**电 子 科 技 大 学**

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

**《五子棋实验报告》**

**BACHELOR THESIS**



**学 院 计算机科学与工程学院**

**专 业 计算机类**

**学 号 2022080901023**

**作者姓名 姚文杰**

**指导教师 俸志刚**

# 实验目的

学习并了解一些棋类的算法，并尝试自己做出一款五子棋游戏出来。

# 实验内容

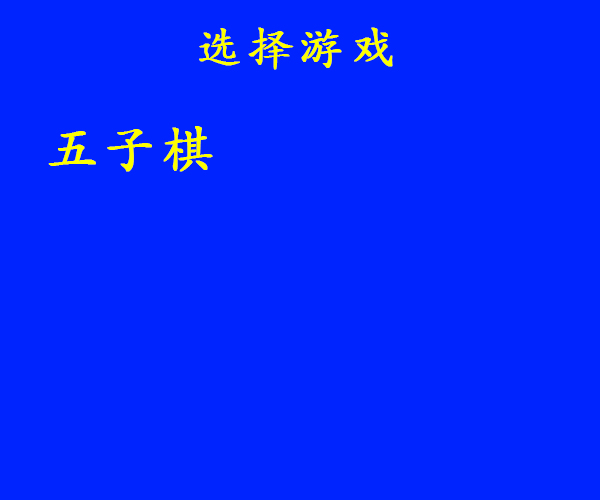
1. 利用easyx图形库构建一个可以用来下棋的平台。
2. 尝试写出玩家对抗
3. 尝试写出简单的人机对战
4. 利用极大值极小值搜索和alpha-beta剪枝实现高难度人机对战

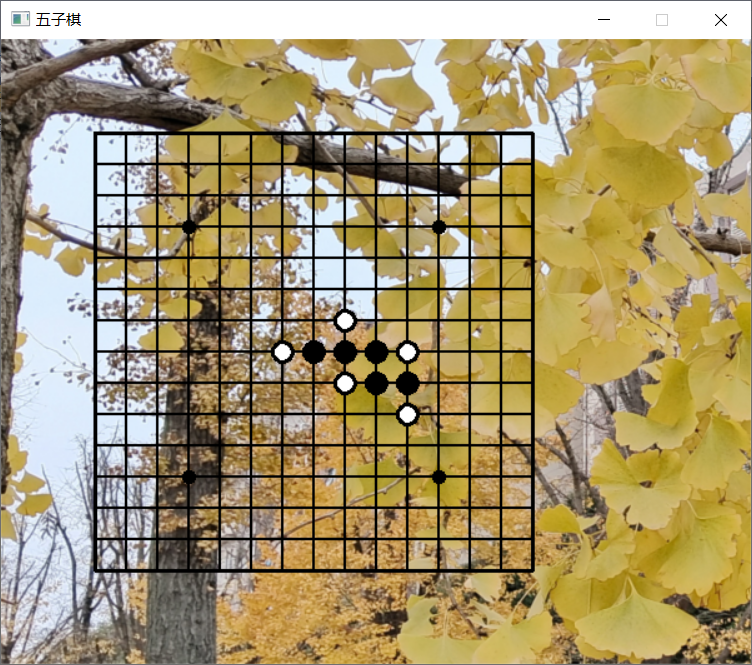
# 实验步骤

1. 问题描述

为了实现玩家对战，我的想法是让玩家用鼠标分别下子，每下完子换一个颜色，以此来实现玩家对抗。为了创建一个可以与棋手抗衡的AI，我的思路是：创建一个打分函数，对于棋盘上的每一个点，利用打分函数计算出棋盘上每个点的分数，AI选择在分数最大的地方下子，以此来达到与棋手对抗的目的。

1. 图形设计和算法分析
2. 利用easyx图形库创建窗口建立棋盘，玩家可以选择游戏（目前只有五子棋，这样的做法是因为如果以后还要做游戏可以将这些放在一起供玩家选择），五子棋对抗方式与对抗难度：





1. 玩家对抗

就如同上面所介绍，每次下完棋后切换棋的颜色，并且每次都判断是否有玩家胜利，此时还不需要利用要打分函数（依靠两位玩家的头脑进行对弈）

1. 落子打分

如果白棋走这个点

|  |  |
| --- | --- |
| 产生效果 | 评分 |
| 连2 | 20 |
| 死2 | 10 |
| 活3 | 200 |
| 死3 | 100 |
| 活4 | 2000 |
| 死4 | 1000 |
| 连5 | 30000 |

如果黑棋走这个点

|  |  |
| --- | --- |
| 产生效果 | 评分 |
| 连2 | 15 |
| 死2 | 8 |
| 活3 | 150 |
| 死3 | 80 |
| 活4 | 1500 |
| 死4 | 800 |
| 连5 | 20000 |

（4）新手难度

搜索层数为0层，即每次玩家下完子后，AI利用搜索函数搜索当层每个位置的分数，在分数最高的地方下子。对于分数高低的判断，采取分别搜索出玩家和AI棋子的最高分和其对应位置，取二者中分数最大的作为落子点。

1. 中级难度

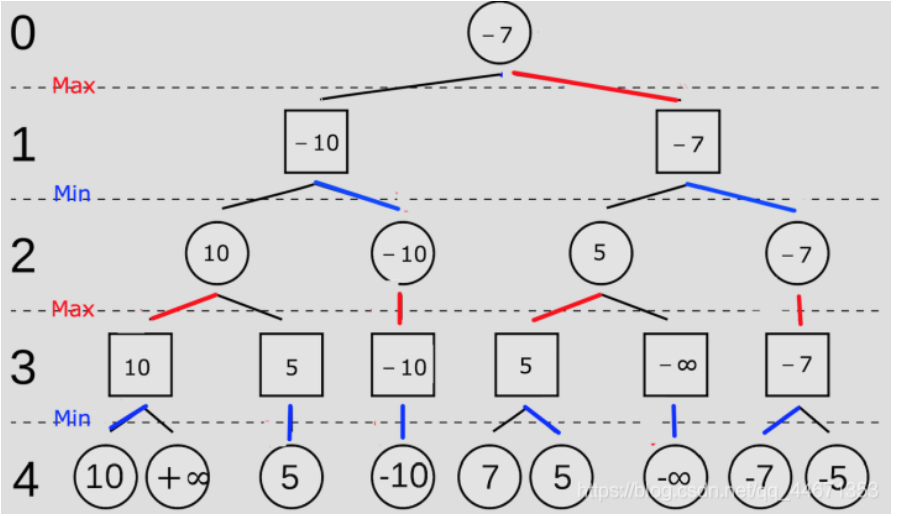
采取极大值极小值搜索，alpha-beta剪枝优化，搜索层数类似迭代加深，对于开始AI在下完3个子之前搜索层数为0层，之后变为2层。

1. 噩梦难度（谨慎选择，耗时较长）

依然采取极大值极小值搜索，alpha-beta剪枝优化。将搜索层数迭代深度加深，即AI在下完3个子之前0层，10子之前2层，随后便是4层。到了4层每次AI下子的时间明显增多。

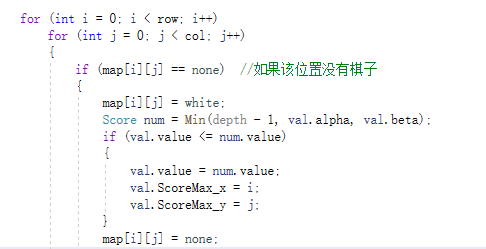
1. minmax搜索，即极大值极小值搜索

所有棋类都可类比为一颗博弈树，树中的每一个节点代表每一种局面，由于五子棋对局是公开透明的，所以我们完全能够通过博弈树计算出所有可能的情况，并且朝着对自己有利的方向下棋，因此就有了极大极小搜索。对于AI来说，它在当下局面要选择对自己最有利的点下棋，即要选出当下局面分值最高的地方。而对于人来说，人也要选择对自己有利的地方下子，所以他要选择对AI最不利的地方落子，即分最低的地方，由此就有了极大极小值搜索。对于AI而言是MAX层，玩家则是MIN层，利用递归直至搜索到叶子结点返回函数值。

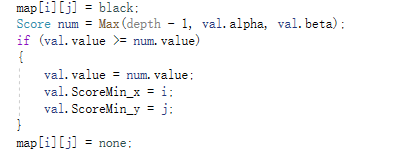


简要实现代码如下

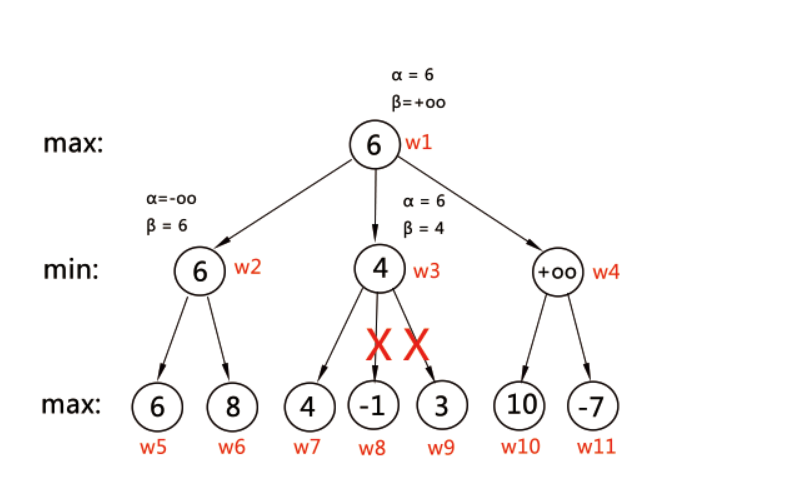
Max层：



Min层：



1. alpha-beta剪枝



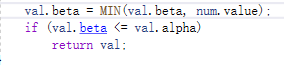
Alpha-beta剪枝在max层只改变alpha值，在min层只改变alpha值。对于max层来说，它的alpha值与beta值与上层的beta值相比较得出最小的那一个，然后将上一层的beta值修改为该值；对于min层来说，它的alpha值与beta值与上层的alpha值相比较得出最大的那一个然后将上一层的alpha值修改为该值。Alpha与beta的初始值分别为-∞的∞，然后依次传递下去，再通过上述方法返回。对于上图来说，在w3处就发生了剪枝，由于通过w7的传递，w3的beta值变为了4此时w3的alpha值大于beta值，发生剪枝，这是因为w3是min层，它会选择最小的作为自己的分数，而通过第一次判断可以得知它的分数会<=4，而w1是max层，他会选择最大的值作为自己的分数，在w2是就已经返回了6，所以w3一定不会被选择，因此发生剪枝，减去w3其他的支点，这样大大减少了计算量。

简要实现代码：

Max层：



Min层：

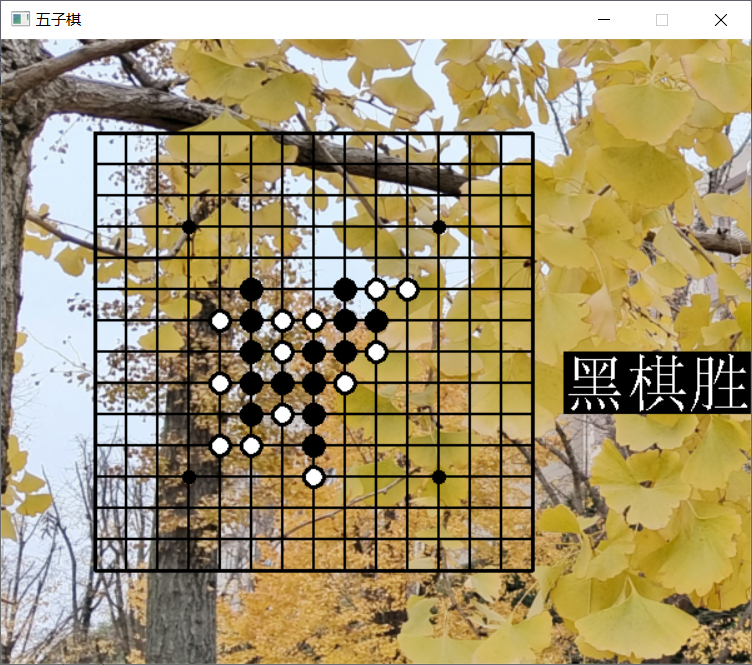


1. 实验优化

因为在刚刚开始游戏的时候，场上的棋子数目都比较少，没有必要去进行过于深度的计算，所以采取了类似于迭代加深的算法，即上述提到过的在AI落子数为5之前都采用0层搜索，随后便一步步加深搜索深度，以此来缩短时间。

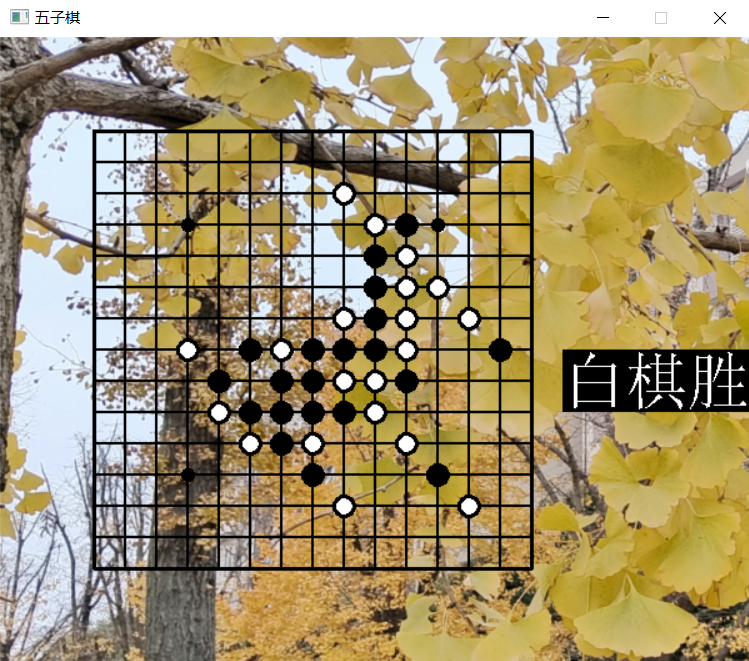
# 实验结果分析

1. 新手难度



可以对付一些对于五子棋棋法了解不深的玩家，但对于一般的棋手而言还是可以战胜新手难度的。

1. 中等难度



下子速度极快，可以控制在一秒左右，中等难度相对于普通难度有了很大的提升，在与一般的棋手玩家PK的时候可以压制一头，甚至可以出其不意地落下神之一手。在与棋力较高的棋手PK过程中可以下的有来有回。