|  |
| --- |
| **计算机专业类课程** |
| **实验报告** |
| **课程名称：程序设计基础**  **学院专业：计算机科学与工程学院**  **学生姓名：赵旭东**  **学　　号：2019081308025**  **指导教师：俸志刚** |
| **日　　期：2020年6月12日** |
|  |

**电 子 科 技 大 学**

**实 验 报 告**

**实验一**

# 一、实验室名称：

电子科技大学清水河校区主楼A2-413-2

# 二、实验项目名称：

实现具备基本功能的五子棋小游戏

# 三、实验目的：

通过构建五子棋游戏，初步对图形界面创建相关操作进行学习，熟悉C语言相关语法和知识。

# 四、实验主要内容：

使用Qt相关构件创建图形界面，通过C语言并运用极大值极小值算法、Alpha-Beta剪枝等算法实现五子棋人机对战，最终构建出完整的五子棋游戏。

# 五、实验器材（设备、元器件）：

(实验器材按照实际使用测试的，硬件平台：计算机配置，CPU内存等，软件平台：操作系统和开发环境，测试环境。**越详细越好。**)

硬件平台：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Page | Group | ItemID | Item | Value |
| 报告 |  | 261 | 计算机 | DESKTOP-G30ECTD |
| 报告 |  | 262 | 用户 | zhaox |
| 报告 |  | 263 | 操作系统 | Microsoft Windows 10 Pro 10.0.19041.329 |
| 报告 |  | 264 | 日期 | 2020/6/13 |
| 报告 |  | 265 | 时间 | 10:07 |
| 系统概述 | 计算机 | 582 | 计算机类型 | 基于 ACPI x64 的电脑 |
| 系统概述 | 计算机 | 513 | 操作系统 | Microsoft Windows 10 Pro |
| 系统概述 | 计算机 | 564 | Internet Explorer | 11.329.19041.0 |
| 系统概述 | 计算机 | 590 | Edge | 44.19041.1.0 |
| 系统概述 | 计算机 | 566 | DirectX | DirectX 12.0 |
| 系统概述 | 计算机 | 514 | 计算机名称 | DESKTOP-G30ECTD |
| 系统概述 | 计算机 | 515 | 用户名称 | zhaox |
| 系统概述 | 计算机 | 516 | 登录域 | DESKTOP-G30ECTD |
| 系统概述 | 计算机 | 0 | 日期/时间 | 2020-06-13 / 10:07 |
| 系统概述 | 主板 | 517 | 处理器名称 | HexaCore Intel Core i7-9750H; 4000 MHz (40 x 100) |
| 系统概述 | 主板 | 518 | 主板名称 | HP OMEN by HP Laptop 15-dh0xxx |
| 系统概述 | 主板 | 519 | 主板芯片组 | Intel Cannon Point HM370; Intel Coffee Lake-H |
| 系统概述 | 主板 | 520 | 系统内存 | 16260 MB (DDR4 SDRAM) |
| 系统概述 | 主板 | 521 | BIOS 类型 | Unknown (03/10/2020) |
| 系统概述 | 显示设备 | 523 | 显示适配器 | GeForce GTX 1660 Ti (6 GB) |
| 系统概述 | 显示设备 | 523 | 显示适配器 | GeForce GTX 1660 Ti (6 GB) |
| 系统概述 | 显示设备 | 523 | 显示适配器 | GeForce GTX 1660 Ti (6 GB) |
| 系统概述 | 显示设备 | 523 | 显示适配器 | GeForce GTX 1660 Ti (6 GB) |
| 系统概述 | 显示设备 | 524 | 3D 加速器 | nVIDIA GeForce GTX 1660 Ti |
| 系统概述 | 显示设备 | 525 | 显示器 | 通用即插即用监视器 [NoDB] |
| 系统概述 | 多媒体 | 526 | 音频适配器 | Intel Cannon Point PCH - cAVS (Audio; Voice; Speech) [B0] |
| 系统概述 | 多媒体 | 526 | 音频适配器 | nVIDIA TU116 HDMI/DP @ nVIDIA TU116 - High Definition Audio Controller |
| 系统概述 | 存储设备 | 580 | IDE 控制器 | Genesys Logic PCIE card reader device |
| 系统概述 | 存储设备 | 581 | 存储控制器 | Intel(R) Chipset SATA/PCIe RST Premium Controller |
| 系统概述 | 存储设备 | 581 | 存储控制器 | Microsoft 存储空间控制器 |
| 系统概述 | 存储设备 | 581 | 存储控制器 | 标准 NVM Express 控制器 |
| 系统概述 | 存储设备 | 528 | 硬盘驱动器 | SAMSUNG MZVLB512HAJQ-000H1 (512 GB; PCI-E 3.0 x4) |
| 系统概述 | 存储设备 | 528 | 硬盘驱动器 | Samsung SSD 860 QVO 1TB (1000 GB; SATA-III) |
| 系统概述 | 存储设备 | 585 | 硬盘 SMART 状态 | OK |
| 系统概述 | 磁盘分区 | 530 | C: (NTFS) | 254036 MB (181438 MB 可用) |
| 系统概述 | 磁盘分区 | 530 | D: (NTFS) | 475999 MB (233631 MB 可用) |
| 系统概述 | 磁盘分区 | 530 | E: (NTFS) | 477851 MB (302106 MB 可用) |
| 系统概述 | 磁盘分区 | 530 | F: (NTFS) | 232386 MB (218497 MB 可用) |
| 系统概述 | 磁盘分区 | 586 | 总大小 | 1406.5 GB (913.7 GB 可用) |
| 系统概述 | 输入设备 | 531 | 键盘 | HID Keyboard Device |
| 系统概述 | 输入设备 | 531 | 键盘 | PS/2 标准键盘 |
| 系统概述 | 输入设备 | 532 | 鼠标 | ELAN Input Device |
| 系统概述 | 输入设备 | 532 | 鼠标 | HID-compliant mouse |
| 系统概述 | 网络设备 | 534 | 网络适配器 | Bluetooth Device (Personal Area Network) |
| 系统概述 | 网络设备 | 534 | 网络适配器 | Intel(R) Wireless-AC 9560 160MHz (192.168.0.105) |
| 系统概述 | 网络设备 | 534 | 网络适配器 | Microsoft Wi-Fi Direct Virtual Adapter #3 |
| 系统概述 | 网络设备 | 534 | 网络适配器 | Microsoft Wi-Fi Direct Virtual Adapter #4 |
| 系统概述 | 网络设备 | 534 | 网络适配器 | Realtek Gaming GbE Family Controller |
| 系统概述 | 网络设备 | 534 | 网络适配器 | VMware Virtual Ethernet Adapter for VMnet1 (192.168.28.1) |
| 系统概述 | 网络设备 | 534 | 网络适配器 | VMware Virtual Ethernet Adapter for VMnet8 (192.168.64.1) |
| 系统概述 | 外围设备 | 536 | 打印机 | Adobe PDF |
| 系统概述 | 外围设备 | 536 | 打印机 | Fax |
| 系统概述 | 外围设备 | 536 | 打印机 | Microsoft Print to PDF |
| 系统概述 | 外围设备 | 536 | 打印机 | Microsoft XPS Document Writer |
| 系统概述 | 外围设备 | 536 | 打印机 | OneNote for Windows 10 |
| 系统概述 | 外围设备 | 568 | USB1 控制器 | nVIDIA TU116 - USB Type-C Port Policy Controller |
| 系统概述 | 外围设备 | 588 | USB3 控制器 | Intel Cannon Point PCH - USB 3.1 xHCI Host Controller [B0] |
| 系统概述 | 外围设备 | 588 | USB3 控制器 | nVIDIA TU116 - USB Type-C xHCI Controller |
| 系统概述 | 外围设备 | 537 | USB 设备 | HP Wide Vision HD Camera |
| 系统概述 | 外围设备 | 537 | USB 设备 | USB Composite Device |
| 系统概述 | 外围设备 | 537 | USB 设备 | USB Composite Device |
| 系统概述 | 外围设备 | 537 | USB 设备 | USB 输入设备 |
| 系统概述 | 外围设备 | 537 | USB 设备 | USB 输入设备 |
| 系统概述 | 外围设备 | 537 | USB 设备 | 英特尔(R) 无线 Bluetooth(R) |
| 系统概述 | 外围设备 | 584 | 电池 | Microsoft AC 适配器 |
| 系统概述 | 外围设备 | 584 | 电池 | Microsoft ACPI 兼容的控制方法电池 |
| 系统概述 | DMI | 548 | DMI BIOS 厂商 | AMI |
| 系统概述 | DMI | 549 | DMI BIOS 版本 | F.24 |
| 系统概述 | DMI | 550 | DMI 系统制造商 | HP |
| 系统概述 | DMI | 551 | DMI 系统产品 | OMEN by HP Laptop 15-dh0xxx |
| 系统概述 | DMI | 552 | DMI 系统版本 |  |
| 系统概述 | DMI | 555 | DMI 主板制造商 | HP |
| 系统概述 | DMI | 556 | DMI 主板产品 | 8600 |
| 系统概述 | DMI | 557 | DMI 主板版本 | 44.41 |
| 系统概述 | DMI | 559 | DMI 主机制造商 | HP |
| 系统概述 | DMI | 560 | DMI 主机版本 | Chassis Version |
| 系统概述 | DMI | 561 | DMI 主机序列号 | CND9203F4G |
| 系统概述 | DMI | 562 | DMI 主机识别标签 |  |
| 系统概述 | DMI | 563 | DMI 主机类型 | Notebook |

操作系统：

|  |  |  |  |
| --- | --- | --- | --- |
| Group | ItemID | Item | Value |
| 操作系统 | 3841 | 操作系统名称 | Microsoft Windows 10 Pro |
| 操作系统 | 3843 | 操作系统语言 | 中文(简体，中国) |
| 操作系统 | 3872 | 操作系统安装语言 | 中文(简体，中国) |
| 操作系统 | 3844 | 操作系统核心(Kernel)类型 | Multiprocessor Free (64-bit) |
| 操作系统 | 3845 | 操作系统版本 | 10.0.19041.329 |
| 操作系统 | 3846 | 操作系统版本升级(SP) | - |
| 操作系统 | 3847 | 操作系统安装日期 | 2020/5/28 |
| 操作系统 | 3848 | 操作系统根目录 | C:\WINDOWS |
| 授权信息 | 3850 | 注册组织 |  |
| 授权信息 | 3870 | 产品激活 (WPA) | 不需求 |
| 当前状态 | 3854 | 计算机名称 | DESKTOP-G30ECTD |
| 当前状态 | 3855 | 用户名称 | zhaox |
| 当前状态 | 3856 | 登录域 | DESKTOP-G30ECTD |
| 当前状态 | 3857 | 已运行时间 | 0001d 01:40:57 |
| 组件版本 | 3858 | Common Controls | 6.16 |
| 组件版本 | 3861 | Windows Mail | - |
| 组件版本 | 3862 | Windows Media Player | 12.0.19041.322 (WinBuild.160101.0800) |
| 组件版本 | 3865 | .NET Framework | 4.8.4084.0 built by: NET48REL1 |
| 组件版本 | 3867 | DirectX | DirectX 12.0 |
| 组件版本 | 3868 | OpenGL | 10.0.19041.1 (WinBuild.160101.0800) |
| 组件版本 | 3869 | ASPI | - |
| 操作系统特征 | 0 | 调试版本 | 否 |
| 操作系统特征 | 0 | DBCS 版本 | 是 |
| 操作系统特征 | 0 | 域控制器 | 否 |
| 操作系统特征 | 0 | 安全性提供 | 否 |
| 操作系统特征 | 0 | 网络提供 | 是 |
| 操作系统特征 | 0 | 远程部分 | 否 |
| 操作系统特征 | 0 | 安全模式 | 否 |
| 操作系统特征 | 0 | 低速处理器 | 否 |
| 操作系统特征 | 0 | 终端服务 | 是 |

开发与测试环境：

Qt Creator 4.11.1

Based on Qt 5.14.1 (MSVC 2017, 32 bit)

From revision b2ddeacfb5

Built on Feb 5 2020 10:36:26

Plugin information:

+ Android 4.11.1

+ AutoTest 4.11.1

AutotoolsProjectManager 4.11.1

BareMetal 4.11.1

+ Bazaar 4.11.1

Beautifier 4.11.1

+ BinEditor 4.11.1

+ Bookmarks 4.11.1

Boot2Qt 4.11.1

+ CMakeProjectManager 4.11.1

+ CVS 4.11.1

+ ClangCodeModel 4.11.1

ClangFormat 4.11.1

ClangPchManager 4.11.1

ClangRefactoring 4.11.1

+ ClangTools 4.11.1

+ ClassView 4.11.1

ClearCase 4.11.1

+ CodePaster 4.11.1

CompilationDatabaseProjectManager 4.11.1

+ Core 4.11.1

+ CppEditor 4.11.1

+ CppTools 4.11.1

Cppcheck 4.11.1

+ CtfVisualizer 4.11.1

+ Debugger 4.11.1

+ Designer 4.11.1

+ DiffEditor 4.11.1

EmacsKeys 4.11.1

+ FakeVim 4.11.1

+ GLSLEditor 4.11.1

+ GenericProjectManager 4.11.1

+ Git 4.11.1

HelloWorld 4.11.1

+ Help 4.11.1

+ ImageViewer 4.11.1

Ios 4.11.1

+ LanguageClient 4.11.1

+ Macros 4.11.1

McuSupport 4.11.1

+ Mercurial 4.11.1

+ ModelEditor 4.11.1

Nim 4.11.1

+ PerfProfiler 4.11.1

Perforce 4.11.1

+ ProjectExplorer 4.11.1

+ Python 4.11.1

+ QbsProjectManager 4.11.1

+ QmakeProjectManager 4.11.1

+ QmlDesigner 4.11.1

+ QmlJSEditor 4.11.1

+ QmlJSTools 4.11.1

+ QmlPreview 4.11.1

+ QmlProfiler 4.11.1

+ QmlProjectManager 4.11.1

+ Qnx 4.11.1

+ QtSupport 4.11.1

+ RemoteLinux 4.11.1

+ ResourceEditor 4.11.1

+ ScxmlEditor 4.11.1

SerialTerminal 4.11.1

SilverSearcher 4.11.1

StudioWelcome 4.11.1

+ Subversion 4.11.1

+ TaskList 4.11.1

+ TextEditor 4.11.1

Todo 4.11.1

UpdateInfo 4.11.1

+ Valgrind 4.11.1

+ VcsBase 4.11.1

WebAssembly 4.11.1

+ Welcome 4.11.1

+ WinRt 4.11.1

# 六、实验步骤：

1. 问题描述

使用Qt Creator 构建五子棋游戏。

2. 算法分析与概要设计

先定义游戏数据结构，在该结构中分别用不同的变量存储游戏棋盘和棋子的情况、各个点位的评分、下棋方、游戏模式等数据，并搭载开始游戏、评分计算、人或电脑下棋、胜负与和棋判断的自定义函数。游戏开始后，程序自动对相关参数进行设置，然后通过几大模块开始游戏，及调用AI与玩家进行对战。该项目包括两大模块，即游戏逻辑模块与游戏界面模块。游戏逻辑模块包括初始化、输赢判断、电脑AI三个模块，而游戏界面包括游戏UI与游戏界面控制两个模块。同时，游戏界面控制又包括游戏启动、鼠标移动监视、下棋动作和绘图四个小模块，均由各自的函数关联实现。

3. 核心算法的详细设计与实现

1. 关键模块实现方法
2. 电脑AI

由于总体规划已在老师的GitHub上给出，此处不再提供更多过于详细的想法，只是大致描述一下我个人的算法与思路。

基本思想：

创建一个表示棋盘格的数组用于评分，遍历整个棋盘，利用极大值极小值算法针对每个空白棋盘格计算该处的评分值，然后查找具有最大评分值的棋盘格落子，如果有多个格的评分相同，就从中随机选取一个。

评分算法：

对某个空白棋盘格，往周围八个方向进行延伸，记录玩家或电脑连成线的棋子的个数，如果遇到空棋盘格则停止循环，并记录空棋盘格个数，随后根据连成线的棋子个数和两端空棋盘格的个数给当前统计的空白格子加分，根据连成棋子的个数来加大小不等的分数，比如对电脑下的棋子“死二”加10分，“活三”加1000分；同理对玩家下的棋子也有相应的判定加分。总体来说，趋势是先尽可能阻止玩家的发展，并在前一前提已经保证之时保持电脑自身的进攻。

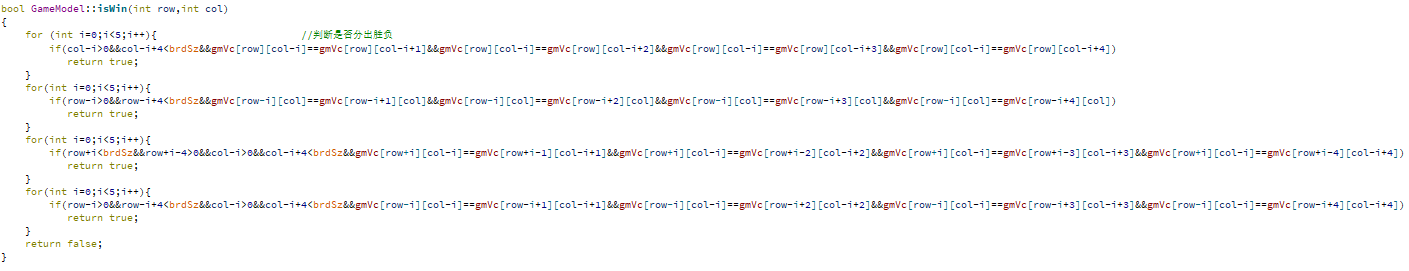
部分代码截图：

AI黑棋评分（玩家白棋评分与之类似）：

# 

1. 判断输赢及判断和棋

每当玩家或电脑下了一个棋子之后，就沿着这个棋子周围的八个方向延伸4个棋子，然后判断有没有连成5个的情况，若有则即该方获胜。若所由棋盘格全部填满仍无胜负即为和棋。

部分代码截图：

1. 游戏界面相关

大部分均使用Qt所提供的内置函数实现，同时进行少量优化。其中有用到鼠标监视，界面绘制等模块。由于与本C语言课程的内容关联程度并不是很大，在此处不再赘述。

1. 部分数据结构细节
2. 游戏数据结构中：

gmVc：存储当前游戏棋盘和棋子的情况,空白为0，白子1，黑子-1

scVc：记录各个点的评分，作为电脑的下棋依据

plOpt：标识下棋方是电脑还是玩家，或是玩家中的哪一方

gameType：选择游戏模式

gameStat：储存游戏状态相关数据

startGame：开始游戏的函数

scrCal：计算评分函数

PlayerAct：玩家下棋函数

AIAct：电脑下棋函数

MapUpd：每次执行落子后更新游戏棋盘

isWin：判断双方是否有一方获胜

isDead：判断是否出现和棋

1. 游戏界面数据结构中：

paintEvent：绘制棋盘

mouseMoveEvent：监听鼠标移动的情况

mouseReleaseEvent：实际落子情况

\*game：游戏数据结构的相关指针

gameType：存储游戏模式

clkRow, clkCol：存储将点击的点的位置

gmInit：游戏初始化函数

gminspec：检查相关

playerChess：玩家落子

AIChess：电脑落子

PVPInit：双人游戏模式初始化

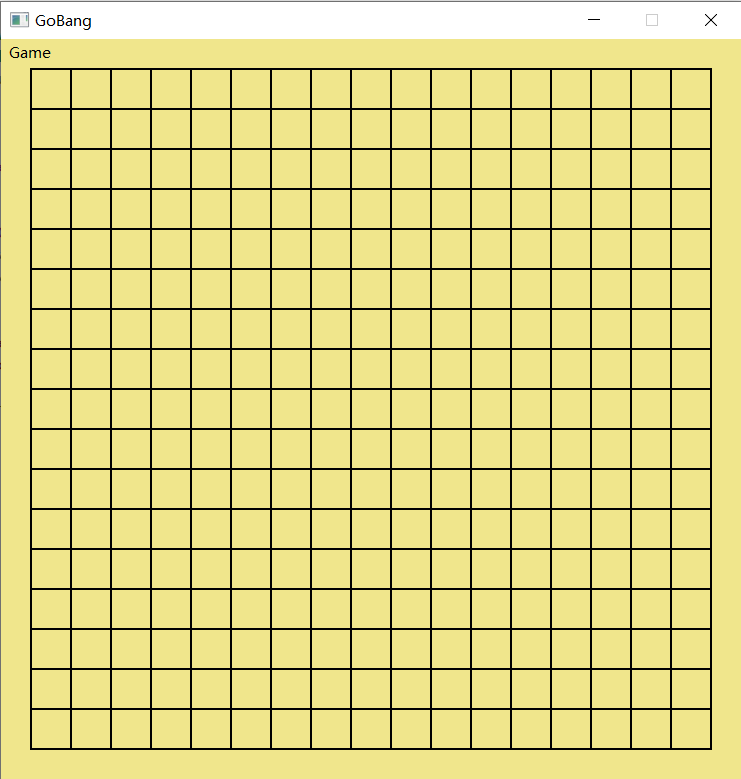
PVEInit：人机对战模式初始化

1. 优化项
2. 大部分变量名均被按照驼峰命名法书写，而且基本能够反映出该变量在整个程序、分函数中的作用是什么，让别人读起来较为浅显易懂
3. 尽可能进行了一些人性化的界面设计，如将棋盘底色设置成了与木制棋盘较为相似的黄色，为游戏加入了较适宜的音效等。
4. 尽可能删除了重复代码，使代码尽可能精简。
5. 尝试修改过几次玩家杀棋的数据权重和电脑自己下棋的局势数据权重，使电脑的棋力尽可能地强（事实上现在仍旧不尽人意）。

# 七、实验数据及结果分析：

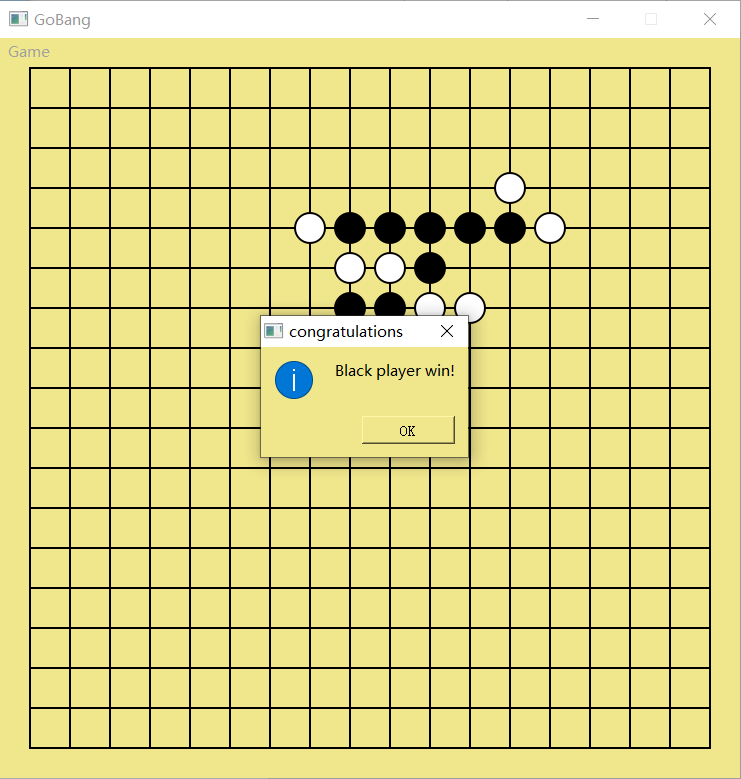
1. 游戏截图

开始主界面：

电脑胜利：

# 

玩家胜利：



1. 棋力评测

经过大致测试，该五子棋游戏能够大致识别玩家的最基本的下棋套路并在较短时间内做出相应的回击，一般会在尽可能确保玩家不取得胜利的前提下尝试获胜，相对来说十分保守。在过去的数十场比赛中，与熟悉少量游戏套路的较低级玩家对战时有一定几率获胜，虽然只有1/4左右；而在网页上参与电脑对战电脑的比赛以及在线人机对战的比赛，只能在极少数情况下取得胜利，并且据观察经常会出现当对手打出冲四的棋时仍不会采取防守、当出现死三时仍会补一子形成死四、没必要进行防守之处仍旧多此一举的场面。总的来说，该五子棋AI虽具备少量五子棋经验，但仍然与玩过五子棋的玩家之间有很大实力差距，在之后的版本维护中应当进行革新。

1. 性能评估



如图所示，因棋力有限，人机对战过程中该五子棋游戏对Intel Core i7-9750H的CPU的峰值占用大约仅有2%，而对NVIDIA Geforce GTX 1660Ti的GPU的占用基本为0，内存占用约为11.4MB左右。而根据笔者测试，在相同环境下的2009年游戏《Plants vs. Zombies Game of the Year》在游戏首页的内存资源占用已达到250MB，在切换进入游戏关卡场景的过程中CPU占用已能突破25%。从这项对比可以看出，构建出的五子棋游戏占用内存空间小、所需配置极低，但是正因如此距离真正的现在意义所说的游戏仍有很长的路要走。

# 八、总结及心得体会：

1. 现阶段完成Qt中的一部分操作我仍旧更像是在照猫画虎，而不是完全对各种函数运用自如。希望之后我能更加深入地学习Qt的知识，在下次完成这类项目的时候能够更加熟练。
2. 感觉现阶段该五子棋游戏的能力仍旧很一般，只能达到一个新手玩家的水平，甚至与大部分普通玩家之间仍然有较大的能力差距。我觉得我还应当在今后的学习中继续调整其中的一些算法以及评分函数的分数权重，让我的游戏的AI水平有所提升。
3. 其中有一些小的特性（其实是Bug）没有被解决，视觉设计在我的心中并没有达到现阶段来说的完美的地步。例如，棋盘被我做成了17×17（如果做成15×15会有Bug），而实际上有效的区域只有15×15，看起来让人很难受。

# 九、对本实验过程及方法、手段的改进建议及展望：

## 相对易于实现的设想——现版本又不是不能用，要什么自行车

1. 将棋盘恢复真正的15×15大小。毕竟五子棋的规范棋盘大小为15×15，作为一个完美主义者我很希望将其恢复成它本该拥有的规格，而且15×15的大小看起来比17×17更加和谐。
2. 体验增强，让下棋感受起来更接近现实生活中的操作感受，增强下棋体验。虽然现在的背景颜色与棋盘格整体设计已经比较贴近生活中的棋盘，但是颜色过于单调不够美观；其次希望在下棋时每次落子声音在六七个声音中随机选择一个并适当调整音量，尽可能还原下棋时的真实感受。现在的落子音效是iOS的虚拟键盘打字声音，胜利音效是Intel广告最后的“灯~等灯等灯~”，和棋音效则源自BMW的广告，部分情况下听起来总是怪怪的。
3. 加入悔棋功能。下棋时悔棋是在平常娱乐游戏中的一个很普通的诉求，算法实现似乎并不困难，如果要将这个项目完善成日用的五子棋游戏，这个功能应当考虑加入。
4. 配备图标。可以看见我的/res目录下还有一个image文件夹，我本想为构建出的游戏增添一个好看的图标，但是最后失败了。我觉得，一个好的程序，没有图标的话会是不完美的吧。希望之后我能成功将其加入。

## 画饼——短期基本不可能实现，但我很希望我的项目适配的功能

1. 联机对战：想法源自与好朋友联机游玩Minecraft

五子棋的最大乐趣在于对战，而没有深度学习的电脑即使战胜玩家也通常在比赛时灵活性低于玩家，并不能完全满足玩家日常娱乐的要求。老师在上课提出之前有大神实现了玩家之间局域网联机的飞机大战游戏，而前段时间与同学使用内网穿透的方式实现Minecraft远程联机来替代服务器的搭建给了我启发。既然游戏能够通过局域网联机实现，为什么不能也使用内网穿透来进行远程联机呢？但是接口、数据处理等方面我仍未曾学习，我觉得有了那些知识之后很有必要为五子棋配备这个功能。

1. 窗口缩放：想法源于游玩植物大战僵尸

植物大战僵尸作为2009年就出现的电脑游戏，许多功能并不完善。当时的电脑分辨率很低，植物大战僵尸启动后无法调整窗口大小，但是其窗口基本能能占满整个屏幕。而到现在，屏幕分辨率不断上升，植物大战僵尸在非全屏模式下的窗口只占到1080P屏幕的1/4左右，很不人性化。我的五子棋也是如此，暂时无法调整窗口大小，虽然五子棋并不考验眼力，但是我认为窗口缩放是一个非全屏游戏该有的功能。而我将其归入“画饼”中而不是前者就是因为，我认为这个功能连植物大战僵尸的开发者都没有实现，一定有些许难度，有一些自己的原因。

1. 高阶五子棋规则：想法源自对五子棋一些技巧的了解

五子棋之所以没有像围棋和象棋那样发展成大规模赛事，在我看来就有它的游戏规则不够明确、规定过于简单这一原因。而后来的规定，如上文所述，黑方指定开局、三手可交换、五手两打、黑方禁手等，能很好的防止一些万金油式技巧的使用，增强五子棋游戏的不确定性与娱乐性，对玩家更为友好。

1. 自定义设置：想法源于游戏Minecraft和手机系统MIUI

Minecraft作为一个游戏，与其他游戏很大的不同之处就在于很多游戏规则可以自行更改，只需要一两条指令就能将游戏规则设定成玩家喜欢的样子。而MIUI也是如此，与原生Android比起来，有着很多实用的功能，并且为用户提供了开启和关闭的选择，能够满足大部分人的需求。我希望我构建出的五子棋也是这样，对于在上面3)中所说的高阶规则，每个都有一个开关能够相应地控制该规则，在每一盘游戏开始前都可以由玩家进行选择，把选择权全部交给玩家。

1. 难度设定：想法源于各种游戏的人机对战时的难度选择

诸如League of Legend和Forza: Horizon 4等游戏，在人机对战时都可以自由在四五个难度等级间自由进行选择，为玩家带来很多便利。比如，我是新手，我就可以尝试“菜鸟”难度，找到玩这个游戏的自信；我很累想放松放松，就可以选择“简单”难度，不用动太多脑子就能来一盘。而我想学习一些游戏中的技巧，我就想挑战“地狱”难度，来看看电脑玩这款游戏的套路是什么。我想，五子棋虽然不像其他游戏那样变幻莫测，但是仍然充满各式各样的套路，不同的玩家对AI的落子灵活度需求也就不同。将难度分开，就要写出不同的AI，且不同的AI之间应当有较为明显的难度差异，在这方面我认为我自己还有很长的路要走。

1. 深度学习：想法源于AlphaGo战胜人类棋手

很早之前AlphaGo击败人类最强围棋选手之时，我就大致听说它有如此之强的棋力原因之一就是它模拟了大量棋盘对弈，在与人类和自己对弈的过程中自己学会了很多东西。虽然我现在对它没有基本的认识，但我希望有一天五子棋也能像AlphaGo那样击败人类，其中应该少不了深度学习。

**报告评分：**

**指导教师签字：**