##### 博弈五子棋AI实验

# 一、 实验背景

## 1.1 基础知识

1. 零和博弈

又称零和游戏，与非零和博弈相对，是博弈论的一个概念，属非合作博弈。指参与博弈的各方，在严格竞争下，一方的收益必然意味着另一方的损失，博弈各方的收益和损失相加总和永远为“零”，双方不存在合作的可能。

1. 极大极小算法

局面估价函数：我们给每个局面规定一个估价函数值 f，评价它对于己方的有利程度。胜利的局面的估价函数值为正无穷，而失败的局面的估价函数值为负无穷。

Max 局面：假设这个局面轮到己方走，有多种决策可以选择，其中每种决策都导致一种子局面（sub-state）。由于决策权在我们手中，当然是选择估价函数值 f 最大的子局面，因此该局面的估价函数值等于子局面 f 值的最大值，把这样的局面称为 max 局面。

Min 局面：假设这个局面轮到对方走，它也有多种决策可以选择，其中每种决策都导致一种子局面（sub-state）。但由于决策权在对方手中，在最坏的情况下，对方当然是选择估价函数值 f 最小的子局面，因此该局面的估价函数值等于子局面 f 值的最小值，把这样的局面称为 max 局面。

终结局面：胜负已分（假设没有和局）。

## 1.2实验内容

利用极大极小搜索实现一个零和博弈的五子棋AI，需要用到启发式搜索方法，评估函数，alpha-Beta剪枝策略等。

## 1.3 参考程序

## <https://github.com/marblexu/PythonGobang>

## <https://github.com/colingogogo/gobang_AI>

## <https://github.com/Kali-Hac/Gomoku-AI>

# 设计思路

棋类博弈AI的设计需要建立统一的棋面评估模型，并评价模型的性能，其中，棋面评估模型包括每一步棋的评价指标和第m步棋的评价指标，需要利用特征工程将n个方向的模式组合，每个匹配到的模式的优劣评估分数需要加权处理。

这里使用博弈树的Min-Max搜索，递归遍历双方下棋的可能，简化合并，评价取反互用。为了大大减小程序的复杂度，采用alpha-beta剪枝算法，将小于beta的分支删除，大于alpha的分支保存。进行简化，使用半径合适的闭包搜索范围，基本覆盖了最佳可能的范围。为了提高性能，使用更简洁的最优化贪心算法构造五子棋程序，即识别双方处境，攻大于守则攻，守大于攻则守。

# 实验环境及运行方式

## 3.1实验环境

Python 3.7

## 3.2运行方式

① 把graphics.py复制到python安装目录下的Lib文件夹中；

② 运行Main.py文件即可进入游戏。

# 实验结果

程序实现了两种难度的AI，分别为简单和困难，同时也设计了人人对战、人机对战、机器对战模式和机器辅助手段，极大的丰富了游戏的趣味性。

对于实现的AI的性能，简单难度的AI在和我的对弈过程中，10战5胜5平，困难难度10战9胜1平，可以说有非常高的智能型，同时，AI的效率也很高，可以及时的对局势做出反应和应对。

缺陷：由于棋盘较小，限制了AI的发挥，增大棋盘网格大小可以使AI的性能获得很大的提升。