

# “可支持人类和 AI 博弈的国际象棋棋盘软件”软件开发课程项目——需求规格说明书

王继成 《软件开发方法》课程 第 6 小组

## 1 项目名称

可支持人类和 AI 博弈的国际象棋棋盘软件。

## 2 引言

### 2.1 目的

本文档旨在界定“可支持人类和 AI 博弈的国际象棋棋盘软件”的功能需求和非功能需求，作为本小组软件系统开发和验证的根本依据。

### 2.2 项目范围

“可支持人类和 AI 博弈的国际象棋棋盘软件”是为 2017-2018 学年度第二学期的《人工智能原理》课程（赵才荣老师）开发的象棋对战平台软件，目标为使得该课程的 AI 国际象棋比赛过程能够自动化进行，以及使得参赛选手调试 AI 程序便利化。

通过应用本软件，希望使得课程象棋比赛能够流畅、节约时间地进行，参赛选手的准备工作能够更为高效地进行。

### 2.3 定义、首字母缩写和缩略语

名词	解释说明
本软件	指本项目的预期开发成果之一，“可支持人类和 AI 博弈的国际象棋棋盘软件”，英文名为 The Chess Board 或 Board。
AI	国际象棋人工智能（Artificial Intelligence）程序的缩略语。 AI 是《人工智能原理》课程的各参赛小组独立设计编写的、能够有效思考并模拟人类作为国际象棋比赛任意一方进行走子博弈的人工智能程序。
比赛	指《人工智能原理》课程的“国际象棋人工智能程序比赛”。
参赛者	指参加比赛的各小组成员（人）。

参赛 AI	指参加比赛的各小组编写完成的 AI，将模拟人进行国际象棋博弈。
组织者	比赛的组织者，也是《人工智能原理》课程的三位课程负责人。
SAN	标准代数记谱法（Standard Algebraic Notation）的缩写。 是一种国际象棋走子的记谱法，采用数字、字母和符号构成的短字符串记录走子的每一步。其详细定义见参考资料 3)与附录 5.1。
PGN	便携式游戏记号（Portable Game Notation）的缩写。 一种在电子计算机上存储国际象棋棋谱的文件格式，其主要部分为 SAN 记谱法的字符串。后文将不对 PGN 和 SAN 作明显的区分。 其详细定义见参考资料 4)。
FEN	福斯夫-爱德华兹记谱法（Forsyth-Edwards Notation）的缩写。 是一种国际象棋局面的记谱法，采用数字、字母、符号构成的长字符串记录一个瞬间的棋盘局面。其详细定义省略，见参考资料 5)。

## 2.4 参考文献

- 1) IEEE 标准文档。
- 2) 《“可支持人类和 AI 博弈的国际象棋棋盘软件”软件开发课程项目——目标规模说明书》。
- 3) “代数记谱法”的 Wikipedia 词条页面：  
<https://zh.wikipedia.org/wiki/%E4%BB%A3%E6%95%B8%E8%A8%98%E8%AD%9C%E6%B3%95>。（可能需要翻墙，下略）
- 4) “Portable Game Notation”的 Wikipedia 词条页面：  
[https://en.wikipedia.org/wiki/Portable\\_Game\\_Notation](https://en.wikipedia.org/wiki/Portable_Game_Notation)。
- 5) “福斯夫-爱德华兹记谱法”的 Wikipedia 词条页面：  
<https://zh.wikipedia.org/wiki/%E7%A6%8F%E6%96%AF%E5%A4%AB%EF%BC%8D%E6%84%9B%E5%BE%B7%E8%8F%AF%E8%8C%B2%E8%A8%98%E8%99%9F%E6%B3%95>。

### 3 项目总体描述

---

#### 3.1 产品背景和前景

##### 3.1.1 背景

见“参考资料 1)——目标规模说明书”的“2.1 项目背景”部分。

概述为：本软件为满足比赛中自动化走棋和参赛者调试 AI 的需求而开发，意在提供可视化的象棋手动与自动对战平台。

##### 3.1.2 业务需求

- BR1：本软件投入使用后，参赛者调试和测试 AI 的时间减少 20%。
- BR2：本软件投入使用后，组织者举办比赛所用时间比原来预期的人工走子方式所用时间减少 50%。

#### 3.2 产品功能

- SF1：图形界面显示控制选项和棋子。
- SF2：加载 AI 作为一个独立进程运行。
- SF3：以不同的游戏模式进行游戏（自动/手动）。
- SF4：与 AI 进程通信，实现自动博弈。
- SF5：与用户通过点击方式交互，实现手动博弈。
- SF6：充当裁判，判断走子是否符合规则。
- SF7：判断游戏结束并给出提示。
- SF8：对走子计时。
- SF9：保存比赛的棋谱。
- SF10：处理和保存用户的设置。

#### 3.3 用户特征

用户	特征
参赛者	比赛的参赛者，一般 4-5 人组成一组。他们大多为计算机系大二或大三学生，有一定的计算机技术水平和操作水平。他们是自己小组的 AI 设计者和编写者，了解 AI 进程的运行方式。即使开发了一个国际象棋 AI，他们大多对于国际象棋的下法、规则和技巧知之甚少。他们苦于手动下子

	的 AI 测试和 AI 参赛方式，愿意适配通信协议以在本软件上实现自动博弈
组织者	比赛的组织者，由 3 人。是计算机系大二或大三学生，拥有较好的计算机技术水平和操作水平，对于国际象棋的规则、比赛的规则基本了解。他们作为比赛的主办方，需要主持开始游戏、宣布游戏是否结束以及游戏的胜负；必要时，还需要保存棋谱。

### 3.4 约束

- CON1：本软件将运行在 x86-64 架构的 CPU 上。
- CON2：本软件将运行在 Windows 7 及以上的 Windows 操作系统上。
- CON3：本软件将采用基于 WinForm 的图形界面。
- CON4：本软件的开发小组需要提交软件需求规格说明书、设计报告说明书和项目小结。

### 3.5 假设和依赖

- AE1：比赛的用户均在使用安装了 .NET Framework 4.6 的 Windows 7 及以上操作系统。
- AE2：参赛 AI 均可在 Windows 7 及以上操作系统运行。
- AE3：参赛 AI 均依照 SAN 字符串的方式（通信协议），处理标准输入输出（Standard Input/output）。

## 4 详细需求描述

---

### 4.1 对外接口需求

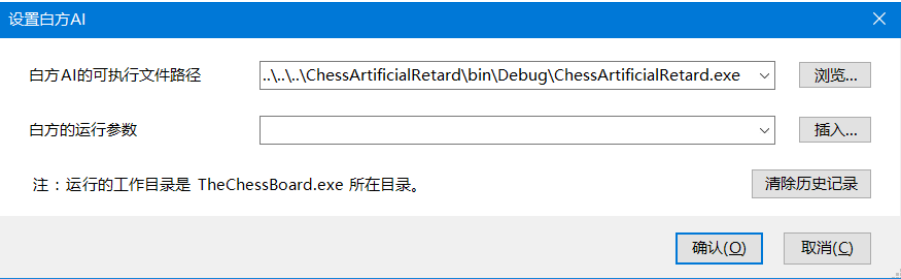
#### 4.1.1 用户界面

- UI1：唯一的一个象棋棋盘和控制窗体主界面。该界面的风格为 WinForm（Windows 风格窗体），方便熟悉 Windows 界面语言的人快速上手和操作。

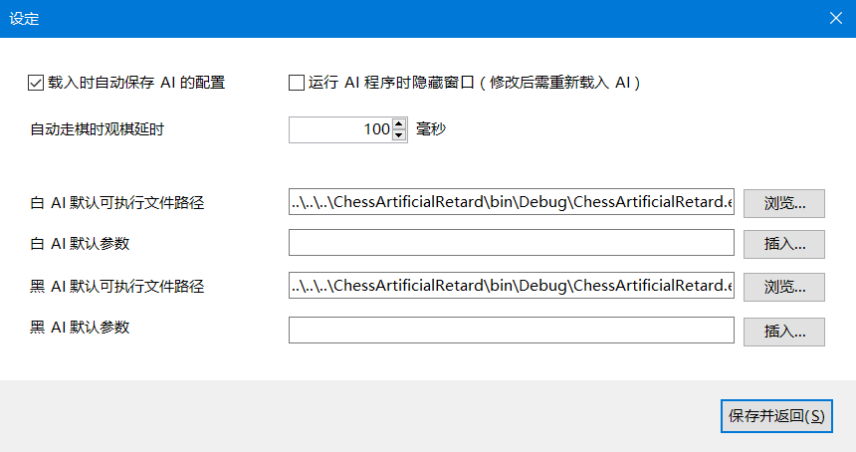
该界面的图示为：



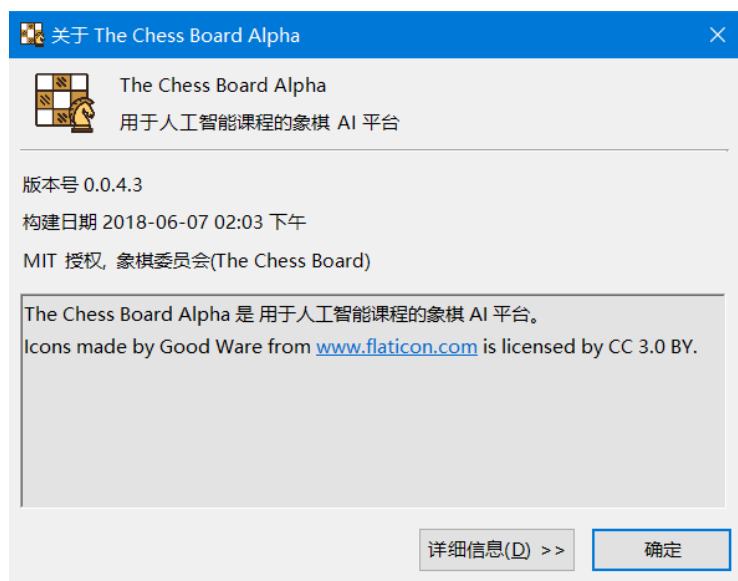
- UI1.1：在载入 AI 时，应弹出填入载入 AI 信息的对话框，如图



- UI1.2：在设置选项时，应弹出含有选项的对话框，如图



- UI1.3：当点击“...”下的“关于”菜单项时，应弹出“关于”开发小组的信息的对话框，如图



#### 4.1.2 通信接口

- CI：本软件与参赛 AI 交互使用的是操作系统的“标准输入输出”；内容协议采用 SAN 格式。

## 4.2 功能需求

### 4.2.1 图形化显示棋盘和棋子

#### 4.2.1.1 特性描述

主界面显示一个 8x8 的正方形国际象棋棋盘。棋子以图形的方式显示在国际象棋棋盘中。参赛者或组织者开始游戏或重置游戏时，应达到初始化棋盘显示的功能；游戏开始后，棋盘显示游戏的棋局，并随着双方走子不断更新显示。如 UI1 所示。

### 4.2.2 加载 AI

#### 4.2.2.1 特性描述

参与比赛的两方参赛者，均可以在比赛开始前在本软件中载入自己的 AI 作为进程运行，准备参与博弈。载入内容包括“是白/黑方”、“可执行文件路径”、“执行参数”。

#### 4.2.2.2 刺激/响应序列

- 刺激：用户点击“载入白方 AI”或载入“黑方 AI”。
  - 响应：系统弹出对话框，用户界面如 UI1.1。
- 刺激：用户在可执行文件文本框中输入可执行文件路径。
  - 响应：文本框成功接受输入。
- 刺激：用户在对话框中点击“浏览”。
  - 响应：系统弹出二级对话框，允许用户浏览文件选择可执行文件。
    - ◆ 刺激：用户选择完毕，点击“确认”。

- 响应：系统关闭对话框，同时填入“可执行文件”文本框的内容。
- ◆ 刺激：用户取消选择文件。
  - 响应：系统关闭对话框。
- 刺激：用户点击“确认”。
  - 响应：系统关闭对话框，尝试验证载入刚刚输入的“可执行文件”文本框的内容。
  - 响应：系统返回验证结果，提示用户“载入成功”或“载入失败”。
- 刺激：用户点击“取消”。
  - 响应：系统关闭对话框，不做任何改动。

#### 4.2.2.3 相关功能需求

LoadAI.onClick	本软件允许用户点击按钮载入 AI
LoadAI.onClick.GameStarted	在游戏正在进行中时，本软件不允许用户点击按钮载入 AI
LoadAI.onClick.InformationPopup	用户点击载入 AI 时，弹出对话框。参见 LoadAI.InformationPopup
LoadAI.InformationPopup	应用户点击弹出的对话框
LoadAI.InformationPopup.Input	允许用户输入可执行文件路径
LoadAI.InformationPopup.Browse	允许用户点击“浏览”，从二级对话框中选择可执行文件
LoadAI.InformationPopup.Confirm	允许用户点击“确定”，确定输入结果。参见 LoadAI.Validate
LoadAI.InformationPopup.Cancel	允许用户点击“取消”，取消输入结果
LoadAI.Validate	用户确认输入的可执行文件后，对其进行检查
LoadAI.Validate.SuccessHint	当检查正确后，提醒用户载入 AI 成功
LoadAI.Validate.FailPopup	当检查错误时，弹框提醒用户载入 AI 失败。

### 4.2.3 以不同的模式进行游戏

#### 4.2.3.1 特性描述

进行游戏可以有以下几种模式：

<div>白方 \ 黑方</div>		人类	AI	
		手动	自动	手动
人类	手动	(人类手动,人类手动)	(人类手动,AI 自动)	(人类手动,AI 手动)
AI	自动	(AI 自动,人类手动)	(AI 自动,AI 自动)	(AI 自动,AI 手动)
	手动	(AI 手动,人类手动)	(AI 手动,AI 自动)	(AI 手动,AI 手动)

其中，“人类手动”指人类利用鼠标点击棋盘上的棋子、选择其移动位置来完成走子，“AI 自动”指本软件不经用户确认直接从 AI 读入信息完成走子，“AI 手动”指本软件需要用户手动点击“从 AI 读入”按钮才从 AI 读入信息。

游戏组织者在开始游戏前，需要可以从本软件中选择其中一种模式。

#### 4.2.3.2 刺激/响应序列

- 刺激：用户点击界面 UI 上的“模式选择”单选按钮。
  - 响应：本软件接受用户的点选输入，并相应地更新 UI。
- 刺激：用户点击“开始游戏”。
  - 响应：本软件进入游戏进行状态，根据选好的模式决定是否向 AI 发送其所在方（黑/白）信息，并开始从 AI 读入走子信息走子。

#### 4.2.3.3 相关功能需求

GameRun.SelectMode.onClick GameRun.SelectMode.Record GameRun.SelectMode.UpdateUI	本软件允许用户点击单选按钮选择模式 用户选择模式后，本软件记住用户选择的模式 用户选择模式后，本软件更新 UI，显示用户选择的模式
GameRun.Start GameRun.Start.SendAICamp GameRun.Start.ReadFromAI GameRun.Start.WaitForUserInput	游戏以用户选好的模式开始 向 AI 发送阵营信息 读取 AI 走子输入 处理用户交互输入

### 4.2.4 与 AI 进程通信、处理用户交互，实现自动、手动博弈；判断走子是否符合规则

#### 4.2.4.1 特性描述

作为棋盘平台，本软件在象棋游戏进行过程中，要根据模式选择的不同，与 AI 进程通信以获得它们的走子信息，以及处理用户交互以获得人工走子信息，实现无组织者人工干预的自动、手动博弈。同时，本软件还应充当裁判的作用，预防不合法的走子产生。

#### 4.2.4.2 刺激/响应序列

- 刺激：用户点击棋盘。
  - 响应：UI 以颜色的方式提醒用户点击是否有效合法；若有效合法，推进走子的流程。
- 刺激：用户成功走子。
  - 响应：更新 UI 为走子后的局面；判断游戏是否结束并给出提醒；若对方为 AI，向其发送走子信息。
- 刺激：AI 发来走子信息。
  - 响应：判断走子信息是否有效合法；若有效合法，更新 UI 为走子后的局面；判断游戏是否结束并给出提醒；且若对方为 AI 则向其发送走子信息。若不合法，则回送特殊信息说明走子错误。
- 刺激：用户点击“停止”。
  - 响应：等待所有附加线程运行完毕，然后停止游戏，更新 UI，给出提示“游戏已停止”。
- 刺激：用户点击“重置”。
  - 响应：等待所有附加线程运行完毕，然后重置游戏，更新 UI。

#### 4.2.4.3 相关功能需求

GameRun.Start GameRun.Start.SendAICamp	游戏以用户选好的模式开始 向 AI 发送阵营信息
---	-----------------------------



GameRun.Start.ReadFromAI	读取 AI 走子输入
GameRun.Start.WaitForUserInput	处理用户交互输入
GameRun.Start.ValidateUserInput	验证用户走子是否正确，并显示，参见 UpdateUI
GameRun.Start.ValidateAIInput	验证 AI 走子是否正确，并显示，参见 UpdateUI
GameRun.Start.UpdateUI	更新 UI 以显示更新后的局面
GameRun.Start.Resend	向 AI 回送信息说明其走子不合法
GameRun.Start.GameEnds	判断游戏是否结束并给出显示提醒
GameRun.Start.SendToOpponentAI	向对方 AI 发送刚刚的走子信息
GameRun.Start.Stop	用户点击“停止”，停止游戏
GameRun.Start.Reset	用户点击“重置”，重置游戏

## 4.2.5 给游戏计时

### 4.2.5.1 特性描述

由于需要控制 AI 走子的时间，故组织者需要对 AI 的走子进行计时。计时的方法是，轮到该 AI 走子时，AI 在走子前等待的时间累计起来，即是该 AI 所用的时间。

### 4.2.5.2 刺激/响应序列

- 刺激：由于上一步的用户手动走子或 AI 的自动走子，将主动权让给了我方 AI。
  - 响应：本软件开始对我方 AI 的走子计时。
- 刺激：我方 AI 思考后走子。
  - 响应：本软件停止对我方 AI 的走子计时，将计时的这段时间累加到我方 AI 所用时间上。
- 刺激：用户点击“重置”。
  - 响应：两方 AI 的计时均归零。

### 4.2.5.3 相关功能需求

GameRun.StopWatch.Start	计时器开始计时
GameRun.StopWatch.Stop	计时器停止计时
GameRun.StopWatch.Accumulate	计时器将刚计的时段累加到本方 AI 的时间上
GameRun.StopWatch.Reset	归零计时

## 4.2.6 保存比赛棋谱

### 4.2.6.1 特性描述

一场游戏结束后，组织者可能需要保存其棋谱。本软件允许组织者将比赛棋谱复制到剪贴板，以便粘贴到别处保存。

### 4.2.6.2 刺激/响应序列

- 刺激：组织者选择历史记录里所有走子。
  - 响应：本软件的 UI 做出相应变化，表示组织者成功选定了这些走子。
- 刺激：组织者按下“Ctrl-C”键。（“复制”操作快捷键）
  - 相应：本软件将已经选定的走子送至操作系统的剪贴板上。

#### 4.2.6.3 相关功能需求

GameRun.HistoryMove.Select	选择历史走子
GameRun.HistoryMove.Copy	将选择的历史走子复制到剪贴板

### 4.2.7 处理和保存用户的设置

#### 4.2.7.1 特性描述

本软件应处理和保存用户的一些参数偏好，包括默认可执行文件路径、是否在运行时隐藏 AI 窗口、观棋时间、是否自动保存 AI 配置。

## 4.3 性能需求

- Performance1：速度：每一步走棋所产生的相应变化需在 1 秒内完成，点击按钮后所产生的变化应在 3 秒内完成。
- Performance2：负载：能够接受两个 AI 同时运行。
- Performance3：适应性：在不同 Windows 版本上能够运行。

## 4.4 数据需求

### 4.4.1 数据定义和格式要求

本软件需要在计算机上存取的数据只有用户设置。用户设置包含如下定义和格式的数据：

数据标识符	数据名称	数据定义	数据格式	默认值
WatchChessTime	观棋时间	在自动模式下，AI 走棋后，在下一步读取走子出发之前，留给用户观看走棋的时间	Unsigned int	100
WhiteDefaultExecPath	白 AI 默认位置	白 AI 的默认位置（路径），在载入 AI 时作为一个默认值	String	..\..\..\ChessArtificialRetard\bin\Debug\ChessArtificialRetard.exe
BlackDefaultExecPath	黑 AI 默认位置	黑 AI 的默认位置（路径），在载入 AI 时作为一个默认值	String	..\..\..\ChessArtificialRetard\bin\Debug\ChessArtificialRetard.exe
HideAIWindow	隐藏 AI	是否在游戏进行时隐藏 AI 进程的窗口	Boolean	False

## 4.5 安全性需求

- Safety1：本软件的运行应对操作系统的完整性无害。
- Safety2：本软件的运行应不违反操作系统规则，不导致操作系统陷入崩溃。
- Safety3：本软件应不对用户的其他文件造成危害，包括修改和删除。

- Safety4：本软件应对每个 Windows 用户的用户设置数据设置屏蔽，防止互通。

## 4.6 可靠性需求

- Reliability1：本软件不应在用户对其正常使用时突然退出、崩溃。
- Reliability2：本软件不应在 AI 程序出现错误、故障、突然退出时发生故障。
  - Reliability2.1：本软件应该检测到 AI 出现异常，并弹窗报告用户，并停止正在进行的游戏。
- Reliability3：本软件应当合理管理分配的内存，防止出现内存泄漏。
- Reliability4：本软件应当线程安全，防止多个线程同时操作一个对象产生的不协调和错误。

## 5 附录及 UML 图

---

### 5.1 《〈人工智能原理〉AI 国际象棋比赛规则书》

该规则书与本说明书配套，但作为公开给参赛者作比赛规则说明用。内部详细解释了 SAN 的诸多规定。

内容：

<p>2018 年《人工智能原理》AI 国际象棋比赛规则</p> <p>一、 下棋规则</p> <p>1. 基本规则参见： <a href="https://zh.wikipedia.org/wiki/%E5%9C%8B%E9%9A%9B%E8%B1%A1%E6%A3%8B">https://zh.wikipedia.org/wiki/%E5%9C%8B%E9%9A%9B%E8%B1%A1%E6%A3%8B</a>。</p> <p>2. 关于主动求和：不允许 AI 主动求和。所以实现的 AI 中不需要考虑求和与接受求和。</p> <p>3. 特殊走法：允许王车易位、吃过路卒、兵的晋升。</p> <p>二、 下棋方式</p> <p>1. 每次下棋时，象棋比赛委员会给出一个 Board 程序（TheChessBoard.exe），比赛双方各给出一个 AI Agent 象棋选手程序。它们通过标准输入输出通信。</p> <p>2. 关于 AI Agent 选手程序的阐述：</p> <p>可以为.exe 文件；也可以为语言脚本等，只要可以通过传递参数方式运行。请在程序制作完毕后向委员会提交文件以及运行使用的命令。如：</p> <p>提交：chess_script.py</p> <p>需求：安装 CPython for Windows 3.6.5；python-chess PIP 包</p> <p>运行命令：python -u ./chess_script.py （-u 参数是为了强制无缓冲输出）</p>
---

3. AI 程序、所有的 Board 程序必须可以在（暂定）断开网络的演示计算机系统 Windows 10 1703（内部版本 15063.966）下正常运行。不限编写语言。若需要安装的额外环境/解释器/运行时库/框架/包，请先向委员会咨询是否可行。一般来说，常见的、下载、安装方便的库（如.NET Framework 4，Windows 已经内置/python 的 pip 包）可以被接受。

目前已被允许使用：

Python 2/3 的 PIP 包 python-chess, numpy

4. 标准输入输出的阐释：

在 Windows 下用 C++ 编译器编译的代码中，

```
cout << "Standard output";
```

编译后将产生标准输出“Standard output”；

```
cin >> a;
```

编译后将读取标准输入存储到变量 a 中。

其他语言进行标准输入输出的方法，请自行查阅。

5. AI 的程序应该如下设置：

1) 每局比赛开始时，AI 默认棋盘上的初始位置放置自己的十六个棋子。没有让子。

2) 两个 AI 均接受 Board 的一个输入，知道自己是执先手（白棋）或是后手（黑棋）。

3) 执先手的 AI 立刻给出一个输出表示自己的走法，之后等待接受标准输入。走法的输入输出依照修正后的标准代数记谱法（SAN，见下文）。执后手的 AI 直接等待接收标准输入。

4) 当先走者给出了标准输出，Board 进行判断。若是不符合游戏规则的走法，将给先走 AI 发送信号“重走”。此时，AI 必须给出一个与之前不同的走法（若再无其他走法，也可重复之前的走法，之后被三次判负）；若该走法仍然不符合规则，三次后将被判负，游戏直接结束。

5) Board 判断该走法符合规则，则将该走法发送给对方 AI。此时对方按照 3) 的方法走子，之后的走子重复 3) 到 5) 的步骤。注意，每个 AI 必须有效地分析标准输入的记谱字符串（见下文），并在程序内部维护一个棋盘的局面。

6) 若某方落子后，Board 判断出现“将死”，则该方胜。

7) 若某方落子后，Board 判断对方出现“欠行”（Stalemate，无法行动），则判和局。

8) 若某方落子后，Board 判断在连续的 50 回合内均没有吃子和动兵，则判和局。

9) Board 不会判断“threefold”和“死局”的情况。由人眼判断，再做决定。

10) 具体的输入/输出格式，再行决定。

6. 关于 SAN：

a) SAN 全称 Standard Algebraic Notation（“标准代数记谱法”），是 Portable Game Notation 采取的棋步记录方式。SAN 的原版规定可在下列网页查看：

<http://cfajohnson.com/chess/SAN/>

<http://www.saremba.de/chessgml/standards/pgn/pgn-complete.htm#c8.2.3>

（Movetext SAN 部分）

其样式与 AN 几乎无异。在本规则内对 SAN 和 AI 的输入/输出方式做一些规定和修正如下：

对 AI 输出的要求：

不得包含任何注释、终局符号（\*等）；

移动兵时不得使用棋子符号 P，必须省略棋子符号；

x、+、#可以省略，也可以输出；

不得有 e.p. 和++出现；

王车易位用 O-O 和 O-O-O（英文字母大写 O）表示；

不得有多余的空格、符号-出现；

行尾必须有且仅有一个'\n'换行符；

兵的晋升用“=棋子”，如“=Q”晋升为王后。

对 AI 接受输入的要求：

必须可以处理符号 x、+、#；

必须可以处理以大写 O 形式表示的王车易位；

正确处理省略原位置坐标的三种情形（全部省略、只省略行、只省略列）；

兵的晋升用“=棋子”，如“=Q”晋升为王后；

行尾必须有且仅有一个'\n'换行符。

以 The Chess Board 程序能处理为最终标准。

7. 会对 AI 的走子做出限时。Board 会对 AI 的走子计时，当一方用完 10 分钟后（暂定，可能视具体各组的运行时间进行调整），将被直接判负。

### 三、 赛制

本次比赛的赛制是瑞士制（积分编排制）。

#### 1. 瑞士制（Swiss-system）

比赛开始首先随机编排进行一轮积分赛。之后轮次的积分赛根据总积分进行编排（在我们的比赛中，用软件进行编排），选择积分相近的队伍作为对手。若在某一轮积分赛中产生平局，则平局的双方即使积分相近也不会下一轮比赛中作为对手。若干轮积分赛后根据积分决定名次。

瑞士制的详细资料：

<https://baike.baidu.com/item/%E7%91%9E%E5%A3%AB%E5%88%B6/6521755?fr=aladdin#2>

#### 2. 计分方法

在棋局中，胜利队伍获得 2 积分，平局双方获得 1 积分，败北队伍获得 0 积分。除积分外，队伍在比完一轮之后还会获得对手分。对手分根据对手的强弱(即对手的积分)由软件来计算。最终用于计算比赛名次的得分由积分和对手分用软件综合计算得出。

#### 3. 比赛轮数

为了充分利用比赛时间，比赛轮数将在之后根据 AI 的实际执行速度、以及小组的个数决定。

#### 4. 先后手

第一轮比赛随机决定先后手，之后的比赛将以每个队伍先后手次数尽量相等为原则，由软件决定先后手。

#### 5. 轮空

因比赛为两两结合，若队伍数为奇数，则每轮比赛都会有一个队伍没有对手，此时该队伍记为胜利，获得 2 积分，但不会得到对手分。每个队伍最多只会有一次轮空。

#### 6. 编排用的软件

本次比赛采用的编排软件为 云蛇比赛编排 v3.3941，下载地址：

<http://www.ysbsbp.com/index.html>

#### 四、截止日期和展示时间

1. 在第 13 周前，各小组完成 AI 的设计。

2. 在第 13 周，提交 AI 的设计报告（包括基本功能、整体构建、整体逻辑、模块设计、每个组员的工作量的描述）、一个演示文稿（与设计报告的内容一致，不一定要当众展示，但也要做）、和一个演示视频（选做）。同时，进行初赛。

3. 在第 14 周，进行复赛。

4. 在第 15 周，前三名进行决赛。（表演性）同时前三名展示演示文稿。

#### 五、关于使用已有的开源代码

暂时定为：对之前的规则进行限制放宽：可以使用与象棋规则、数学有关的库；要实现的人工智能有关的功能的剩余代码必须自己写出。

如：

使用某库内的一个数据结构 Move 表示从某处到某处的一个走子——允许；

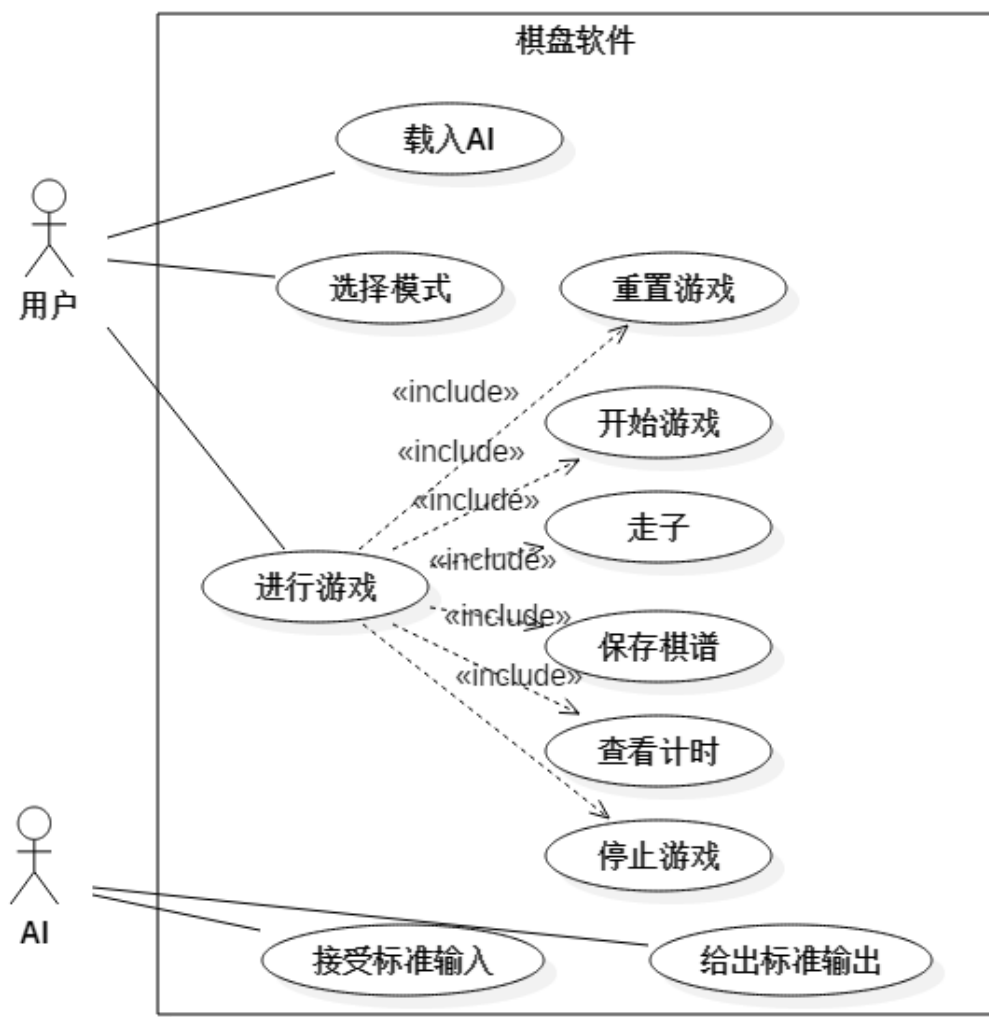
使用某数学库内的某算法进行大矩阵运算——允许；

使用某人工智能库中的某个算法函数——不允许；

自己实现某人工智能算法，但实现为该算法而特别设计的一个数据结构，或可用于该算法的某个非常见的、复杂的数据结构时，使用库——不允许。

## 5.2 用例图

注：由于 AI 并不属于本软件，并且模拟人类进行对弈，所以也将其看作一个“参与者”；简洁起见，只列出了最为重要的用例。



## 5.3 用例描述

### 5.3.1 选择模式

UC1 选择模式	
参与者	用户，通常为组织者
触发条件	组织者希望切换游戏模式
前置条件	当前载入 AI 的情况满足该游戏模式的前提
后置条件	可以开始游戏
正常流程	<ol style="list-style-type: none"> <li>1. 用户点击单选按钮，切换模式</li> <li>2. 本软件响应点击，切换模式，并更新 UI 以符合新的模式</li> </ol>
扩展流程	<ol style="list-style-type: none"> <li>1a. 用户选择的模式当前不允许使用： <ol style="list-style-type: none"> <li>1. 本软件不响应点击</li> </ol> </li> </ol>
特殊需求	

### 5.3.2 进行国际象棋游戏

UC2 进行国际象棋游戏	
参与者	用户，通常为组织者
触发条件	用户点击“开始”
前置条件	用户已经正确载入了 AI，选择了正确的模式。
后置条件	进行游戏的各项操作。
正常流程	<ol style="list-style-type: none"> <li>1. 若有 AI 参与则载入 AI 程序，设置 AI 方为自动</li> <li>2. 开始游戏，棋盘界面初始化</li> <li>3. 国际象棋棋盘程序在有人工参与时接受鼠标界面点击走子，显示所选棋子可行落子，在有 AI 参与游戏时接受 AI 标准输出，并判断合法性</li> <li>4. 判断游戏局面，及时修改棋盘对战界面，在存在 AI 时向 AI 发送对手走子信息，同时对 AI 方走子进行计时并在界面进行显示以及更新，并且在日志窗体以及历史窗体进行走子、游戏状态更新。</li> <li>5. 重复 4~5 步直至出现胜负或和局局面，显示结果，并等待游戏重置</li> </ol>
扩展流程	<p>3a.AI 输出不合法：</p> <ol style="list-style-type: none"> <li>1. 系统提示错误并向该 AI 发送重新走子信息</li> </ol> <p>3b.AI 因未知原因退出：</p> <ol style="list-style-type: none"> <li>1. 系统提示错误信息并等待游戏重置</li> </ol> <p>4a.需要查看全部或部分走子历史步骤：</p> <ol style="list-style-type: none"> <li>1. 用户可以在历史窗体拖动选择需要查看的历史走子，棋盘回放走子，期间不中断正在进行的对战</li> </ol> <p>4b.用户需要重置游戏</p> <ol style="list-style-type: none"> <li>1. 用户按下“重置”按钮</li> <li>2. 游戏重置</li> </ol> <p>4c.用户需要停止游戏</p> <ol style="list-style-type: none"> <li>1. 用户按下“停止”按钮</li> <li>2. 游戏停止</li> </ol> <p>5a.需要保存棋谱：</p> <ol style="list-style-type: none"> <li>1. 用户选定所有历史走子</li> <li>2. 用户按下“Ctrl-C”</li> </ol>
特殊需求	1.玩家知悉执棋方

### 5.3.3 AI 接受系统标准输入

UC3 接受系统标准输入	
参与者	AI
触发条件	AI 等待对方走棋
前置条件	整个游戏至今为止正常进行
后置条件	AI 可以进行思考并给出己方走子
正常流程	<ol style="list-style-type: none"> <li>1. AI 进入等待输入的函数</li> <li>2. 本软件给出一个标准输入到 AI，表示对方的走子</li> <li>3. AI 接受输入，并解析应用到自身内部维护的棋盘</li> </ol>
扩展流程	无
特殊需求	

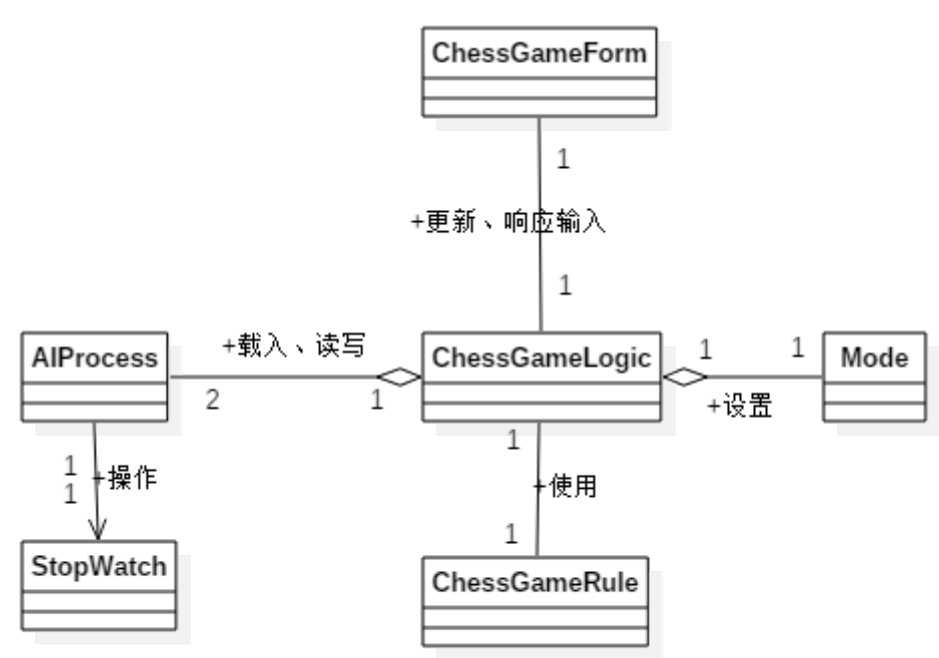


5.3.4 AI 给出标准输出

UC4 向系统标准输出	
参与者	AI
触发条件	AI 思考完毕，可以给出走棋
前置条件	整个游戏至今为止正常进行
后置条件	AI 等待对方的走子输入
正常流程	1. AI 给出一个 SAN 字符串表示自己的走子 2. 本软件读入该字符串 3. 本软件解析该字符串 4. 该字符串正确，本软件将走子信息应用到内部棋盘，并更新 UI，开始处理下一步
扩展流程	4a.该字符串错误（或走子不符合规则） 1. 本软件向 AI 发信，说明该字符串错误 2. AI 重新给出一个字符串 3. 回到正常流程-2
特殊需求	

5.4 概念类图

由于本软件从需求来看功能要求不多，故对于所有用例，统一绘制一个概念类图于此。带有属性、方法的详细的类图与此图有所差别，请参阅《软件设计报告》。

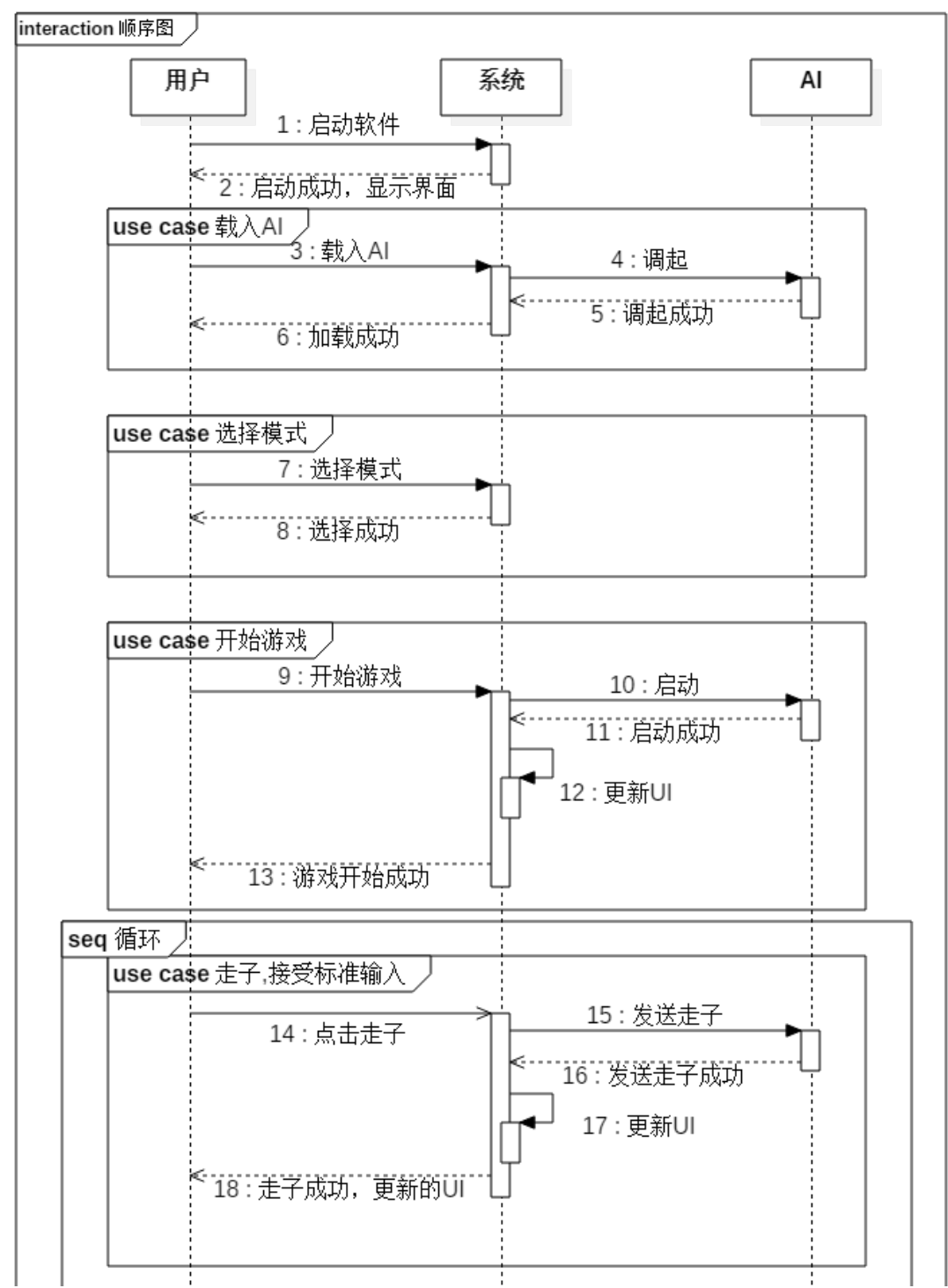


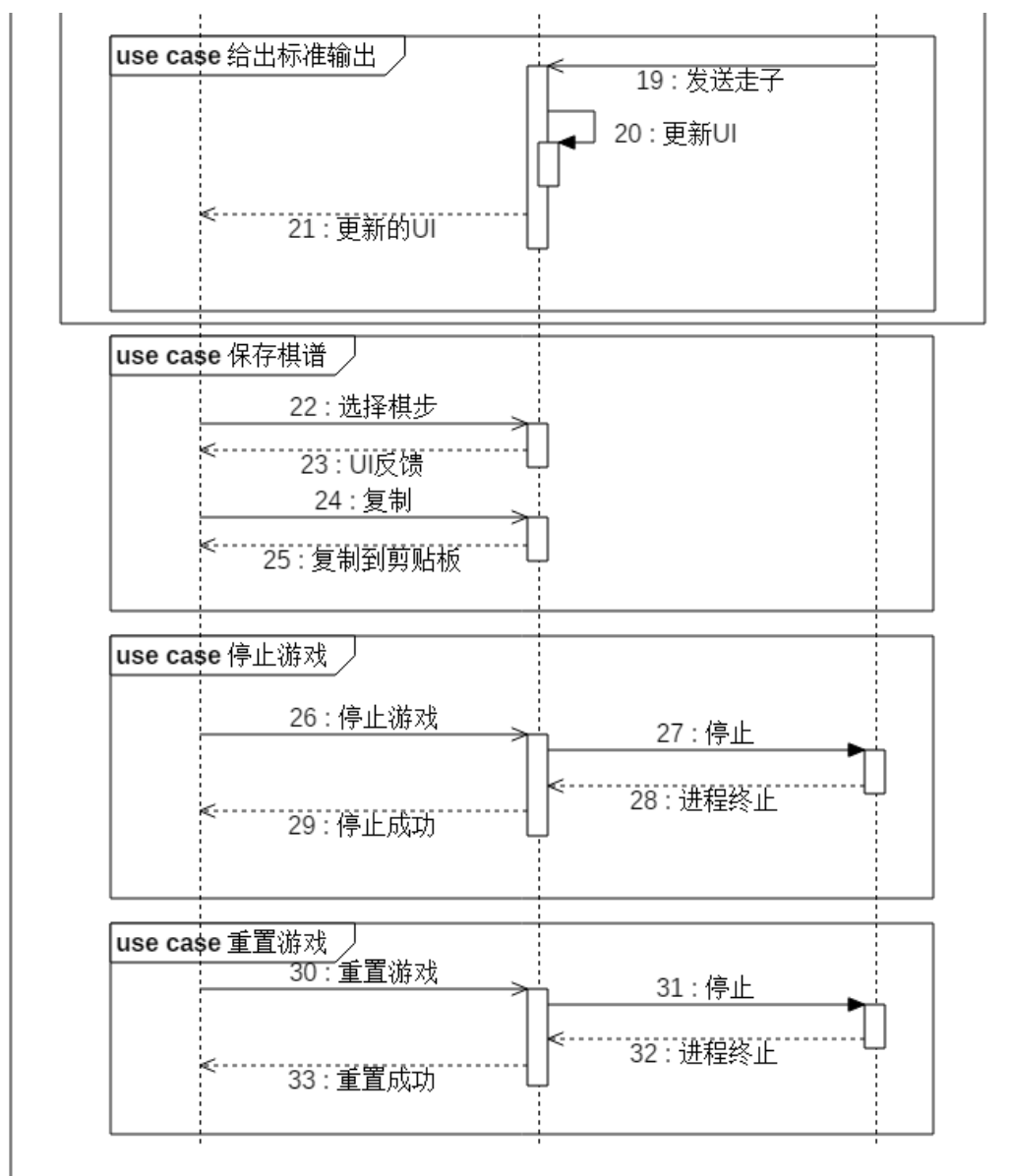
解释：ChessGameLogic 是关于游戏运行时逻辑的类，为软件的核心；  
ChessGameForm 是游戏运行的窗体，充当 UI；ChessGameRule 是判断走子是否合

法、游戏是否结束时用到的象棋规则类；AIProcess 和 Stopwatch 分别为 AI 进程类和计时类。

### 5.5 系统顺序图

同概念类图，我们将所有用例集合在一起，以一个用户开启软件到进行完一局游戏的全过程绘制了系统顺序图。





## 5.6 状态图

