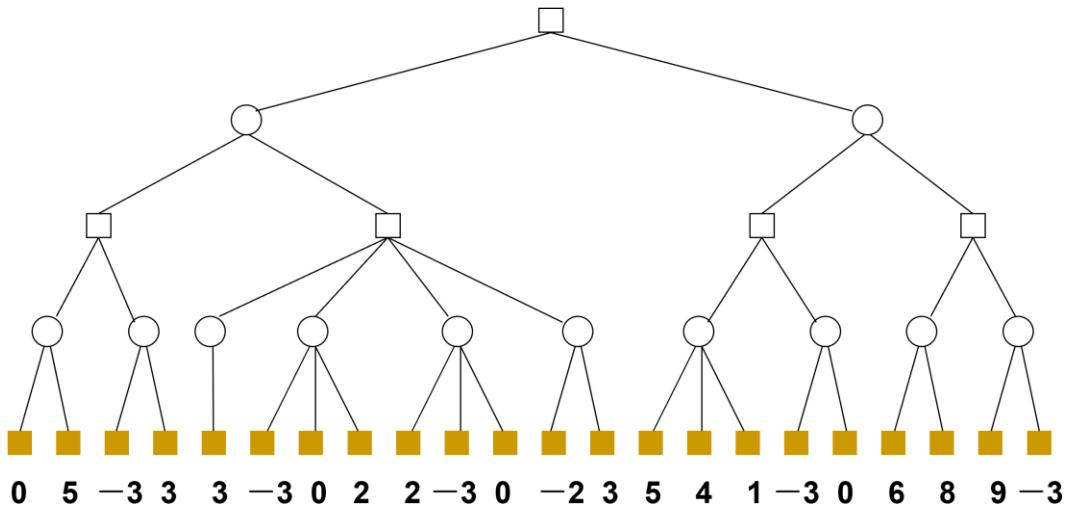


项目二、六子棋对弈机器人

一、 α - β 剪枝

下图中的树结构以结点三元组(结点序号,父节点序号,结点估值)的形式,存放于 tree.txt 文本文件中。



1. 请编写一个程序，输出以下内容：

第一行是搜索得到的走步及其估值，用三个空格分隔的数字表示。如 1 3 1；表示从结点 1 到 3 的走步，其估值为 1。

后面若干行，代表剪掉的枝子。每行前面两个数字分别代表被剪掉的枝子的父结点和子结点，如果是 α 剪枝，则两个数字后面是字符串 alpha，否则是 beta。数字和字符串之间都用空格分隔。如

5 6 alpha

10 11 beta

10 12 beta

10 13 beta

• • • • •

2. 实验报告：请融入到项目研发报告中的 α - β 剪枝的介绍中。

二、随机棋手-V0

使用博弈机器人开发平台，开发以下三个不同随机策略的机器人。

走法 1: 两个子的位置均通过随机掷骰子的方式确定，在整个棋盘（19*19）范围内掷骰子。

走法 2: 通过随机掷骰子确定第一个子的位置，第二个子下在与第一个子相邻的空位上，和第一个子相邻的空位最多有 8 个，要求随机选择一个空位落子，若相邻的 8 个位置都有子，则在整个棋盘上随机选择一个空位落子。

走法 3: 两个子的位置均通过随机掷骰子的方式确定，在棋盘的中心区域(13*13)随机掷骰子，若连续 10 次不中(没有找到空位)，则在整个棋盘(19*19)范围内掷骰子。

要求：① 请预估走法 2 和走法 3 哪种走法的胜率更高。② 编程实现走法 2 和走 3 分先(轮换先后手)各比 500 场，共 1000 场。根据比赛结果检验你的预估是否正确。③ 走法 2 和走法 3 可以采用先将下棋范围内的所有位置打散(洗牌)，然后一个一个取出作为落子位置，请分析这种走法的胜率会不会提高，算法的时间复杂度是否会降低。请用这种方法创建你们组的随机棋手。④ 将上述结果及其分析，写入项目二的研发报告中。

三、不断精进

1. 代码要求：

请使用课程提供的博弈树及六子棋对弈的基本框架，使用 Board, Move, Player, Game 等超类，不允许修改类的公共接口。代码要添加相应的注释，符合 Java 程序的编程规范。

2. 性能要求：

(1) **第一阶段 (V1):** 偏重于着法生成，只看一步。(a) 己方有胜着时，能够找到；(b) 设计能够快速计算威胁个数的算法，当对方威胁不多于两个时，能够防守得住；(c) 选点有一定智能；(d) 下棋过程中不会出现异常。

(2) **第二阶段 (V2):** 偏重于盘面估值、盘面更新及博弈树搜索。(a) 能够快速找出当前局面下双方的“有效路”，并能对局面进行快速评估；(b) 改进着法生成策略，能够高效展开博弈树搜索，进行正确的 α - β 剪枝。

(3) **第三阶段 (V3):** 能够进行“威胁空间搜索 (TBS)”或“蒙特卡洛树搜索”，对前面阶段的程序保持高胜率。

四、项目研发报告要求：

对六子棋进行介绍；论述棋盘上“路”的定义及其对棋局评估的作用；论述 α - β 剪枝和“威胁空间搜索 (TBS)”的原理和算法实现；进行实验数据对比分析，阐明不断改进的各个算法的优缺点。重点论述博弈树搜索、路表维护、局面评估及着法生成等算法，阐明算法所使用的关键数据结构及其在算法实现和算法性能等方面的作用。

研发报告大致内容目录如下：

题目：六子棋博弈程序研究开发报告

封面

目录

- 一、计算机博弈的历史及最新进展（参考文献）
 - 二、六子棋及六子棋博弈算法研究综述（参考文献）
 - 三、我的博弈程序的主要思路及特点
 - 四、上一个版本的主要数据结构及关键算法
 - 4.1 主要数据结构
 - 4.2 关键算法及其实现（尽量用伪代码描述，忽略细节。不要直接拷贝源代码）
 - 4.2.1 盘面估值
 - 4.2.2 着法生成
 - 4.2.3 盘面更新（主要数据结构所保存数据的修改）
 - 4.2.4 博弈树搜索（如何调用盘面估值及着法生成）
 - 五、对上一个版本的改进
 - 5.1 主要数据结构的改进
 - 5.2 关键算法的改进
 - 5.3 威胁空间搜索或蒙特卡洛搜索等其他改进（可选）
 - 5.4 改进前后博弈搜索效率及胜率的对比（实验数据及性能指标相关图表）
 - 六、总结与展望
 - 七、参考文献
- 附录：人工智能课程感言