



链滴

2048 小游戏开发

作者: [MuMuloveU](#)

原文链接: <https://ld246.com/article/1562982628943>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

2048 小游戏文档

只有一个2048类

- 暴露两个接口：构造函数 play函数
- 游戏主体是一个4x4的二维数组
- display函数展示这个二维数组，和一些提示信息
- rand_map函数负责在数组中不是0的地方随机刷新2，有百分之二十的概率刷新的是4
- key_process函数用来响应用户的按键，就是用户按下方向键，数字会移动
- play函数根据游戏逻辑调用这些函数
- full函数用来判断游戏区域是否还有空白，如果还有空白返回 1，没有空白返回0，没有空白意味着不能再随机刷新数字，等价于游戏结束，程序终止
- up down left right 这四个函数实现了用户在按下了不同的方向键之后数字的移动
- 构造函数实现了数组的初始化，并且调用rand_map在两个位置生成了两个数字（2 或者 4）

使用方法

```
#Linux平台
g++ 2048.cpp main.cpp -o game
./game
```

下面是代码

main.cpp

```
/*
 * File Name: main.cpp
 * Author:muyuan
 * Mail: muyuanhuck@163.com
 * Created Time: 2019年07月12日 星期五 08时36分59秒
 */

#include<iostream>
#include "2048.h"
using namespace std;

int main()
{
    _2048 game;
    game.play();
    return 0;
}
```

2048.h

```
/*
```

```
> File Name: 2048.h
> Author:muyuan
> Mail: muyuanhuck@163.com
> Created Time: 2019年07月12日 星期五 08时36分38秒
```

```
*****/
```

```

[]
#ifndef _2048_H
#define _2048_H
#endif
[]
#define N 4
[]
class _2048
{
private:
int map[N][N];
int score;
void display();
void rand_map();
int key_process();
int full();
int up();
int down();
int left();
int right();
public:
int play();
_2048();
};
[]
```

2048.cpp

```
/******
```

```
> File Name: 2048.cpp
> Author: muyuan
> Mail: muyuanhuck@163.com
> Created Time: 2019年07月12日 星期五 08时36分30秒
```

```
*****/
```

```
#include <iostream>
#include <iomanip>
#include <ctime>
//#include <cstdio>
#include "2048.h"
using namespace std;
[]
_2048::_2048()
{
score = 0;
for (int x = 0; x < N; x++)
{
for (int y = 0; y < N; y++)
{
map[x][y] = 0;

```

```

}
}
rand_map();
rand_map();
}
[]
void _2048::display()
{
cout << "+-----+-----+-----+-----+" << endl;
for (int y = 0; y < N; y++)
{
for (int x = 0; x < N; x++)
{
if (map[y][x] == 0)
{
cout << "|" << setw(4) << " ";
}
else
cout << "|" << setw(4) << map[y][x];
}
cout << "|" << endl;
cout << "+-----+-----+-----+-----+" << endl;
}
cout << "+-----+-----+-----+-----+" << endl;
cout << "|   k           |" << endl;
cout << "|h<---+--->l     |" << endl;
cout << "|   j           |" << endl;
cout << "|请输入方向, q退出 |" << endl;
cout << "| 分数           " << setw(4) << score << " |" << endl;
cout << "+-----+-----+-----+-----+" << endl;
}
[]
void _2048::rand_map()
{
srand(time(0));
int numx = rand();
numx = numx % 4;
int numy = rand();
numy = numy % 4;
//cout << numx << " " << numy << endl;
[]
int flag = rand();
flag = flag % 5;
if (flag > 3)
{
while (map[numy][numx] != 0)
{
numx = rand();
numx = numx % 4;
numy = rand();
numy = numy % 4;
}
map[numy][numx] = 4;
}
}

```

```

else
{
while (map[numy][numx] != 0)
{
numx = rand();
numx = numx % 4;
numy = rand();
numy = numy % 4;
}
map[numy][numx] = 2;
}
}
[]
int _2048::key_process()
{
char c;
cout << "请输入方向" << endl;
//c = getchar();
//scanf("%*c");
cin >> c;
if (c == 'q')
{
cout << "quit" << endl;
return 0;
}
else if (c == 'h')
{
cout << "左" << endl;
left();
}
else if (c == 'j')
{
cout << "下" << endl;
down();
}
else if (c == 'k')
{
cout << "上" << endl;
up();
}
else if (c == 'l')
{
cout << "右" << endl;
right();
}
else
{
cout << "输入错误" << endl;
}
return c;
}
[]
int _2048::play()
{

```

```

int flag = 1;
display();
while (flag)
{
flag = key_process();
if (flag == 'h' || flag == 'j' || flag == 'k' || flag == 'l')
{
rand_map();
display();
}
else
{
continue;
}
flag = full();
}
}
int _2048::full()
{
int aflag = 0;
for (int y = 0; y < N; y++)
{
for (int x = 0; x < N; x++)
{
if(map[y][x] != 0)
{
aflag++;
}
}
}
if (aflag == 16)
{
return 0;
}
else
{
return 1;
}
}
int _2048::up()
{
int x,y;
for (y = 3;y > 0;y--)
{
for(x = 0;x < 4; x++)
{
if(map[y-1][x] == 0)
{
map[y-1][x] =map[y][x];
map[y][x] = 0;
}
}
else
{
if(map[y-1][x] == map[y][x])

```

```

{
score +=map[y-1][x];
map[y-1][x] *=2;
map[y][x] = 0;
}
}
}
}
int _2048::down()
{
int x,y;
for (y = 0;y < 3;y++)
{
for(x = 0;x < 4; x++)
{
if(map[y+1][x] == 0)
{
map[y+1][x] =map[y][x];
map[y][x] = 0;
}
else
{
if(map[y+1][x] == map[y][x])
{
score +=map[y+1][x];
map[y+1][x] *=2;
map[y][x] = 0;
}
}
}
}
}
int _2048::left() //左
{
int x,y;
for (x = 3;x > 0;x--)
{
for(y = 0;y < 4; y++)
{
if(map[y][x-1] == 0)
{
map[y][x-1] =map[y][x];
map[y][x] = 0;
}
else
{
if(map[y][x-1] == map[y][x])
{
score +=map[y][x-1];
map[y][x-1] *=2;
}
}
}
}
}
}

```

```
map[y][x] = 0;
}
}

}
}
}
int _2048::right()
{
int x,y;
for (x = 0;x < 3;x++)
{
for(y = 0;y < 4; y++)
{
if(map[y][x+1] == 0)
{
map[y][x+1] =map[y][x];
map[y][x] = 0;
}
else
{
if(map[y][x+1] == map[y][x])
{
score +=map[y][x+1];
map[y][x+1] *=2;
map[y][x] = 0;
}
}
}
}
}
}
```