

面向对象程序设计（C++） 课程设计报告

题目：2048 游戏的开发

专业：计算机科学与技术

设计日期：2022 年 5 月 2 日 至 2022 年 5 月 27 日

小组成员人数：2 人

小组成员名单：

组长 班级：计 2003 学号：20204535 姓名：韩燚 成绩：

组员 班级：计 2003 学号：20204977 姓名：李文昊 成绩：

报告撰写的内容与要求

1. 项目简介：介绍本次课程设计选题的目的、意义、任务概况、本人在项目中的分工等内容。
2. 项目内容：系统的设计与实现的全面描述，介绍系统整体结构、系统框架图、UML 类图、程序流程图、重点难点分析及解决方案、调试难点及解决方法、系统交互界面及结果展示等。本部分内容应以记叙或白描手法为基调，在完整叙述的基础上，对自己认为有重要意义或需要研究解决的问题进行重点叙述，其它内容则可简述。
3. 总结或体会：对课设效果进行综合评价，着重介绍自身的收获与体会，内容较多时可列出小标题，逐一列举。总结或体会的最后部分，应针对实习中发现的自身不足，简要地提出今后学习，努力的方向。
4. 报告正文一律采用计算机排版、A4 纸**双面**打印，正文字体为**小四号**宋体，**1.35 倍**行距，正文页数不少于 10 页、不多于 20 页(其中代码不超过 3 页，**中文字数**不少于 5000 字)。要求语句通顺、论述严谨、规范、正确。
5. 请注意封面页、扉页、评语页等的打印及装订顺序。分别为①封面页（其背面为扉页“实习报告的内容与要求”）、②目录、③正文、④其他附件（如有）。

目录

一 . 项目简介	1
1.1 项目内容	1
1.2 文件与执行环境.....	1
1.2.1 电子文档打包文件名及文件列表	1
1.2.2 编译执行环境与运行步骤.....	2
1.3 任务分工	2
二 . 项目内容.....	3
2.1 系统的整体架构.....	3
2.2 UML 类图	4
2.3 系统的整体设计	5
2.4 功能模块设计	6
2.5 重点难点分析及解决方案	7
2.5.1 重难点分析.....	7
2.5.2 解决办法	7
2.6 调试难点及解决方法	8
2.6.1 调试难点	9
2.6.2 解决方案	9
2.7 结果展示	9
三 . 总结与体会.....	14

一. 项目简介

1.1 项目内容

项目简介：

利用 C++ 的 OOP 编程思想、C++ 语言的特性和 QT 以及一些自行拓展的新知识，实现了键盘操控的 2048 游戏。

在实现课程设计的过程中，巩固并深入理解课内学习的 C++ 语言的语法以及一些特性，同时开发面向对象编程的思想和能力；扩展知识面，在理解并熟练使用课上所学习的内容的基础上，学习了一些课外内容，例如 C++ 图形库——QT、lambda 表达式等；增加自己开发实际项目的经验，进行合理的需求分析，对所学的内容进行取舍，使代码更加简洁高效。同时通过阅读一定量他人的代码，使自己更加熟练的看懂别人的编程思想及算法思维，并从中学习，提高自己的编程能力。

主要功能：

- (1) 欢迎界面：开始游戏、关于游戏、排行榜
- (2) 登录界面：用户注册、用户登录
- (3) 游戏界面：基本游戏与游戏信息、AI 帮助、重新开始、存档
- (4) 结算界面：再来一局、退出、排行榜
- (5) 存档界面：保存存档、读取存档、查看存档、删除存档、返回游戏

1.2 文件与执行环境

1.2.1 电子文档打包文件名及文件列表

文件清单如下：

\\转专业第 3 组 2048 游戏的开发

\\工程文件（包含项目源代码、资源文件、.pro 工程文件）

\\可执行文件（包含可执行程序.exe、动态库.dll）

转专业第 3 组 2048 游戏的开发.doc

转专业第 3 组 2048 游戏的开发.pdf

1.2.2 编译执行环境与运行步骤

- (1) 编译环境: Windows10 QT 5.14.2 MinGW 64-bit
- (2) 运行环境: Windows10 Qt Creator
- (3) 数据文件: ① archivelist.txt 存档信息文件 ② scorelist.txt 分数信息文件 ③ user.txt 用户信息文件
- (4) 运行步骤: ①点击工程文件 2048game.pro 启动 Qt Creator, 构建并运行 ②运行打包文件下的 2048game.exe

1.3 任务分工

任务概况:

本次课程设计时间周期较短、任务量大, 出于图形化界面的考虑, 需要在一定的时间内完成 QT 的学习, 并且能应用代码进行编写窗口。

将课程设计分为四个周期:

- ① 第一周: 学习 QT 的基本使用方法与机制, 对任务进行规划并分工
- ② 第二周: 完成基本游戏及各项功能的编写, 能够在控制台运行游戏
- ③ 第三周: 将游戏迁移到 QT 编写 GUI 界面上, 进行基本的调试
- ④ 第四周: 测试 bug, 界面美化

将任务分为以下内容:

- ① 欢迎界面: 开始游戏 (跳转到登录界面)、游戏基本说明、排行榜、退出游戏
- ② 登录界面: 用户注册、用户登录、基本的查重提示
- ③ 游戏界面: 显示游戏信息 (游戏主体、分数、步数、用时)、重新开始、AI 帮助、游戏存档
- ④ 结算界面: 显示结算信息 (游戏用时、胜利或失败提示、分数、步数)、重新开始、排行榜、退出游戏
- ⑤ 存档界面: 显示存档列表 (存档日期)、增加存档、删除存档、查询存档信息 (鼠标悬停显示)、读取存档、返回游戏

任务分工:

项目的任务分工如表一所示

表 1 任务分工表

姓名	完成内容
----	------

韩焱	①游戏界面：核心游戏实现，重新开始，存档，界面设计美化；②结算界面：胜利结算，失败结算，界面设计美化；③负责统筹组员完成自己相关的部分；
李文昊	①开始界面：进入游戏，排行榜，游戏信息，界面设计美化；②登录界面：用户登录，用户注册，界面设计美化；③游戏界面：AI 帮助

二. 项目内容

2.1 系统的整体架构

根据课设题目要求，规划出系统的整体架构。

系统分为 5 个部分：欢迎界面、登录界面、游戏界面、存档界面、结算界面。

欢迎界面，应当有开始游戏、游戏说明、排行榜三个功能。开始游戏，跳转到登录界面；游戏说明、排行榜，弹出新的界面。

登录界面，主要的功能是用户注册与用户登录。成功登录或者注册后，跳转到游戏界面。

游戏界面，是游戏的主体部分。游戏界面应当显示游戏的信息，包括 2048 的 4*4 游戏矩阵、分数、步数和用时等；使用 W、A、S、D 进行游戏的操作，并在游戏矩阵上实时显示；存档、AI 帮助、重新开始等其他功能。

存档界面，储存之前的游戏存档。此界面应该显示已有的存档列表，能够实现保存当前游戏、读取存档、删除存档和返回游戏界面等功能。

结算界面，应有重新开始游戏、查看排行榜、退出游戏的功能。

系统整体结构如图 1 所示：

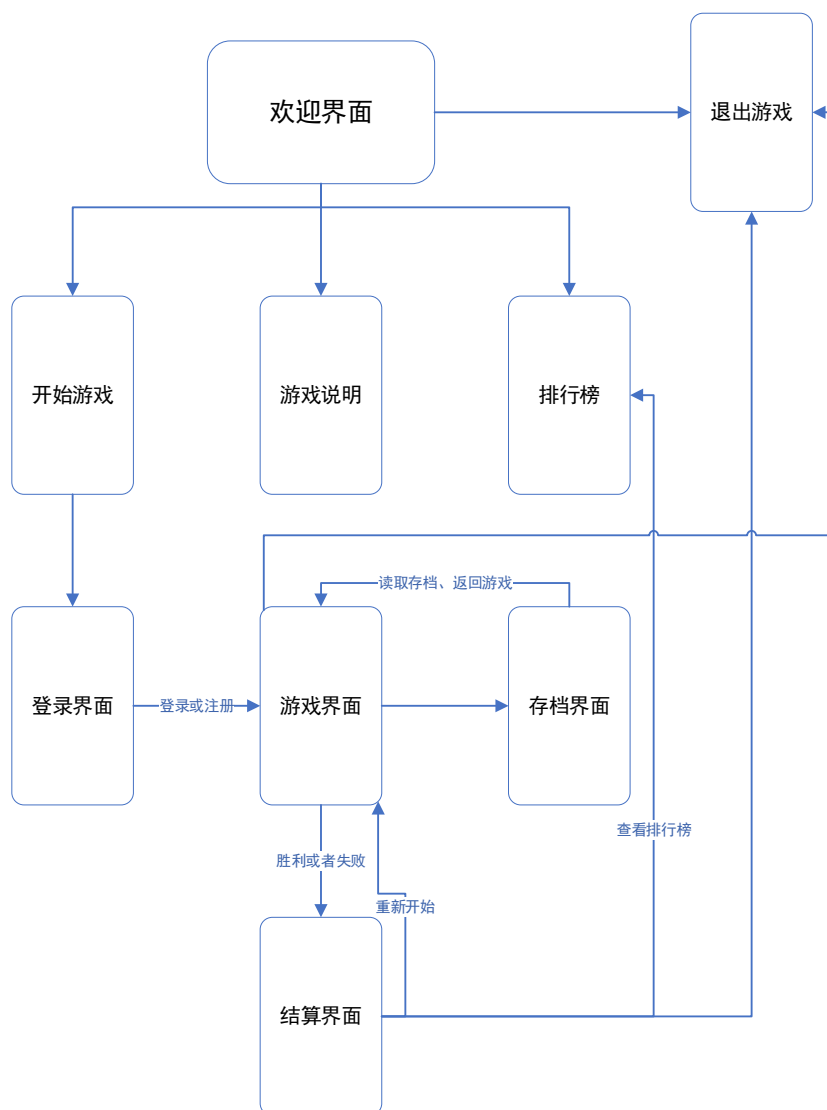


图 1 系统整体结构图

2.2 UML 类图

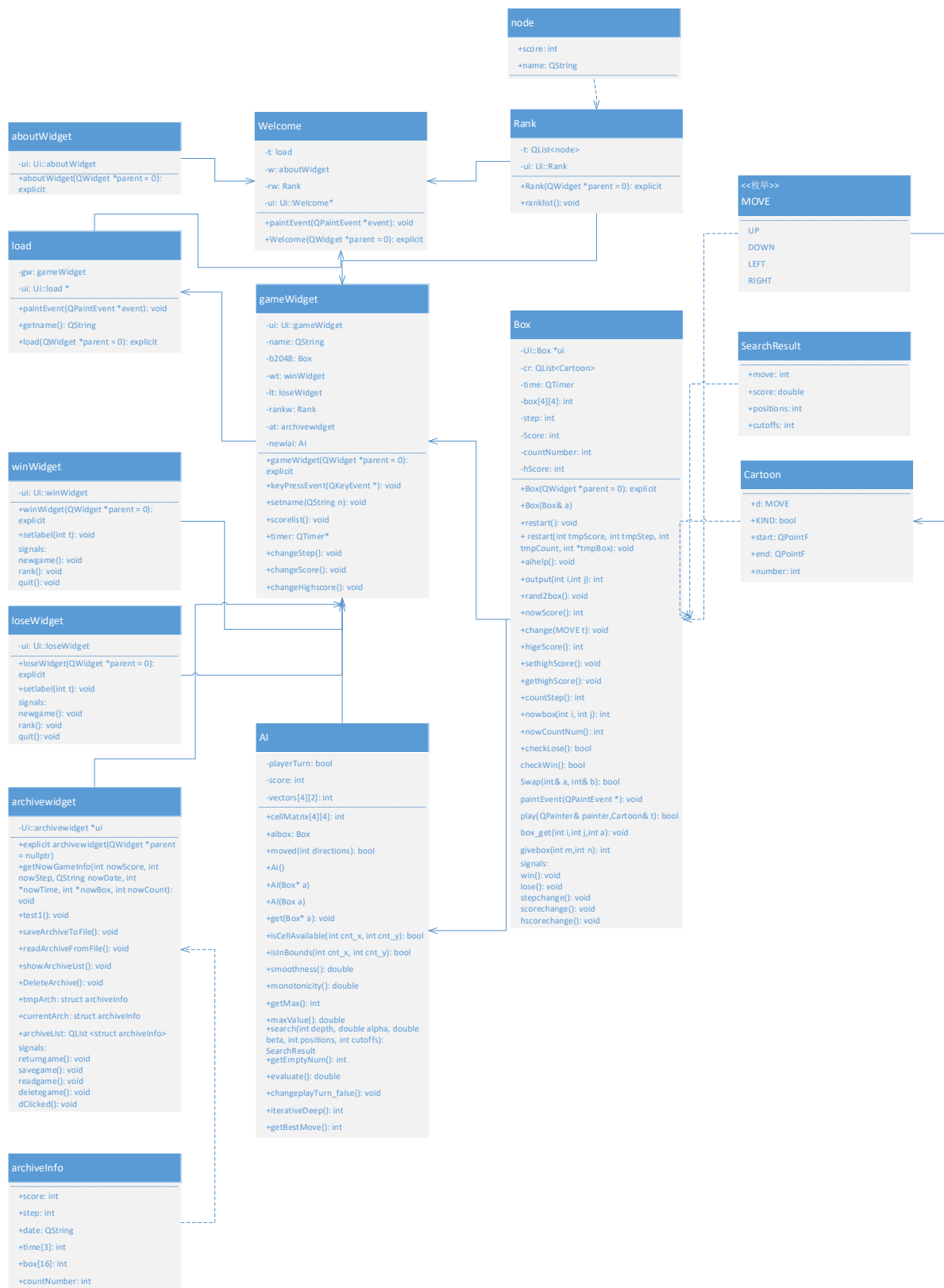


图 2 UML 类图

2.3 系统的整体设计

系统的整体程序流程图如图 3 所示

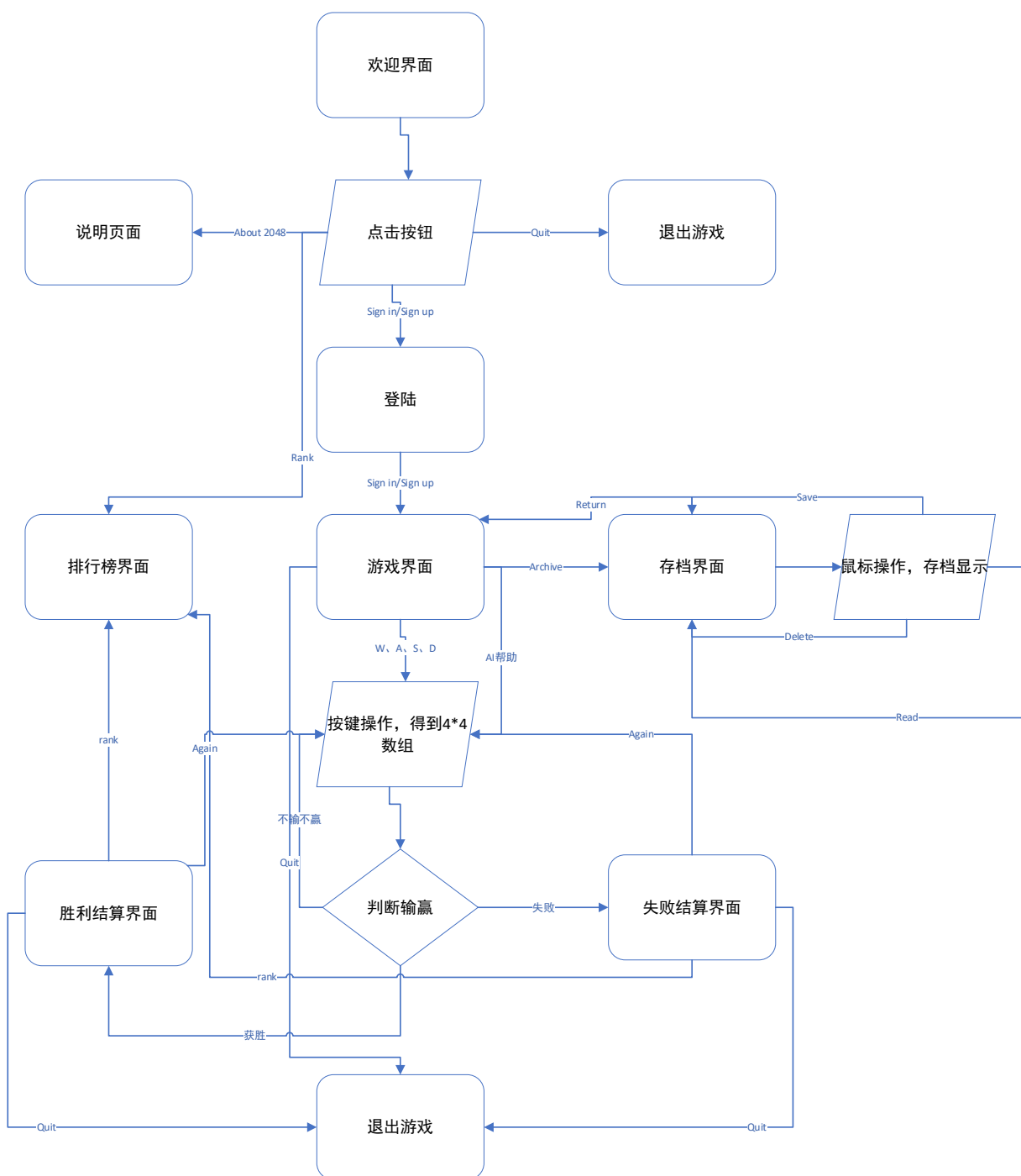


图 3 系统整体程序流程图

2.4 功能模块设计

系统总体分为三大功能模块：开始模块、登录模块、游戏模块。

开始模块，如图 2、图 3 所示，包含 Welcome 类，此功能模块是游戏的初始界面，主要作用为游戏的基本功能说明，查看排行榜等。

登录模块，如图 2、图 3 所示，包含 load 类，此功能模块是游戏的登录界面，主要作

用为用户注册、用户登录游戏。

游戏模块，如图 2、图 3 所示，主体为 gameWidget 类。游戏模块下属 4 个子模块，分别为游戏实现模块、存档模块、结算模块、AI 帮助模块。游戏实现部分为 Box 类，存档部分为 archivewidget 类，结算界面为 win/losewidget 类，AI 帮助为 AI 类。

系统中使用的数据表如表 2 所示。包括如下字段，其意义为游戏矩阵、存档信息、排行榜信息、用户信息。

表 2 数据表

字段	关于该字段描述
Box	游戏矩阵，储存游戏运行信息，是一个 4*4 的矩阵
archInfo	存档信息，包括存档时间、日期、分数、步数、游戏矩阵等信息
scorelist	排行榜信息，包括用户名与分数
user	用户信息，包括用户名和密码

2.5 重点难点分析及解决方案

2.5.1 重难点分析

1. 如何实现界面的切换
2. 如何绘制移动动画
3. 如何实现一个基本的 2048 游戏（随机生成新出现数字、如何实现数字移动、如何实现数字相加、如何加分）
4. 如何实现存档功能
5. AI 功能的实现。

2.5.2 解决办法

1. 如何实现界面的切换：
每个界面都设计成一个类，并继承 Qt 中的 Widget 类，点击按钮后，显示该类，并利用 this 指针关闭原有的类。
2. 如何绘制移动动画：
全部动画都通过 paintEvent 事件一帧一帧来绘制，使用 Qt 提供的 QList 链表模板，建立记录动画效果、移动的起始结束坐标及该动画对应的数字的结构体的链表。
3. 如何实现一个基本的 2048 游戏：
随机生成新出现数字：通过随机函数 rand，和随机数种子 srand，还有 time 函数来实

现。在主函数中设置随机数种子，目的是防止伪随机，及每次随机连续随机出的数都相同。

实现数字移动和相加：玩了网页上最经典的 2048 游戏([2048 Game - Play 2048 Game Online](#))，查询了网络上关于 2048 规则的一些资料得出结论。总体来说游戏分为两个部分，一个是方块移动，一个是方块相加，其中方块移动优先度高于方块相加。每次先将所有非零方块进行移动，然后检查该方块的移动方向上是否可以进行相加操作。

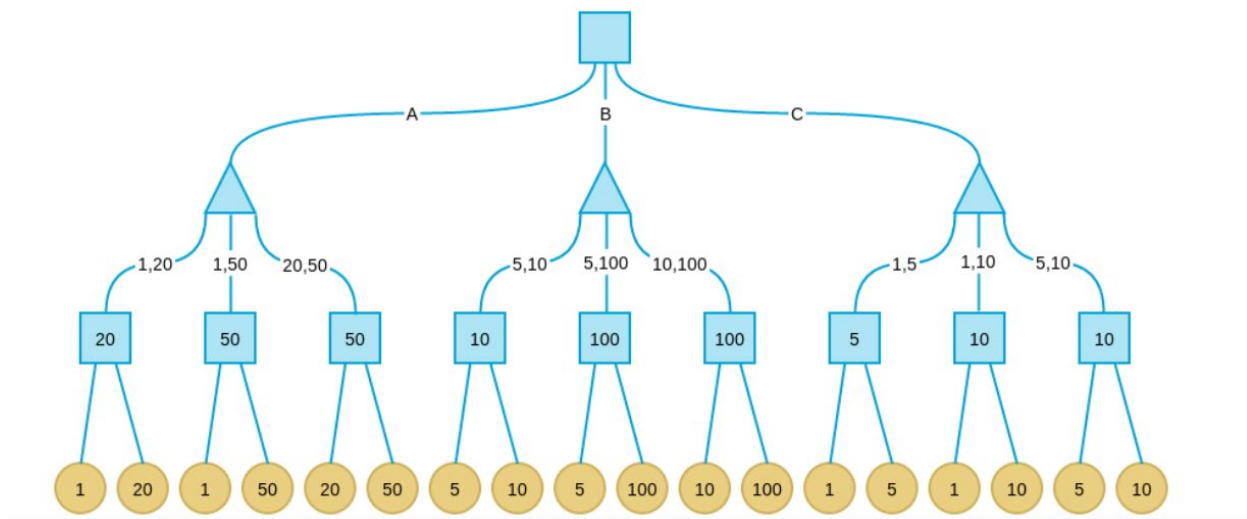
如何加分：经过观察，发现每次加的分是每次合成的方块数值。

4. 如何实现存档功能：

使用一个临时的 archInfo 类型变量记录每次的存档信息，通过 Qt 自带的类 QFile 将存档信息分行分块写入文件中。重载了开始游戏函数，通过 Qt 的信号与槽的连接机制，在读取存档信息后，传入信息参数，重新开始游戏。

5. 如何实现 AI 功能：

参考了五子棋最开始 AI 的算法，类比为信息双向透明的人机对弈，我方负责上下左右移动，电脑方向负责在某个位置添加 2，我方胜利条件为合成 2048，电脑方胜利条件为让我方无法走动。由此可以类比五子棋采用 Min-Max 算法，即为设定一个判定函数，在假设电脑方走动目标是让我方判定函数得分最小的情况得到最大分数，同时采用 $\alpha - \beta$ 剪枝算法进行速度优化。判定函数则由最大值、平滑变化度、空格数三个方面按照一定权重进行分配（权重配比由调试时具体表现进行动态变化）



2.6 调试难点及解决方法

2.6.1 调试难点

1. 在进行课程设计任务的过程中，我们小组采用的是 Qt Creator 这个 QT 公司开发的，与 QT 图形库适配的 IDE。不同于以往的诸如 CLion、VisualStudio、VisualStudioCode 等 IDE 和编辑器，由于 Qt 的 GUI 图形库的特殊性，界面只有当全部代码被编译完后才会出现，很难在 Qt Creator 里采取设置断点的方法进行单步调试。

2. 由于本次课设并未采用 git 等版本控制系统，给代码回退和组装调试带来了一定程度上的困难。

3. 动画在方块在移动的时候显示正常，而当方块到达位置的时候来回移动。

4. 调试 AI 时为确定参数需进行多局游戏然后评价每局的好坏不容易量化且浪费时间。

2.6.2 解决方案

1. 基于断点调试的困难，我们采取了另一种调试方式。起初，我们打算使用 cout 在控制台输出程序信息，从而判断传参等过程是否正确运行，但是却发生了莫名其妙的错误，经过查询资料，我们认为这是 Qt 的机制问题。无奈之下，我们开始寻找另一种方案。我们在 Qt 的库里面找到了 qDebug 这个函数，可以在程序运行的过程中在控制台打印内容，且程序运行正常。

2. 对于这个问题，我们进行了合理的分工，事先规定好接口的标准，确保彼此的代码互不影响；我们严格遵守面向对象程序设计的要求，所有的代码都进行了封装，便于使用对方写的类里的方法。

3. 一开始利用 qDebug 调试了很久都没有得出结论，偶然间，在检查动画移动代码的时候，发现移动到终点的判断是“==”，但实际上，由于误差，这个是很难实现的，于是，将终点判断改为“<=”，并在到达终点时，令当前坐标等于终点坐标，从而解决了这个问题。

4. 通过得分确定好坏量化指标且调试时让 AI 自行启动减少时间。

2.7 结果展示

欢迎界面如图 4 所示，点击 About 2048 进入说明界面，如图 5 所示；点击 View RankList，进入排行榜界面，如图 6 所示；点击 Quit Game 退出游戏。

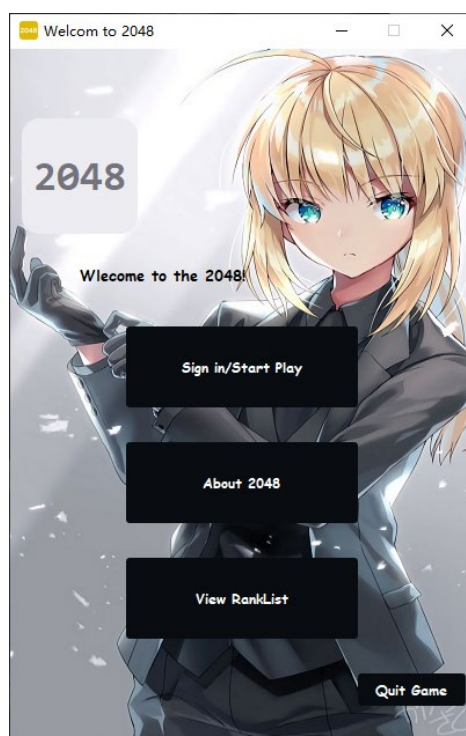


图 4 欢迎界面

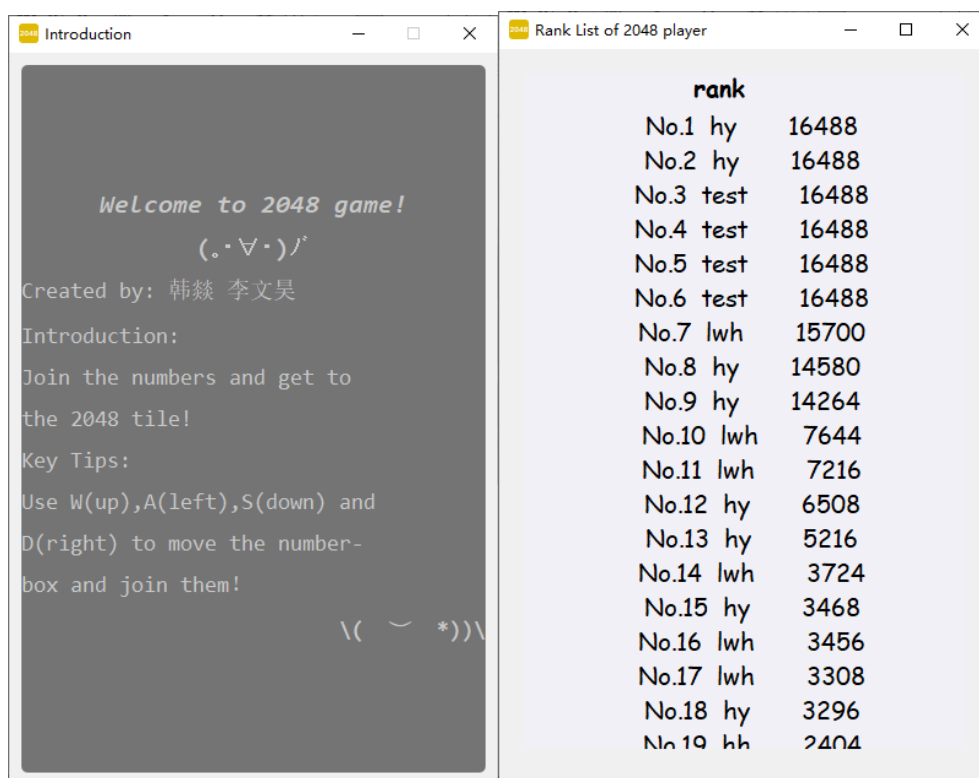


图 5 说明界面

rank		
No.1	hy	16488
No.2	hy	16488
No.3	test	16488
No.4	test	16488
No.5	test	16488
No.6	test	16488
No.7	lwh	15700
No.8	hy	14580
No.9	hy	14264
No.10	lwh	7644
No.11	lwh	7216
No.12	hy	6508
No.13	hy	5216
No.14	lwh	3724
No.15	hy	3468
No.16	lwh	3456
No.17	lwh	3308
No.18	hy	3296
No.19	hh	2404

图 6 排行榜

在欢迎界面点击 Sign in/Start Play，进入登陆界面，如图 7 所示。

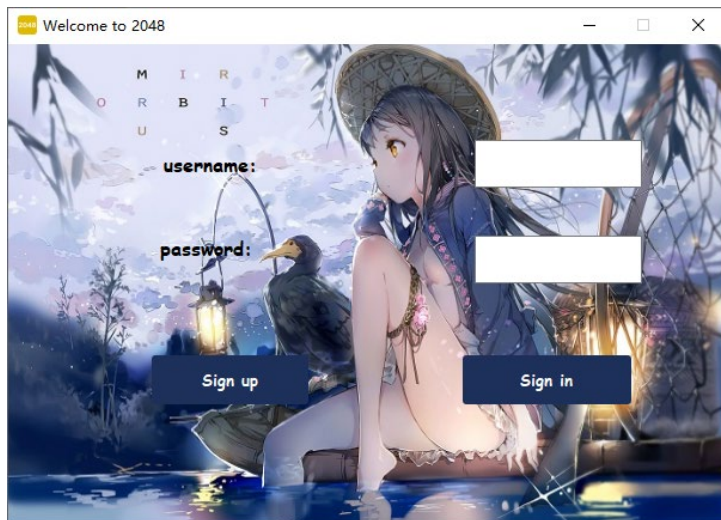


图 7 登陆界面

登陆或者注册，进入游戏界面，如图 8 所示。

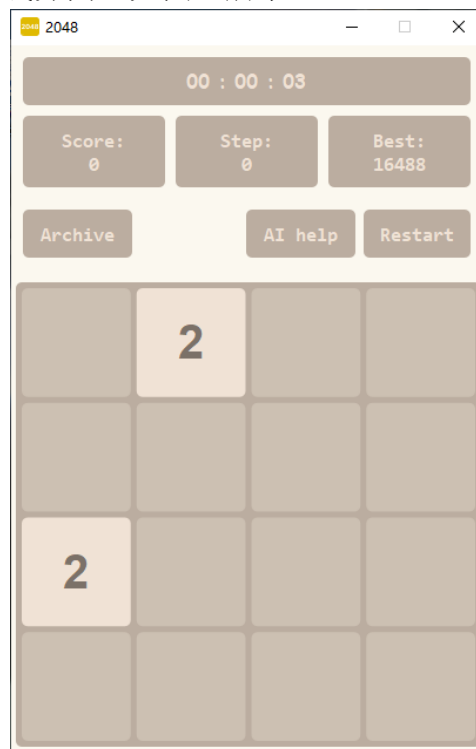


图 8 游戏界面

通过 W、A、S、D 移动方块、合成方块，如图 9 所示。

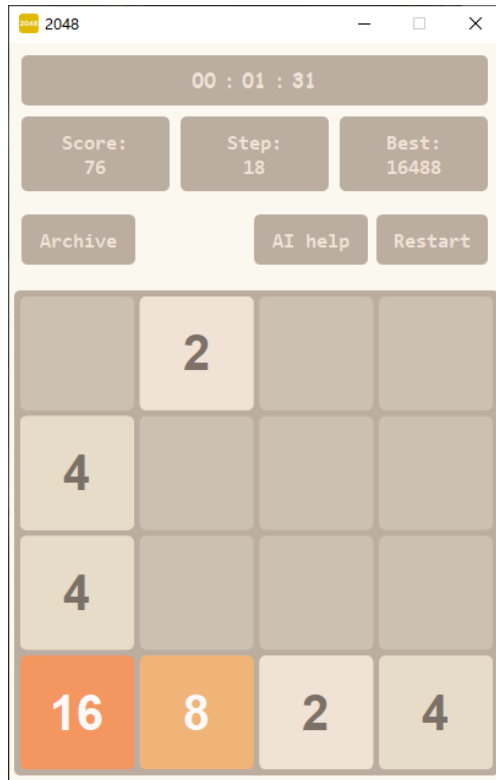


图 9 游戏实现

点击 AI help，调用算法，实现帮助移动，如图 10 所示。

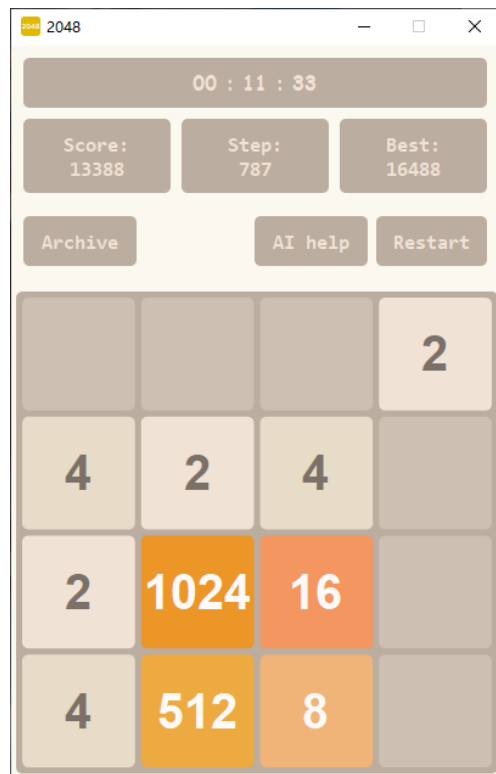


图 10 用 AI help 打出的存档

失败结算页面，如图 11 所示；胜利结算界面，如图 12 所示。



图 11 失败结算界面

图 12 胜利结算界面

点击 **Archive**，进入存档界面，如图 13 所示。点击返回按钮，返回游戏界面；点击 **save**，保存当前存档；点击 **read**，读取选中存档；点击 **delete**，删除选中存档。双击存档，可以直接读取存档，鼠标悬停在存档上方，可以查看存档信息，如图 14 所示。

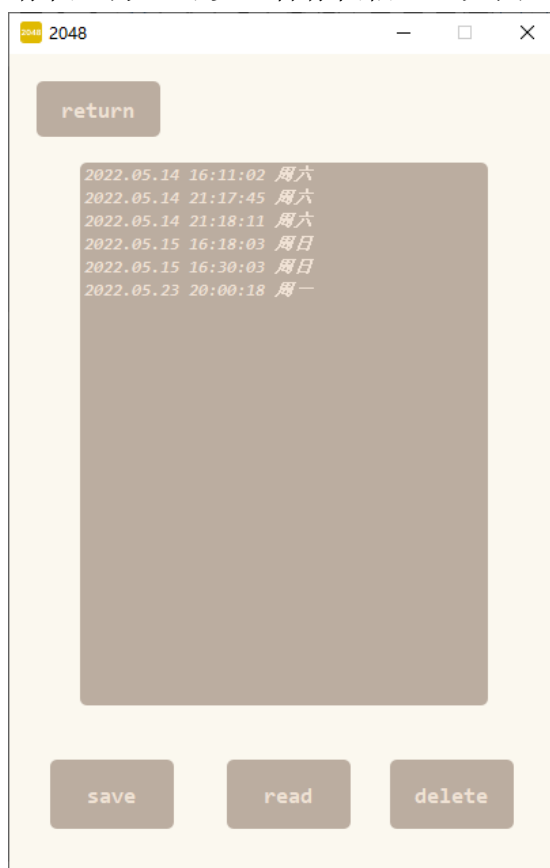


图 13 存档界面

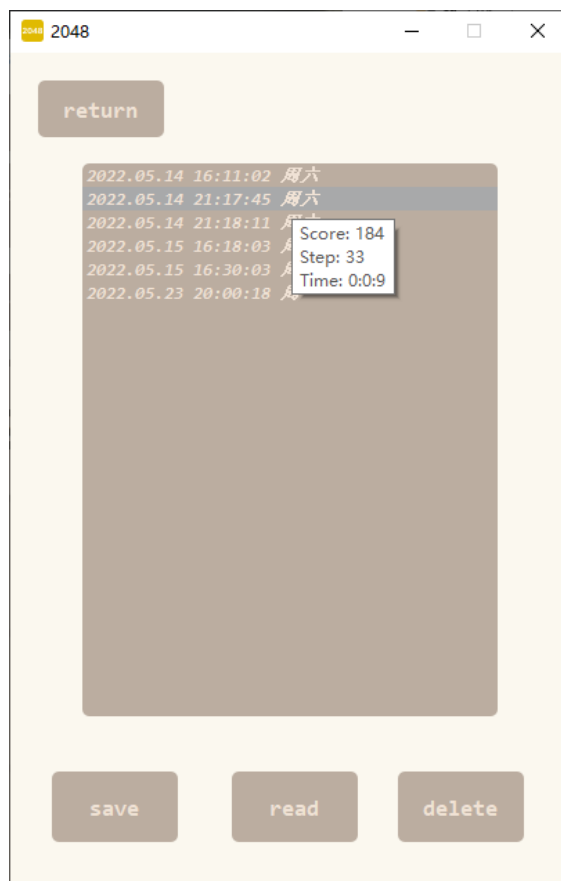


图 14 鼠标悬停显示存档信息

三. 总结与体会

韩燧

收获和体会:

由于此前做了太多的管理系统，本次课设便想着做点不一样的东西，一来二去，就选择了做 2048 游戏。课设结束之际，有太多感受和体会。

一是学习能力上。课设之前，玩过很多游戏，但是从来没有做过游戏。此时选择做 2048 游戏，有种面对熟悉的陌生人的奇妙感觉，就像是平常去 KFC 吃饭，直到有一天去了 KFC 打工那种新鲜感。首先为了做 2048 游戏，先去玩了很多次，在玩的过程感受游戏机制，结合查找的资料，弄懂了 2048 游戏的算法原理；其次，由于 C 语言课设上使用 easyX 图形库给我留下了不好的印象，决定此次课设使用 QT 这个功能强大的 GUI 库。学习 QT 时，面对各种各样陌生的机制，加上时间紧张，学起来非常的痛苦，但是毕竟是自己选择的，只好硬着头皮去学。在此过程中，逐渐熟悉起了 QT 的基础机制，使用起来不像刚开始那样滞涩。这个过程，让我明白了如何去自学计科这方面新的东西，对我有着莫大的裨益。

二是团队合作上。一开始，想着独立完成这个游戏，因为一个人写起来肯定会流畅些，代码也好理解。后来还是找了队友，我们相互约定好了代码规范，交流如何实现功

能的想法。沟通的过程带来了平常自己一个人写项目从未有过的满足感，也许这就是团队合作的乐趣。

三是锻炼耐心上。就像前文调试问题所说的那样，遇到了抓耳挠腮，苦思冥想而不得解的 bug 上，渐渐地不再急躁，而是从头审视自己的想法，合理设置打印信息的地方，出去散心转换心情回来再接着写代码。

不足之处：

做课设，收获了许多，也暴露了不少问题。比如对 C++ 面向对象的机制还不熟悉，导致一开始出现许多可笑的错误；最初任务分工不明确，导致重复工作，浪费时间等等诸如此类的问题。所幸的是，大多数问题都在过程中顺利解决，余下来的，知道了问题所在，相信在以后也能顺利的克服。

改进与建议：

在学习新知识的过程，使用思维导图等方式记录自己学习过的内容，多写使用文档，方便以后进行复习回忆；代码格式一定要规范，这样既让自己看的舒服，也能够减少别人阅读代码的痛苦；事先做好规划，能够事半功倍，不至于出大问题。

李文昊

收获和体会：

当时最开始接触计算机的时候，游戏便是最直观的印象，而 2048 也算曾今占据我大把时间的游戏，当时是我与电脑“斗智斗勇”，而如今设计的 AI 却可以看作我让电脑与电脑“斗智斗勇”，想想其中也有几分趣味。其中的艰难险阻也曾动摇我是否要把它做下去的决心，但想想从前不管是五子棋还是象棋见到 AI 的“神之手”时是多么的欣喜若狂，而现在一个简简单单的 AI 也将从我的手上诞生，一切困难也成为了我一路上披荆斩棘的动力，“踏浪碧海真情在，又见东风卷潮来”，苦其心志，劳其筋骨，也只为最好的它动起来的那一瞬间的灿烂烟火，明明是机器的无情指令，但在我看来此时全是一幅沙鸥翔集，锦鳞游泳的勃勃生机万物竞发之态！第一次看它拼出 512，第一次看它拼出 1024，第一次看它拼出 2048！那种激动之情甚至比我自己玩时更为激动，踏遍青山，独登危楼，心中怀着炬火，满眼尽是星光！第一次感觉 0 与 1 的交响诗从我手中奏响，二进制的乐章在我耳边回荡，计算机的歌声绕梁的千年，而我如今终于为它增添了一个小小的音符，或许它是点点尘雾，但仍能益补山海；或许它是点点星光，但仍能增辉日月！

不足之处：

AI 是基于电脑最坏情况实现的，但对于实际情况电脑走的为随机情况，大部分情况下并不会走到最坏情况使得 AI 走动会略显保守，基于此改动应该可以使 AI 性能更加提升。

改进与建议：

在在学习新知识的过程，提前搭好框架，确定每个函数的意义，方便以后进行复习回忆；代码格式一定要规范，可以使用 clang 等通用格式，这样既让自己看的舒服，也能够减少别人阅读代码的痛苦；工欲善其事必先利其器。