**中山大学计算机学院**

**人工智能**

**本科生实验报告**

课程名称：Artificial Intelligence

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 |  | 姓名 |  |

# 实验题目

1. 与对手AI共博弈2次，其中先手、后手各评估一次（ 在main.py中未实现算法的红黑机指定代码，需自行实现 ）。积分规则：胜一局记3分，平一局记1分，负一句记0分。

# 实验内容

1. 算法原理

Alpha和bate剪枝博弈树：

从当前棋局出发预测下面棋局发展可能情况，计算棋局得分

1. 当轮到己方下棋时，进行alpha剪枝：

* 己方节点是max节点，下面的分支是或的关系，为了得到对己方有利的最大收益，应该找寻下面分支中棋局最大得分的分支。
* 遍历全部分支，时间和空间开销大，因此要进行alpha剪枝。能够alpha剪枝的依据：己方节点处于上一层对方节点（min节点）的与分支下,min节点要取分支中的最小值为节点值（beta），因此当己方节点的值大于上一层的beta值时，beta值不会在因为己方节点的值改变，可以停止搜索己方max节点的剩余分支
* 在象棋AI中具体表现：预测棋盘走向时记录上一层alpha，beta值，轮到己方下棋时，遍历己方可走的每一步棋的分支，计算棋盘得分，若棋盘得分大于beta值，则到达终止遍历的条件。且不断更新当层棋盘得分最大值为当层的alpha值

1. 当轮到对方下棋时，进行beta剪枝：

* 对方节点是min节点，下面的分支是与的关系，对方为了得到对自己有利的最大收益，应该找寻下面分支中棋局最小得分的分支。
* 遍历全部分支，时间和空间开销大，因此要进行beta剪枝。能够beta剪枝的依据：对方节点处于上一层己方节点（max节点）的或分支下,max节点要取分支中的最大值为节点值（alpha），因此当对方节点的值小于上一层的alpha值时，alpha值不会在因为对方节点的值改变，可以停止搜索对方min节点的剩余分支
* 在象棋AI中具体表现：预测棋盘走向时记录上一层alpha，beta值，预测到对方下棋时，遍历对方可走的每一步棋的分支，计算棋盘得分，若棋盘得分小于alpha值，则到达终止遍历的条件。且不断更新当层棋盘得分最小值为当层的beta值
* 总：预测下一步时，进入己方（max）节点，进行alpha剪枝，在获得max节点以下的分支值时，由于max和min层是交替的；进入对方（min）节点，进行beta节点，获得min节点以下的分支值时，进入己方（max）节点；如此循环往复下去，知道到达最后一层，然后逐层返回到第一层，在第一层得到得分最大的下一步；

注意：由于计算到最后一层开销太大，这里加以剪枝层数的限制，达到最大的剪枝层数则返回

* 伪代码（来自ppt）

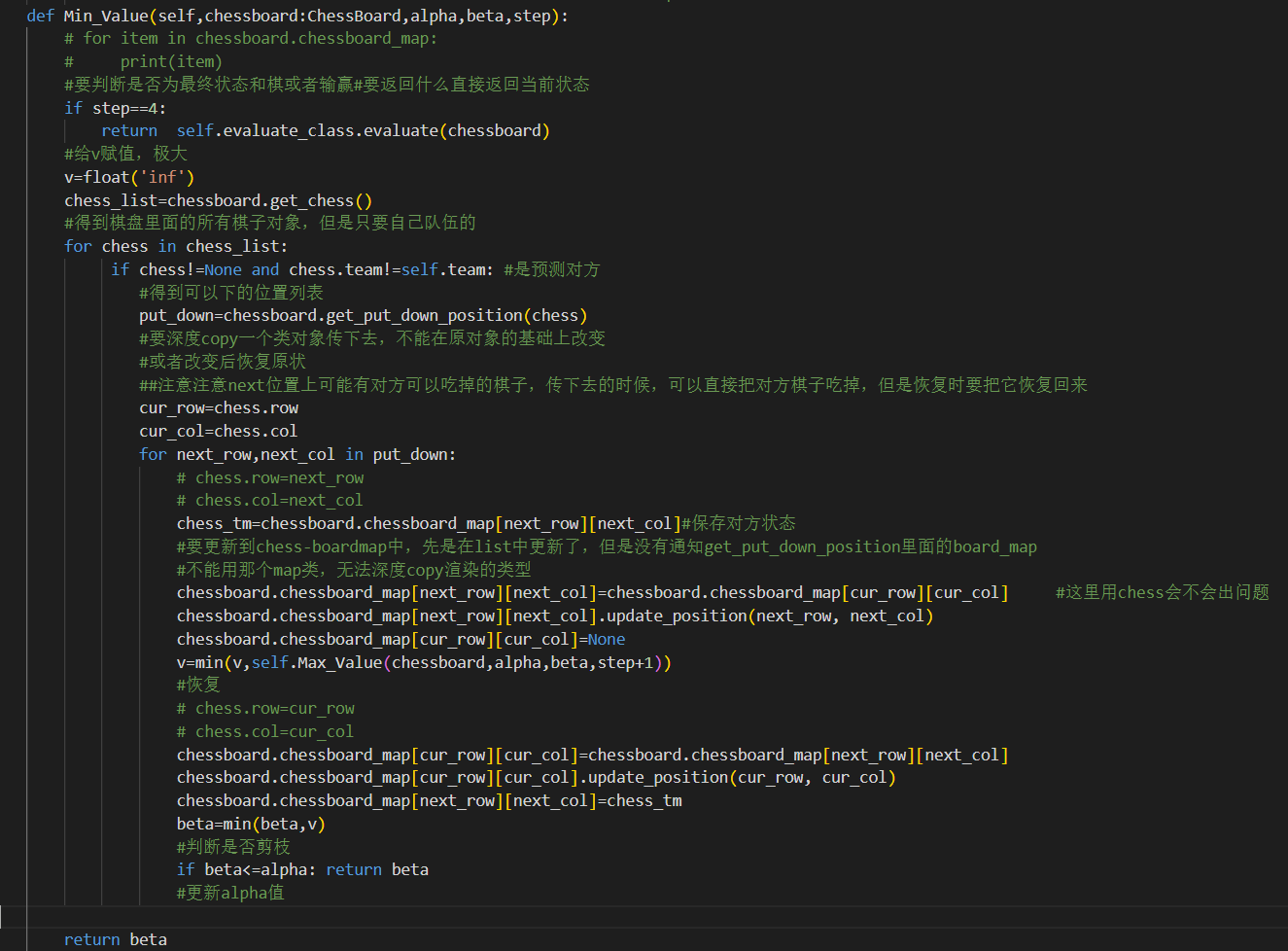


1. 关键代码展示（可选）
2. Beta剪枝：

* Beta剪枝时是在预测对方的下一步，因此注意遍历所有可下的棋子时，是遍历对方的棋子（chess.team!=self.team）
* 由于是不停地搜索到指定层数后逐层返回，因此要注意搜索完某个分支时要恢复本层的原来的样子，再进行下一个分支的搜索。
  + 在象棋AI中要注意，保存好新位置的旧棋子，一个分支搜索完就要恢复。新位置可能原来没有棋子为None，可能有棋子但是分支预测时被吃掉。

则不要直接恢复新棋子的位置为None，不然后续遍历往下搜索时可能会出现访问为空的错误

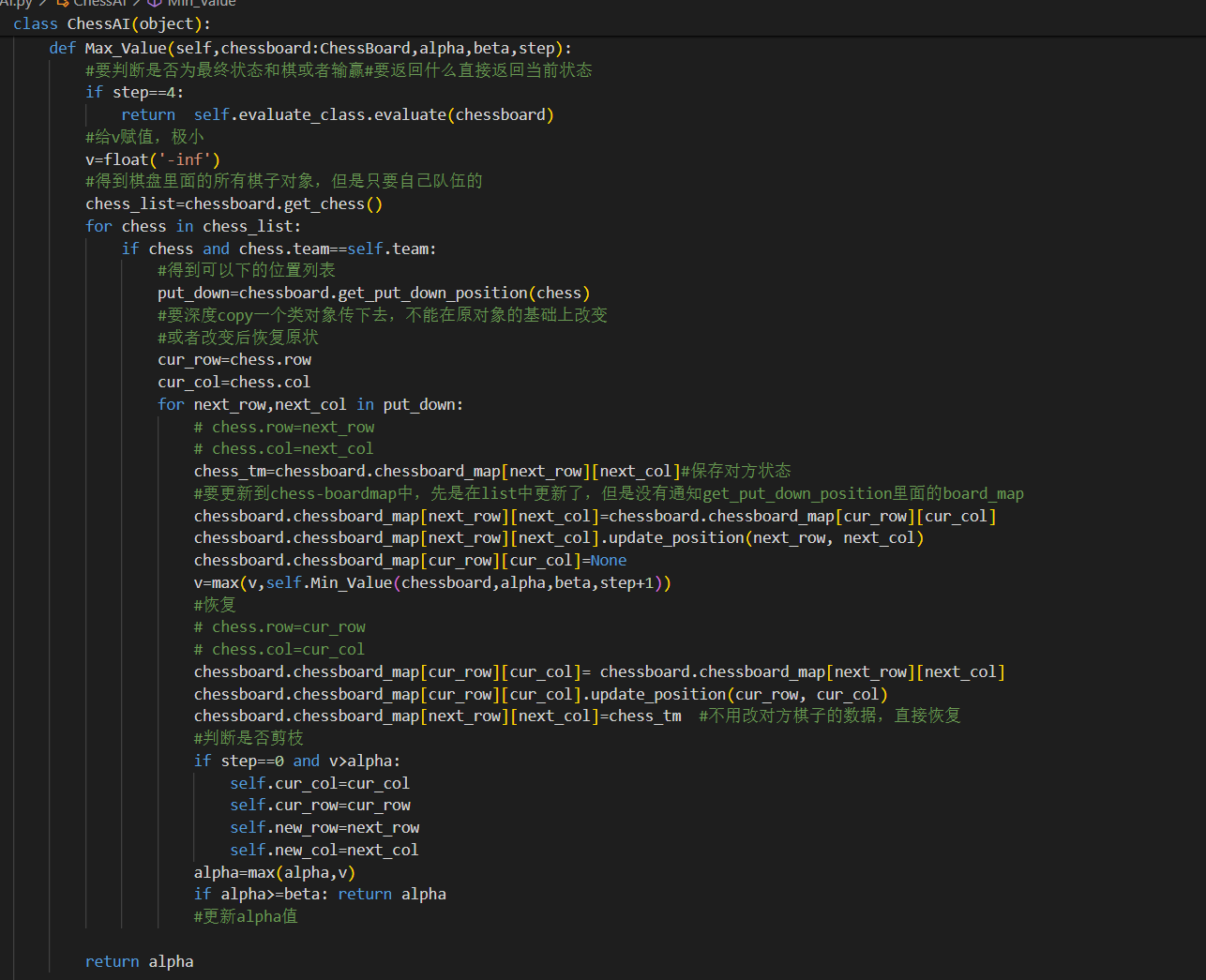
* 在更新旧棋子的行列时，可以直接用update\_position，如果直接更改行列，可能会出现走一步有三颗棋子移动的bug
* 返回给上一层的代表当前节点的分支值的变量，最好直接是beta，而不是临时记录beta的变量v，有时候v和beta的值不相同



1. Alpha剪枝:

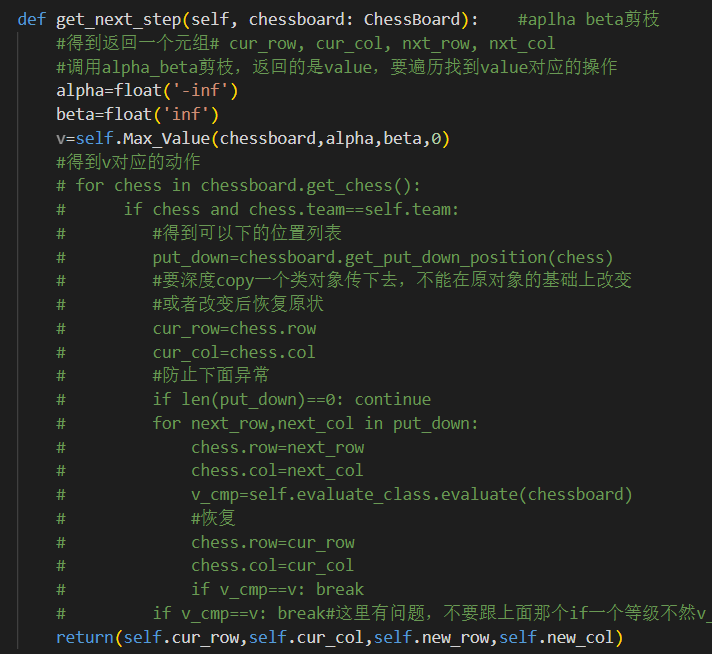
* 这里和ppt的伪代码略微不同：为了提高算法效率，加了一个在第一层且alpha值更新的时候，应该存储此时分支更新的情况（走的棋子原位置和新位置）。

如果按照伪代码，alpha剪枝在第0层时将最大的alpha值传给总函数（get\_next\_step），然后让总函数根据得到的最大的棋盘得分，遍历可走的下一步，获得棋盘得分，寻找最大棋盘对应的下一步，在总函数需要再次遍历第一层。所以直接在alpha剪枝的第0层实时更新。



1. 总get\_next\_step():

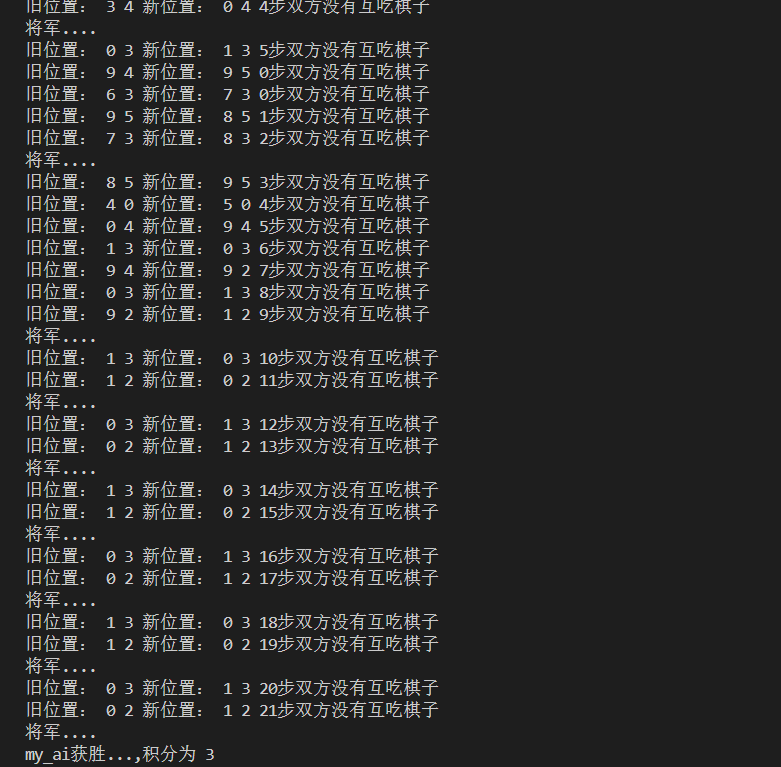
进入第0层，进行alpha剪枝



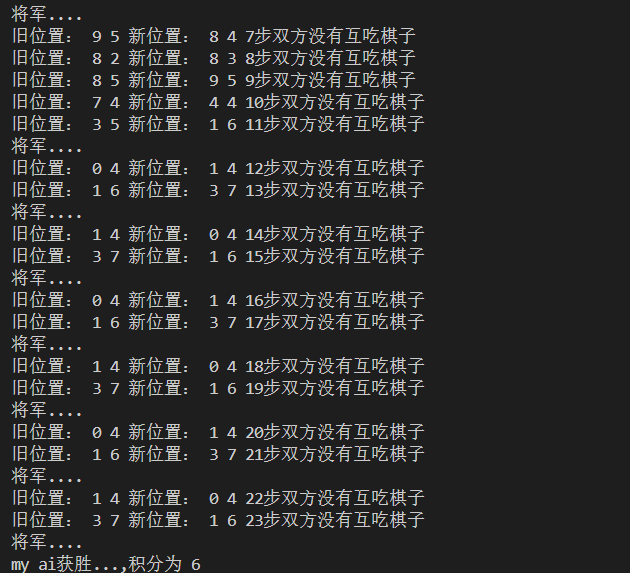
# 实验结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）

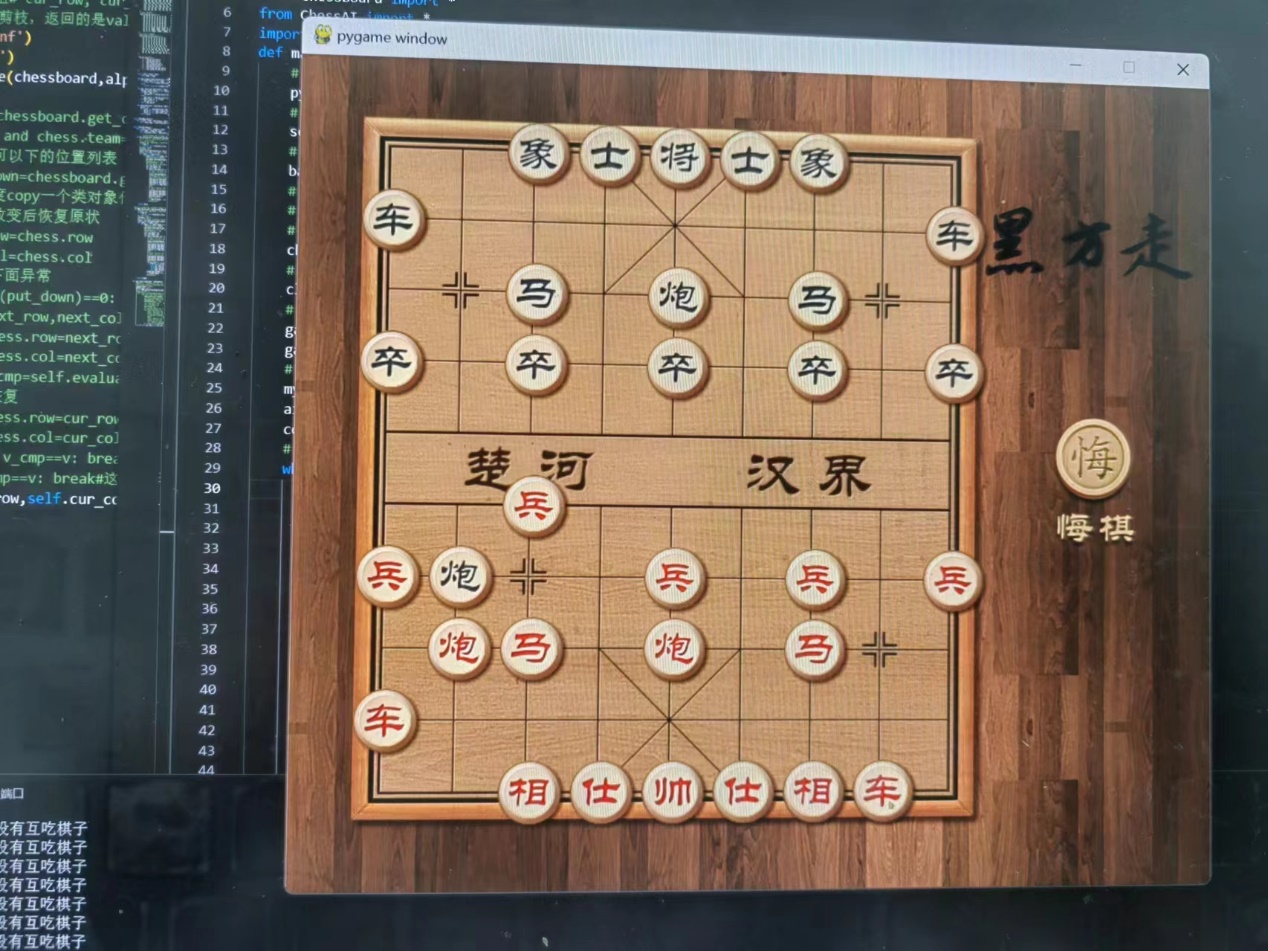
第一局：黑棋为my\_ai，先手，获胜积分为3



第二局：my\_ai,黑棋，后手，积分为6



运行时相关截图：





2. 评测指标展示及分析（机器学习实验必须有此项，其它可分析运行时间等）

* My\_ai和内置ai都各搜索五层时：两局所需时间：约14分钟：My\_ai平均一步时间为40s
* 当my\_ai搜索六层而内置ai搜索五层时：效果和my\_ai搜索五层几乎一样，但是用时更长
* My\_ai搜索七层而内置ai搜索五层时：my\_ai响应时间过久，一步响应四五分钟都没有结果

# 参考资料

**课程ppt**