**CERTIFICATE**

This is to certify that this computer science project is a bonified project of \_\_\_

As a part of fulfilment of the practical examination, he has completed this project under my supervision to the best of my satisfaction.

**Ms. \_\_**

**(Computer Teacher)**

**ACKNOWLEDGEMENT**

I express my sincere thanks and gratitude to my computer science teacher**, Ms. \_\_**, for her valuable guidance, constant supervision and personal interest taken at every stage in my studies.

**OBJECTIVE AND APPLICATION DESCRIPTION**

1. **OBJECTIVE**

To create a two player game using C++.

1. **DESCRIPTION**

This is a turn based Role playing game. This game is played by 2 users/players. Each user is required to create their own character. This character then gets stored in the database.

Each character can be given a name and a type can be selected from the options. The users can also select a pre-saved character. Each character has different types of attacks.

1. **CONCEPT USED**

Handling of binary files using Object Oriented Program in C++ language.

**SOURCE** **CODE**

#include<fstream.h>

#include<conio.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

fstream file,temp;

class game // Class Declaration

{

private:

char nm[20];

int level;

long int exp;

char type[20];

char attack[3][20];

int deaths,kills;

float kdr;

int ap1[5];

int ap2[5];

int ap3[5];

public:

void assatt();

void getstat() // class input function

{

cout<<"\nEnter name of player: ";

gets(nm);

cout<<"\nEnter type: ";

gets(type);

kills=0,deaths=0,kdr=0,level=0,exp=0;

assatt(); //type of attacks

}

void showstat() // class output function

{

cout<<"\n The stats are as follows:";

cout<<"\nPlayer name: "<<nm;

cout<<"\nType: "<<type;

cout<<"\nLevel: "<<level;

cout<<"\nExperience: "<<exp;

cout<<"\nKills: "<<kills;

cout<<"\nDeaths: "<<deaths;

}

char\* retnm()

{

return (nm);

}

char\* reta1() //Return attack name

{

return attack[0];

}

char\* reta2()

{

return attack[1];

}

char\* reta3()

{

return attack[2];

}

int retap1() //return attack power for the game

{

randomize();

int j=random(5);

return ap1[j];

}

int retap2()

{

randomize();

int j=random(5);

return ap2[j];

}

int retap3()

{

randomize();

int j=random(5);

return ap3[j];

}

void win() //Condition for win

{

kills++;

if(deaths==0)

cout<<"\n Kill Death Ratio of "<<nm<<" is infinite as the player's deaths are 0";

else if(kills!=0&&deaths!=0)

{ kdr=float(kills)/float(deaths);

cout<<"\n Kill Death ratio of "<<nm<<" is "<<kdr;

}

exp+=10;

if (exp>=100)

{

exp=0;

level++;

}

}

void lose() //Condition for lose

{

deaths++;

if(kills==0)

cout<<"\n Kill Death Ratio of "<<nm<<" is 0 as the player's kills are 0";

else if(kills!=0&&deaths!=0)

{ kdr=float(kills)/float(deaths);

cout<<"\n Kill Death Ratio of "<<nm<<" is "<<kdr;

}

if(level>=0)

{

exp-=5;

if (exp<=0)

{

exp=99;

level--;

}

}

}

}s1,s2;

void game :: assatt() // Assigning the attack for each type

{

if (strcmpi(type,"Foot Soldier")==0)

{

strcpy(attack[0],"Punch");

strcpy(attack[1],"Shoot");

strcpy(attack[2],"Stab");

int k=0;

for(int x=0;x<5;x++)

{

ap1[x]=k+5;

}

ap2[0]=0;

ap2[1]=20;

ap2[2]=0;

ap2[3]=40;

ap2[4]=70;

ap3[0]=0;

ap3[1]=0;

ap3[2]=0;

ap3[3]=50;

ap3[4]=80;

}

else if(strcmpi(type,"ninja")==0)

{

strcpy(attack[0],"Star");

strcpy(attack[1],"Sword");

strcpy(attack[2],"Sneak from behind");

int k=0;

for(int x=0;x<5;x++)

{

ap1[x]=k+5;

}

ap2[0]=0;

ap2[1]=15;

ap2[2]=0;

ap2[3]=50;

ap2[4]=60;

ap3[0]=0;

ap3[1]=0;

ap3[2]=0;

ap3[3]=50;

ap3[4]=80;

}

else if(strcmpi(type,"wizard")==0)

{

strcpy(attack[0],"Stupefy");

strcpy(attack[1],"Crucio");

strcpy(attack[2],"Avada Kedavra");

int k=0;

for(int x=0;x<5;x++)

{

ap1[x]=k+5;

}

ap2[0]=0;

ap2[1]=30;

ap2[2]=0;

ap2[3]=50;

ap2[4]=55;

ap3[0]=0;

ap3[1]=0;

ap3[2]=0;

ap3[3]=40;

ap3[4]=100;

}

else if (strcmpi(type,"viking")==0)

{

strcpy(attack[0],"Tackle");

strcpy(attack[1],"Sheild Bash");

strcpy(attack[2],"Axe Charge");

int k=0;

for(int x=0;x<5;x++)

{

ap1[x]=k+5;

}

ap2[0]=0;

ap2[1]=20;

ap2[2]=0;

ap2[3]=45;

ap2[4]=65;

ap3[0]=0;

ap3[1]=0;

ap3[2]=0;

ap3[3]=50;

ap3[4]=80;

}

else if (strcmpi(type,"Robot")==0)

{

strcpy(attack[0],"Launch Fists");

strcpy(attack[1],"Grab and Throw");

strcpy(attack[2],"Gun Down");

int k=0;

for(int x=0;x<5;x++)

{

ap1[x]=k+5;

}

ap2[0]=0;

ap2[1]=20;

ap2[2]=0;

ap2[3]=40;

ap2[4]=70;

ap3[0]=0;

ap3[1]=0;

ap3[2]=0;

ap3[3]=50;

ap3[4]=80;

}

}

void create() // File Creation

{

char ans;

file.open("game.dat",ios::out|ios::binary);

do

{

s1.getstat();

file.write((char\*)&s1,sizeof(s1));

cout<<"\nEnter another player?(y/n) ";

cin>>ans;

}while(ans=='y'||ans=='Y');

file.close();

}

void addp() // Adding new player in the file

{

char ans;

file.open("game.dat",ios::app|ios::binary);

do

{

s1.getstat();

file.write((char\*)&s1,sizeof(s1));

cout<<"\nEnter another player? ";

cin>>ans;

}while(ans=='y'||ans=='Y');

file.close();

}

void editp() // modifying player data

{

file.open("game.dat",ios::in|ios::binary);

temp.open("temp.dat",ios::out|ios::binary);

char nm[30];

cout<<"\nEnter player name ";

gets(nm);

while(file.read((char\*)&s1,sizeof(s1)))

{

if(strcmpi(s1.retnm(),nm)==0)

{

cout<<"\nEnter new details \n";

s1.getstat();

temp.write((char\*)&s1,sizeof(s1));

}

else

temp.write((char\*)&s1,sizeof(s1));

}

file.close();

temp.close();

remove("game.dat");

rename("temp.dat","game.dat");

}

void show() // Output data

{

file.open("game.dat", ios::in|ios::binary);

int ch;

cout<<"\n Please choose an option \n1. Show data of all players \n2. Show data of a particular player ";

cin>>ch;

if(ch==1)

{

while(file.read((char\*)&s1,sizeof(s1)))

{

s1.showstat();

}

}

if(ch==2)

{

cout<<"\nEnter name: ";

char name[20];

gets(name);

while(file.read((char\*)&s1,sizeof(s1)))

{

if(strcmpi(name,s1.retnm())==0)

{

s1.showstat();

}

}

}

file.close();

}

void del() // Deleting a player

{

file.open("game.dat",ios::in|ios::binary);

temp.open("temp.dat",ios::out|ios::binary);

char name[20];

int f=0;

cout<<"\nEnter player name to be deleted ";

gets(name);

while(file.read((char\*)&s1,sizeof(s1)))

{

if(strcmpi(s1.retnm(),name)!=0)

temp.write((char\*)&s1,sizeof(s1));

else

f=1;

}

if(f==0)

cout<<"\nPlayer not found ";

else

cout<<"\nDeleted ";

file.close();

temp.close();

remove("game.dat");

rename("temp.dat","game.dat");

}

void duel(game p1, game p2) //DUEL, The brain behind the whole working of the game

{

int p=1,p1h=100,p2h=100,miss;

cout<<"\n Player one starts! \n";

do

{

randomize();

clrscr();

cout<<' '<<p1.retnm()<<" health : "<<p1h;

cout<<' '<<p2.retnm()<<" health : "<<p2h;

cout<<"\n Choose your attack player "<<p;

if (p==1)

{

cout<<"\nPress 1 for "<<p1.reta1();

cout<<"\nPress 2 for "<<p1.reta2();

cout<<"\nPress 3 for "<<p1.reta3();

cout<<"\nAttack chosen : ";

miss=p2h;

int ach;

cin>>ach;

switch(ach)

{

case 1: p2h-=p1.retap1();

break;

case 2: p2h-=p1.retap2();

break;

case 3: p2h-=p1.retap3();

break;

}

if(miss==p2h)

cout<<" MISS \n";

else

cout<<" HIT \n";

getch();

p=2;

}

else

{

cout<<"\nPress 1 for "<< p2.reta1();

cout<<"\nPress 2 for "<< p2.reta2();

cout<<"\nPress 3 for "<< p2.reta3();

cout<<"\nAttack chosen : ";

miss=p1h;

int ach;

cin>>ach;

switch(ach)

{

case 1: p1h-=p2.retap1();

break;

case 2: p1h-=p2.retap2();

break;

case 3: p1h-=p2.retap3();

break;

}

if(miss==p1h)

cout<<" MISS \n";

else

cout<<" HIT \n";

cout<<endl;

getch();

p=1;

}

}while (p1h>0&&p2h>0);

if(p1h<=0)

{

cout<<"\n Player two wins ";

p2.win();

p1.lose();

}

else

{

cout<<"\n Player one wins ";

p1.win();

p2.lose();

}

cout<<endl;

p1.showstat();

p2.showstat();

file.open("game.dat",ios::in|ios::binary);

temp.open("temp.dat",ios::out|ios::binary);

while(file.read((char\*)&s1,sizeof(s1)))

{

if(strcmpi(s1.retnm(),p2.retnm())==0)

{

s1=p2;

}

else if(strcmpi(s1.retnm(),p1.retnm())==0)

{

s1=p1;

}

temp.write((char\*)&s1,sizeof(s1));

}

file.close();

temp.close();

remove("game.dat");

rename("temp.dat","game.dat");

}

void main()

{

clrscr();

game p1,p2;

clrscr();

char ans;

int ch;

do

{

clrscr();

cout<<"\n \t\t WELCOME TO RPG SIMULATOR 2017! ";

cout<<"\n\n\n";

cout<<"\n1. Database options ";

cout<<"\n2. Duel ";

cout<<"\nEnter Option : ";

cin>>ch;

switch(ch)

{

case 1: cout<<"\n1.Create the file from start? "; //Database options

cout<<"\n2.Add new player ";

cout<<"\n3.Delete a player ";

cout<<"\n4.show data ";

cout<<"\n5.Modify player data ";

cout<<"\nEnter your choice : ";

int choice;

cin>>choice;

switch(choice)

{

case 1: cout<<"\nChoose one from the following by typing your choice below:\n1.Foot Soldier\n2.Ninja\n3.Wizard\n4.Viking\n5.Robot";

cout<<"\nEach type has its own set of attacks. Each type has his own strengths and weakness. All types will have 3 attacks each";

cout<<"\nThe first attack causes the least damage but has 100% accuracy. Where as the last attack causes the heaviest damage but has a low success rate.";

create();

break;

case 2: cout<<"\nChoose one from the following by typing your choice below:\n1.Foot Soldier\n2.Ninja\n3.Wizard\n4.Viking\n5.Robot";

cout<<"\nEach type has its own set of attacks. Each type has his own strengths and weakness. All types will have 3 attacks each";

cout<<"\nThe first attack causes the least damage but has 100% accuracy. Where as the last attack causes the heaviest damage but has a low success rate.";

addp();

break;

case 3: del();

break;

case 4: show();

break;

case 5: editp();

break;

default: cout<<"Enter an option ";

break;

}

break;

case 2: cout<<"\n Player one enter your name: "; //Game options

char pnm[20];

gets(pnm);

int f1=0, f2=0;

file.open("game.dat",ios::in|ios::binary);

while(file.read((char\*)&s1, sizeof(s1)))

{

if (strcmpi(s1.retnm(),pnm)==0)

{

p1=s1;

f1=1;

break;

}

}

if(f1==0)

{ cout<<"Player not found ";

getch();

file.close();

break;

}

cout<<"\n Player two enter your name: ";

gets(pnm);

file.seekp(0);

while(file.read((char\*)&s2, sizeof(s2)))

{

if (strcmpi(s2.retnm(),pnm)==0)

{

p2=s2;

f2=1;

break;

}

}

if(f2==0)

{ cout<<"Player not found ";

getch();

file.close();

break;

}

file.close();

cout<<' '<<p1.retnm()<<" VERSUS "<<' '<<p2.retnm();

cout<<"\n LET THE DUEL BEGIIINNNNN!!!!!! \n";

getch();

clrscr();

duel(p1,p2);

break;

default: cout<<"\nENTER SOMETHING sensible ";

}

cout<<"\n\nDo you want to continue? y/n ";

cin>>ans;

}while(ans=='y'||ans=='Y');

getch();

}

**BIBLIOGRAPHY**

* Computer Science Study Material, Delhi Public School, Vasant Kunj, New Delhi
* “Computer Science with C++” by Sumita Arora
* www.google.com