Project 1 – 五子棋 AI 算法

1 说明

五子棋是一项规则比较简单的棋类游戏，其器具与围棋通用，起源于中国古代传统的黑白棋种之一。通常双方分别使用黑白两色的棋子，下在棋盘直线与横线的交叉点上，轮流下子，先形成五子连线者获胜。

本作业中用到的棋盘默认为 15*15 的棋盘（管理员可根据需要修改设置），学生需要按照接口要求实现五子棋的 AI 算法，按照要求提交到系统进行可用性测试和积分赛。

2 评分规则

将评估分为 2 个阶段：

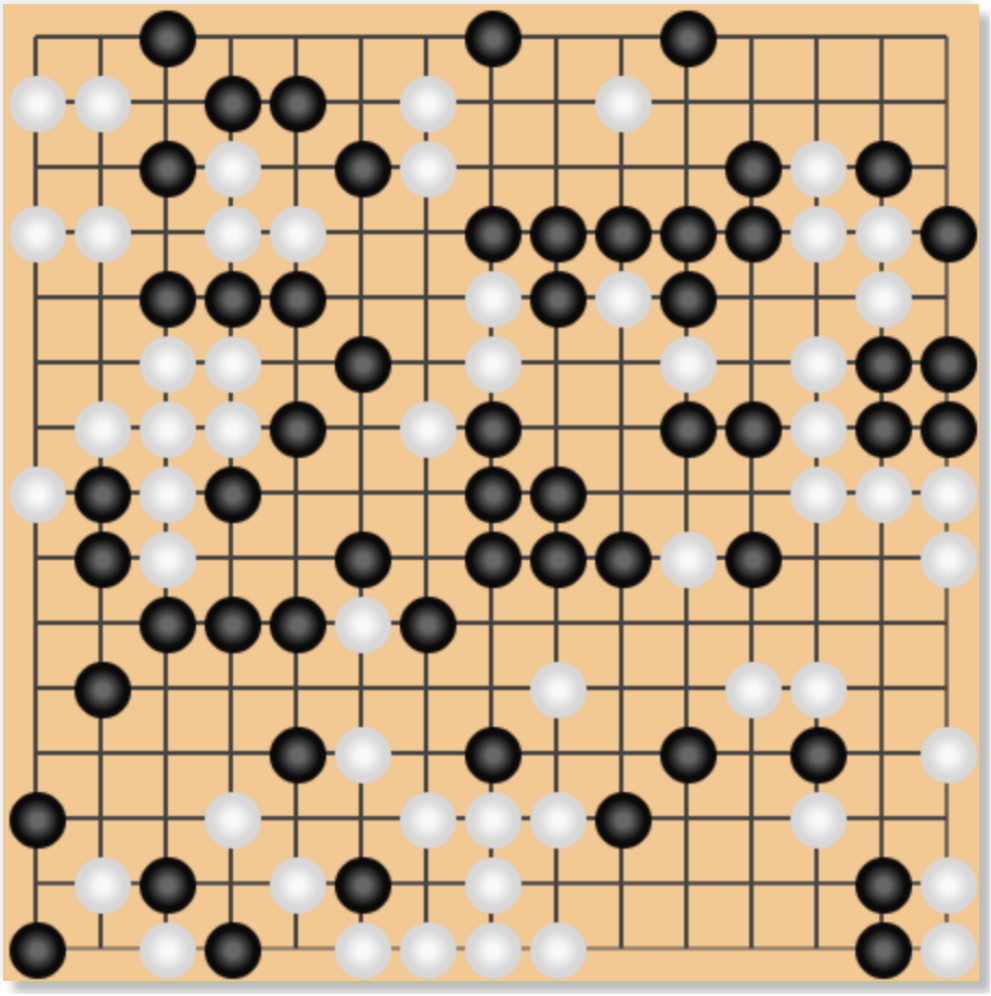
可用性测试：在这项测试中，我们将使用一些简单的棋盘测试用例，学生需要找到最佳落子的位置。只有通过可用性测试的作业才能及格。

积分阶段：通过可用性测试的学生，才能参加积分赛。具体比赛规则为：学生可将其 AI 算法提交到五子棋对战平台上，提交（提交后进行可用性测试，通过了才算提交成功）成功后，选择排名比自身靠前一名的选手 PK，如果战胜了对手，则其积分增加 10，如果平局则积分保持不变，如果战败，则积分减 10 分。为了避免先手优势，每次 PK 均对战 2 局，双方轮流先手，如果两次均获胜，或一胜一平，则胜；如果一胜一负，则为平局；如果两次均败，或一败一平，则负。

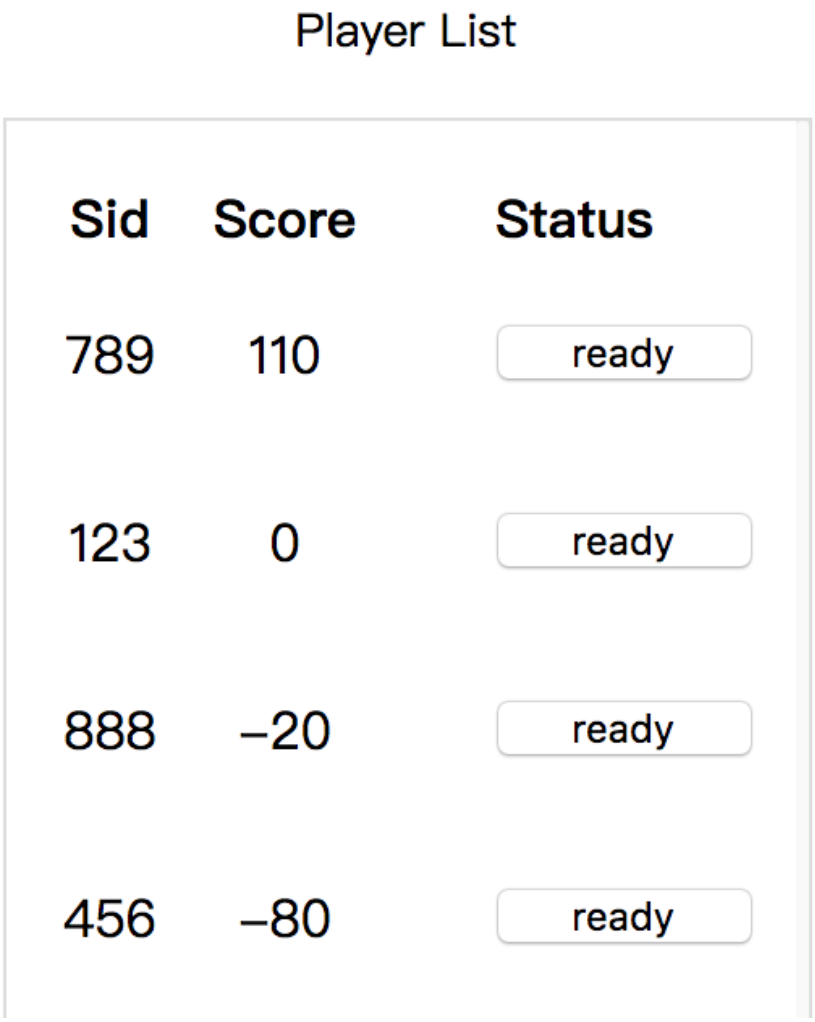
在 DDL 之前，学生均可通过提交代码，观测积分来评估自己的算法效果。最终的成绩按照 DDL 之后，系统进行循环赛积分的排名给分。

如下棋局就是两个 AI 算法对战的战况，当然这个示例很简单，为两个随机下子的 AI 算法的模拟结果。

Status： 游戏结束，123获胜！ Go Play



如下为积分榜示例：



五子棋对战平台目前还处理开发内测阶段，各方面还有很大提升空间，等系统稳定后将开放给所有同学使用。目前招募优秀的同学参与开发和测试，欢迎大家积极报名。请有意者发邮件给：zhaoy6@sustc.edu.cn

3 代码要求

1、Python version: 3.6

2、Code template:

```
import numpy as np
import random
import time

COLOR_BLACK=-1
COLOR_WHITE=1
COLOR_NONE=0
random.seed(0)
#don't change the class name
class AI(object):
    #chessboard_size, color, time_out passed from agent
    def __init__(self, chessboard_size, color, time_out):
        self.chessboard_size = chessboard_size
        #You are white or black
```

```
15          self.color = color
16          #the max time you should use, your algorithm's run time must not exceed the time
    limit.
17          self.time_out = time_out
18          # You need add your decision into your candidate_list. System will get the end of
    your candidate_list as your decision .
19          self.candidate_list = []
20
21
22      # The input is current chessboard.
23      def go(self, chessboard):
24          # Clear candidate_list
25          self.candidate_list.clear()
26          #===============================================================
27          #Write your algorithm here
28          #Here is the simplest sample:Random decision
29          idx = np.where(chessboard == COLOR_NONE)
30          idx = list(zip(idx[0], idx[1]))
31          pos_idx = random.randint(0, len(idx)-1)
32          new_pos = idx[pos_idx]
33          #==============Find new pos=======================================
34          # Make sure that the position of your decision in chess board is empty.
35          #If not, return error.
36          assert chessboard[new_pos[0],new_pos[1]]== COLOR_NONE
37          #Add your decision into candidate_list, Records the chess board
38          self.candidate_list.append(new_pos)
39
```

3、时间测量

start = time.time()

… algorithm…

run_time = (time.time() - start)

4 文档要求

1）参见《ProjectReport_Template.docx》

2）或者是《Transactions-instructions-only.pdf》和《ieeecitationref.pdf》

3）去年优秀实验报告《Lab1_report_11510506.pdf》

4）文档占比每个实验评分的 30%