南开大学 计算机大类

姓名 刘孙延

学号 2311062

班级 2.1班

2024年5月13日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 1](#_Toc21286)

[一. 作业题目 2](#_Toc29969)

[二. 开发软件 2](#_Toc18364)

[三. 课题要求 2](#_Toc742)

[四. 主要流程 2](#_Toc25240)

[1． 整体流程 2](#_Toc20452)

[2． 算法或公式 2](#_Toc25354)

[（1） 菜单界面 3](#_Toc22508)

[（2） 扫雷 4](#_Toc28665)

[（3） 2048 6](#_Toc32428)

[（4） 五子棋 8](#_Toc23860)

[（5） 头文件 11](#_Toc13349)

[3． 单元测试 11](#_Toc24674)

[五. 单元测试 11](#_Toc28265)

[六. 收获 15](#_Toc23116)

高级语言程序设计大作业实验报告

1. **作业题目**

经典小游戏合集

1. **开发软件**

Visual Studio 2022（easyx库）

1. **课题要求**
2. 面向对象。
3. 单元测试。
4. 模型部分
5. 验证
6. **主要流程**
   1. **整体流程**

Main.cpp：菜单界面

2048.cpp：2048游戏代码

Gomoku.cpp：五子棋游戏代码

Minesweeper.cpp：扫雷游戏代码

* 1. **算法或公式**
     1. **菜单界面**

initgraph(600, 400);//窗口

//菜单图片导入

IMAGE img;

loadimage(&img,"images/StartMenu.jpg");

putimage(0,0,&img);

//设置消息结构体获取鼠标点击信息

while (true)

{

ExMessage msg;

if (peekmessage(&msg, EX\_MOUSE))

{

if (msg.message == WM\_LBUTTONDOWN)

{

stratupScene(&msg);

initgraph(600, 400);

putimage(0, 0, &img);

cout << "returntoMenu" << endl;

}

}

}

//判断点击区域 调用对应游戏函数cpp文件

void stratupScene(ExMessage\* msg)

{

if (isInrect(msg, 242, 170, 352-242, 246-170))

{

exit(-1);

}

else if (isInrect(msg, 90, 65, 200-90, 140-65))

{

Minesweeper();

cout << "closeMinesweeper" << endl;

return;

}

else if (isInrect(msg, 240, 65, 355-240, 140-65))

{

\_2048();

cout << "close\_2048" << endl;

return;

}

else if (isInrect(msg, 390, 65, 520 - 390, 140 - 65))

{

Gomoku();

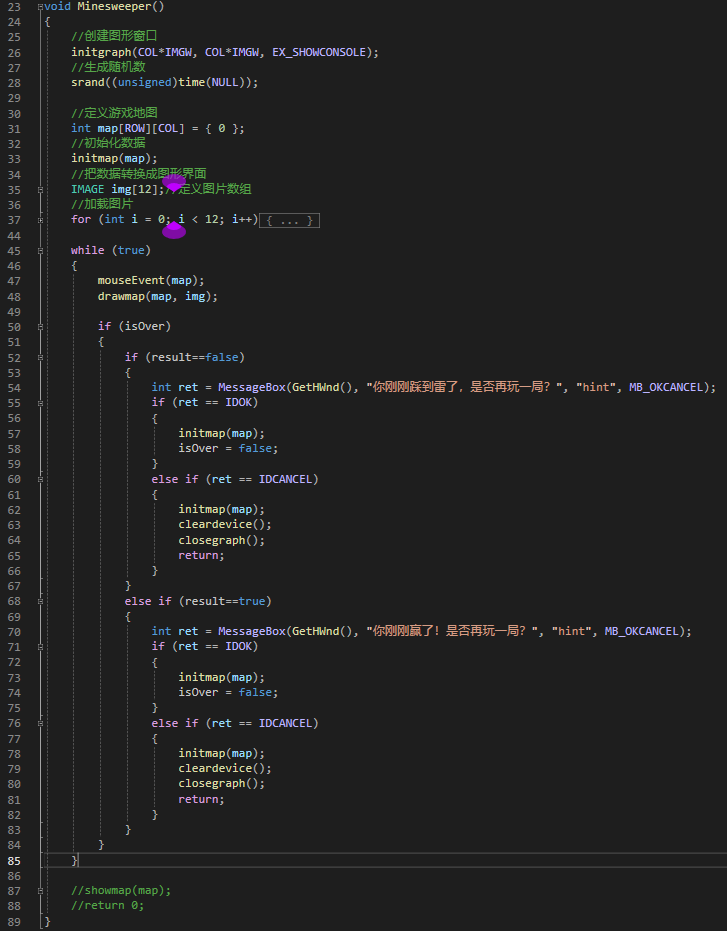
cout << "closeGomoku" << endl;

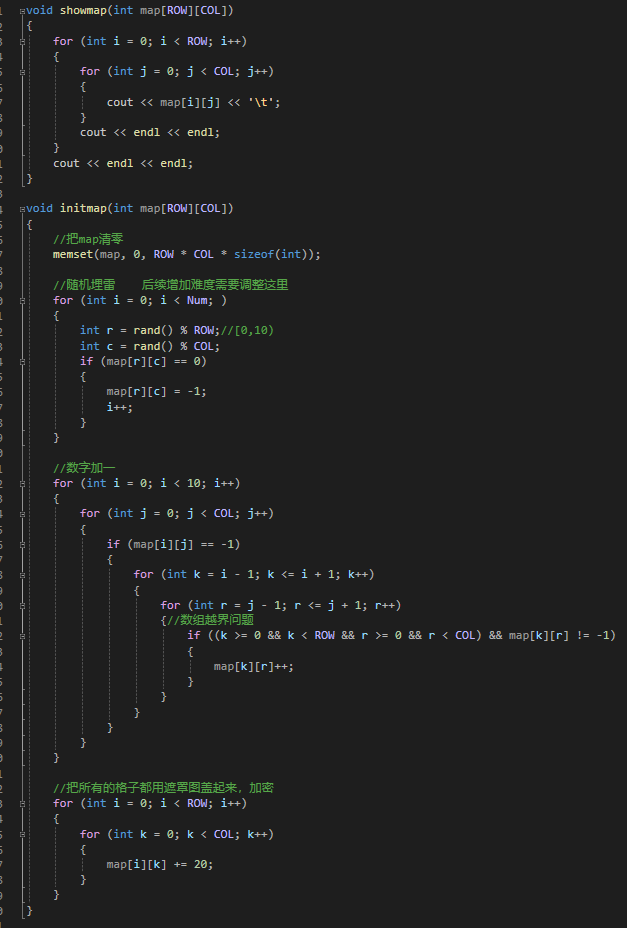
return;

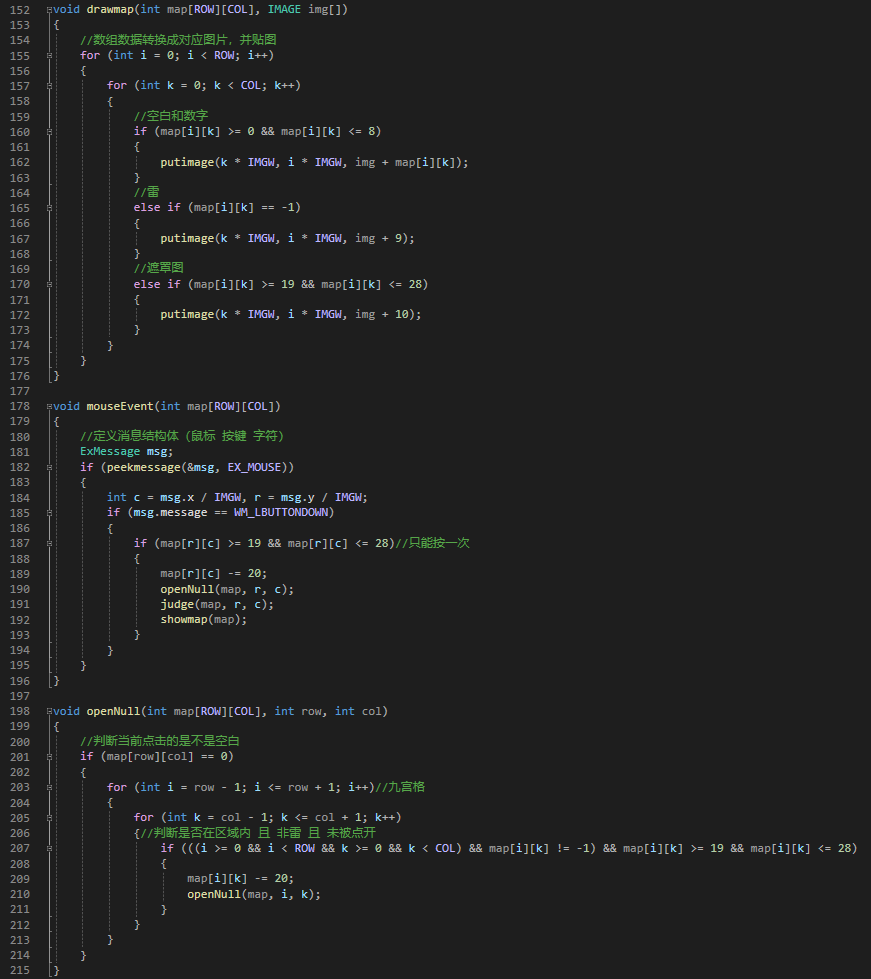
}

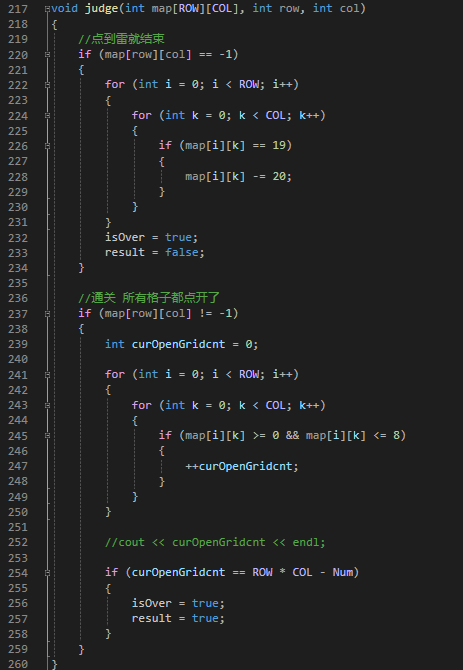
}

* + 1. **扫雷**

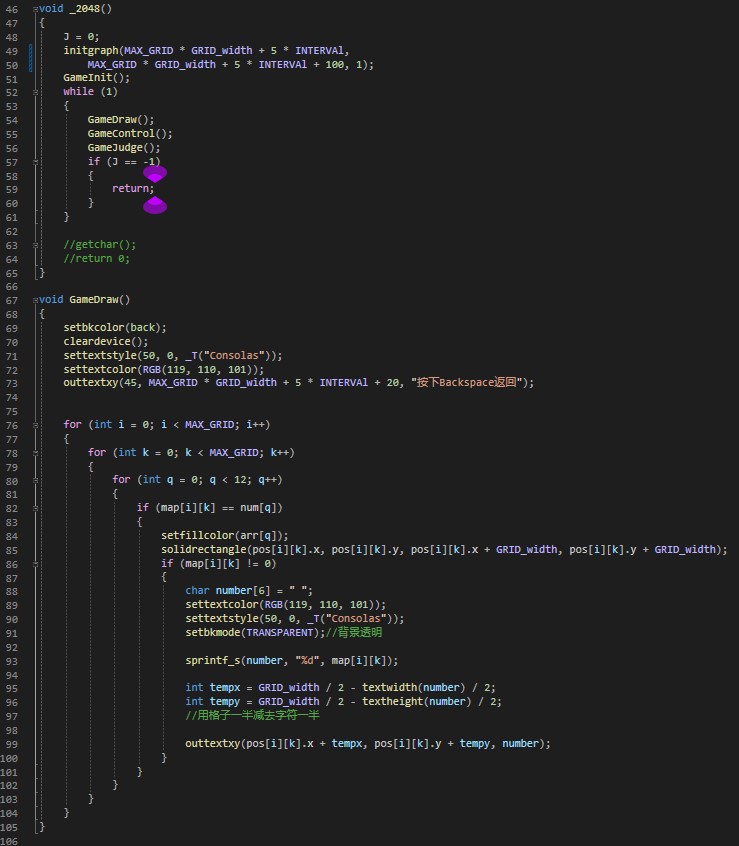




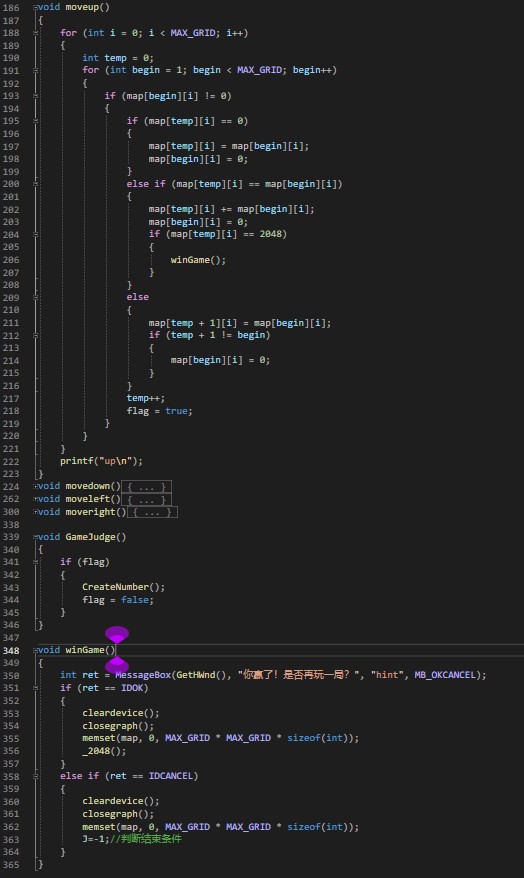




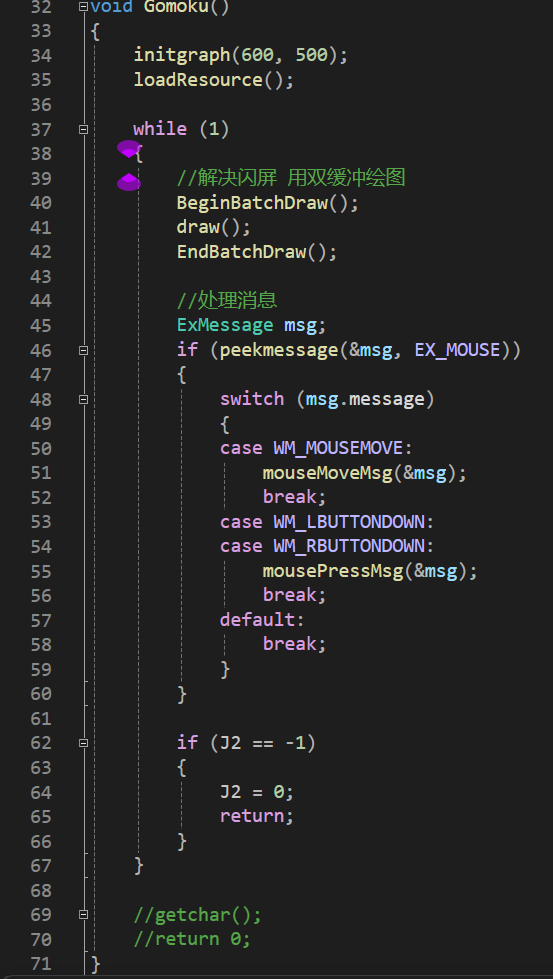
* + 1. **2048**

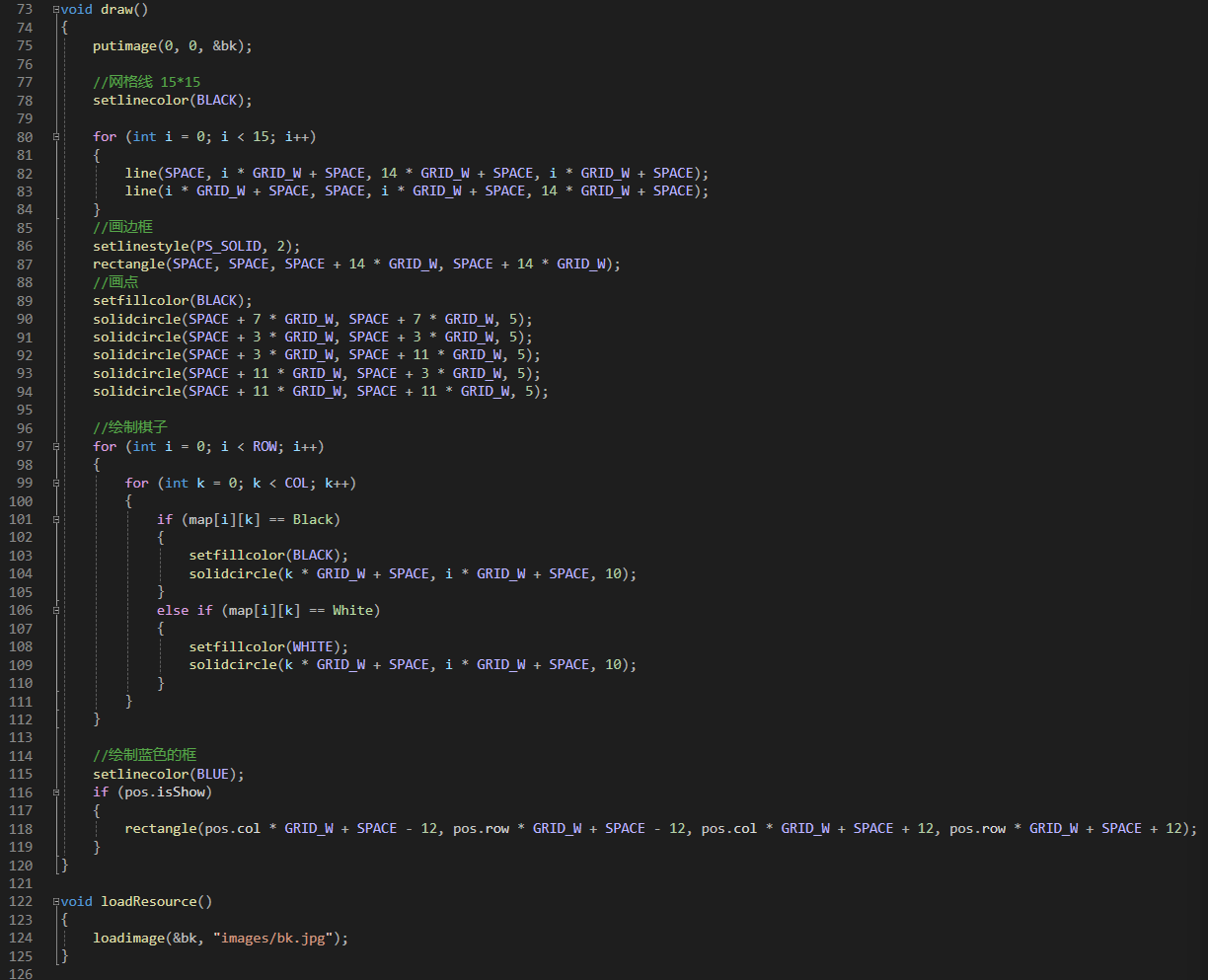


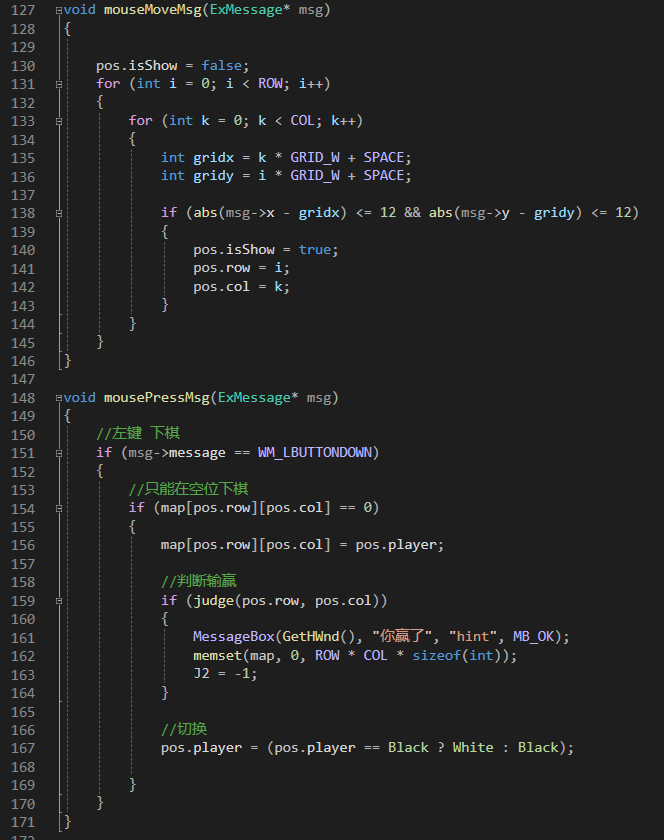


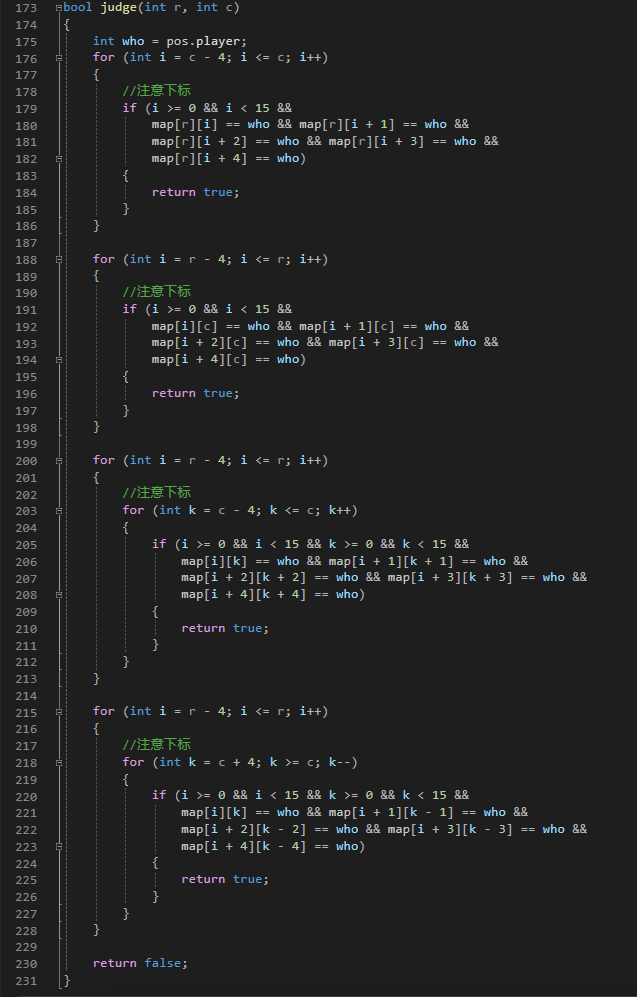


* + 1. **五子棋**

****

****

****

****

* + 1. **头文件**

#pragma once

void Minesweeper();

void \_2048();

void Gomoku();

* 1. **单元测试**

针对每个cpp文件，单独测试游戏。

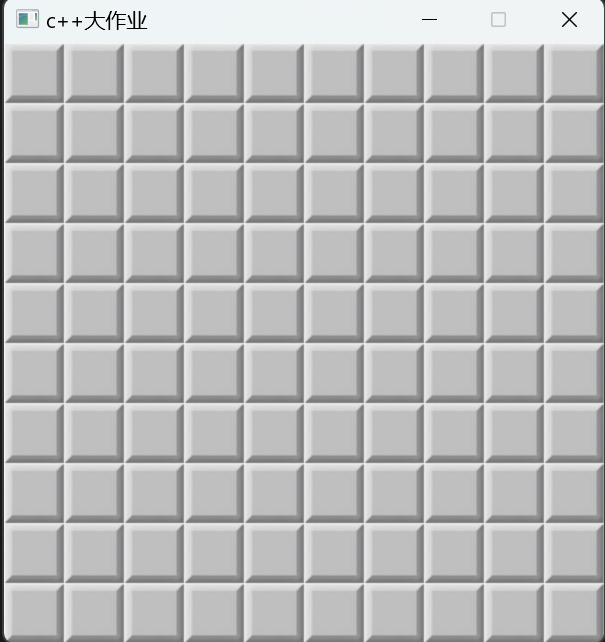
针对菜单界面，测试是否能够调用正确函数。

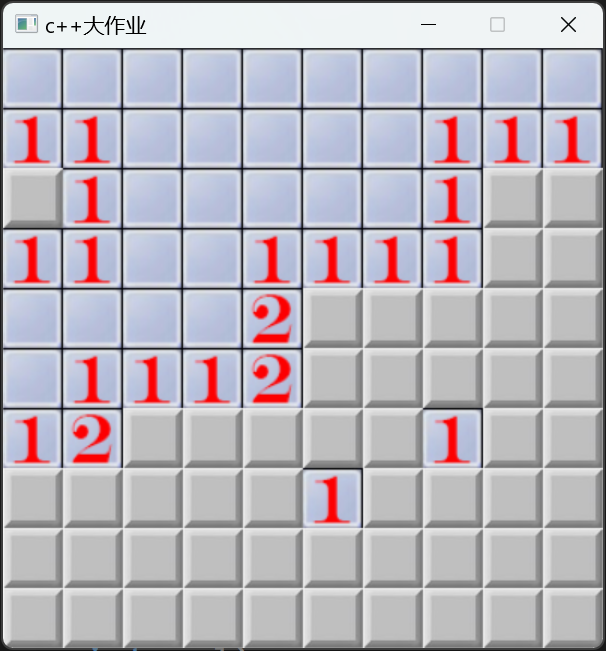
1. **单元测试**

**菜单**



**扫雷**



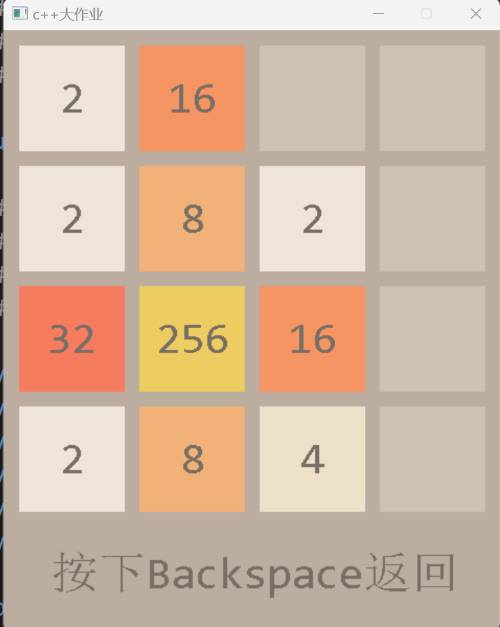




**2048**

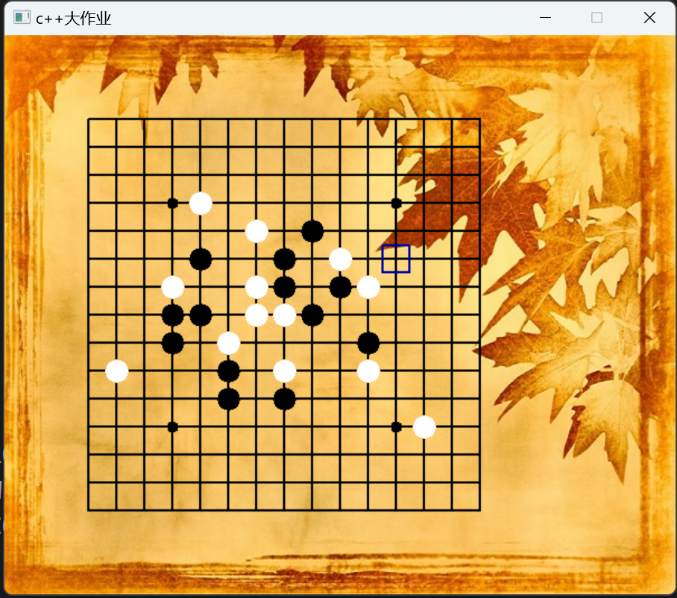


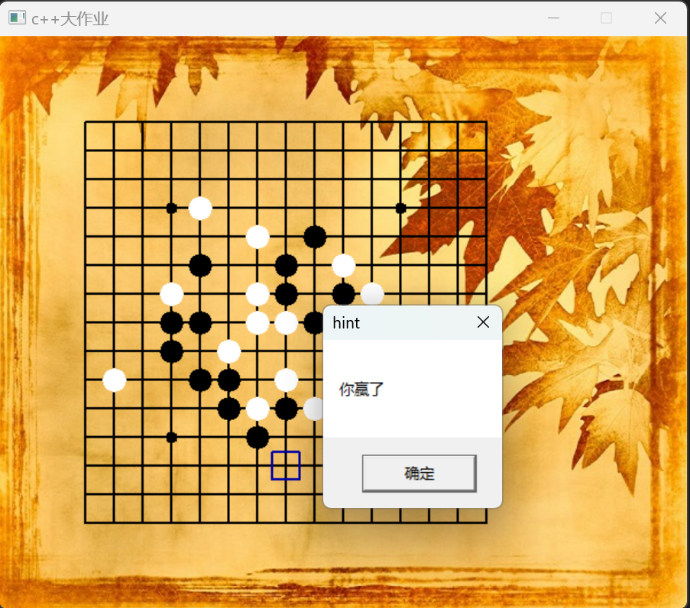




**五子棋**







**均无问题**

1. **收获**

1. 使用有意义的变量和函数命名：确保变量和函数的命名能够清晰地表达其用途，避免使用无意义的命名或缩写。

2. 编写注释：在代码中添加注释，解释代码的逻辑、目的和特殊情况，有助于其他人理解和维护代码。

3. 遵循代码规范：遵循统一的代码风格和规范，使代码易于阅读和维护。例如，缩进、命名规范、代码结构等。

4. DRY原则（Don't Repeat Yourself）：避免重复代码，将重复的代码抽象成函数或模块，提高代码的可维护性和复用性。

5. 单一职责原则（Single Responsibility Principle）：确保每个函数或类只负责完成一个明确的功能，避免功能耦合过高。

6. 测试：在编写代码之前先编写测试用例，确保代码的质量和稳定性。

7. 代码优化：优化代码结构、算法和性能，避免不必要的循环或重复操作，提高代码的效率。