

黑白棋程序报告

学号: 2310764

姓名: 王亦辉

一、问题重述

(简单描述对问题的理解, 从问题中抓住主干, **必填**)

实现用蒙特卡洛树搜索策略下黑白棋的 AI。该 AI 在决定每次下的位置时, 会进行一定次数的采样并选择效果最好的下法, 这与 minimax 或者 alpha-beta 剪枝是完全不同的想法, 大大降低了计算复杂度。

二、设计思想

(所采用的方法, 有无对方法加以改进, 该方法有哪些优化方向(参数调整, 框架调整, 或者指出方法的局限性和常见问题), 伪代码, 理论结果验证等... **思考题, 非必填**)

三、代码内容

(能体现解题思路的主要代码, 有多个文件或模块可用多个"===="隔开, **必填**)

进行蒙特卡洛树搜索的选择、扩展、模拟、反向传递。

```
def uct(self, max_times, root):
    for i in range(max_times):
        selected_node = self.select(root)
        leaf_node = self.extend(selected_node)
        reward = self.stimulate(leaf_node)
        self.backup(leaf_node, reward)
    max_node = None
    max_ucb = -sys.maxsize
    for child in root.children:
        child_ucb = child.get_ucb(self.ucb_param)
        if max_ucb < child_ucb:
            max_ucb = child_ucb
            max_node = child
    return max_node.action
```

计算 UCB 值。

```
def get_ucb(self, ucb_param):
    if self.visits == 0:
        return sys.maxsize
```

```
explore = math.sqrt(2.0 * math.log(self.parent.visits) / float(self.visits))
now_ucb = self.reward/self.visits + ucb_param * explore
return now_ucb
```

四、实验结果

(实验结果, **必填**)

=====
先后手:黑棋先手; 棋局难度:中级。黑棋获胜, 领先棋子数: 4

五、总结

(自评分析(是否达到目标预期, 可能改进的方向, 实现过程中遇到的困难, 从哪些方面可以提升性能, 模型的超参数和框架搜索是否合理等), **思考题, 非必填**)