武汉大学计算机学院
本科生实验报告

基于C++Qt的原创游戏编程 实验

专 业 名 称 ： 计算机科学与技术

课 程 名 称 ： 高级语言程序设计

指 导 教 师 ： 祝园园

学 生 学 号 ： 2018302120108

学 生 姓 名 ： 叶子扬

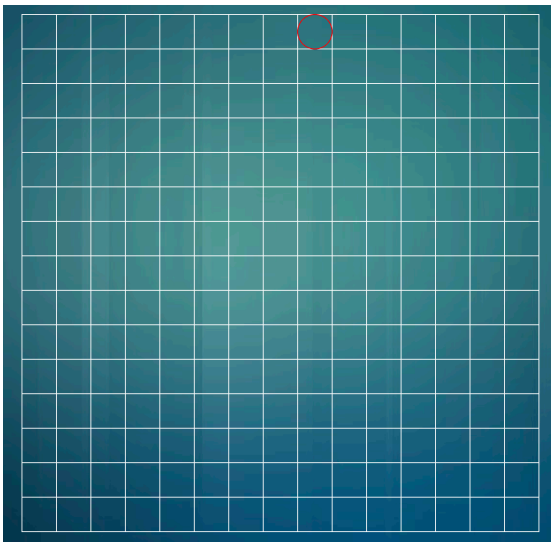
二〇一九年五月

1 实验目的

1. 熟悉C++语言基本语法
2. 了解图形化编程
3. 掌握跨平台GUI库Qt相关操作
4. 熟悉小游戏编程逻辑

2 实验内容

本次试验的内容来自我大一上学期的一个创意，在一个如图所示的棋盘上，会出现黑子和白子



它们的坐标会根据关卡的变化而变化，同时会设置一个出口，白子会朝向黑子靠近，黑子需要绕开白子走向终点即为胜利。

为了实现这个功能，我将使用传统的五子棋游戏的制作方法，即通过不断更新棋盘来实现格子的不同状态，同时我还借助Qt优秀的GUI编程来设计界面，以及使用paint3D来制作logo和棋子模型

3 实验步骤及分析

1. 编程环境：Qt Creator 4.8.2 Based on Qt 5.12.1 (Clang 10.0 (Apple), 64 bit)
2. 首先进行类的设计，根据功能，游戏的类大致分为下面几个

（1）Mainwindow类：继承自Qt原生类QMainwindow，执行游戏界面的初始化，对游戏界面的绘制，以及对鼠标事件的监控及相关操作

（2） GameCountroller类：顾名思义，游戏控制类，执行游戏控制相关操作如重新设置棋盘，开始游戏，判定胜利及失败等功能。

（3）Worrior类：勇士类，即黑子，记录黑子当前坐标及实现移动函数

（4）Enermy类： 敌人类，即白子，继承自Worrior类，重写了移动函数

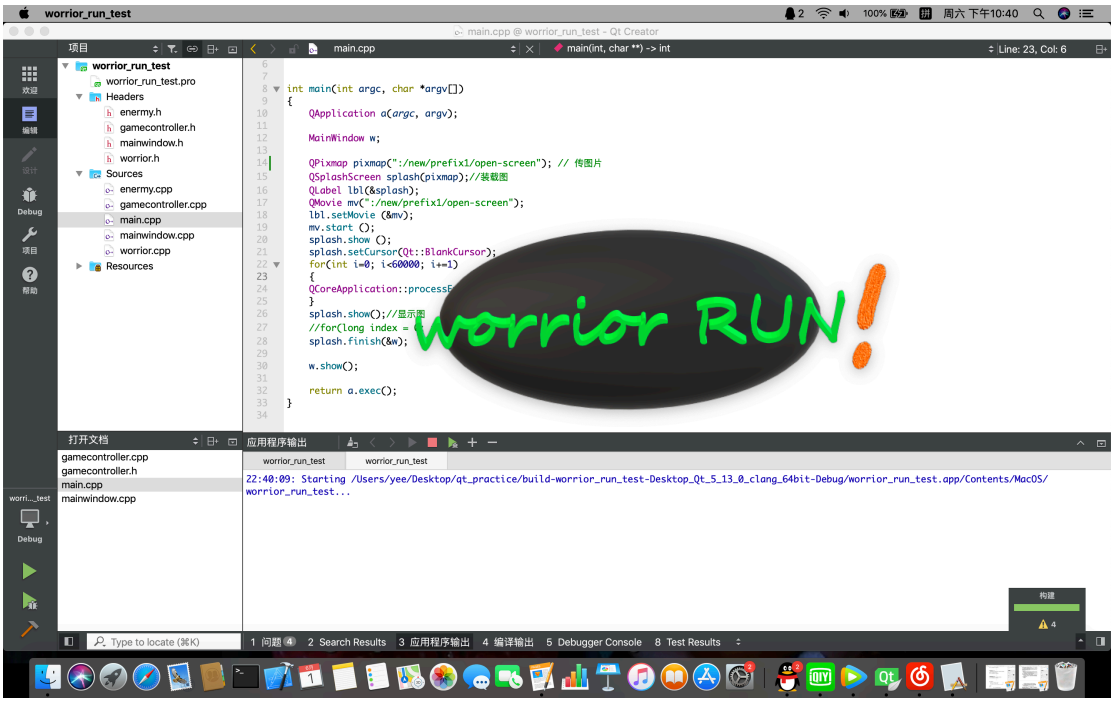
3. 各个功能的具体实现：

1. Main 函数

Main函数

Main函数的主要功能是创建窗口以及实现游戏开场动画功能

首先用建模软件设计好游戏logo并制作成图片，用QPixmap装载，先创建一个QLabel然后用QMovie装载图片后可以实现播放动态图的功能



2. GameController类的相关功能

（1）用enum存储游戏状态和关卡

（2）包含四个公有数据成员：Worrior指针worrior，Enermy指针数组enermies，walls及exit分别表示黑子，白子，墙和出口，其中墙和出口是一个pair<int, int>数值对用以存储坐标

（3）用一个二维向量来存储棋盘状态

（3）功能函数：

开始游戏：根据关卡布局初始化棋盘状态，不同状态的格子分配不同的值

更新棋盘：根据新的棋盘布局重新对二维向量赋值

判赢、判输：黑子与白子接触则游戏失败，黑子到达终点则游戏胜利

移动动作：调用黑子白子中的移动函数

3. Mainwindow类相关功能

（1）屏幕大小设置：在MainWindow的构造函数中使用

setFixedSize(kBoardMargin * 2 + kBlockSize * kBoardSizeNum + KButtonMargin, kBoardMargin * 2 + kBlockSize * kBoardSizeNum);（kblocksize等是事先声明好的全局常量）

（2）设置屏幕背景：同样在构造函数中实现

通过**QPixmap**装载背景图片，将调色板的画刷**QBrush**设置成图片样式，代码如下：

```
QPixmap pixmap = QPixmap(":/new/prefix1/back-ground").scaled(this->size());
QPalette  palette (this->palette());
palette .setBrush(QPalette::Background, QBrush(pixmap));
this-> setPalette( palette );
```

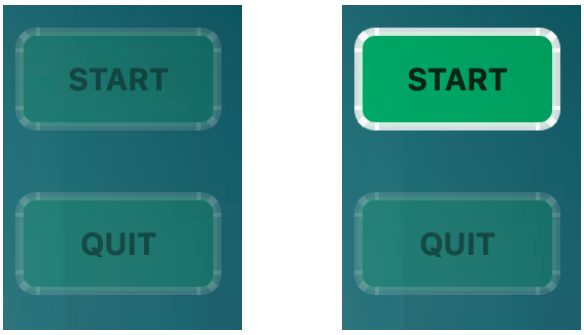
（3）状态栏设置：

向状态栏中添加选项，声明新的**QAction**作为关卡选择器，用图片适当美化并将其添加到状态栏选项中，同时声明关卡的**initial**函数，通过**Qt**的信号槽机制，将**QAction**的点击信号作为信号，将**initial**函数作为槽，以将二者**connect**，这样通过点击即可实现关卡转换，效果如图

（4）添加按钮：添加开始，退出按钮，并用**setstylesheet**函数美化

```
//添加按钮
QPushButton* Start_button = new QPushButton(this); //开始按钮
Start_button->setStyleSheet(
                    //正常状态样式
                    "QPushButton{"
                    "background-color:rgba(100,225,100,30);"//背景色（也可以设置图片）
                    "border-style:outset;"                  //边框样式（inset/outset）
                    "border-width:4px;"                     //边框宽度像素
                    "border-radius:10px;"                   //边框圆角半径像素
                    "border-color:rgba(255,255,255,30);"    //边框颜色
                    "font:bold 15px;"                       //字体，字体大小
                    "color:rgba(0,0,0,100);"                //字体颜色
                    "padding:6px;"                          //填衬
                    "}"
                    //鼠标按下样式
                    "QPushButton:pressed{"
                    "background-color:rgba(100,255,100,200);"
                    "border-color:rgba(255,255,255,30);"
                    "border-style:inset;"
                    "color:rgba(0,0,0,100);"
                    "}"
                    //鼠标悬停样式
                    "QPushButton:hover{"
                    "background-color:rgba(100,255,100,100);"
                    "border-color:rgba(255,255,255,200);"
                    "color:rgba(0,0,0,200);"
                    "}");
Start_button->setText(tr("START")); //设置文本
Start_button->setDefault(true); //默认选中
Start_button->move(663,100); //设置坐标
Start_button->resize(100,50); //设置大小
//Start_button->setIcon(QPixmap(":/new/prefix1/Start-button"));
connect(Start_button, &QPushButton::clicked, this, &MainWindow::StartGame);
```

效果如图：（鼠标未悬浮时半透明，鼠标悬浮时显形）



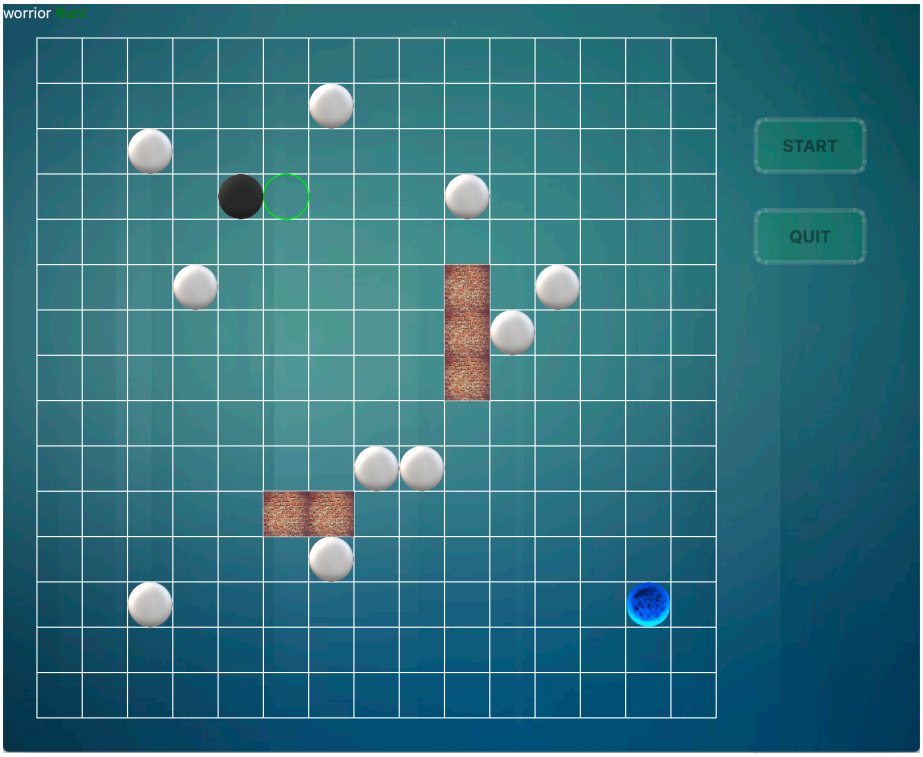
（5）添加文字：用QLabel将文字显示出来

（6）重写paintEvent：

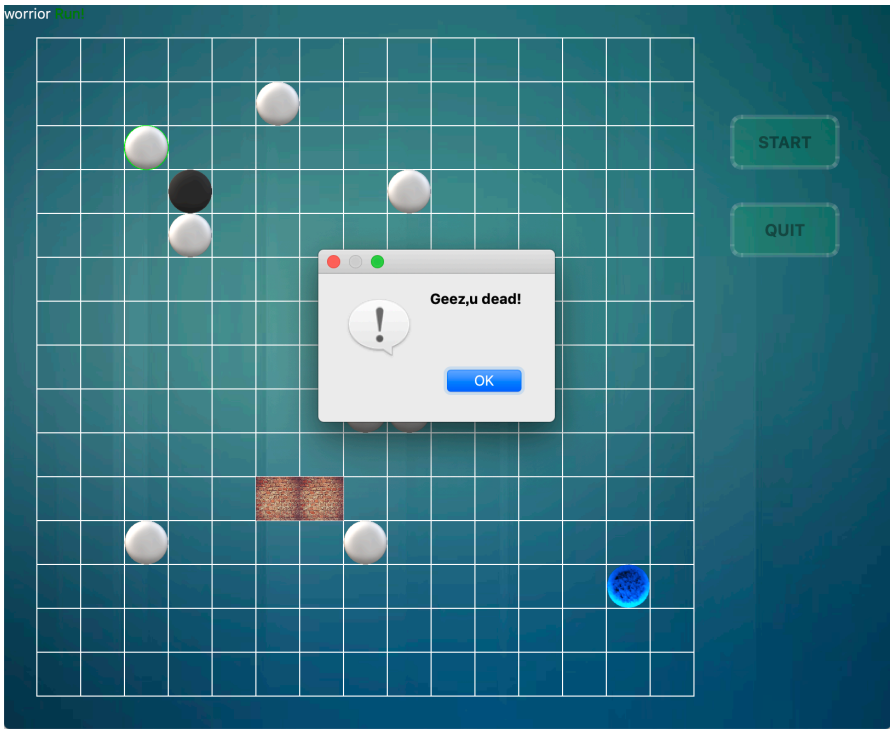
I. 当鼠标移动到某个格子时显示落子标记，可以移动用绿色表示，不可以用红色表示

II. 对于二维向量不同的值绘制不同的原件

绘制效果图：



III.判赢判输：修改游戏状态并弹出对话框提示游戏结束，并初始化新游戏避免无限循环



（7）重写mouseMoveEvent：当鼠标移动时存储坐标，以实现落子标记的绘制

（8）重写mouseRleaseEvent：当鼠标释放时调用移动函数实现棋子移动

4. Worrior及Enermy类相关功能

移动功能：当鼠标点击的格子在黑子周边的时候黑子可以移动，同时白子向黑子移动

部分bug及debug方案

1.在没有使用该方案前，我的想法是在GameController中安装事件过滤器，我想这样可以有效避免控制层与绘图层功能耦合，但是不管我怎么尝试，我的事件过滤器都无法过滤鼠标事件，键盘事件倒是能照常过滤。

错误原因及解决方案：，事件过滤器安装位置错误，应该安装到相应鼠标操作的view上，而我安装到了窗体类MainWindow的this指针上，安装后能正常响应，但我最后仍选择重写相关事件而不是使用事件过滤器。一来paintEvent是在Mainwindow类中，将Event放在一起代码结构比较清晰，二来重写事件比使用事件过滤器容易操作许多，我也更加熟悉

2.程序异常退出：程序往往在运行时毫无征兆地异常退出
错误原因及解决方案：作为**c++**语言框架，**Qt**对于指针的要求异常严格，程序异常退出的原因，往往是由于指针使用不当造成的，一般而言比较容易犯的错误有：忘记检查指针是否为空就使用，没有及时删除并令其指向空地址等

4.实验结果及总结
最后，程序终于正常地跑了起来，界面的美观程度也差强人意。做一款完全自己原创的游戏其实并非简单，很多问题都没有源码可以对照，只好自己硬着头皮上，但编程的过程是自由而又快乐的。游戏功能游戏界面游戏**logo**的设计与实现也让人乐在其中，程序不大，代码也不多，但潜力无限。时间有限，更多的优化我会在之后继续完善，同时我也期待它最后的高度。