高级语言程序设计 实验报告

南开大学 计算机科学与技术

姓名：何闻硕

学号：2411783

班级：计算机科学卓越班

2025年5月8日

目录

[高级语言程序设计大作业实验报告 3](#_Toc2018946449)

[一、 作业题目 3](#_Toc43862096)

[二、 开发软件 3](#_Toc603356551)

[三、 课题要求 3](#_Toc195771523)

[四、 主要流程 3](#_Toc387039857)

[1. 根据需求设计封装对应类 3](#_Toc250909836)

[2. 收集图像素材 4](#_Toc1531214591)

[3. 处理窗口绘制问题 4](#_Toc1827088936)

[4. 实现棋子、道具生成逻辑 4](#_Toc1015078899)

[5. 人机模式实现 4](#_Toc820963725)

[6. 实现结算逻辑 4](#_Toc354894100)

[五、 收获 5](#_Toc1143050981)

高级语言程序设计大作业实验报告

# 一、 作业题目

基于 QT 的简单棋盘二维游戏

# 二、 开发软件

QT6

# 三、 课题要求

（1）面向对象

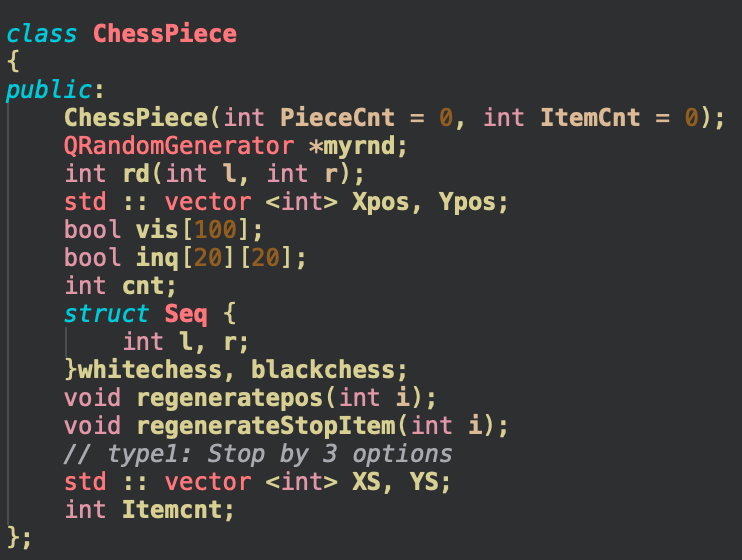
（2）使用 c++

（3）实现图形化

# 四、 主要流程

## 1. 根据需求设计封装对应类

窗口设计类是平凡的，以棋子对象为例，在类内部实现了坐标向量、标记数组、道具数等内容，以及随机数函数、生成棋子和道具的函数声明。



其中，rd(l, r) 函数负责生成值域在 [l, r] 范围内的整数随机数，regeneratepos(i) 函数和 regenerateStopItem(i) 则是分别用于重新生成一个棋子和道具。

使用的时候，声明一个 ChessPiece Chess 对象的负责控制全局的棋子和道具的生成工作。

## 收集图像素材

对于背景、人物头像等素材，自行从网络上搜取加工：



## 处理窗口绘制问题

通过写入窗口类的 QPaintEvent 函数并定时调用 update() 函数来保证画面的实时刷新。但是这样绘制出的窗口会遇到屏幕闪烁问题，严重影响游戏体验。

实现上，通过双缓冲机制解决了这个问题，大大优化了动画的流畅度。

## 实现棋子、道具生成逻辑

对于黑白棋子，按照双方对应颜色棋子各 5 个在 19 路棋盘上随机生成。通过标记数组以及边界条件判定保证生成的棋子不会重叠且不会生成在棋盘外。

对于道具生成，同样需要处理棋盘关系和位置关系，保证不交叉、不出现在盘外。位置完全随机。

## 人机模式实现

为了适应单人的游戏情景，需要实现一个简单的机器供玩家对抗。实现中选择了简单的贪心策略：每次找到当前位置距离最近的一个棋子，然后规划好路径走过去。为了避免被玩家预判机器玩家的行走路线，在方向选则上增加了随机处理，保证能达到终点的同时，路径的具体实现是随机的。

反应时间给机器玩家设定了固定的 200ms 。只需要在每次移动的时候，检测当前是否已经确认好路径，如果确认好了就继续走，否则重新考察全局棋子排布，确认下一个要到达的目标棋子即可。

从实际表现上，简单的策略对玩家已经初具难度。然而机器玩家还可以有更优秀的决策路径，比如一次性规划好五个棋子的最短路，并且将道具纳入决策函数的考虑范畴等。

## 实现结算逻辑

为了让玩家更好地了解整局游戏中的表现数据，需要选取若干对对局起到重要作用的因素加以统计量化并呈现。代码选取了剩余棋子数、变向反应时间、同向反应时间、道具利用个数、陷阱放置个数、陷阱成功个数这些变量来刻画游戏进程。

# 收获

深刻认识到了中开发过程中封装类的重要性，同时也加深了我对整个开发过程的认知和理解；对于动画绘制等方面有个更深入的认识，也对游戏设计方面有了一个基础的认知。