



“高级语言程序设计” 课程设计项目报告

Jesse Senior

Xshellye

一、 项目目标

应用面向对象编程的相关知识，设计一个简单五子棋游戏。

具体要求：

1. 实现五子棋人人对战、人机对战；
2. 支持对战历史记录显示、复盘、删除，支持对战信息统计；
3. 充分应用面向对象思维，合理应用类、对象、封装、继承、多态等特性，其中：
 - (a) 类的数量应不少于 5；
 - (b) 继承层次结构的数量应不少于 2；
4. 界面设计精美，交互设计友好；
5. 支持随机文件处理（读、写、改）；
6. 源代码总行数不少于 2000。

二、 项目设计

2.1 前期分析

考虑到五子棋人机对战存在应用神经网络算法的可能，以及语言对于功能的封装程度、编译环境搭建难度、跨平台性等各种因素，由于 C++ 自身封装程度低，需要自己“重新发明轮子”（如 OpenGL），且使用第三方库过程复杂、开发过程中容易增加沟通阻碍（如 OpenCV），因此本项目并未选择使用 C++ 作为项目开发语言，而最终决定选择封装程度高、开发难度相对更小的 python 作为项目开发语言。

在此基础上，考虑到游戏相对于其他软件的特殊性，最终选择封装相对底层、自由度更高的 pygame 作为项目 GUI 渲染引擎，并按照项目实际需要对其重新进行封装。

由于项目涉及到对战历史记录的存储，考虑到数据自身结构性强，增删改查规范，最终选择轻量化本地数据库 SQLite 以支持高稳定性的随机文件读写。

2.2 总体设计

2.2.1 系统组成

根据项目目标，我们将项目划分为四个主要模块：

模块名	项目路径
核心逻辑模块	src/core.py
数据库模块	src/database.py
AI 模块	src/ai.py
GUI 模块	src/display/

核心逻辑模块： 对于五子棋棋局进行抽象并进行封装，以供 GUI 模块和数据库模块使用。

数据库模块： 负责实现数据库增删改查，并维护游戏本体与数据库间信息交换以及数据库自身合法性。

AI 模块： 游戏人机对战核心模块，实现 Min-Max 搜索以及 Alpha-Beta 剪枝。

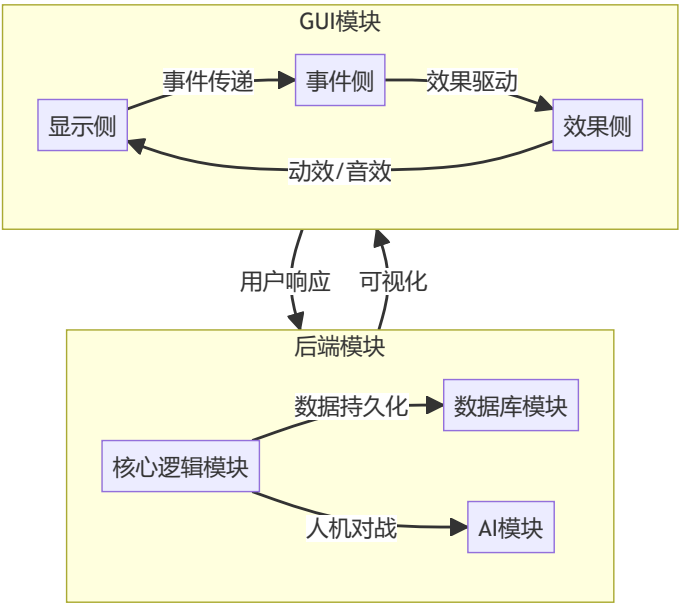
GUI 模块： 游戏图形界面模块，游戏引擎主体，包含交互设计。

2.2.2 GUI 设计

GUI 部分主要分为三条设计方向：**显示、事件、效果**。总体上参考了 Unreal 的组件式设计以及 Qt 的信号槽机制，同时引入效果机制，实现动画功能。

GUI 部分合理发挥面向对象设计的思想，对 pygame 的各个模块进行了深度定制以及合理抽象，增强其易用性的同时有效降低模块间耦合程度，遵循了依赖倒置原则以及开放封闭原则，具体设计细节将在详细设计部分进一步解释。

2.2.3 模块依赖关系



2.3 详细设计

2.3.1 核心逻辑模块

考虑到项目目标是实现一个简单五子棋游戏，因此对于五子棋棋局的模型抽象显得尤为重要。本项目设计class Board，其具有以下几个主要属性以及方法：

参数/方法名	含义
timestamp	执黑/白方玩家名
competitor_black/ompetitor_white	执黑/白方玩家名
board_size	棋盘尺寸
current_side	执子侧
winner	胜利侧
kifu	棋谱
place(column, row)	落子
cancel()	悔棋

核心逻辑模块有效实现了棋局状态的表示以及转移，并进行操作合法性检查，为项目其他模块的运行提供了基础。

2.3.2 数据库模块

如前文所述，考虑到五子棋对局数据自身结构性强，增删改查规范，使用 SQLite 进行数据存储其可用性要明显高于直接进行二进制文件读写。下表为数据库各列名称以及数据类型：

列名	数据类型
timestamp__	TEXT (PRIMARY KEY)
competitor_black	TEXT
competitor_white	TEXT
board_size	BLOB (NOT NULL)
kifu	BLOB (NOT NULL)
winner	INTEGER

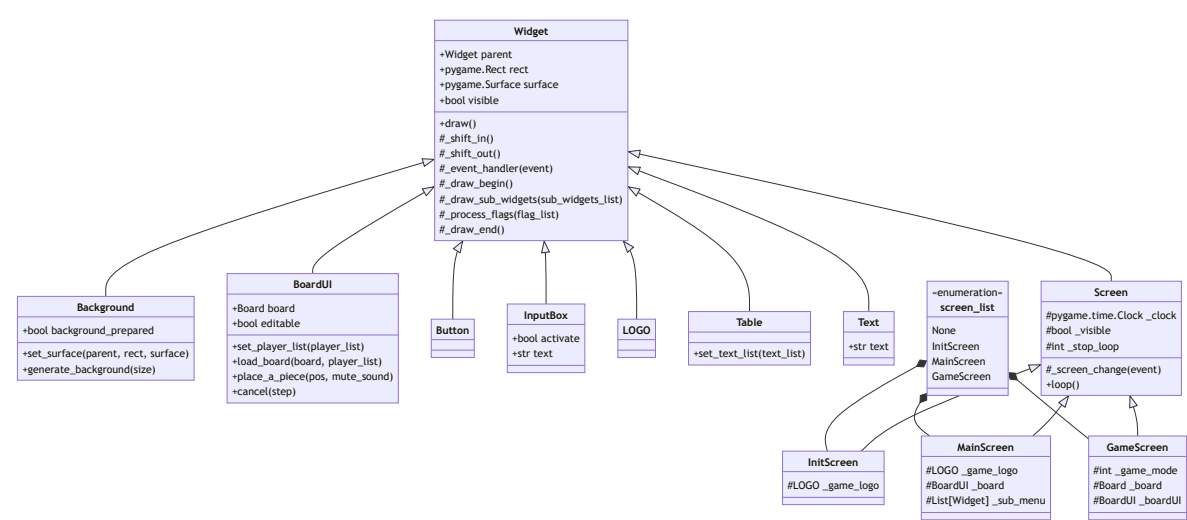
表 1: board_table

本项目设计class BoardDatabase，其具有以下几个方法：

方法名	含义
append(board_to_save)	增加棋局记录
export()	列出数据库中全部棋局记录
erase(board_timestamp)	删除指定棋局记录

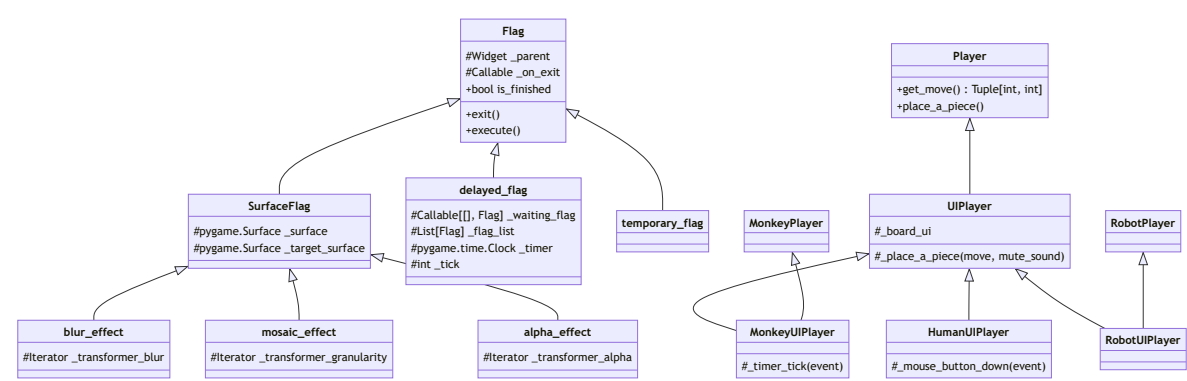
2.3.3 GUI 模块

关于显示方向，本项目设计class Widget以及继承子类，如下图所示：



关于事件方向，本项目设计_handlers[event]=function机制，在Screen类执行loop() 循环时自动提取游戏全部事件并向下层Widget传播，而各层Widget在接受事件并处理后会尝试向下层传播。

关于效果方向，本项目设计class Flag机制，用于每帧计算画面变化内容并作用到Widget上，其子类如下图所示：



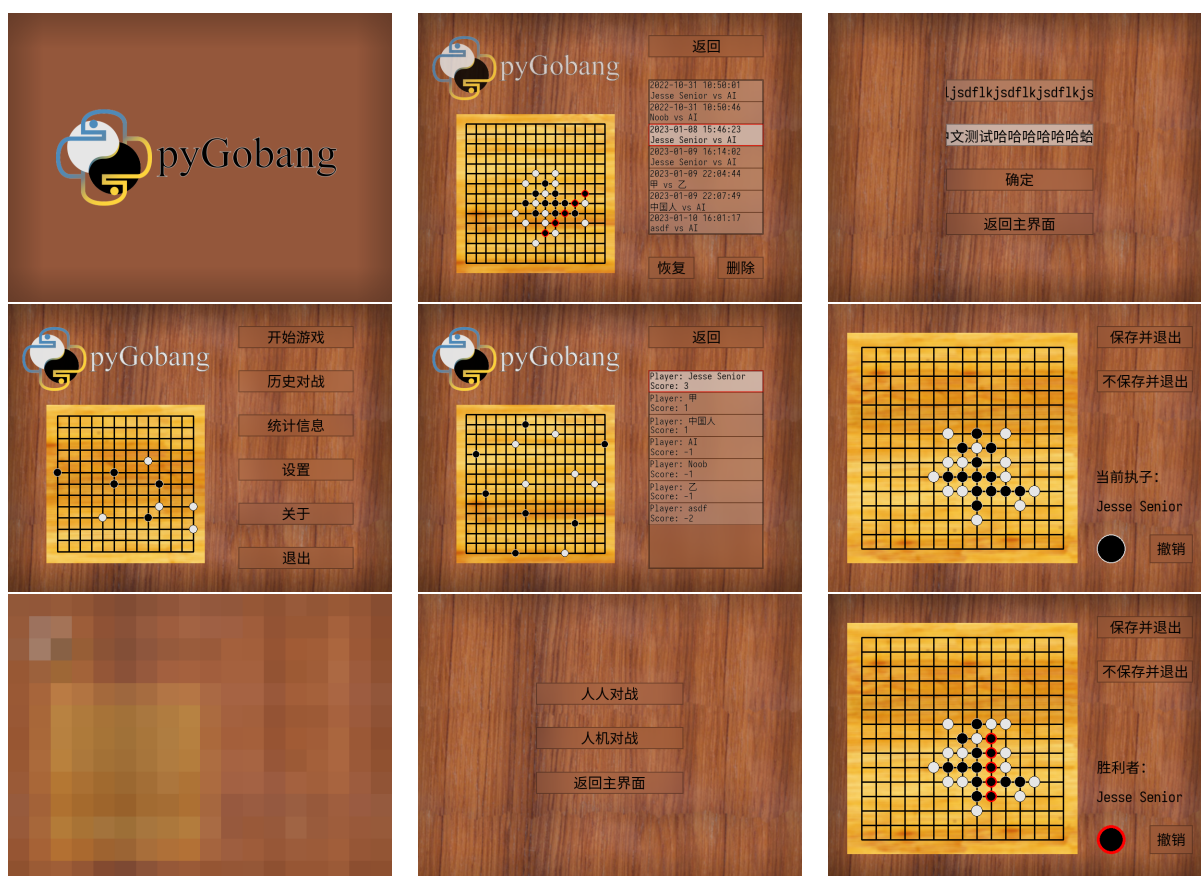
注： 由于 python 语言自身特性，项目设计类数量较多、结构复杂（如闭包、嵌套类等），部分细节并未在上述两张图中给出。

2.3.4 AI 模块

考虑到神经网络等智能算法对于计算机性能要求较高，而五子棋采用优化后的 Min-Max 搜索即可达到足够高的棋力，因此本项目最终决定参考第三方 Github 项目 [lihongxun945/gobang](#)，基于 Python 再次实现其功能。

三、 项目实施

以下为项目实际截图：



截止本文档完成时，本项目为 v0.2.0 版本。项目源代码以及预编译二进制文件可以从 [JesseSenior/pyGobang](#) 获得，同时可以从 Bilibili 观看本项目 v0.1.1 版本的演示视频。

四、 项目目标完成情况

- ✓ 实现五子棋人人对战、人机对战；
- ✓ 支持对战历史记录显示、复盘、删除，支持对战信息统计；
- ✓ 充分应用面向对象思维，合理应用类、对象、封装、继承、多态等特性；
- ✓ 类的数量应不少于 5；

- ☑ 继承层次结构的数量应不少于 2;
- ☑ 界面设计精美，交互设计友好;
- ☑ 支持随机文件处理（读、写、改）;
- ☑ 源代码总行数不少于 2000。

4.1 源代码总行数统计结果

LINES	FILE
28	.\pyGobang.py
239	.\src\ai.py
103	.\src\constants.py
223	.\src\core.py
117	.\src\database.py
50	.\src\main.py
77	.\src\players.py
0	.\src__init__.py
205	.\display\effect.py
175	.\display\texture.py
49	.\display\tool.py
0	.\display__init__.py
577	.\screen\game_screen.py
137	.\screen\init_screen.py
620	.\screen\main_screen.py
99	.\screen__init__.py
138	.\widget\background.py
448	.\widget\board.py
179	.\widget\button.py
351	.\widget\input_box.py
116	.\widget\logo.py
286	.\widget\table.py
90	.\widget\text.py
191	.\widget__init__.py
115	.\test\core_test.py
67	.\test\database_test.py
0	.\test__init__.py
Total Lines: 4680	