#include<iostream.h>

#include<fstream.h>

#include<conio.h>

#include<stdlib.h>

#include<process.h>

#include<iomanip.h>

#include<dos.h>

#include<time.h>

#include<stdio.h>

#define loop(i,j) for(i=0;i<j;i++)

int add[3]={2,4,2};

class score

{

public:

int score;

char pname[30];

void disp(){cout<<"Name:"<<pname<<"\tScore:"<<score<<endl;}

};

class g2048

{

int a[4][4];

score S;

int temp,i,j,k,q,r,g,f1,f2,f3;

char p[10];

public:

g2048();

void up();

void down();

void left();

void right();

void scoresave();

int ARN(); //ARN-addition of random no. to the game

void output();

int check();

void in();

int retf3()

{

return f3;

}

};

void g2048::in()

{

f1=0;

do

{

f1=check();

if(!f1)

{

f3=0;

break;

}

cin>>p;

if(p[0]=='w')

up();

else if(p[0]=='s')

down();

else if(p[0]=='d')

right();

else if(p[0]=='a')

left();

else

{

cout<<"\nwrong input";

}

}while(f1);

}

g2048::g2048()

{

f3=1;

loop(i,4)

loop(j,4)

a[i][j]=0;

cout<<"press w to move the tiles up \ns to move them down \na to move them left \nd to move them right\n";

cout<<"Enter your name"<<endl;

cin>>S.pname;

S.score=0;

output();

}

int g2048::ARN()

{

int k=0;

loop(i,4)

loop(j,4)

if(a[i][j]==0)

k=1;

if(k==0)

return 0;

q=random(4);

r=random(4);

while(a[q][r]!=0)

{

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

{

for(j=0;j<4;j++)

{

if(a[i][j]==0)

{

q=i;r=j;

}

}

}

}

if(k==1)

{

g=random(3);

a[q][r]=add[g];

}

return 0;

}

void g2048::left()

{

loop(i,4)

{

loop(j,3)

{

if(a[i][j]==a[i][j+1])

f1=0;

}

}

loop(i,4)

{

loop(j,4)

{

if(a[i][j]==0)

f1=0;

}

}

if(f1)

{

cout<<"Invalid move";

}

if(!f1)

{

loop(i,4)

{

loop(k,3)

{

j=2-k;

if(a[i][j]==0)

while((j+1)<=3)

{

a[i][j]=a[i][j+1];

a[i][j+1]=0;

j++;

}

}

loop(k,3)

{

j=k;

if(a[i][j]==a[i][j+1])

{

a[i][j]+=a[i][j+1];

S.score+=a[i][j];

a[i][j+1]=0;

j++;

while((j+1)<=3)

{

a[i][j]=a[i][j+1];

a[i][j+1]=0;

j++;

}

}

}

}

}

}

void g2048::right()

{

loop(i,4)

{

loop(j,3)

{

if(a[i][j]==a[i][j+1])

f1=0;

}

}

loop(i,4)

{

loop(j,4)

{

if(a[i][j]==0)

f1=0;

}

}

if(f1)

{

cout<<"Invalid move";

}

if(!f1)

{

loop(i,4)

{

loop(j,3)

{

k=j+1;

if(a[i][k]==0)

while((k-1)>=0)

{

a[i][k]=a[i][k-1];

a[i][k-1]=0;

k--;

}

}

loop(j,3)

{

k=3-j;

if((a[i][k]==a[i][k-1])&&(a[i][k]!=0))

{

a[i][k]+=a[i][k-1];

S.score+=a[i][k];

a[i][k-1]=0;

k--;

while((k-1)>=0)

{

a[i][k]=a[i][k-1];

a[i][k-1]=0;

k--;

}

}

}

}

}

}

void g2048::up()

{

loop(i,4)

{

loop(j,3)

{

if(a[j][i]==a[j+1][i])

f1=0;

}

}

loop(i,4)

{

loop(j,4)

{

if(a[i][j]==0)

f1=0;

}

}

if(f1)

{

cout<<"Invalid move";

}

if(!f1)

{

loop(i,4)

{

loop(k,3)

{

j=2-k;

if(a[j][i]==0)

while((j+1)<=3)

{

a[j][i]=a[j+1][i];

a[j+1][i]=0;

j++;

}

}

loop(k,3)

{

j=k;

if(a[j][i]==a[j+1][i])

{

a[j][i]+=a[j+1][i];

S.score+=a[j][i];

a[j+1][i]=0;

j++;

while((j+1)<=3)

{

a[j][i]=a[j+1][i];

a[j+1][i]=0;

j++;

}

}

}

}

}

}

void g2048::down()

{

loop(i,4)

{

loop(j,3)

{

if(a[j][i]==a[j+1][i])

f1=0;

}

}

loop(i,4)

{

loop(j,4)

{

if(a[i][j]==0)

f1=0;

}

}

if(f1)

{

cout<<"Invalid move";

}

if(!f1)

{

loop(i,4)

{

loop(j,3)

{

k=j+1;

if(a[k][i]==0)

while((k-1)>=0)

{

a[k][i]=a[k-1][i];

a[k-1][i]=0;

k--;

}

}

loop(j,3)

{

k=3-j;

if((a[k][i]==a[k-1][i])&&(a[k][i]!=0))

{

a[k][i]+=a[k-1][i];

S.score+=a[k][i];

a[k-1][i]=0;

k--;

while((k-1)>=0)

{

a[k][i]=a[k-1][i];

a[k-1][i]=0;

k--;

}

}

}

}

}

}

int g2048::check()

{

loop(i,4)

loop(j,4)

if (a[i][j]==2048)

{

cout<<"\nyou have finished the game"<<endl;

return 0;

}

loop(i,4)

loop(j,3)

if((a[i][j]==a[i][j+1]))

return 1;

loop(i,4)

loop(j,3)

if(a[j][i]==a[j][i+1])

return 1;

loop(i,4)

loop(j,4)

if(a[i][j]==0)

return 1;

return 0;

}

void g2048::output()

{

clrscr();

loop(i,4)

{

loop(j,4)

{

cout<<"|";

if(a[i][j]!=0)

{cout<<setw(4)<<a[i][j];}

else if(a[i][j]==0)

{cout<<" ";}

}

cout<<"|"<<endl;

}

delay(300);

gotoxy(10,7);

cout<<"Score: "<<S.score<<endl;

}

void g2048::scoresave()

{

ofstream fout("data.dat",ios::app|ios::binary);

fout.write((char\*)&S,sizeof(S));

fout.close();

}

void opensave()

{

ifstream fin("data.dat",ios::binary);

score ob;

clrscr();

while(fin.read((char\*)&ob,sizeof(ob)))

{

ob.disp();

}

fin.close();

}

void instructions()

{ clrscr();

int x;

cout<<"\t\t\tINSTRUCTIONS\n";

cout<<"The primary objective of this game is to achieve the highest number possible by adding same numbers.";

cout<<"\nThe numbers that you will be adding are the exponents of the number 2 ";

cout<<"\nPress 1 to go back";

cin>>x;

if(x==1)

clrscr();

}

void menu()

{

int i;

do

{

cout<<"\t\t\tMENU:\n";

cout<<"\t\t\t1.Play Game\n";

cout<<"\t\t\t2.Leaderboards\n";

cout<<"\t\t\t3.Instructions\n";

cout<<"\t\t\t4.Exit\n";

cin>>i;

switch(i)

{

case 1: clrscr();

g2048 newgame;

for(;;)

{

newgame.ARN();

newgame.output();

newgame.in();

if(!newgame.retf3())

break;

}

cout<<"Game Over !!!"<<endl;

newgame.scoresave();

break;

case 2: opensave();

break;

case 3:instructions();

break;

case 4: exit(0);

}

}while (i!=4);

}

void spacing()

{

clrscr();

int i,x,y,j;

//this for 2

for(i=1;i<=3;i++)

{ gotoxy(18,i\*6);

loop(j,9)

{delay(40); cout<<"=";}

}

for(i=7;i<=11;i++)

{

gotoxy(26,i);

delay(40);

cout<<"||";

}

for(i=13;i<18;i++)

{

gotoxy(17,i);

delay(40);

cout<<"||";

}

//this for 0

for(i=1;i<=3;i+=2)

{ gotoxy(29,i\*6);

loop(j,9)

{delay(40); cout<<"=";}

}

for(i=7;i<18;i++)

{

gotoxy(37,i);

delay(40);

cout<<"||";

}

for(i=7;i<18;i++)

{

gotoxy(28,i);

delay(40);

cout<<"||";

}

//this is for 4

for(i=7;i<=15;i++)

{

gotoxy(40,i);

delay(40);

cout<<"||";

}

gotoxy(41,16);

loop(j,9)

{delay(40);cout<<"=";}

for(i=13;i<=18;i++)

{

gotoxy(46,i);

delay(40);

cout<<"||";

}

//this for 8

for(i=1;i<=3;i++)

{ gotoxy(54,i\*6);

loop(j,9)

{delay(40);cout<<"=";}

}

for(i=7;i<18;i++)

{

gotoxy(62,i);

delay(40);

cout<<"||";

}

for(i=7;i<18;i++)

{

gotoxy(53,i);

delay(40);

cout<<"||";

}

gotoxy(62,21);

cout<<"1 to enter game";

gotoxy(62,22);

cout<<"0 to exit";

cin>>x;

if(x==1)

menu();

else

exit(0);

getch();

}

int main()

{

clrscr();

spacing();

getch();

return 0;

}

CLASSES USED:

* Score

To store the details of the player name and the score.

## DATA MEMBERS:

* int – score.
* Char – pname[30]

## MEMBER FUNCTIONS :

* **Void disp()**

A function to display pname and score.

* g2048

Contains the various functions required for the implementation of the code of the game.

## DATA MEMBERS:

* int - a[4][4], temp,i,j,k,q,r,g,f1,f2,f3.
* score - S.
* char - p[10].

## MEMBER FUNCTIONS:

* g2048 ()

A constructor used to initialise the variables belonging to the class g2048 to their default values.

* void up ()

A function which enables the user to move the tiles up.\*

* void down ()

A function which enables the user to move the tiles down.\*

* void left ()

A function which enables the user to move the tiles left.\*

* void right ()

A function which enables the user to move the tiles right.\*

* void scoresave()

A function to save the details of the player name and his respective score.

* int ARN()

A function to add a random number.

* **Void output()**

A function to display the game screen.

* **Void check()**

A function which checks if moves are available.

* **Void in()**

A function to get the input from the user.

* **Int retf3()**

A function to return the value of f3.

\* - If moves are available.

FUNCTIONS USED:

* **Void opensave()**

A function to display the players and their score . It implements it using file handling.

* **Void instructions()**

A function that displays the instructions on how to play the game.

* **Void menu()**

This function displays an interactive menu that helps the user to choose from various options available in the game.

* **Void spacing()**