**高级语言程序设计实验报告**

姓名：罗铃

学号：2412002

班级：工科试验班模拟3-2班

2025.5.15

**一、作业题目**

扫雷游戏

**二、开发软件**

Qt Creator

C++编译器

**三、课题要求**

1.面向对象

代码中定义了  MainWindow  类和  QMineSweeper  类， MainWindow  类继承自  QMainWindow ，用于管理游戏界面和相关操作； QMineSweeper  类继承自  QPushButton ，用于表示扫雷游戏中的方块，体现了面向对象的继承特性。

类中封装了各自的属性和方法，如  MainWindow  类的  gameOption 、 customOption  等私有槽函数， QMineSweeper  类的  openBlock 、 updateStyle  等私有方法，实现了数据和操作的封装。

2.单元测试

对游戏的各个功能模块进行单元测试，确保功能正确。

3.模型部分

构建了扫雷游戏模型，通过  QMineSweeper  类来表示游戏中的方块，定义了方块的各种状态码（如  BLOCK\_STATE\_AREA  等），以及游戏状态信号码（如  GAME\_STATE\_SWEEP  等）。

利用静态成员变量和方法来管理游戏地图相关信息，如  mineMap  存储所有方块对象， mapRow 、 mapCol  等表示地图的行列等属性。

4.验证

功能验证：通过实际运行游戏，检查游戏是否能正常进行，包括方块的点击操作、标记操作、游戏结束和胜利判断等功能是否正确。例如，点击雷方块时游戏应结束并显示相应提示，成功扫除所有非雷方块时游戏应判定胜利并提示。

边界条件验证：可测试地图行列数、雷数等边界情况，如最小和最大行列数设置、雷数接近行列数乘积等情况，确保游戏在各种合理边界条件下能稳定运行。

四、主要流程

1.整体流程

实现思路：以 Qt 为开发框架，利用其提供的图形界面组件和信号槽机制。通过创建  MainWindow  类作为游戏主窗口，管理游戏界面布局、菜单选项等； QMineSweeper  类继承  QPushButton  实现游戏方块的逻辑和显示。通过信号槽机制实现方块点击等操作与游戏状态更新的交互。

虚析构函数：无

其他继承关系：MainWindow  继承自  QMainWindow ，获得了主窗口的基本功能； QMineSweeper  继承自  QPushButton ，在按钮的基础上扩展了扫雷方块的特定功能，如方块状态管理、游戏逻辑处理等。

2.算法或公式

在  QMineSweeper::initMap  函数中，使用随机算法生成雷方块的位置。通过  myrand  函数（文档未展示具体实现）生成指定数量（ mineNum ）的随机方块 id 列表，确定雷的分布。

计算方块相邻雷数时，通过遍历相邻方块（在  initMap  函数中，通过双重循环遍历相邻方块坐标），判断相邻方块是否为雷，来确定当前方块的危险数值  selfN 。

3.单元测试

对QMineSweeper`类的各个方法进行单元测试，确保功能正确。

五、单元测试

测试结果

 QMineSweeper  类  openBlock  方法：在测试场景中，打开普通方块时，方块状态正确更新为已打开，相邻空白方块也能递归打开；打开雷方块时，游戏状态正确更新为失败，测试通过。

 QMineSweeper  类  updateStyle  方法：设置不同状态和数值后，按钮样式能按照预期更新，文字颜色和背景颜色符合定义的样式表，测试通过。

 MainWindow  类  gameOption  方法：触发不同游戏模式选项，游戏地图能按参数创建，窗口大小调整正确，但在边界参数下（如最大行列数），界面显示有轻微错位，需进一步调整，测试基本通过。

 MainWindow  类  gameState  方法：传入不同游戏状态码，能正确显示对应的胜利或失败提示信息，测试通过。

六、收获

1.虚析构函数

无

2.单元测试

通过思考和编写单元测试用例，深刻认识到单元测试在程序开发中的关键作用。它能帮助在开发过程中及时发现代码中的逻辑错误和潜在问题，提高程序的稳定性和可靠性。在后续开发中，会更加重视单元测试环节，在代码编写初期就规划好测试用例，实现测试驱动开发，提升开发效率和代码质量。