

# 五子棋AI大赛参赛手册

目录：

- 五子棋AI大赛参赛手册
  - 比赛说明
  - 比赛规则
  - 五子棋AI规则
    - 基本交互规则
    - 五子棋比赛规则
    - 限制
    - 示例AI
    - 注意
  - 对战平台使用说明
  - 如何调试你的AI
  - 参考资料

## 比赛说明

- 这个比赛并不要求强制参加。
- 这个比赛应该只占用你们个人的空闲时间，不允许以参加这个比赛为借口不写作业或者不做家务等。
- 这个比赛的主要目的是鼓励大家用所学的知识解决实际问题。
- 比赛期间允许交流允许查资料。
- 友谊第一比赛第二。

## 比赛规则

- 比赛内容为五子棋，选手提交自己写的AI代码和其他选手的AI进行对战。
- 比赛从4月30号开始，5月2号晚上10点结束。
- 比赛期间每1小时会进行一次两两对战，两两对战会将所有选手的最新AI进行两两对战。每两个AI之间的对战会进行两局，会交换先后手。
- 以5月2号晚上10点的两两对战为最终排名，如果对战过程出现平台BUG导致对战没有顺利完成，则该两两对战中的所有对局都会重新进行。
- 我也会参与比赛，但是不参与最终排名。
- 最终解释权归平台放所有。

两两对战的说明：

假设有三个选手提交了AI，分别是 AI1，AI2 和 AI3。那么这三个选手两两会进行对战。并且每两个选手之间会对战两局，交换先后手。

因此总共会对战 6 局，分别是：AI1 vs AI2，AI2 vs AI1，AI1 vs AI3，AI3 vs AI1，AI2 vs AI3 和 AI3 vs AI2。

每局对战，胜者AI获得 2 分，败者AI获得 0 分，平局则双方各获得 1 分。

上面的 6 局对战中，假设 AI1 最厉害，它参与的对局都获胜了，则它可以获得  $4 * 2 = 8$  分。  
最终排名以每个AI获得的总得分从高往低排，得分相同的AI具有相同的排名。

5月2号晚上10点的两两对战决定最终排名，在之后平台还能提交AI代码并且对战，但不会影响最终排名，5月3号晚上10点会临时关闭对战平台，五一假期结束之后再开启。

最终所有参赛选手都能获得一个奖杯，第一名的奖杯会刻上“五子棋AI大赛第一名”，第二名的奖杯会刻上“五子棋AI大赛第二名”，依次类推。奖杯的初步方案如下图：

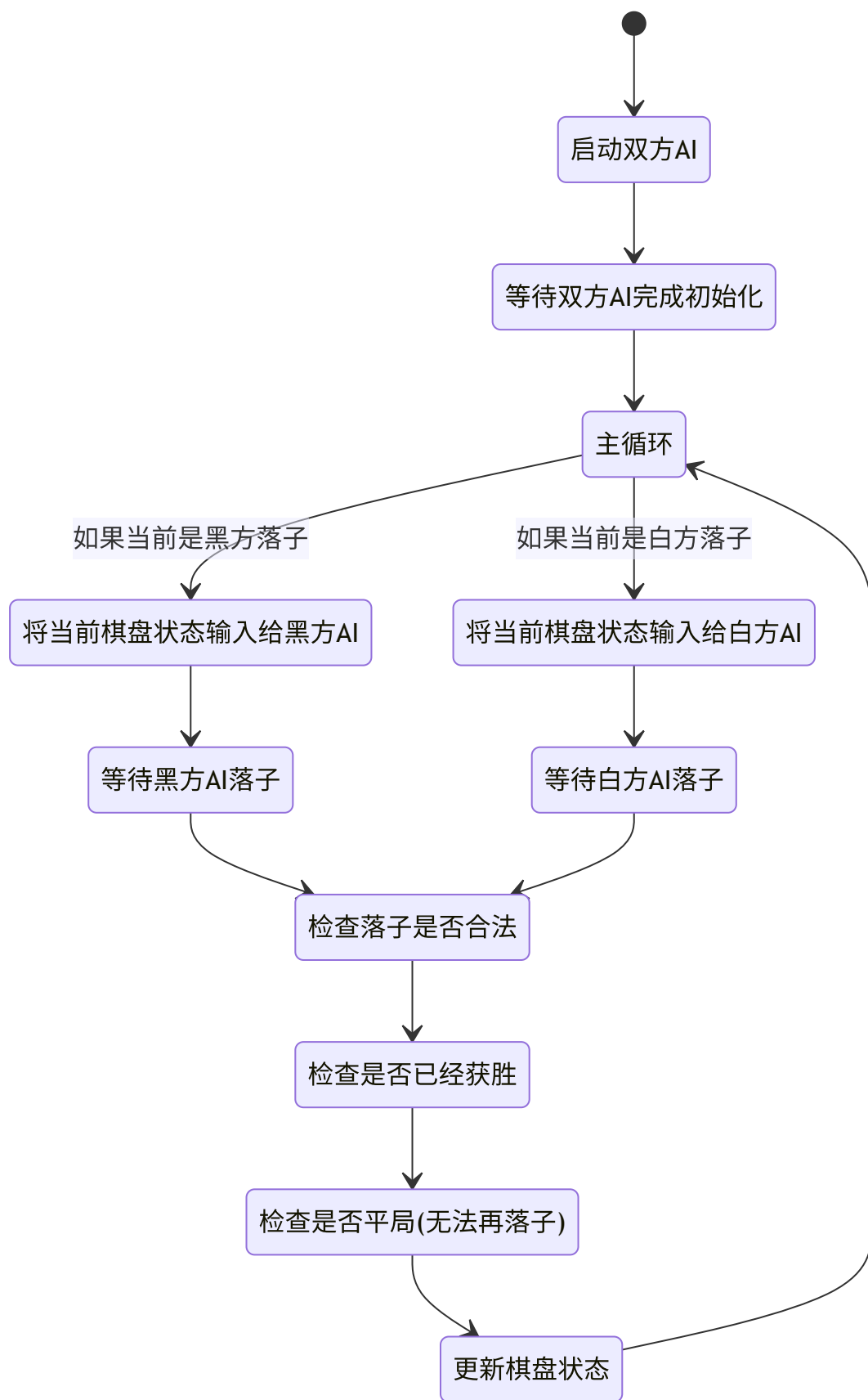


## 五子棋AI规则

### 基本交互规则

选手可以提交C++代码，平台会将其命名为 `code.cpp`，并使用 `g++ --std=c++14 -O2 code.cpp -o ai.exe` 编译命令将选手提交的代码编译为AI。

交互流程如下，整个对局过程中双方AI会被平台所启动，并且一直运行到对局结束。



## 五子棋比赛规则

在本次活动中，五子棋的棋盘大小为  $11 \times 11$ ，棋盘的行从上往下依次编号为 0 到 10，棋盘的列从左往右依次编号为 0 到 10。

黑方为先手，白方为后手，无禁手限制。先在横线、竖线或斜对角线上形成至少 5 子连线者获胜。

对战期间分为两个阶段：(1) 初始化阶段 和 (2) 落子阶段。

在 初始化阶段 ，AI可以什么都不做，也可以对所使用到的一些数据进行初始化，初始化结束时应该输出一行 `init end` 表示结束。

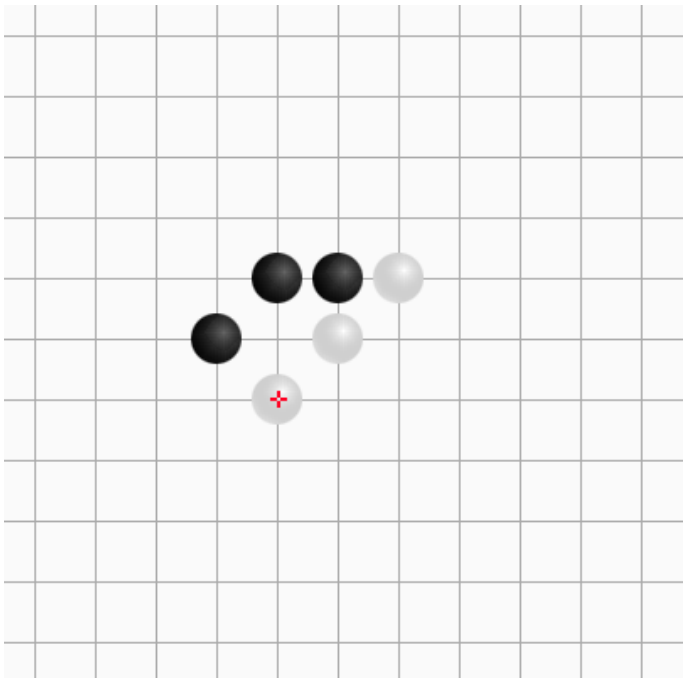
在 落子阶段 ，平台会通过输入将当前的棋盘局面传递给AI，格式如下：

```
player
chessboard
```

其中 `player` 表示自己是黑方还是白方，`1` 表示黑方，`2` 表示白方。这个参数在整局对战过程中都会保持不变。

`chessboard` 是一个 `11 x 11` 的整数矩阵，`0` 表示该位置空白，`1` 表示该位置是黑方的棋子，`2` 表示该位置是白方的棋子，不会出现其他数字。

比如已经进行了 `3` 回合对战，当前的局面如下：

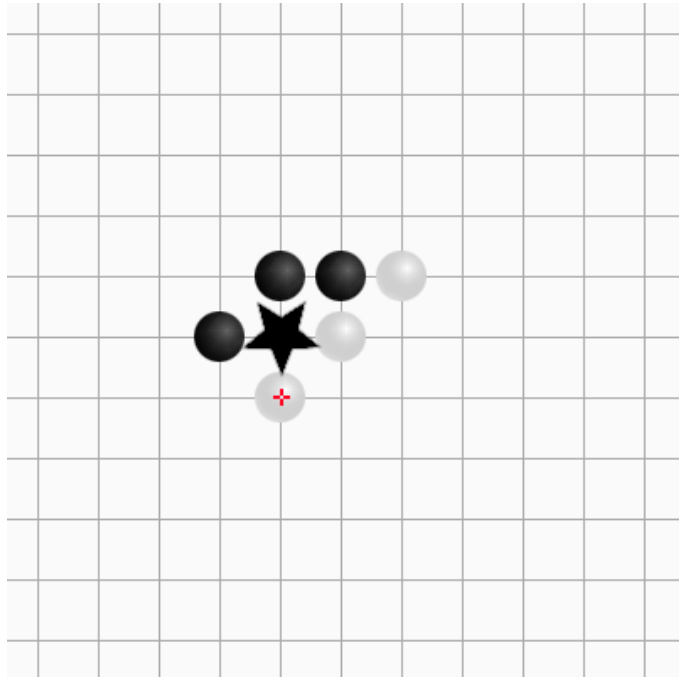


下一步是黑方AI进行落子，因此输入给黑方AI的局面就是：

```
1
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 2 0 0 0 0
0 0 0 1 0 2 0 0 0 0 0
0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
```

如果黑方AI决定再第 `5` 行第 `4` 列落子，那么可以输出：

落子位置如图：



## 限制

0. AI的编译时间不能超过 30 秒。
1. 初始化应该不超过 30 秒。
2. 每个AI从读入局面到输出落子应该不超过 5 秒。
3. 每个AI在输出落子之后到读入下一次局面之前不应该进行运算。
4. 只允许单线程运行，不能访问网络。

## 示例AI

以下AI是一个示例AI，该AI在每一回合会随机选择一个空白位置落子。

```
#include <iostream>
#include <cstdlib>
#include <cstdio>

using namespace std;

const int MAXN = 11;
int player;
int chessboard[MAXN][MAXN];

int main()
{
    // 如果有什么初始化，可以写在这里
    // 初始化结束输出`init end`
    srand(time(0));
    cout << "init end" << endl;
```

```
while (cin >> player)
{
    for (int i = 0; i < MAXN; i++)
    {
        for (int j = 0; j < MAXN; j++)
        {
            cin >> chessboard[i][j];
        }
    }

    // 随机选择一个没有棋子的位置输出
    int x, y;
    while (true)
    {
        x = rand() % MAXN;
        y = rand() % MAXN;
        if (chessboard[x][y] == 0)
            break;
    }
    cout << x << " " << y << endl;
}

return 0;
}
```

## 注意

为了使得平台能够获取到AI的输出，你的AI最好遵循以下原则：

1. 如果使用 `cout` 输出，那么最好使用 `endl` 换行。
2. 如果使用 `printf` 输出，那么使用 `\n` 换行，并且在输出之后运行 `fflush(stdout);`。

## 对战平台使用说明

---

第1步，提交AI：

\* AI代码：

```
#include <iostream>
#include <cstdlib>
#include <cstdio>

using namespace std;

const int MAXN = 11;
int player;
int chessboard[MAXN][MAXN];
```

第2步，和其他选手进行对战：

进行一次对战：

我要打十个!!!



	代码长度	1
	731	2

第3步，等待对战完成：

<

1

>

编号	发起人	对局数量	对战状态
33	admin	4	对战完成

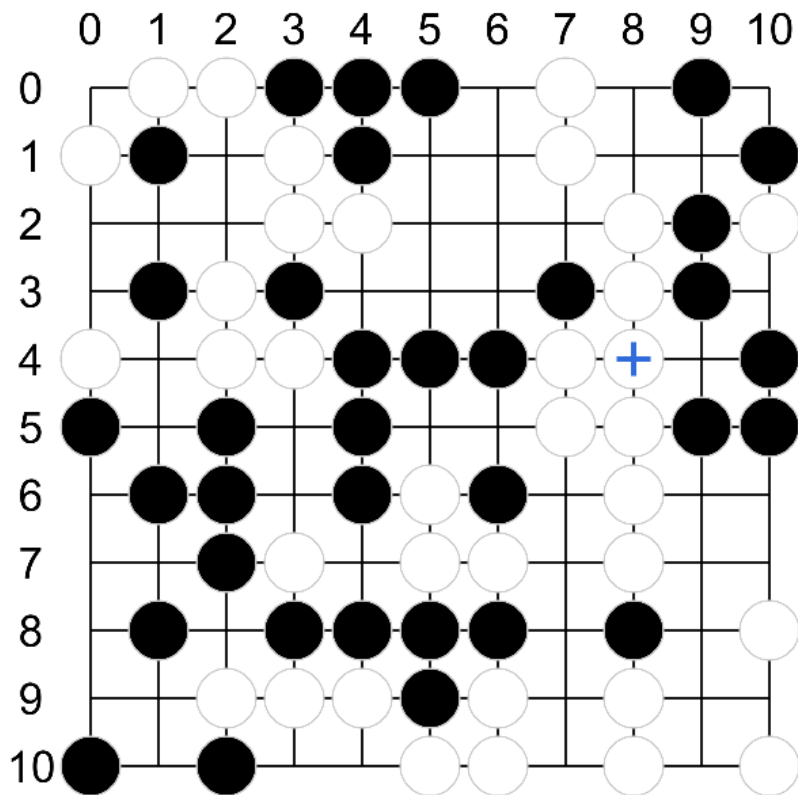
第4步，查看对战结果：

发起人	admin
发起时间	2022-04-28 00:18

编号	AI1	AI2	对局状态
534	admin (版本 9)	user1 (版本 4)	AI1赢两局
535	admin (版本 9)	user2 (版本 6)	各赢一局
536	admin (版本 9)	user3 (版本 7)	各赢一局

第5步，播放对战录像：



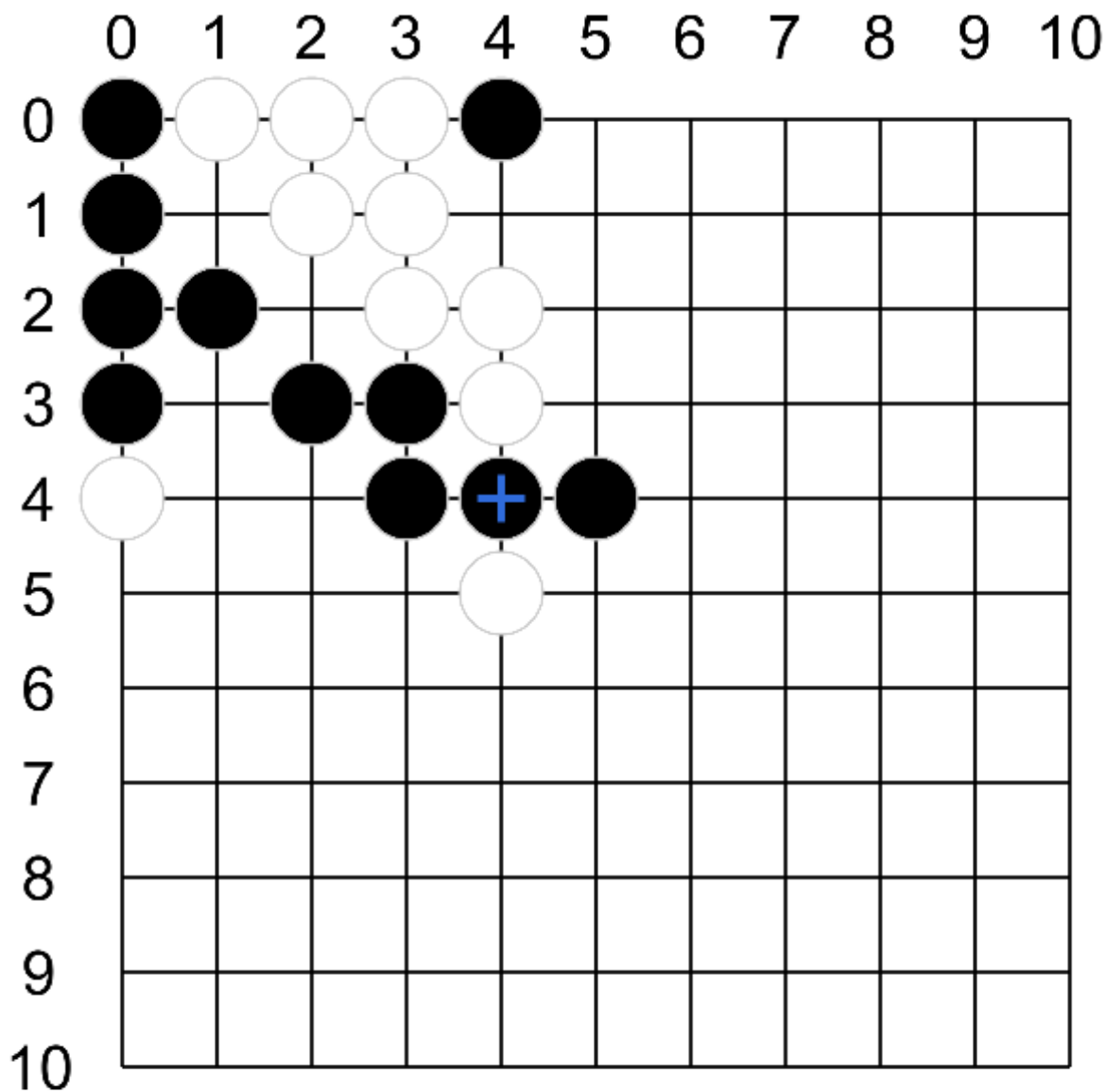


回合35:白方落子(4, 8)
回合35:黑方落子(10, 0)
回合34:白方落子(4, 7)
回合34:黑方落子(5, 4)
回合33:白方落子(3, 2)
回合33:黑方落子(0, 4)
回合32:白方落子(6, 8)
回合32:黑方落子(6, 1)
回合31:白方落子(1, 7)
回合31:黑方落子(8, 4)
回合30:白方落子(9, 8)
回合30:黑方落子(4, 6)
回合29:白方落子(9, 4)

## 如何调试你的AI

假如你的AI是白方，在一个局面下你的AI执行了一个意料之外的落子。

你可以首先将局面移动到你的AI落子之前的局面：



清空棋盘

移动到最后

复制当前局面到剪贴板

然后点击下方的“复制当前局面到剪贴板”，然后打开一个记事本，按Ctrl-V就能够将当前的局面输出：

```

2
1 2 2 2 1 0 0 0 0 0 0
1 0 2 2 0 0 0 0 0 0 0
1 1 0 2 2 0 0 0 0 0 0
1 0 1 1 2 0 0 0 0 0 0
2 0 0 1 1 1 0 0 0 0 0
0 0 0 0 2 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0

```

你可以将该局面输入给你的AI程序，从而对你的AI进行调试。

如果你无法将局面复制到剪贴板，请尝试使用谷歌浏览器或者Edge浏览器。

## 参考资料

---

- [五子棋AI设计心得 —— 上篇](#)
- [一步步看 \$\alpha\$ - \$\beta\$ 剪枝算法](#)
- [蒙特卡洛树搜索](#)

