**C Language Project Report**

**2019\_TAQ\_C\_Project\_10**

​​

·

|  |  |
| --- | --- |
| **Name ：** | 2048 |
| **Group ：** | 10 |
| **Members：** | 邰桂田 郭腾宇 宋旭 赵博骞 |
| **Class ：** | 2018 JLU TAQ class of chemistry |
| **Date ：** | July 6th，2019 |

**Content**

[Abstract 3](#_Toc422388671)

[1 Problem Statement 3](#_Toc422388672)

[2 Group 5](#_Toc422388673)

[3 Analysis](#_Toc422388674) 6

4 Design 11

5 Implementation 13

[6 Test and Conclution](#_Toc422388678) 20

[附件：（Code）](#_Toc422388679) 22

**Abstract**

As we all know, recently, the artificial Intelligence is very popular all around the world. Therefore, this term we studied C language, and we design a project as our outcome. The project objective is aimed to create a game that has most of teenagers want to play --- 2048. 2048 is a puzzle game, which is very suitable and popular for all ages. Therefore, we also want to design such a game to enrich our university life.

We all come to an agreement that clearly understanding that making a complete C language project successfully and cooperating with teammates well is more vital than just simply finish it as a class assignment. And we also have the confidence to achieve our goal.

However, we all know that the road to success is not always smooth. During the project, we encountered many difficulties that we had never thought of, and we even wanted to give up at one time. However, through the cooperation and efforts of team members, we finally completed the project.

**1.Problem Statement**

When we selected the project and began to discuss implementation, we found that doing a c project was not as easy as we thought. We had a lot of tough problems. It mainly focuses on the following four aspects:

**(1) How to initialize an interface？**

Interface is the face of the project and it is also the first part. Interface to do good or bad will directly determine the aesthetic degree of the project. So how to design and make a beautiful and useful interface is the first mission we must solve.

**(2) How to control the movement of Numbers with a mouse or keyboard？**

This part is the core of the whole project, which is related to the normal operation of the project. It is also where knowledge and technology are most needed in the whole project. How to make digital bits move across the interface is a big problem that we have to solve.

**(3) How to accumulate and record scores？（Including the current score and the highest score)**

This part is related to the accumulation and calculation of scores and the game decision and other aspects, so it is also very important. It also includes the storage and judgment of two scores (the highest score and the current score).

**(4) How to determine the winner and the end of the process？**

We need to design a conditional program to help us determine whether the game is successful or not. This is also the final part of the whole project. Its completion also marks the end of the project.

After asking these questions, the only thing we need to do is to discuss the research and work together to figure out the solution to the problem and complete our project. Comparing to other similar project，there are plenty of unique advantages that our application contains. First of all, we believe that our app fits the majority’s taste. Our interface is more succinct without any ads or complex pictures and patterns. Players are always tired of dealing with glaring screens and our interface can let them directly get to the information they need. Our Interface has four parts: “Start” to play game, ”Introduction” to introduce our game, ”Instruction” to let you know how to play, ”Exit” to drop out. Second, our game is very similar to the game “2048” we played before. What’s more, our code is succinct. None of them is useless. Last but not least, we used the knowledge we learned in class or after class, such as array, handle, static priority-driven scheduling and so on.

**2.Group Division**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student number | Major | Work Proportion |
| 邰桂田 | 12180905 | Chemistry | 50% |
| 郭腾宇 | 19180223 | Chemistry | 35% |
| 宋旭 | 12180423 | Chemistry | 10% |
| 赵博骞 | 12180316 | Chemistry | 5% |

**Work Proportion**

宋旭 Songxu

Provided a good start for the whole project.

Finish the interface production and design work.

邰桂田 Tai Guitian

Leader of the project and the one with the best coding ability in our group.

In charge of completing the code of movement control.

Deal with the accumulation of scores and the judgment of winning or losing.

Check the whole project

赵博骞 Zhao Boqian

Integration and interface beautification.

Putting the pieces together then make the interface beautiful.

郭腾宇 Guo Tengyu

Rechecking the code

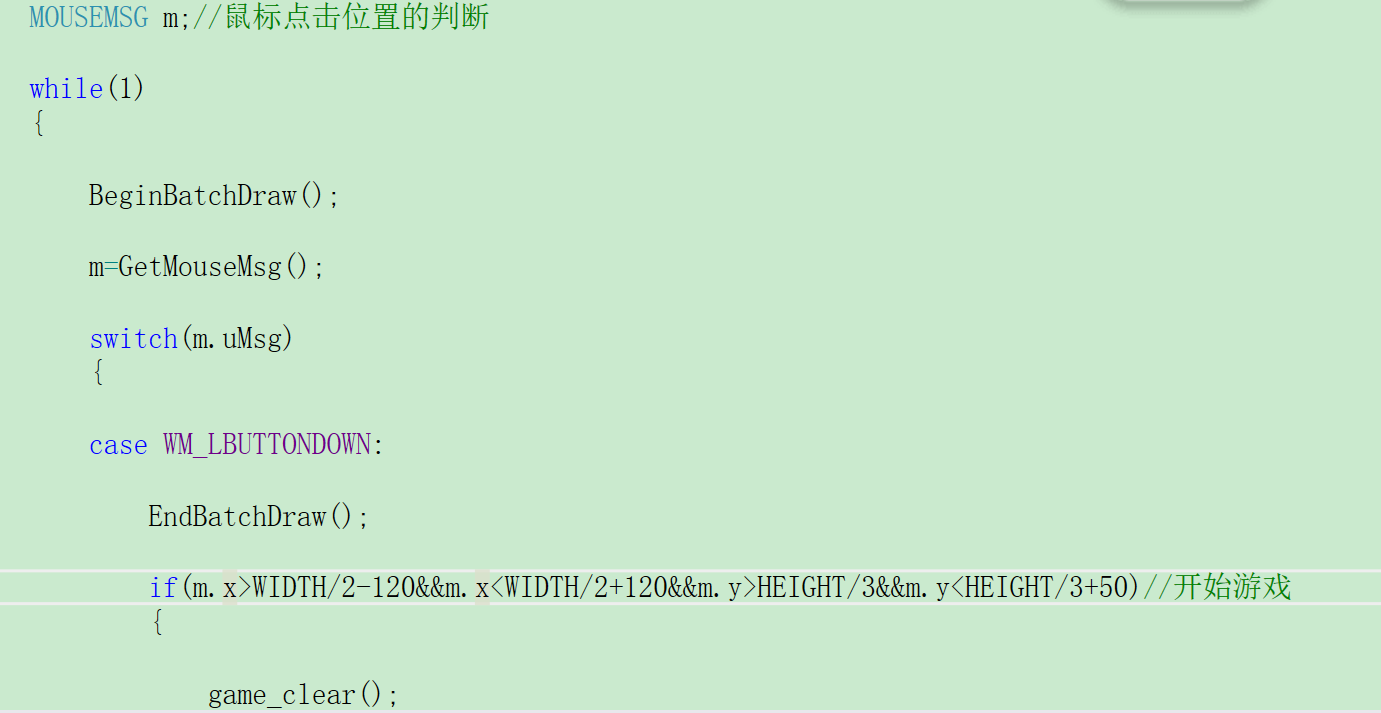
Giving the entire program logic

**3.Analysis**

"2048" is a popular digital game, first released on March 20, 2014.  The original 2048 was first released on GitHub, the original author was Gabriele Cirulli, and was later ported to various platforms.  This game is a new digital game based on the gameplay of "1024" and "Little 3 Legend".

In this game, you can select one of the up, down, left and right directions to slide each time. Each time you slide, all the digital squares will move closer to the sliding direction. The system will also display a number square in the blank space. The same number of squares is close together. When they collide, they will add up. Constantly stacking and finally figuring out the number of 2048 is a success.

**3.1 Login Interface**

When compiling the codes, an initial interface should be shown. The interface shall convey the message of the name of the game which is Welcome to the World of 2048. The login or initial interface has four choices for the player. At this part, we used cursor click to control progress.

**3.2 Main Interface**

For such a classic grid game, we first need to create a matrix.

int map[4][4]={

{0,0,0,0},

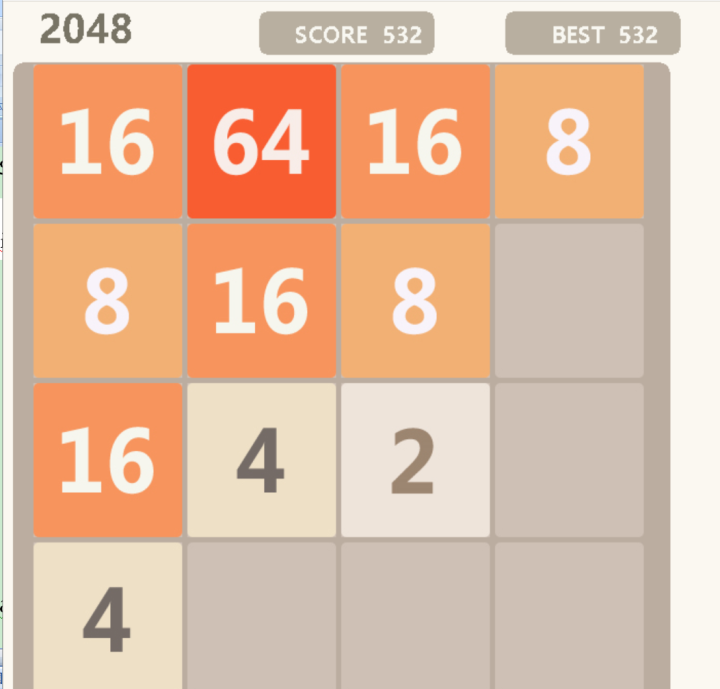
{0,0,0,0},

{0,0,0,0},

{0,0,0,0}

};

For the 16 small parts we take the method of picking up the images, we use the corresponding function “loadimage ,putimage”, put them in the corresponding position. A gray blank picture is equivalent to zero.



For the change in score, we are using a random function “game\_rand”.

while(1)

{

score=0;

game\_rand();/

game\_rand();

cirloop();

void game\_rand()//随机2 4

{

int x,y,temp,num;

srand((int)time(0));

while(1)

{

x=rand()%4;

y=rand()%4;

temp=rand()%5;

if(temp==1)num=4;

else num=2;

if(map[y][x]==0)

{

map[y][x]=num;

break;

}

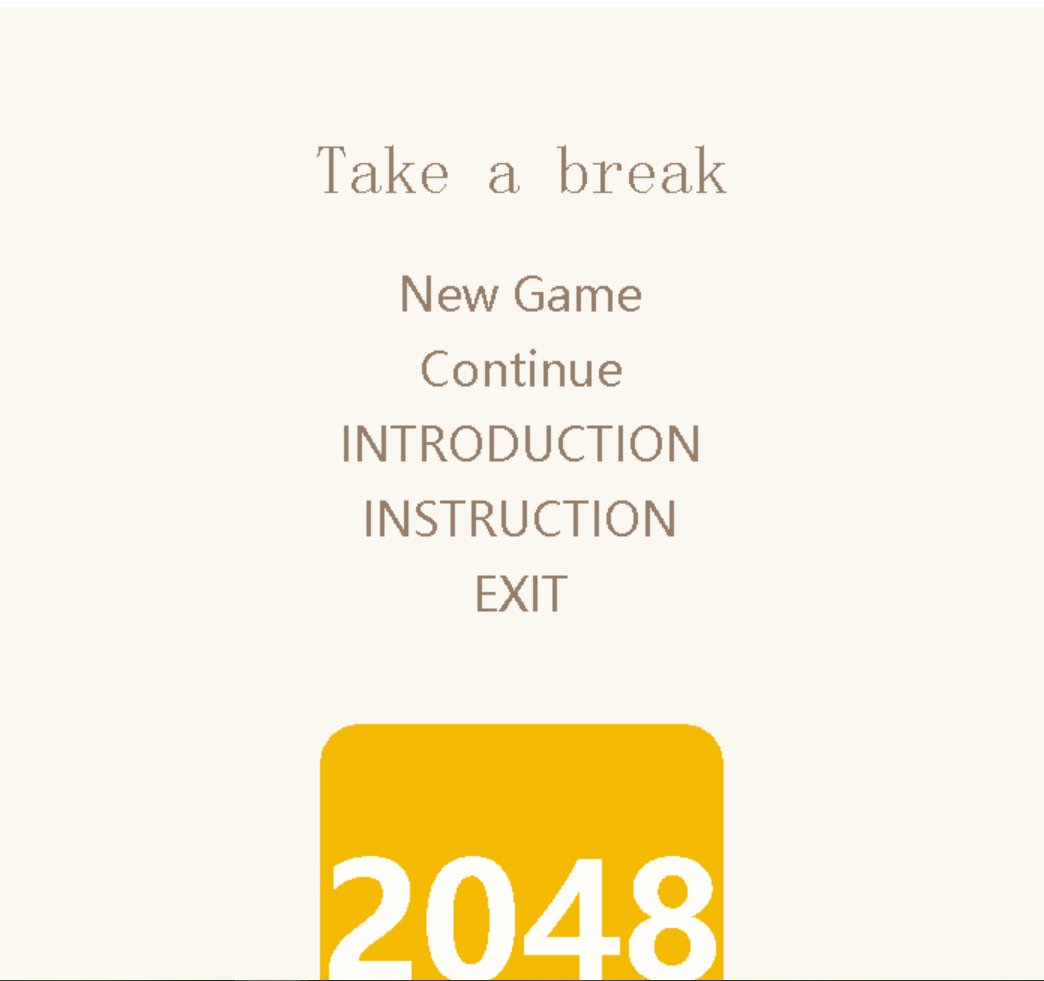
}

}

If you can skillfully compile such a program, you can completely replace the image of this game with the following type.

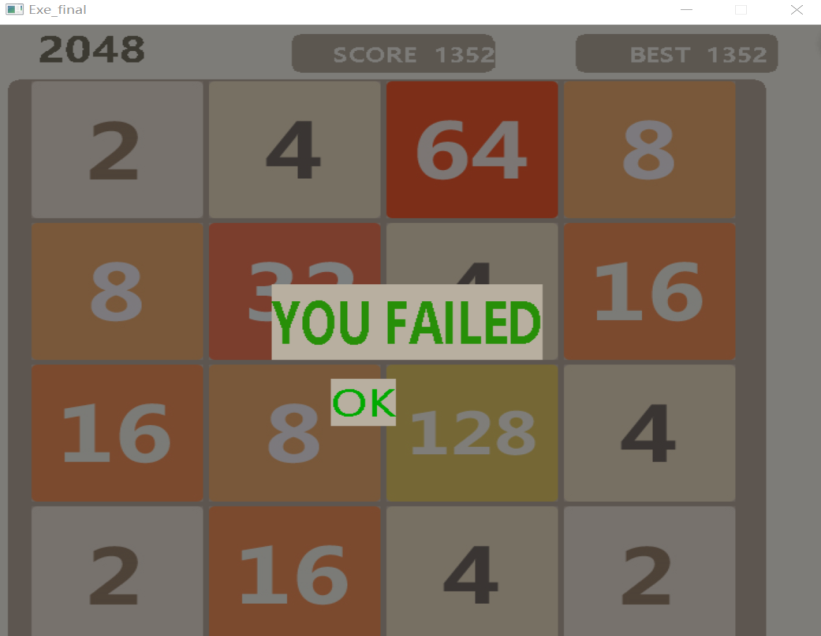


In this game, you just need to use direction keys to add up to your target score. And if you would like to have a break, just press esc key and you will see another interface like this.



You could choose to click “Continue” to continue your game or just click new game to restart a game. For the part of “New Game”, we used game restart function. It will be clarified in the following.

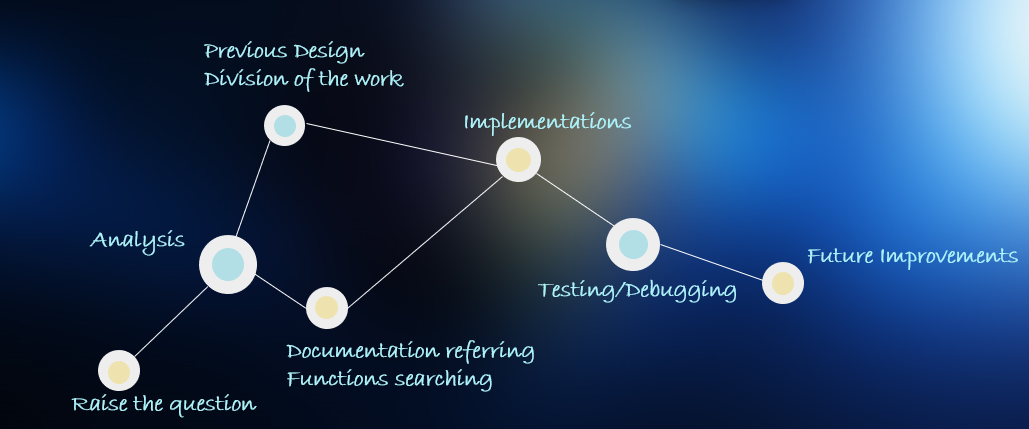
If the entire interface is full of sections that cannot be moved, then you lose.



The screen will show “You Failed” and then you can click the “OK” button to restart a game.

If you succeed, you will see an interface just like the “lost” interface. We only changed “you failed” to “you win”. What’s more, we also add the music file. You might heart some voices when you play our game.

As is shown above, these are merely the core functions of this game, there are severe other designs we have made in order to make the game operable and playable.

**4.Design**

P4.1 Overall Design

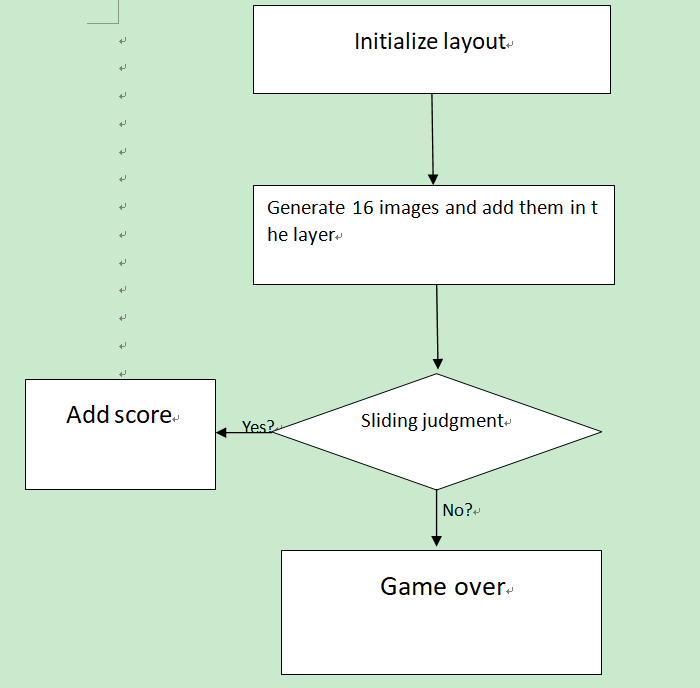
As is shown above, it is our Overall Design.

The initial proposal of the program is raised by a group member. The used code for basketball shooting was abandoned for an error we can’t solve. After the group discussion, we found the game “2048” quite intriguing and amusing. We searched the relative documents, the prospect of imitating this online game seems rather promising. Because the algorithm is quite clear to be seen.

We first raised the question. And then we did the analysis over the whole topic, which made us determined to devote to the program.

Later, we did previous design, division of work, documentation referring and functions searching part simultaneously. And then we each wrote the assigned codes as our group work division, we helped each other through tough times.

Below comes a rough flow chart of our project.



After compiling the codes, the game started, if the assigned region is clicked, an initial interface is shown. You can see a 4 by 4 multicolored grid map. If you press the arrow keys, you can see that the same number will be eliminated after the collision, and a larger number will be generated at the corresponding position.

This is actually a simple summation or an elementary transformation of the matrix.

And here comes to the whole program flow chart.



**5.Implementation**

After showing our design, we would like to analysis the implementation of this program. During the implementation part, we put our design and analysis into real codes with the help of all the information we have obtained about different header fills and different function names. But most importantly, we used the techniques we have learned from the C kiss to think about our program. Since the code is way too long, we would display 3 most important part of the code to show our methodology. However, since the codes have been clearly displayed, further illustration would seem tedious and unnecessary. We would label the important parts of the codes, therefore benefit the readers.

This program includes two important head files, “easy. h” and “graphics. h”. Many important functions are based on the head files, such as “drawtext”,” settextstyle” and so on.

**5.1 Control direction function**

Take the function of controlling the movement to the left as an example.

void control\_left()

{

int n,i,j,k;

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

{

k=0;

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

{

if(map[i][n]!=0)

{

for(j=n;j>k;j--)

{

if(map[i][j-1]==0)

{

map[i][j-1]=map[i][j];

map[i][j]=0;

pass=1;

cirloop();

Sleep(1);

}

else if(map[i][j-1]==map[i][j])

{

map[i][j-1]=2\*map[i][j-1];

score+=map[i][j-1];

map[i][j]=0;

k=j;

cirloop();

Sleep(1);

pass=1;

break;

}

else break;

}

}

}

}

}

We know that it changes by controlling the column vector of the two-dimensional array.

**5.2 Game check function**

For the 2048 game you have to judge the success or failure of the game. Therefore, we need a function to check the game. Here comes the code.

int game\_check()

{

int success=0;

int i,n;

int a=0;//Used for counting numbers in the grid which is not near the same number

if((map[0][0]!=0)&&(map[0][1]!=0)&&(map[0][0]!=0)&&(map[0][0]!=map[0][1])&&(map[0][0]!=map[1][0]))a++;

if((map[3][3]!=0)&&(map[3][2]!=0)&&(map[2][3]!=0)&&(map[3][3]!=map[3][2])&&(map[3][3]!=map[2][3]))a++;

if((map[0][3]!=0)&&(map[0][2]!=0)&&(map[1][3]!=0)&&(map[0][3]!=map[0][2])&&(map[0][3]!=map[1][3]))a++;

if((map[3][0]!=0)&&(map[3][1]!=0)&&(map[2][0]!=0)&&(map[3][0]!=map[3][1])&&(map[3][0]!=map[2][0]))a++;

if((map[0][1]!=0)&&(map[0][0]!=0)&&(map[0][2]!=0)&&(map[0][1]!=map[0][0])&&(map[0][1]!=map[0][2]))a++;

if((map[0][2]!=0)&&(map[0][1]!=0)&&(map[0][3]!=0)&&(map[0][2]!=map[0][1])&&(map[0][2]!=map[0][3]))a++;

if((map[3][1]!=0)&&(map[3][0]!=0)&&(map[3][2]!=0)&&(map[3][1]!=map[3][0])&&(map[3][1]!=map[3][2]))a++;

if((map[3][2]!=0)&&(map[3][1]!=0)&&(map[3][3]!=0)&&(map[3][2]!=map[3][1])&&(map[3][2]!=map[3][3]))a++;

if((map[1][0]!=0)&&(map[0][0]!=0)&&(map[2][0]!=0)&&(map[1][0]!=map[0][0])&&(map[1][0]!=map[2][0]))a++;

if((map[2][0]!=0)&&(map[3][0]!=0)&&(map[1][0]!=0)&&(map[2][0]!=map[1][0])&&(map[2][0]!=map[3][0]))a++;

if((map[1][3]!=0)&&(map[0][3]!=0)&&(map[2][3]!=0)&&(map[1][3]!=map[0][3])&&(map[1][3]!=map[2][3]))a++;

if((map[2][3]!=0)&&(map[3][3]!=0)&&(map[1][3]!=0)&&(map[2][3]!=map[1][3])&&(map[2][3]!=map[3][3]))a++;

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

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

if((map[n][i]!=0)&&(map[n-1][i]!=0)&&(map[n+1][i]!=0)&&(map[n][i-1]!=0)&&(map[n][i+1]!=0)&&(map[n][i]!=map[n+1][i])&&(map[n][i]!=map[n-1][i])&&(map[n][i]!=map[n][i+1])&&(map[n][i]!=map[n][i-1]))a++;

if(a==16)success=-1;//Lose

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

for(n=0;n<4;n++)if(map[n][i]==2048)//Many 2048 show

success=1;//Winning conditions

return success;

}

It means if the number of digital images in the grid reaches 16, then the game fails. Otherwise the game can continue

**5.3 Cycle show function and game clear function**

If you want to refresh the score, you must have a loop function. We chose to design a function called “cirloop”. It is also used to do through arrays.

void cirloop()//Cycle show

{

int i,n;

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

{

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

{

show(i,n,map[n][i],score);// The last parameter is passed in the score.

}

}

}

And game check function is similar to the “cirloop”. It is used to replace all digital pictures with empty pictures.

void game\_clear()//Clear the screen

{

int i,n;

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

{

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

{

map[n][i]=0;

}

}

}

**5.4 Rand function**

At the beginning of the game we need to randomly generate a few “two” or “four”. Here we introduce a random function called game\_rand.

void game\_rand()//Putting number “2”or“4” randomly

{

int x,y,temp,num;

srand((int)time(0));

while(1)

{

x=rand()%4;

y=rand()%4;

temp=rand()%5;

if(temp==1)num=4;//Putting number “4” randomly

else num=2;//Putting number “2” randomly

if(map[y][x]==0)

{

map[y][x]=num;//Putting numbers randomly

break;

}

}

}

After random generation we can start to eliminate.

**5.5 Game function**

I think this part is really important. This part of code will make this game run normally and achieve the most important functions.

void game()//Game initialization and operation

{

loadSound();

while(1)

{

score=0;//Initialize the scores

game\_rand();//Appearing two numbers randomly

game\_rand();

cirloop();//Putting numbers

while(1)

{

if(\_kbhit)//The keys

{

pass=0;

control();//give direction

fflush(stdin);//Clear the input cache

if(pass)game\_rand();//Determine if it can continue to generate random numbers after moving.

if(game\_check())//check to check function

{

if(game\_check()==1)// check to check function

{

if(game\_win()==2)break;// Interface of victory——play again

}

else if(game\_check()==-1)// check to check function {

cirloop();

game\_failed();

break;

}

}

cirloop();//Putting numbers

}

else Sleep(1);

}

game\_clear();//Clear the screen

continue;

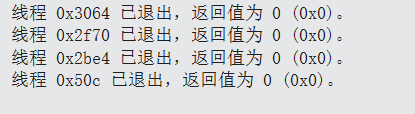
}

}

**6. Test and Conclusion**

After a regular basis of programming, compiling and debugging, we can say that we have mostly finished our initial idea of our project, which is to use our C language skills to try to imitate the online-version. It turned out that we have done it quite well. Except of the definition of the background pictures is rather lower than the commercial one.

Now, we give the test for the project.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project | 2048Game | | | |
| Function | Make the game is same as the traditional “2048” game, with simple operation to get a lot fun from the game | | | |
| Test purpose | Make sure that use that operations can really make the movements. For example, counting the scores/elements’ movement/… | | | |
| Contents | Description | Expected result | Actual result | Test result |
| 001 | Use↑.↓.←.→to control the cursor | The cursor can move with the press of↑.↓.←.→keys. | Same as the expected result | Normal |
| 002 | Make the 2 same elements collide | The two elements will disappear, last a 0 element and a sum element | Same as the expected result | Normal |
| 003 | Same as 002 | Score will be renewed, plus the number of the sum | Same as the expected result | Normal |
| 004 | Play the game until break the record | The “best score” will be renewed | Same as the expected result | Normal |

All in all. So far we have achieved all effects we want to reach. In my opinion, this game is very interesting with high playability. It is not just a puzzle game, it is also a game with some fashion elements, such as we chose the popular music “Chicken, you are so beautiful” as our background music. When you click “start”, you will enter the game and enjoy the experience we offer to you. After testing and debugging, we can say this program is finished as our expectation.

During our C programming, we have learned to cooperate with our teammates, everyone is an essential part within our system, and we can only make progress when everyone is doing their best for the group. Specially thanks to our captain, Tai Guitian, who has undertaken a lot of work alone.

We have also learned quite a lot about the understanding the basic uses of the C language and we have practiced quite a lot about debugging and other knowledge. While researching the functions and header files, we have gained many techniques about programming. Most importantly, we have learned to think like the debugger.

Code

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

#include<time.h>

#include<iostream>

#include<mmsystem.h>

#pragma comment(lib,"WINMM.LIB")

int map[4][4]={

{0,0,0,0},

{0,0,0,0},

{0,0,0,0},

{0,0,0,0}

};//The initial value of the 2048 interface

int WIDTH=700; //Width of the interface

int HEIGHT=650; //Height of the interface

int score;//Calculate the score

int flag=0;//Use for filling the background

int pass;//Detect if it need putting new numbers

void start\_1(); //head interface

void start\_2(); //Game introduction

void start\_3(); //Game instruction

void start\_4(); //Game pulse, etc.

void show\_bk2(); //Renew the score

void show\_bk1(); //Interface background

void show(); //Put elements

void show\_1(); //Initialize the interface

void game\_rand(); //Put number “2” or “4” randomly

void control(); //give direction

void control\_up();//Detect “up” movement

void control\_down();//Detect “down” movement

void control\_left();//Detect “left” movement

void control\_right();//Detect “right” movement

void game();//Initialize the game and run the game

void game\_restart();//Continue the game

int game\_win();//Win interface

void game\_failed();//Fail interface

int game\_check();//Check the game is win or lose

void game\_clear();//The function used for clearing the screen

void cirloop();//Basic loop function, draw on the display function

void show\_bk1()//Background of the game interface

{

setbkcolor(RGB(251,248,241));//”2048”on the top left corner

setfont(50,0,"Microsoft Yahei UI Bold");

settextcolor(RGB(120,116,101));

outtextxy(35,0,"2048");

setfillcolor(RGB(184,175,160));// “SCORE” frame and “BEST” frame on the top right corner

solidroundrect(250,10,420,50,15,15);

setfont(28,0,"Microsoft Yahei UI Bold");

setbkcolor(RGB(184,175,160));

settextcolor(RGB(248,247,241));

outtextxy(285,18,"SCORE");

solidroundrect(490,10,660,50,15,15);

outtextxy(535,18,"BEST");

setfillcolor(RGB(187,173,160));//below game area

solidroundrect(10,58,650,780,20,20);

}

void show\_bk2(int score)//Score renewed

{

char str[10],best\_str[10];

setbkcolor(RGB(184,175,160));

itoa(score,str,10);

outtextxy(370,18,str);//Your score

if(str>=best\_str)itoa(score,best\_str,10);

outtextxy(600,18,best\_str);//Best score

if(flag==0)//The background is only used to read once.

show\_bk1();

flag++;

}

void show(int x,int y,int num,int score)

{

setbkcolor(RGB(251,248,241));//Color of the background

IMAGE img0,img2,img4,img8,img16,img32,img64,img128,img256,img512,img1024,img2048;

show\_bk2(score);

switch(num)//Draw on pictures

{

case 0:loadimage(&img0,"pic\\0.jpg");

putimage(x\*150+30,y\*150+60,&img0);

break;

case 2:loadimage(&img2,"pic\\2.jpg");

putimage(x\*150+30,y\*150+60,&img2);

break;

case 4:loadimage(&img4,"pic\\4.jpg");

putimage(x\*150+30,y\*150+60,&img4);

break;

case 8:loadimage(&img8,"pic\\8.jpg");

putimage(x\*150+30,y\*150+60,&img8);

break;

case 16:loadimage(&img16,"pic\\16.jpg");

putimage(x\*150+30,y\*150+60,&img16);

break;

case 32:loadimage(&img32,"pic\\32.jpg");

putimage(x\*150+30,y\*150+60,&img32);

break;

case 64:loadimage(&img64,"pic\\64.jpg");

putimage(x\*150+30,y\*150+60,&img64);

break;

case 128:loadimage(&img128,"pic\\128.jpg");

putimage(x\*150+30,y\*150+60,&img128);

break;

case 256:loadimage(&img256,"pic\\256.jpg");

putimage(x\*150+30,y\*150+60,&img256);

break;

case 512:loadimage(&img512,"pic\\512.jpg");

putimage(x\*150+30,y\*150+60,&img512);

break;

case 1024:loadimage(&img1024,"pic\\1024.jpg");

putimage(x\*150+30,y\*150+60,&img1024);

break;

case 2048:loadimage(&img2048,"pic\\2048.jpg");

putimage(x\*150+30,y\*150+60,&img2048);

break;

}

}

void show\_1()//Initialize the interface

{

initgraph(WIDTH,HEIGHT);// Width and height of the interface

setbkcolor(RGB(251,248,241));// Background color

settextcolor(RGB(150,126,104));

start\_1();

}

void start\_1()//Head interface

{

cleardevice();

setbkcolor(RGB(251,248,241));

settextcolor(RGB(150,126,104));

setfont(45,0,"方正行黑简体"); //Some word frames

RECT r1 = {0, 0, WIDTH, HEIGHT/3};

drawtext("Welcome to the World of 2048", &r1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

setfont(40,0,"微软雅黑");

RECT r2={WIDTH/2-120,HEIGHT/3,WIDTH/2+120,HEIGHT/3+50};

drawtext("START", &r2, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r3={WIDTH/2-120,HEIGHT/3+50,WIDTH/2+120,HEIGHT/3+100};

drawtext("INTRODUCTION", &r3, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r4={WIDTH/2-120,HEIGHT/3+100,WIDTH/2+120,HEIGHT/3+150};

drawtext("INSTRUCTION", &r4, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r5={WIDTH/2-120,HEIGHT/3+150,WIDTH/2+120,HEIGHT/3+200};

drawtext("EXIT", &r5, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

setfillcolor(RGB(244,186,1));//Paint “2048’ logo

solidroundrect(216,458,484,706,50,50);

setfont(145,0,"Microsoft Yahei UI Bold");

setbkcolor(RGB(244,186,1));

settextcolor(RGB(254,253,249));

outtextxy(216,500,"2048");

setbkcolor(RGB(251,248,241));//Initialize the color of the background

settextcolor(RGB(150,126,104));

setfont(30,0,"Microsoft Yahei UI");

MOUSEMSG m;//Detect the mouse click position

while(1)

{

BeginBatchDraw();

m=GetMouseMsg();

switch(m.uMsg)

{

case WM\_LBUTTONDOWN:

EndBatchDraw();

if(m.x>WIDTH/2-120&&m.x<WIDTH/2+120&&m.y>HEIGHT/3&&m.y<HEIGHT/3+50)//开始游戏

{

game\_clear();

cleardevice();

flag=0;

game();

break;

}

else if(m.x>WIDTH/2-120&&m.x<WIDTH/2+120&&m.y>HEIGHT/3+50&&m.y<HEIGHT/3+100)//游戏介绍Game introduction

{

start\_2();

break;

}

else if(m.x>WIDTH/2-120&&m.y>HEIGHT/3+100&&m.x<WIDTH/2+120&&m.y<HEIGHT/3+150)//操作说明

{

start\_3();

break;

}

else if(m.x>WIDTH/2-120&&m.y>HEIGHT/3+150&&m.x<WIDTH/2+120&&m.y<HEIGHT/3+200)//退出游戏

{

exit(0);

}

break;

}

}

getch();

}

void start\_2()//Game int

{

cleardevice();

RECT C2={60,60,640,790};

drawtext("Game introduction: You can select one of the up, down, left and right directions to slide each time. Each time you slide, all the digital squares will move closer to the sliding direction. The system will also display a number square in the blank space. The same number of squares are close together. When they collide, they will add up. Constantly stacking and finally figuring out the number of 2048 is a success. \n\n　Try to create new records! ! You are too young！！\n\n　　Developer：邰桂田 / 宋旭 / 郭腾宇/ 赵博骞\nPS： For technical reasons, temporarily can't reach 4096 or higher\nThere are still all sorts of weird mistakes, and you don't panic", &C2, DT\_WORDBREAK);

RECT R1={WIDTH-WIDTH+400,HEIGHT-200,WIDTH-200,HEIGHT-2};

drawtext("Return", &R1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

MOUSEMSG m;

while(1)//Return key settings

{

m=GetMouseMsg();

switch(m.uMsg)

{

case WM\_LBUTTONDOWN:

if(m.x>WIDTH-WIDTH+400&&m.x<WIDTH-200&&m.y>HEIGHT-200&&m.y<HEIGHT-2)

start\_1();

}

}

}

void start\_3()//Operation instruction

{

cleardevice();

RECT C3={60,60,640,790};

drawtext("Operation instructions: \n\nLeft shift：A Key/direction key←\n\n Right shift：D Key/direction key→\n\nMove up：W Key/direction key ↑\n\nMove down：S Key/direction key ↓\n\nMain menu：Esc\n\n(Return：Author Robin:Don't think about things too simple(￣^￣)ゞ)", &C3, DT\_WORDBREAK);

RECT R1={WIDTH-WIDTH+200,HEIGHT-200,WIDTH-2,HEIGHT-2};

drawtext("Return", &R1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

MOUSEMSG m;

while(1)// Return key settings

{

m=GetMouseMsg();

switch(m.uMsg)

{

case WM\_LBUTTONDOWN:

if(m.x>WIDTH-WIDTH+200&&m.x<WIDTH-2&&m.y>HEIGHT-200&&m.y<HEIGHT-2)

start\_1();

}

}

}

void start\_4()//Game pulse, etc.

{

cleardevice();

setbkcolor(RGB(251,248,241));

settextcolor(RGB(150,126,104));

setfont(45,0,"方正行黑简体");

RECT r1 = {0, 0, WIDTH, HEIGHT/3};

drawtext("Take a break", &r1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

setfont(40,0,"微软雅黑");

RECT r2={WIDTH/2-120,HEIGHT/3,WIDTH/2+120,HEIGHT/3+50};

drawtext("Continue", &r2, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r3={WIDTH/2-120,HEIGHT/3+50,WIDTH/2+120,HEIGHT/3+100};

drawtext("INTRODUCTION", &r3, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r4={WIDTH/2-120,HEIGHT/3+100,WIDTH/2+120,HEIGHT/3+150};

drawtext("INSTRUCTION", &r4, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r5={WIDTH/2-120,HEIGHT/3+150,WIDTH/2+120,HEIGHT/3+200};

drawtext("EXIT", &r5, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r6={WIDTH/2-120,HEIGHT/3-50,WIDTH/2+120,HEIGHT/3};

drawtext("New Game", &r6, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

setfillcolor(RGB(244,186,1));

solidroundrect(216,478,484,746,50,50);

setfont(145,0,"Microsoft Yahei UI Bold");

setbkcolor(RGB(244,186,1));

settextcolor(RGB(254,253,249));

outtextxy(216,535,"2048");

setbkcolor(RGB(251,248,241));

settextcolor(RGB(150,126,104));

setfont(30,0,"Microsoft Yahei UI");

MOUSEMSG m;

while(1)

{

BeginBatchDraw();

m=GetMouseMsg();

switch(m.uMsg)

{

case WM\_LBUTTONDOWN:

EndBatchDraw();

if(m.x>WIDTH/2-120&&m.x<WIDTH/2+120&&m.y>HEIGHT/3-50&&m.y<HEIGHT/3)

{

cleardevice();

flag=0;

game\_clear();

game();

break;

}

else if(m.x>WIDTH/2-120&&m.x<WIDTH/2+120&&m.y>HEIGHT/3&&m.y<HEIGHT/3+50)

{

cleardevice();

flag=0;

game\_restart();

break;

}

else if(m.x>WIDTH/2-120&&m.x<WIDTH/2+120&&m.y>HEIGHT/3+50&&m.y<HEIGHT/3+100)

{

start\_2();

break;

}

else if(m.x>WIDTH/2-120&&m.y>HEIGHT/3+100&&m.x<WIDTH/2+120&&m.y<HEIGHT/3+150)

{

start\_3();

break;

}

else if(m.x>WIDTH/2-120&&m.y>HEIGHT/3+150&&m.x<WIDTH/2+120&&m.y<HEIGHT/3+200)

{

exit(0);

}

break;

}

}

getch();

}

int game\_win()//Screen of winnings

{

COLORREF ref;

int r,g,b;

int x,y;

char \*yes,\*no,\*win;

srand((int)time(0));

BeginBatchDraw();

for(x=0;x<700;x++)

{

for(y=0;y<900;y++)

{

ref=getpixel(x,y);

r=GetRValue(ref);

g=GetGValue(ref);

b=GetBValue(ref);

putpixel(x,y,RGB(r\*0.5,g\*0.5,b\*0.5));

}

}

FlushBatchDraw();

EndBatchDraw();

yes="YES";

no="NO";

win="YOU WIN !";

for(x=0;x<10;x++)//Show the colorful color of “win”

{

settextcolor(RGB(rand()%255,rand()%255,rand()%255));

settextstyle(100,50,NULL);

outtextxy(50,50,win);

Sleep(100);

}

settextstyle(50,25,"微软雅黑");

settextcolor(GREEN);

outtextxy(200,150,"Continue？");

settextcolor(BLUE);

outtextxy(120,250,yes);

outtextxy(320,250,no);

setfillcolor(RED);

MOUSEMSG m;//Click “continue” or “Exit” by mouse

while(1){//Click “continue” clear the screen and back to 1，click “restart” clear the screen and back to 2

FlushMouseMsgBuffer();

m=GetMouseMsg();

if(m.uMsg==WM\_LBUTTONDOWN)

{

x=m.x;y=m.y;

if(x>120&&x<220&&y>250&&y<300)

{

cleardevice();

return 1;

}//Draw on game function

if(x>320&&x<420&&y>250&&y<300)

{

cleardevice();

return 2;

}// Draw on game function

}

}

}

void game\_failed()//Screen of losing

{

COLORREF ref;

int r,g,b;

int x,y;

srand((int)time(0));

BeginBatchDraw();

for(x=0;x<700;x++)

{

for(y=0;y<900;y++)

{

ref=getpixel(x,y);

r=GetRValue(ref);

g=GetGValue(ref);

b=GetBValue(ref);

putpixel(x,y,RGB(r\*0.5,g\*0.5,b\*0.5));

}

}

FlushBatchDraw();

EndBatchDraw();

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

{

settextcolor(RGB(rand()%255,rand()%255,rand()%255));

settextstyle(80,20,NULL);

outtextxy(WIDTH/3,HEIGHT/2-50,"YOU FAILED");

Sleep(1000);

}

settextstyle(50,0,"微软雅黑");

settextcolor(GREEN);

outtextxy(WIDTH/3+50,HEIGHT/2+50,"OK"); //Show “YOU FAILED”, click “yes” then clear the screen

MOUSEMSG m;

while(1)

{

FlushMouseMsgBuffer();

m=GetMouseMsg();

if(m.uMsg==WM\_LBUTTONDOWN)

{

x=m.x;

y=m.y;

if(x>WIDTH/3+50&&x<WIDTH/3+100&&y>HEIGHT/2+50&&y<HEIGHT/2+100)

{

cleardevice();

break;

}

}

}

}

int loadSound()

{

sndPlaySound("g:\KuGou\SWIN - 只因你太美", SND\_ASYNC);

return true;

}

void game()//Game initialization and operation

{

loadSound();

while(1)

{

score=0;//Initialize the scores

game\_rand();//Appearing two numbers randomly

game\_rand();

cirloop();//Putting numbers

while(1)

{

if(\_kbhit)//The keys

{

pass=0;

control();//give direction

fflush(stdin);//Clear the input cache

if(pass)game\_rand();//Determine if it can continue to generate random numbers after moving.

if(game\_check())//check to check function

{

if(game\_check()==1)// check to check function

{

if(game\_win()==2)break;// Interface of victory——play again

}

else if(game\_check()==-1)// check to check function {

cirloop();

game\_failed();

break;

}

}

cirloop();//Putting numbers

}

else Sleep(1);

}

game\_clear();//Clear the screen

continue;

}

}

void game\_restart()

{

loadSound();

show\_bk1();

cirloop();

while(1)

{

if(kbhit)//The keys

{

pass=0;

control();//give direction

fflush(stdin);//Clear the input cache

if(pass)game\_rand();//Determine if it can continue to generate random numbers

if(game\_check())//check to check function {

if(game\_check()==1)// check to check function {w

if(game\_win()==2)break;//Winning interface——Restart the game

}

else if(game\_check()==-1)// check to check function {

cirloop();

game\_failed();

break;

}

}

cirloop();//Putting numbers

}

else Sleep(1);

}

game\_clear();//Clear the screen

}

void game\_rand()//Putting number “2”or“4” randomly

{

int x,y,temp,num;

srand((int)time(0));

while(1)

{

x=rand()%4;

y=rand()%4;

temp=rand()%5;

if(temp==1)num=4;//Putting number “4” randomly

else num=2;//Putting number “2” randomly

if(map[y][x]==0)

{

map[y][x]=num;//Putting numbers randomly

break;

}

}

}

void control()

{

char keyboard;

keyboard=\_getch();

if(keyboard=='w'||keyboard=='W'||keyboard==72)control\_up();//up

if(keyboard=='s'||keyboard=='S'||keyboard==80)control\_down();//down

if(keyboard=='a'||keyboard=='A'||keyboard==75)control\_left();//left

if(keyboard=='d'||keyboard=='D'||keyboard==77)control\_right();//right

if(keyboard==27)//Esc the main menu

{

flag=0;

setbkcolor(RGB(251,248,241));

start\_4();

}

}

void control\_up()//Move up

{

int n,i,j,k;

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

{

k=0;

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

{

if(map[i][n]!=0)

{

for(j=i;j>k;j--)

{

if(map[j-1][n]==0)//There is no number above

{

map[j-1][n]=map[j][n];

map[j][n]=0;

pass=1;

cirloop();

Sleep(1);

}

else if(map[j-1][n]==map[j][n])// It has numbers above

{

map[j-1][n]=2\*map[j-1][n];

score+=map[j-1][n];

map[j][n]=0;

k=j;

pass=1;

cirloop();

Sleep(1);

break;

}

else break;

}

}

}

}

}

void control\_down()

{

int n,i,j,k;

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

{

k=3;

for(i=4;i>=0;i--)

{

if(map[i][n]!=0)

{

for(j=i;j<k;j++)

{

if(map[j+1][n]==0)

{

map[j+1][n]=map[j][n];

map[j][n]=0;

pass=1;

cirloop();

Sleep(1);

}

else if(map[j+1][n]==map[j][n])

{

map[j+1][n]=2\*map[j+1][n];

score+=map[j+1][n];

map[j][n]=0;

k=j;

cirloop();

Sleep(1);

pass=1;

break;

}

else break;

}

}

}

}

}

void control\_left()

{

int n,i,j,k;

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

{

k=0;

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

{

if(map[i][n]!=0)

{

for(j=n;j>k;j--)

{

if(map[i][j-1]==0)

{

map[i][j-1]=map[i][j];

map[i][j]=0;

pass=1;

cirloop();

Sleep(1);

}

else if(map[i][j-1]==map[i][j])

{

map[i][j-1]=2\*map[i][j-1];

score+=map[i][j-1];

map[i][j]=0;

k=j;

cirloop();

Sleep(1);

pass=1;

break;

}

else break;

}

}

}

}

}

void control\_right()

{

int n,i,j,k;

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

{

k=3;

for(n=3;n>=0;n--)

{

if(map[i][n]!=0)

{

for(j=n;j<k;j++)

{

if(map[i][j+1]==0)

{

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

map[i][j]=0;

pass=1;

cirloop();

Sleep(1);

}

else if(map[i][j+1]==map[i][j])

{

map[i][j+1]=2\*map[i][j+1];

score+=map[i][j+1];

map[i][j]=0;

k=j;

cirloop();

Sleep(1);

pass=1;

break;

}

else break;

}

}

}

}

}

int game\_check()

{

int success=0;

int i,n;

int a=0;//Used for counting numbers in the grid which is not near the same number

if((map[0][0]!=0)&&(map[0][1]!=0)&&(map[0][0]!=0)&&(map[0][0]!=map[0][1])&&(map[0][0]!=map[1][0]))a++;

if((map[3][3]!=0)&&(map[3][2]!=0)&&(map[2][3]!=0)&&(map[3][3]!=map[3][2])&&(map[3][3]!=map[2][3]))a++;

if((map[0][3]!=0)&&(map[0][2]!=0)&&(map[1][3]!=0)&&(map[0][3]!=map[0][2])&&(map[0][3]!=map[1][3]))a++;

if((map[3][0]!=0)&&(map[3][1]!=0)&&(map[2][0]!=0)&&(map[3][0]!=map[3][1])&&(map[3][0]!=map[2][0]))a++;

if((map[0][1]!=0)&&(map[0][0]!=0)&&(map[0][2]!=0)&&(map[0][1]!=map[0][0])&&(map[0][1]!=map[0][2]))a++;

if((map[0][2]!=0)&&(map[0][1]!=0)&&(map[0][3]!=0)&&(map[0][2]!=map[0][1])&&(map[0][2]!=map[0][3]))a++;

if((map[3][1]!=0)&&(map[3][0]!=0)&&(map[3][2]!=0)&&(map[3][1]!=map[3][0])&&(map[3][1]!=map[3][2]))a++;

if((map[3][2]!=0)&&(map[3][1]!=0)&&(map[3][3]!=0)&&(map[3][2]!=map[3][1])&&(map[3][2]!=map[3][3]))a++;

if((map[1][0]!=0)&&(map[0][0]!=0)&&(map[2][0]!=0)&&(map[1][0]!=map[0][0])&&(map[1][0]!=map[2][0]))a++;

if((map[2][0]!=0)&&(map[3][0]!=0)&&(map[1][0]!=0)&&(map[2][0]!=map[1][0])&&(map[2][0]!=map[3][0]))a++;

if((map[1][3]!=0)&&(map[0][3]!=0)&&(map[2][3]!=0)&&(map[1][3]!=map[0][3])&&(map[1][3]!=map[2][3]))a++;

if((map[2][3]!=0)&&(map[3][3]!=0)&&(map[1][3]!=0)&&(map[2][3]!=map[1][3])&&(map[2][3]!=map[3][3]))a++;

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

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

if((map[n][i]!=0)&&(map[n-1][i]!=0)&&(map[n+1][i]!=0)&&(map[n][i-1]!=0)&&(map[n][i+1]!=0)&&(map[n][i]!=map[n+1][i])&&(map[n][i]!=map[n-1][i])&&(map[n][i]!=map[n][i+1])&&(map[n][i]!=map[n][i-1]))a++;

if(a==16)success=-1;//Lose

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

for(n=0;n<4;n++)if(map[n][i]==2048)//Many 2048 show

success=1;//Winning conditions

return success;

}

void cirloop()//Cycle show

{

int i,n;

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

{

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

{

show(i,n,map[n][i],score);// The last parameter is passed in the score.

}

}

}

void game\_clear()//Clear the screen

{

int i,n;

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

{

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

{

map[n][i]=0;

}

}

}

int main()

{

show\_1();

mciSendString("open G:\\KuGou\\1.mp3", NULL, 0, NULL);

mciSendString("play G:\\KuGou\\1.mp3", NULL, 0, NULL);

while (1);

return 0;

}