

C++课程设计项目报告：2048小游戏

一、小组分工

二、游戏规则

三、游戏界面设计

四、程序设计

（一）数据结构设计

（二）程序流程设计

（三）交互方式设计

五、系统实现

（一）初始化函数initialize

（二）游戏运行函数run

（三）操作函数move

（四）检验函数over

六、完整源代码

七、结果展示

C++课程设计项目报告：2048小游戏

使用集成开发环境为Visual Studio 2022社区版，C++编写

一、小组分工

二、游戏规则

2048作为一个经久不衰的小游戏，游戏规则较为基础

- 初始地图为一个4*4的网格，每一个网格均为空
- 开始游戏后，每轮随机在一个空网格内生成一个数字2
- 玩家有四种操作，分别为上移、下移、左移、右移
- 每次操作，地图中所有数字都会向所选方向移动，补足该方向一侧的空格
- 当该方向相邻两格中数字相同，两数字合成为原数字数值的两倍，每个数字每轮仅合并一次
- 每次合并，玩家获得合并后数值的分数
- 胜利条件：成功合成出2048(10^{11})，玩家获胜
- 游戏结束条件：玩家成功合成2048，或所有网格被数字填满，无法再移动，游戏结束，玩家失败

三、游戏界面设计

由于时间比较紧张，并未使用easyx等图形库，主要通过控制台输出

1. 标题“2048 game”
2. 玩家得分、用户名
3. 游戏主界面，包含一个4*4的网格，通过for循环输出'|'与'-'构建。数据结构是一个二维数组Map[i][j]，储存每格中的数值

```
1  for (int i = 1; i <= 5 * N + 1; i++)cout << '-';
2      cout << endl;
3      for (int i = 1; i <= N; i++) {
4          cout << " ";
5          cout << '|';
6          for (int j = 1; j <= N; j++) {
7              if (Map[i][j]) {
8                  cout << setw(4) << Map[i][j];
9                  cout << '|';
10             }
11             else cout << setw(5) << '|';
12         }
13         cout << endl;
14         cout << " ";
15         for (int i = 1; i <= N * 5 + 1; i++) cout << '-';
16         cout << endl;
```

4. 为了更好的游戏体验，我们采用了windows.h库内部的光标操作，隐藏了控制台的光标，hideCursor函数如下：

```
1  void hideCursor() { //隐藏光标操作，这里参考了一些网上的做法
2      CONSOLE_CURSOR_INFO cursor;
3      cursor.bVisible = 0;
4      cursor.dwSize = 1;
5      HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
6      SetConsoleCursorInfo(hOut, &cursor);
7  }
```

四、程序设计

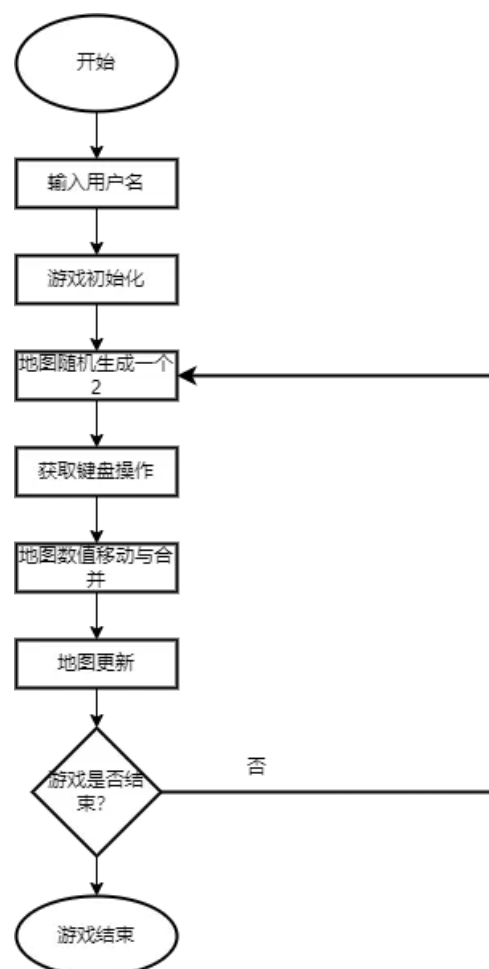
（一）数据结构设计

游戏类 `game2048`, 包括数据成员整型二维数组 `Map[N+2][N+2]`, 用于存储网格中的数字, 整型变量 `score`、`maxsq`、`sumsq`, 分别表示玩家得分、场上最大方块数值、场上方块数目总和, 以及一个字符串 `player`, 存储玩家用户名

访问权限均为 `protected`

```
1 class game2048{
2     protected:
3         int Map[N + 2][N + 2]; //地图二维数组
4         int score; //分数
5         string player;
6         int maxsq; //最大方块数值
7         int sumsq; // 场上方块总数
8     };
```

（二）程序流程设计



（三）交互式设计

1. 用户名输入

2. 通过_getch()获取用户键盘按键，且_getch()无回显，对应按键数值如下

上: 72
下: 80
左: 75
右: 77

3. 游戏结束结算

```
1  if (sumsq == N * N && over()) { //满了，检测是否可以移动
2      cout << "GAME OVER! You got " << score << " points"
   << endl; //无法移动，游戏结束
3      system("pause");
4      return;
5  }
6  if (maxsq == 2048) { //合并到2048，游戏胜利
7      cout << "You win!" << endl;
8      system("pause");
9      return;
10 }
```

4. 光标隐藏（见上）

五、系统实现

（一）初始化函数initialize

```
1  void initializing(string str) { //类的初始化
2      memset(Map, 0, sizeof(Map)); //初始化地图二维数组
3      score = 0;
4      maxsq = 0;
5      sumsq = 0;
6      player = str;
7  }
```

（二）游戏运行函数run

```
1 void run() { //游戏运行函数
2     system("title 2048"); //利用system设置窗口名称
3     hideCursor(); //隐藏光标，否则影响观感
4     generate();
5     sumsq++;
6     while (1) { //开始游戏
7         hideCursor();
8         if (sumsq == N * N && over()) { //满了，检测是否可以移动
9             cout << "GAME OVER! You got " << score << " points"
10            << endl; //无法移动，游戏结束
11            system("pause");
12            return;
13        }
14        if (maxsq == 2048) { //合并到2048，游戏胜利
15            cout << "You win!" << endl;
16            system("pause");
17            return;
18        }
19        print(); //更新地图
20        move(); //移动操作
21        system("cls");
22    }
```

（三）操作函数move

```
1 void move() {
2     int key = _getch();
3     key = _getch(); //读取键盘输入，无回显
4     switch (key) { //键盘输入了什么呢，72上箭头，80下箭头，75左箭头，77右
        箭头
5         case 72: up(); break;
6         case 80: down(); break;
7         case 75: left(); break;
8         case 77: right(); break;
9     }
10 }
```

(四) 检验函数over

```
1  bool over() { //判断是否有可合并项存在
2      for (int i = 1; i <= N; i++) {
3          for (int j = 1; j <= N; j++) {
4              if (Map[i][j] == Map[i - 1][j] || Map[i][j] == Map[i
+ 1][j] || Map[i][j] == Map[i][j + 1] || Map[i][j] == Map[i][j -
1]) {
5                  return 0; //有两相同项相邻, 可操作
6              }
7          }
8      }
9      return 1; //无法操作, 游戏结束
10 }
```

六、完整源代码

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include<iostream>
3  #include<iomanip>
4  #include<cstdlib>
5  #include<time.h>
6  #include<string>
7  #include<windows.h> //与windows以及窗口相关的头文件, 用到system里的
   cls、pause、命名窗口
8  #include<conio.h> //控制台输入输出, _getch()函数
9  using namespace std;
10 const int N = 4; //边长
11 class game2048 { //创建游戏相关的类
12 public:
13     void initializing(string str) { //类的初始化
14         memset(Map, 0, sizeof(Map)); //初始化地图二维数组
15         score = 0;
16         maxsq = 0;
17         sumsq = 0;
18         player = str;
19     }
20     void hideCursor() { //隐藏光标操作, 这里参考了一些网上的做法, 用到头
   文件windows.h中的函数
21         CONSOLE_CURSOR_INFO cursor;
22         cursor.bVisible = 0;
23         cursor.dwSize = 1;
```

```

24     HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
25     SetConsoleCursorInfo(hOut, &cursor);
26 }
27 bool over() { //判断是否有可合并项存在
28     for (int i = 1; i <= N; i++) {
29         for (int j = 1; j <= N; j++) {
30             if (Map[i][j] == Map[i - 1][j] || Map[i][j] ==
Map[i + 1][j] || Map[i][j] == Map[i][j + 1] || Map[i][j] ==
Map[i][j - 1]) {
31                 return 0; //有两相同项相邻, 可操作
32             }
33         }
34     }
35     return 1; //无法操作, 游戏结束
36 }
37 void generate() {
38     srand(time(0));
39     int x = rand() % N + 1, y = rand() % N + 1;
40     while (Map[x][y]) {
41         x = rand() % N + 1;
42         y = rand() % N + 1;
43     } //随机寻找一个位置, 知道该位置为空, 填入2
44     Map[x][y] = 2;
45 }
46 void print() { //打印地图, 由 '-' 和 '|' 构成
47     cout << setw((5 * N + 4)/2+5) << "2048 game" << endl;
48     cout << "score: " << score << " Player: " << player <<
endl;
49     cout << " ";
50     for (int i = 1; i <= 5 * N + 1; i++) cout << '-';
51     cout << endl;
52     for (int i = 1; i <= N; i++) {
53         cout << " ";
54         cout << '|';
55         for (int j = 1; j <= N; j++) {
56             if (Map[i][j]) {
57                 cout << setw(4) << Map[i][j];
58                 cout << '|';
59             }
60             else cout << setw(5) << '|';
61         }
62         cout << endl;

```

```

63         cout << " ";
64         for (int i = 1; i <= N * 5 + 1; i++) cout << '-';
65         cout << endl;
66     }
67 }
68 void up() { //up,down,right,left操作类似, 以up为例子
69     bool ismove = 0;
70     int i = N; //从最下方向上扫
71     for (int j = 1; j <= N; j++) {
72         int k = i + 1, flag = 0, y1 = 0, ismerge = 0, isag
= 0; //flag表示该循环下是否移动, y1表示是否找到了非零点, ismerge是否有合并
操作, isag是否已经扫描过一遍
73         while (1) {
74             k--; //向上寻找
75             if (k == 0) { //到顶端
76                 if (flag) k = i, flag = 0, y1 = 0, isag =
1; //移动过, 再扫一次
77                 else break; //无法再移动, 跳出
78             }
79             if (Map[k][j]) y1 = 1; //找到一处非零, 开始处理
80             if (!Map[k][j] && y1 == 1) { //若此行为零, 将下面一
行上移
81                 Map[k][j] = Map[k + 1][j];
82                 Map[k + 1][j] = 0;
83                 flag = 1; //此次循环已移动
84                 ismove = 1; //整个过程已经移动
85                 continue;
86             }
87             if (Map[k][j] == Map[k + 1][j] && y1 == 1 &&
(!ismerge && !isag)) { //可与下面一行合并, 是第一次循环 (若多次循环合并出
现多次合并现象), 最近一次未合并 (否则连续合并多次)
88                 Map[k][j] *= 2;
89                 Map[k + 1][j] = 0;
90                 score += Map[k][j];
91                 maxsq = max(maxsq, Map[k][j]);
92                 sumsq--;
93                 flag = 1;
94                 ismove = 1;
95                 ismerge = 1;
96                 continue;
97             }
98             if (ismerge) ismerge++;

```



```

99         if (ismerge >= 2) ismerge = 0; //未合并，下次可以合并
100     }
101 }
102     if (sumsq != N * N && ismove) { //已经移动过，且没有满，继续随机
产生一个2
103         sumsq++;
104         generate();
105     }
106 }
107 void down() {
108     bool ismove = 0;
109     int i = 1;
110     for (int j = 1; j <= N; j++) {
111         int k = i - 1, flag = 0, y1 = 0, ismerge = 0, isag
= 0;
112         while (1) {
113             k++;
114             if (k == N + 1) {
115                 if (flag) k = i, flag = 0, y1 = 0, isag = 1;
116                 else break;
117             }
118             if (Map[k][j]) y1 = 1;
119             if (!Map[k][j] && y1 == 1) {
120                 Map[k][j] = Map[k - 1][j];
121                 Map[k - 1][j] = 0;
122                 flag = 1;
123                 ismove = 1;
124                 continue;
125             }
126             if (Map[k][j] == Map[k - 1][j] && y1 == 1 &&
(!ismerge && !isag)) {
127                 Map[k][j] *= 2;
128                 Map[k - 1][j] = 0;
129                 score += Map[k][j];
130                 maxsq = max(maxsq, Map[k][j]);
131                 sumsq--;
132                 flag = 1;
133                 ismove = 1;
134                 ismerge = 1;
135                 continue;
136             }
137             if (ismerge) ismerge++;

```

```

138         if (ismerge >= 2) ismerge = 0;
139     }
140 }
141 if (sumsq != N * N && ismove) {
142     sumsq++;
143     generate();
144 }
145 }
146 void left() {
147     bool ismove = 0;
148     for (int i = 1; i <= N; i++) {
149         int j = N;
150         int k = j + 1, flag = 0, y1 = 0, ismerge = 0, isag
= 0;
151         while (1) {
152             k--;
153             if (k == 0) {
154                 if (flag) k = j, flag = 0, y1 = 0, isag = 1;
155                 else break;
156             }
157             if (Map[i][k]) y1 = 1;
158             if (!Map[i][k] && y1 == 1) {
159                 Map[i][k] = Map[i][k + 1];
160                 Map[i][k + 1] = 0;
161                 flag = 1;
162                 ismove = 1;
163                 continue;
164             }
165             if (Map[i][k] == Map[i][k + 1] && y1 == 1 &&
(ismerge == 0 && !isag)) {
166                 Map[i][k] *= 2;
167                 Map[i][k + 1] = 0;
168                 score += Map[i][k];
169                 maxsq = max(maxsq, Map[i][k]);
170                 sumsq--;
171                 flag = 1;
172                 ismove = 1;
173                 ismerge = 1;
174                 continue;
175             }
176             if (ismerge) ismerge++;
177             if (ismerge >= 2) ismerge = 0;

```

```

178         }
179     }
180     if (sumsq != N * N && ismove) {
181         sumsq++;
182         generate();
183     }
184 }
185 void right() {
186     bool ismove = 0;
187     for (int i = 1; i <= N; i++) {
188         int j = 1;
189         int k = j - 1, flag = 0, y1 = 0, ismerge = 0, isag
= 0;
190         while (1) {
191             k++;
192             if (k == N + 1) {
193                 if (flag) k = j, flag = 0, y1 = 0, isag = 1;
194                 else break;
195             }
196             if (Map[i][k]) y1 = 1;
197             if (!Map[i][k] && y1 == 1) {
198                 Map[i][k] = Map[i][k - 1];
199                 Map[i][k - 1] = 0;
200                 flag = 1;
201                 ismove = 1;
202                 continue;
203             }
204             if (Map[i][k] == Map[i][k - 1] && y1 == 1 &&
(!ismerge && isag == 0)) {
205                 Map[i][k] *= 2;
206                 Map[i][k - 1] = 0;
207                 score += Map[i][k];
208                 maxsq = max(maxsq, Map[i][k]);
209                 sumsq--;
210                 flag = 1;
211                 ismove = 1;
212                 ismerge = 1;
213                 continue;
214             }
215             if (ismerge) ismerge++;
216             if (ismerge >= 2) ismerge = 0;
217         }

```

```

218
219     }
220     if (sumsq != N * N && ismove) {
221         sumsq++;
222         generate();
223     }
224 }
225 void move() {
226     int key = _getch();
227     key = _getch(); //读取键盘输入，无回显
228     switch (key) { //键盘输入了什么呢，72上箭头，80下箭头，75左箭
头，77右箭头
229         case 72: up(); break;
230         case 80: down(); break;
231         case 75: left(); break;
232         case 77: right(); break;
233     }
234 }
235 void run() { //游戏运行函数
236     system("title 2048"); //利用system设置窗口名称
237     hideCursor(); //隐藏光标，否则影响观感
238     generate();
239     sumsq++;
240     while (1) { //开始游戏
241         hideCursor();
242         if (sumsq == N * N && over()) { //满了，检测是否可以移动
243             cout << "GAME OVER! You got " << score << "
points" << endl; //无法移动，游戏结束
244             system("pause");
245             return;
246         }
247         if (maxsq == 2048) { //合并到2048，游戏胜利
248             cout << "You win!" << endl;
249             system("pause");
250             return;
251         }
252         print(); //更新地图
253         move(); //移动操作
254         system("cls");
255     }
256 }
257 protected:

```

```
258     int Map[N + 2][N + 2]; //地图二维数组
259     int score; //分数
260     string player;
261     int maxsq; //最大方块数值
262     int sumsq; // 场上方块总数
263 };
264 int main() {
265     srand(time(0));
266     game2048 g;
267     string username;
268     cout << "请输入你的用户名:" << endl;
269     getline(cin, username);
270     g.initializing(username);
271     cout << "按下任意键开始游戏" << endl;
272     system("pause");
273     system("cls");
274     g.run();
275     return 0;
276 }
```

七、结果展示

见视频