大作业第一阶段设计文档

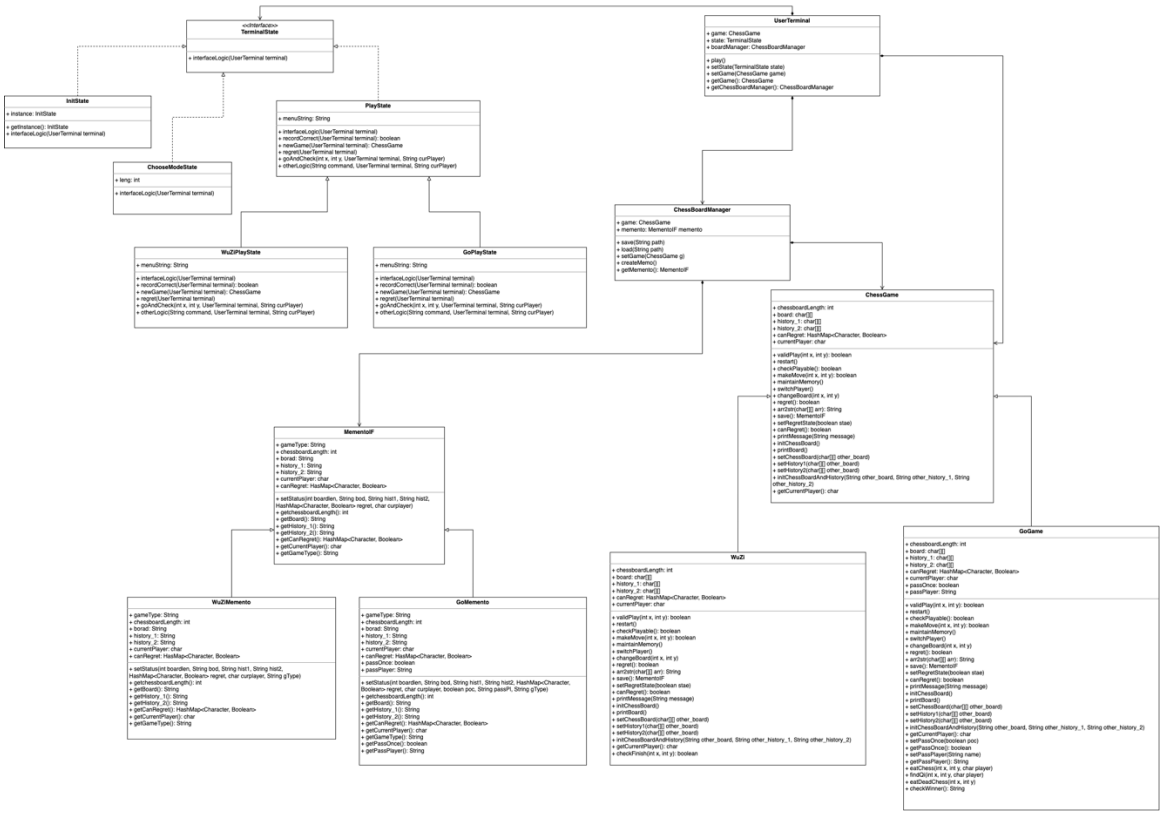
刘星翰 2023214259

设计思路：

在本阶段中，我用到的设计模式、设计思路，以及一些对游戏规则的实现介绍如下：

1. 客户端类为 UserTerminal，负责前端与用户的交互逻辑，考虑到交互逻辑中的状态转移比较复杂，不适合直接用条件判断语句处理，故我采用了状态模式（State Pattern）来设计这一部分。具体来说，以 TerminalState 接口为代表，通过继承和实现方式衍生出多个子类，用于处理不同的逻辑：包括初始化棋盘（选择尺寸、游戏类型）、进行对局中的各项操作（重新开始、落子、悔棋、提子、认输、胜负判断、虚着）等。
2. 由于 InitState 类的对象是全局唯一的，且不会产生变化，故我将其实现为单例（Singleton Pattern），避免重复创建和释放。
3. 考虑到存档、读档的需求，故设计并实现了备忘录模式（Memento Pattern），包括 MementoIF 抽象类以及衍生出的具体备忘录类，另外还实现了 ChessBoardManager 类作为备忘录模式中的 Caretaker 角色，负责以窄接口的形式管理数据的保存和读取。
4. 具体实现细节方面：五子棋白棋先手，围棋黑棋先手。对于悔棋，我规定了每个玩家每局只有一次机会。对于五子棋，没有特别实现禁手的规定。对于围棋，当两边各虚着一次后，认为双方同意进入判断胜负阶段。目前判断胜负的手段是比较双方在棋盘上的棋子数量，相同时判后手的白棋获胜。
5. 围棋的可下点判断算法方面，我实现了通过宽度优先遍历与落子点 4-连通的同色棋子块，并考察其是否有“气”，如果遍历结束时仍没有找到“气”，则判断为不可落子。对于提子的逻辑，则是在判断某棋子块没有“气”之后，再走一遍遍历顺序，并依次提去对应位置的棋子。

UML 图：（大作业 1.png）



代码仓库：

https://git.tsinghua.edu.cn/xh-liu23/oop_design_final

演示视频：

https://cloud.tsinghua.edu.cn/d/97153ff107e2449d9977/