses);

if(njob==49){

moves1=MAGEF;

moves2=MAGEH;

moves3=MAGEP;

moves4=MAGEE;

}

else if(njob==50){

moves1=RANGEF;

moves2=RANGEH;

moves3=RANGEP;

moves4=RANGEE;

}

else if(njob==51){

moves1=FIGHTF;

moves2=FIGHTH;

moves3=FIGHTP;

moves4=FIGHTE;

}

atk=atkcalc(njob,xp);

compsc();

cont=gfight(rclass, hp, atk, cd, xp,

score, moves1, moves2, moves3, moves4,bosses, easy); //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, hp, atk, cd, xp, score, moves1, moves2, moves3, moves4,

bosses, easy);//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, hp, atk, cd, xp,

score, moves1, moves2, moves3, moves4, bosses, easy);//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, hp, atk, cd, xp,score, moves1, moves2, moves3,

moves4, bosses, easy);

//^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, hp, atk, cd, xp, score, moves1, moves2, moves3,

moves4, bosses, easy);

//^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, hp, atk, cd, xp, score, moves1, moves2, moves3, moves4,

bosses, easy);

//^Dragon fight

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

exit(0);

}

else compsc();

destroy(easy);

}

void destroy(Mobs \*x){

delete x;

}

void display(fstream &file, string name, char njob, float score){

string role=(njob==49)?"Mage":(njob==50)?"Ranger":"Fighter";

file.seekg(0L, ios::beg);

file << fixed;

file << "Name: " << name << endl;

file << "Class " << role << endl;

file << "Score " << score << endl <<endl ;

cout << fixed;

cout << "Name: " << name << endl;

cout << "Class: " << role << endl;

cout << "Score: " << score << endl << "0/" << endl;

}