C++大作业总结报告

物理系 基科31 蒋文韬 2013011717

一、主要技术难点与算法设计：

这个程序在程序语言本身的功能的应用上并没有用到较为复杂的知识，如模版，STL等，但也仍有一些有一定难度的地方。在程序的编程实现过程中，遇到的较大的难点主要有三个，分别是用户点击时的操作，用户鼠标信息的获取，对用户输入的限制以及文件的读写。

在处理用户左键点击的相应操作这个难点时，在有雷，或与雷相邻的格子里点击时的情况较为简单，而在四周都没有雷的情况下，玩过扫雷的玩家应该比较了解，游戏将自动探索相连通的所有四周无雷的区域。因此程序需要实现寻找满足该条件的连通区域的算法。对这种情况，我的算法是首先标记四周无雷的被点击的点，随后多次遍历雷阵中的每一个点，判断该点是否与被标记的点相邻且满足四周无雷，是则标记，循环遍历直到一次遍历后没有新的点被标记则结束。实现这个方法后，仍有一些问题，经检查与debug均为细节上的漏误，更正后程序工作正常。在继续之后代码的编写时我又想到也可以用递归的算法解决这个问题，即左键点击函数在点击的位置四周无雷时，先标记该位置已探索，然后对四周的八个位置也调用左键点击函数，递归到点击到周围有雷的格子时便自动进入周围有雷时的处理方法。于是也对这个想法进行了实现，但程序运行情况为超出迭代深度。由于之前已经有正确的实现，因此我也没有太花费时间深究，最终仍采取了前一种方法。

在获取鼠标信息时，因为这部分涉及方面较广，与windows系统的消息机制等知识相关，我了解甚少，因此采用了在网上查阅资料与参考、修改他人代码的方式。实现这部分功能的代码主要修改自网上查找到的一段70行左右的代码。该段代码能够在控制台中让光标跟踪鼠标位置，并在鼠标左键点击时结束运行。我仔细阅读了这段代码，大致理解了其含义，其大部分的代码都是用于故障处理。我保留了它的光标跟踪鼠标的功能，并找到了它用来存储鼠标当前位置的变量，将鼠标位置接收到我的变量中以供之后使用。随后，我通过在解决方案资源管理器中搜索代码中出现的FROM\_LEFT\_1ST\_BUTTON\_PRESSED，找到了其在外部文件wincon.h中的定义，也通过阅读在其中找到了我所需的RIGHTMOST\_BUTTON\_PRESSED。最终，我将这段代码修改为能将鼠标实时位置传输给我的引用参数，并在用户点击后返回左键或右键点击的信息的函数。

用户输入这一部分，为防止用户输入乱七八糟的东西造成bug，我在需要用户输入其选择或数组的部分均使用了getch()函数，一次只接收一个字符，并通过循环条件控制只接收我想要的字符。在让用户输入自定义雷阵大小时，这种方式较为麻烦，需要挨个接收用户输入的数字字符，再将其转换为十进制数字的相应值，不过实现后程序工作良好，完全保证了在接收用户输入方面的正确性。

在文件读写这部分，涉及到三种游戏规模的前三名，以及字符串的读入与输出等，较为繁琐且容易出错，也耗费了我一定的时间，暴露出了我在字符串操作方面的不足，也通过这一部分功能的实现在这方面进行了练习与提升。

二、感想与不足：

这次大作业的编写，让我体会到了面向对象的编程方法的强大之处，以及自己亲手编出一个比较像样的成就感，也是对这一学期课程学习的应用与考验。通过这次大作业的编写，我对C++ 的相关知识点的理解与记忆更加深刻，如MATRIX类中对()，=操作符的重载，以及类的继承等。实际编写程序的经验也更加丰富，更认识到了规划，设计程序架构，处理程序运行的异常情况等方面能力的重要性。

而这次的程序也还有许多不足，比如在游戏开始时的菜单选择界面，还是使用了面向过程的编程思路。程序整体代码有一千余行，应该有不少冗余之处。游戏具体运行过程中的一些算法也应该有改进的余地。在实际的使用体验中，雷阵较大时，如高级雷阵与自定义较大雷阵时，每次点击后程序均会重新打印出用户看到的雷阵，屏幕闪烁对使用体验有所不良影响。在电脑运行较慢时，如配置较低又未接电源的笔记本电脑，会在点击鼠标右键标注雷的同时击发控制台自身的右键菜单，而电脑运行速度较快时则无此问题。

这次扫雷我原打算用MFC等图形编程工具实现而非控制台，但在进行了一些了解之后感觉仍较为复杂，最终还是选择了在控制台中实现。希望以后在这方面能有所进步。