# 棋类对战系统大作业实验报告

https://cloud.tsinghua.edu.cn/f/31311cff257f4055a8e8/

git@github.com:xidao4/chess.git

## 设计思路

*1 开始游戏前的功能有：*

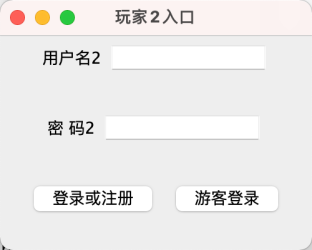
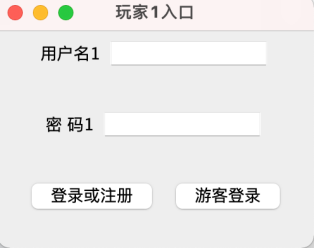
**1.1 选择游戏模式**



**1.2 玩家选择登录或注册，或以游客身份进入。**

其中登录与注册共用一个按钮

（1）玩家-玩家模式



（2）玩家-AI模式



1. AI-AI模式无需登录，直接进入棋盘

*2 三种棋类共同的功能有：*

**2.1 选择游戏类型**

**2.2 重新开始游戏**

**2.3 设定棋盘大小**

**2.4 投负：一方投负立刻结束游戏**

**2.5 回放模式：**

按“上一步”“下一步”可以来回观看每一步的局面

**2.6 显示玩家账号和战绩：**

每一局结束后更新战绩

**2.7 局面保存：**

保存当前的游戏类型等游戏局面，包括回放模式中保存的每一步。只有登录的玩家才能保存的局面

**2.8 局面读取：**

恢复最新保存的局面，包括局面的每一步。登录的玩家读取自己最新保存的局面



*3 围棋的其他功能有：*

**3.1 落子位置不合法：**

落子后，看是否一片子都无气了，如果是就不能下在这里

**3.2 提子**

**3.3 虚着**

*4 五子棋*

**4.1判断胜负**

*5 黑白棋*

**5.1 虚着**

一方无法下棋后，被迫虚着

**5.2 判断胜负**

*6 五子棋AI*

**6.1 一级AI：**

随机落子

**6.2 二级AI：**

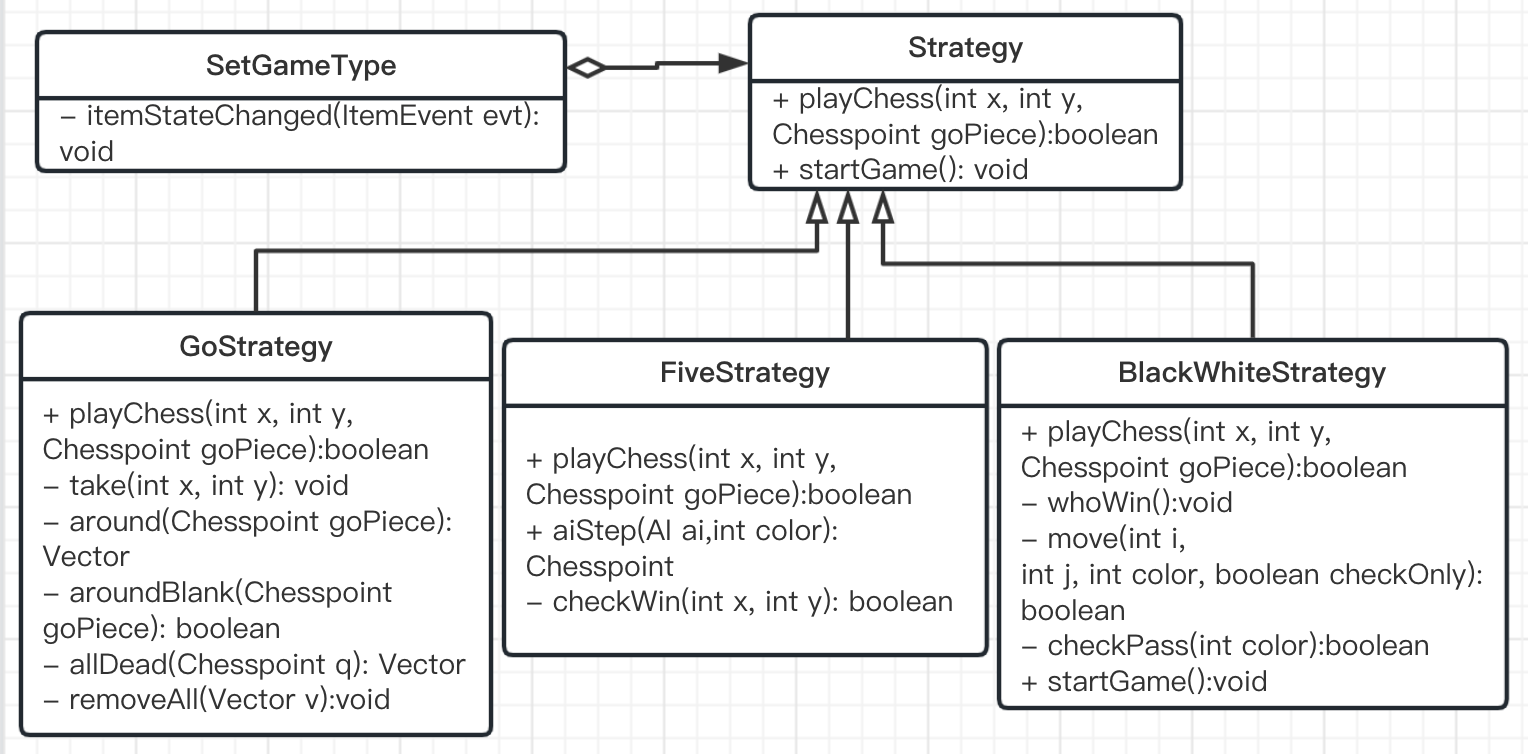
设计“连五”“活四”“死三”等各种连子情况及对应的评分，结合棋局选择得分最高的位置落子。

**设计模式、设计原则与UML图**

1. **简单工厂模式和策略模式。**

选择三种不同游戏类型后，下棋的规则不同，而且界面初始化不同（黑白棋初始化四颗棋子），所以可以采用策略模式，且与简单工厂模式相结合。

Strategy接口中的playChess方法用于实现不同的落子和胜负规则，startGame方法用于初始化参数和界面。



GoStrategy为围棋策略，take方法为进行提子操作，需要通道aroundBlank, allDead, removeAll等方法。playChess为父类的方法，实现鼠标点击落子后围棋的操作，包括判断是否有气、是否提子等。

FiveStrategy为五子棋策略，playChess方法中与围棋不同的是要进行胜负判断。

BlackWhiteStrategy为黑白棋策略，move和checkPass方法实现了当前落子夹棋判断及翻转操作。同时重写父类的startGame，初始化两白两黑四颗棋子。

1. **备忘录模式。**

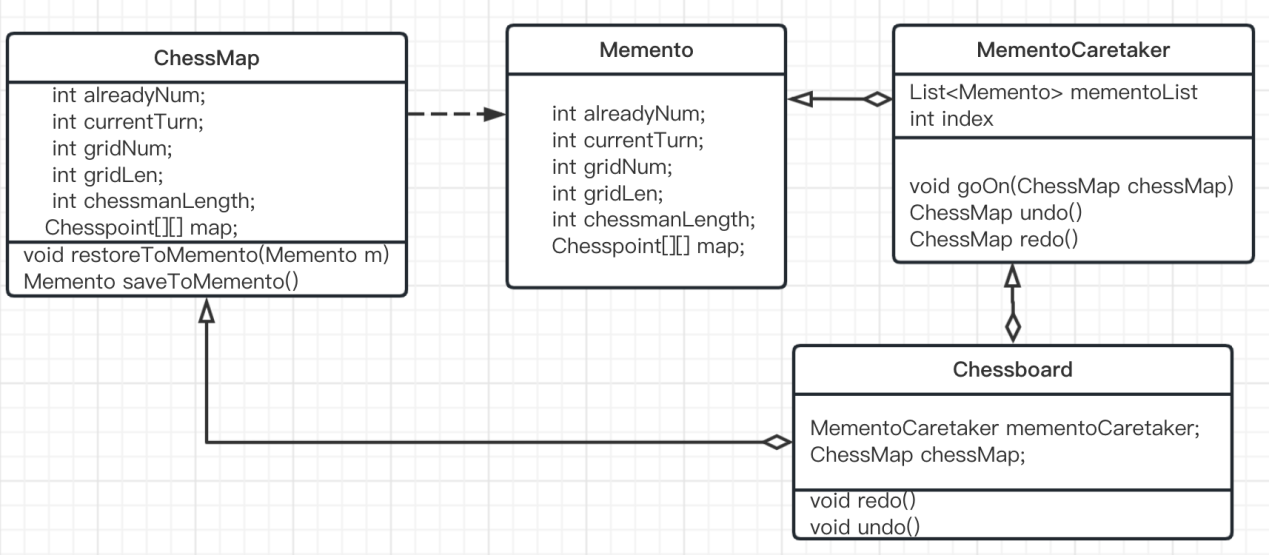
回放模式就是将造成当前局面的每一步过程都记录下来，可以通过“上一步”“下一步”按钮来回观看。同时“保存局面”和“恢复局面”时，不仅针对当前步，也需要记录过去所有步。

Originator原发器：棋局的主要参数都封装在**ChessMap类**中，是要保存状态的类。有restoreToMemento和saveToMemento方法。封装的内容有棋子在棋盘上的位置Chesspoint[][] map，当前轮到哪方下currentTurn，下了多少手alreadyNum等等。

**Memento**备忘录：包含ChessMap类的所有属性。

Caretaker负责人：**MementoCaretaker类**可以存储一个或多个备忘录对象，它只负责存储对象，而不能对备忘录内容进行操作或检查，也无须知道其实现细节。有undo上一步、redo下一步、goOn落子三个方法。

Client客户端：**Chessboard**持有MementoCaretaker和ChessMap引用，可以进行undo和redo操作。



3、**设计原则**

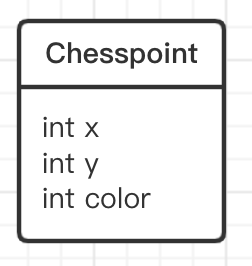
里氏代换原则：所有引用其（基类）父类的地方都能够透明的引用其子类。

依赖倒置原则：抽象不应该不依赖于细节，细节应该依赖于抽象。针对接口编程，而不是针对实现编程。

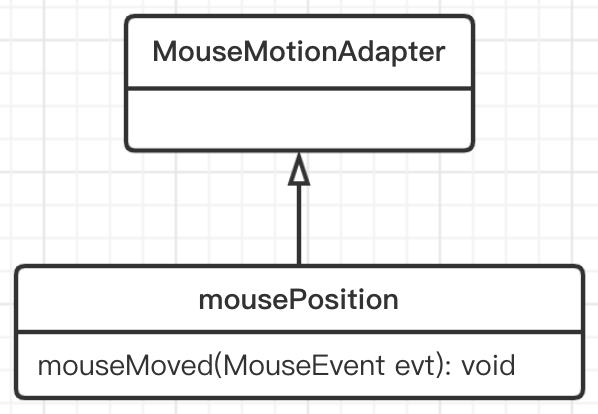
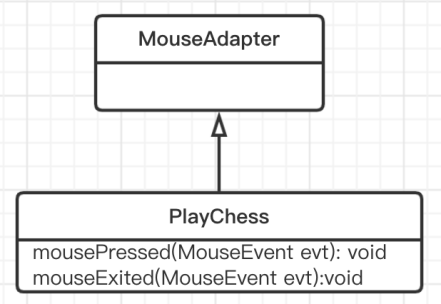
合成复用原则：尽量使用对象组合，而不是继承来达到复用的目的。

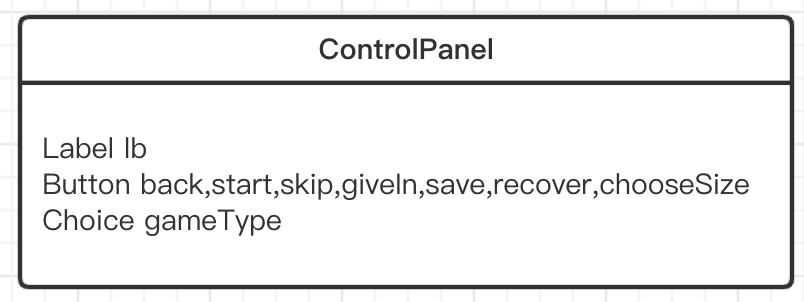
**关键类和函数功能说明**

1. Chesspoint表示棋子类，有坐标和颜色三个参数。



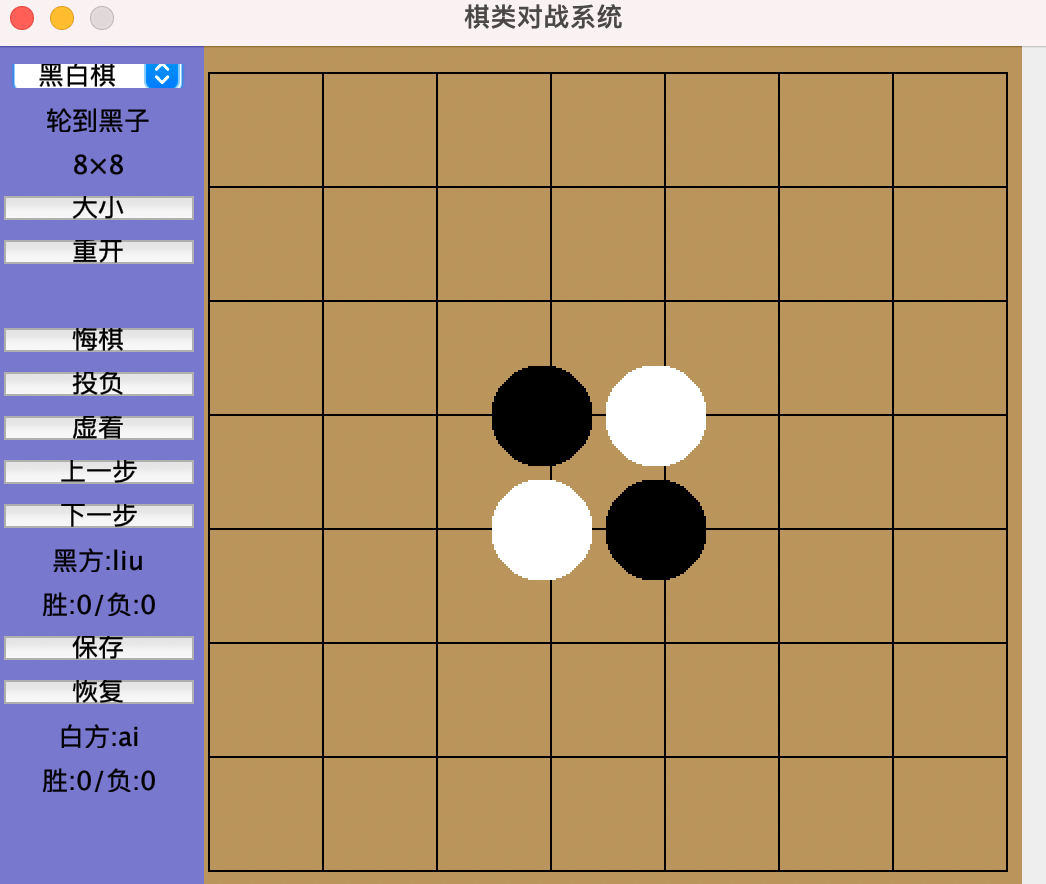
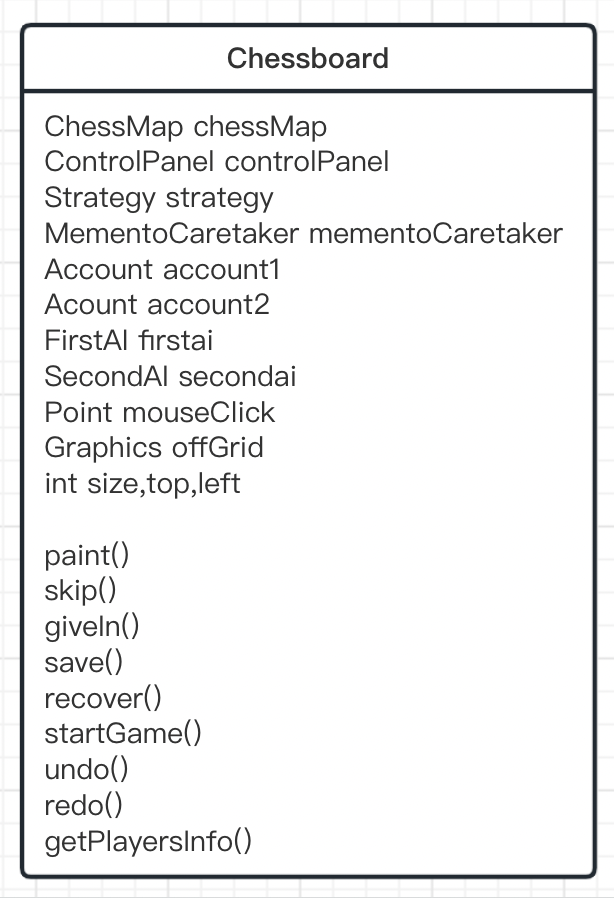
2、PlayChess类继承MouseAdapter，监听鼠标点击事件。mousePressed方法中调用strategy.playChess方法，不同棋类使用不同策略。  
MousePosition类继承MouseMotionAdapter，监听鼠标移动事件，进行落子位置的提示。



3、ControlPanel是控制栏类，包括标签、多个按钮和选择棋类型的选择标签。

4、Chessboard类存储当前棋局的基本信息，包含ChessMap、其他一些绘制棋盘的参数、

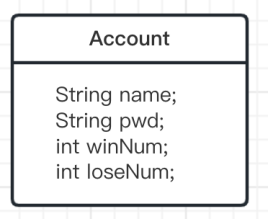
当前玩家信息、五子棋AI、备忘录负责人。负责绘制棋盘paint()。投负、保存棋局、恢复棋局、获取玩家信息等不同棋类的共同功能，也在其中实现。



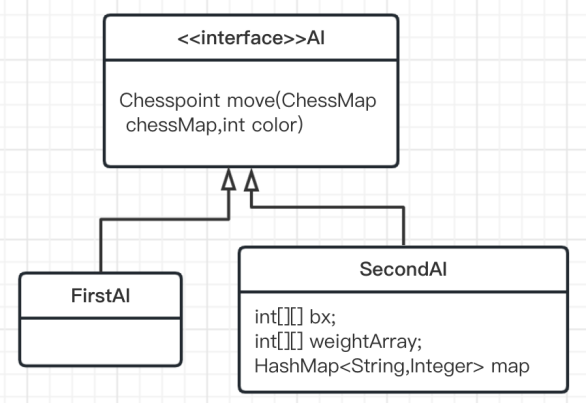
1. Lfr类是游戏入口界面



6、Account

包括玩家的昵称、密码、胜场、负场。

7、AI接口及两级AI



二级AI使用map来存储设定的“活一连”“活二连”“活三连”“活四连”“死一连”等八种模式和评分。使用棋盘大小的weightArray来记录每一个坐标位置的权重。每次AI进行落子前，循环遍历整个存储棋子的数组，对每个位置横向向左、横向向右、竖向向上、竖向向下、左斜向上、左斜向下、右斜向上、右斜向下八个方向分别根据棋子的相连情况，获取HashMap中存储的权值，然后加到权值数组中。最后选择权值最大的坐标作为落子位置。



**测试输入和输出**

