

AI BATTLE——Rodirou

2021 PPCA

游戏规则

棋盘由 9×9 个格子构成，格子间有凹槽。

每方各有一棋子，大小为 1×1 。各有10片木板（墙），长度为2格，用来放在凹槽中。

两人玩时，两子需放在相对侧。

先将自己的棋子移动到对方**底线**（即对方棋子的起点所在行）的获胜。

轮到回合的玩家，需要进行以下两动作之一：

移动：移动至邻边四格之一，但不可穿过墙。若与对方的棋子相邻，则可跳过对方的棋子到它的后方格。若与对方的棋子相邻，但是对方的棋子后面是墙或是棋盘边界，则不可以跳到后方格，只可以跳到对方棋子的左边或右边（当然需保证对方棋子的左边或右边无墙）。

放墙：放木板至凹槽。木板不能交叉或重叠。放置后不可以使自己或对方的棋子永远无法到达对侧底线。

内容

实现 `sample.cpp` 文件，你需要实现 `init` 与 `action` 函数。

`init` 函数只会在程序开始执行时调用一次。

`action` 函数中你会收到对手的决策，你需要返回你的决策。

`ai_side` 表明你的先后手，0表示先手，1表示后手。

接口

棋盘的表示法基于玩家0的视角。

初始时，玩家0在下方中央，玩家1在上方中央。

从玩家1一侧到玩家0一侧的行依次标记为0到8。

从左侧到右侧，列依次标记为0到8。

一个位置的坐标被表示为(行号, 列号)，例如，玩家0的起点为(8, 4)，玩家1的起点是(0, 4)。

轮到你的回合时，你需要提交一个**动作**，**动作**定义为(类型代码, (行坐标, 列坐标))（在C++代码中表示为：`pair<int, pair<int, int>>`）。

类型代码定义如下：

—1：您不会使用到，但是如果您接受到此种类型代码表明您是全局的第一步

0：移动棋子

1：放置垂直木板

2：放置平行木板

关于**坐标**，棋子移动的坐标被表示为**棋子占据的新坐标**，放置木板的坐标被表示为**该木板中心位置左上角的正方形的坐标**。

您需要分析对手的动作并返回自己的**动作**，抢先一步将自己的棋子移动到对方的底线（玩家1需将自己的棋子移动到第8行，玩家0需将自己的棋子移动到第0行）。

如何进行本地测试

运行 `judge.py`。

后接两个参数表示ai(可执行文件)路径，如果ai路径为 `human` 则表示为人机或人人对战。

下面是例子

```
1 python3 judge.py ./sample_ai ./sample_ai
2 python3 judge.py human ./sample_ai
3 python3 judge.py human human
```

如何进行在线测试

<http://ai.cong258258.xyz/>

注册并登陆OJ，上传你的AI。

每个AI会有一个rating，初始为1500，进行比赛后rating会变动，变动规则遵循ELO算法。

每天凌晨会进行循环赛，参加循环赛的AI是baseline与所有选手最后提交的AI版本。

请注意：OJ的rating并非您的最终成绩,因为最后我们还要进行统一测试，但是从一定程度上来说，rating也反映了你AI的强度。

请注意：请不要向 `stdout` 中输出任何信息，否则将导致您输掉比赛。如有输出调试信息需要，请输出到 `stderr`。

时空限制

内存限制：512M

时间限制：单步时限2s

服务器配置

```
1 sys: Ubuntu 20.04(WSL on WIndows Server 2019 Datacenter)
2 cpu: Intel(R) Xeon(R) Platinum 8269CY CPU @ 2.50GHz 2.50GHz
3 mem: 2.00 GB
```

评分细则

请在DDL之前确认您最后提交的AI是您参与评分的版本。

- 工科
 - 实现**Minimax Search** && **Alpha-beta Prune**(code review) 60%
 - 击败baseline 20%
 - 每击败任意一个baseline获得5%，共六个，不会溢出。
 - 击败是指**胜率**> 70%。
 - 最终排名 20%
 - 我们会在第四周结束之后进行循环赛。每两个人会进行若干局（≥ 10根据评测压力而定）对战，一半先手一半后手。

循环赛结束后会根据胜场数与击败选手的胜场数将所有人排名。

排名越高，此部分分数越高。

- ACM班
 - 同工科全部内容 70%
 - ML-related Task 30%