《人工智能及应用》

实验报告

姓 名： 王浩

学 号： 2020204246

专业班级： 计算机科学技术学院图灵班

指导教师： 杨环

提交日期：

实验一：

实验题目利用α-β搜索的博弈树算法编写一字棋游戏

1. 实验目的及要求

理解和掌握博弈树的启发式搜索过程，能够用选定的编程语言实现简单的博弈游戏。

(1) 学习极大极小搜索及α-β 剪枝。

(2) 利用学到的算法实现一字棋

## 实验内容

#### 2.1 游戏规则

"一字棋"游戏（又叫"三子棋"或"井字棋"），是一款十分经典的益智小游戏。"井字棋 " 的棋盘很简单，是一个 3×3 的格子，很像中国文字中的"井"字，所以得名"井字棋"。"井 字棋"游戏的规则与"五子棋"十分类似，"五子棋"的规则是一方首先五子连成一线就胜利； "井字棋"是一方首先三子连成一线就胜利。 井字棋（英文名 Tic-Tac-Toe） 井字棋的出现年代估计已不可考，西方人认为这是由古罗马人发明的；但我们中国人 认为，既然咱们都发明了围棋、五子棋，那发明个把井字棋自然是不在话下。这些纯粹是口 舌之争了，暂且不提。

#### 2.2 极小极大分析法

设有九个空格，由 MAX，MIN 二人对弈，轮到谁走棋谁就往空格上放一只自己的棋子， 谁先使自己的棋子构成"三子成一线"(同一行或列或对角线全是某人的棋子)，谁就取得了胜 利。 用圆圈表示 MAX，用叉号代表 MIN。 比如左图中就是 MAX 取胜的棋局。 估价函数定义如下： 设棋局为 P，估价函数为 e(P)。 (1) 若 P 对任何一方来说都不是获胜的位置，则 e(P)=e(那些仍为 MAX 空着的完全 的行、列或对角线的总数)-e(那些仍为 MIN 空着的完全的行、列或对角线的总数) (2) 若 P 是 MAX 必胜的棋局，则 e(P)＝+∞ （实际上赋了 60）。 (3) 若 P 是 B 必胜的棋局，则 e(P)＝-∞ （实际上赋了-20）。 比如 P 如下图示,则 e(P)=5-4=1 需要说明的是，+∞赋 60，-∞赋-20 的原因是机器若赢了，则不 论玩家下一步是否会赢，都会走这步必赢棋。 《人工智能及其应用》实验指导书 6

#### 2.3 α -β剪枝算法

上述的极小极大分析法，实际是先生成一棵博弈树，然后再计算其倒推值，至使极小极 大分析法效率较低。于是在极小极大分析法的基础上提出了α-β 剪枝技术。 α -β 剪枝技术的基本思想或算法是，边生成博弈树边计算评估各节点的倒推值，并且 根据评估出的倒推值范围，及时停止扩展那些已无必要再扩展的子节点，即相当于剪去了博 弈树上的一些分枝，从而节约了机器开销，提高了搜索效率。 具体的剪枝方法如下：

(1) 对于一个与节点 MIN，若能估计出其倒推值的上确界 β，并且这个 β 值不大于 MIN 的父节点(一定是或节点)的估计倒推值的下确界 α，即 α≥β，则就不必再扩展该 MIN 节点的其余子节点了(因为这些节点的估值对 MIN 父节点的倒推值已无任何影响了)。这一 过程称为 α 剪枝。

(2) 对于一个或节点 MAX，若能估计出其倒推值的下确界 α，并且这个 α 值不小于 MAX 的父节点(一定是与节点)的估计倒推值的上确界 β，即 α≥β，则就不必再扩展该 MAX 节点的其余子节点了(因为这些节点的估值对 MAX 父节点的倒推值已无任何影响 了)。这 一过程称为 β 剪枝。 从算法中看到： (1) MAX 节点(包括起始节点)的 α 值永不减少； (2) MIN 节点(包括起始节点)的 β 值永不增加。 在搜索期间，α 和 β 值的计算如下： (1) 一个 MAX 节点的 α 值等于其后继节点当前最大的最终倒推值。 (2) 一个 MIN 节点的 β 值等于其后继节点当前最小的最终倒推值。

#### 2.4 输赢判断算法设计

因为每次导致输赢的只会是当前放置的棋子,输赢算法中只需从当前点开始扫描判断是 否已经形成三子。对于这个子的八个方向判断是否已经形成三子。如果有，则说明有一方胜 利，如果没有则继续搜索，直到有一方胜利或者搜索完整个棋盘。

## 3.实验结果及分析

Please enter 1 to select human first or enter 0 to select AI first

1

You have chosen human first!

-----

| |

-----

| |

-----

| |

-----

\*\*\*\*\*\* YOUR ROUND \*\*\*\*\*\*

Please enter the position you want place

Such as: 1,2 means you want to place a piece in the 1st row and 2nd column

1 1

Successful placement

-----

O| |

-----

| |

-----

| |

-----

\*\*\*\*\*\* AI ROUND \*\*\*\*\*\*

-----

O| |

-----

|X|

-----

| |

-----

\*\*\*\*\*\* YOUR ROUND \*\*\*\*\*\*

Please enter the position you want place

Such as: 1,2 means you want to place a piece in the 1st row and 2nd column

3 1

Successful placement

-----

O| |

-----

|X|

-----

O| |

-----

\*\*\*\*\*\* AI ROUND \*\*\*\*\*\*

hh,You wish

-----

O| |

-----

X|X|

-----

O| |

-----

\*\*\*\*\*\* YOUR ROUND \*\*\*\*\*\*

Please enter the position you want place

Such as: 1,2 means you want to place a piece in the 1st row and 2nd column

2 3

Successful placement

-----

O| |

-----

X|X|O

-----

O| |

-----

\*\*\*\*\*\* AI ROUND \*\*\*\*\*\*

-----

O|X|

-----

X|X|O

-----

O| |

-----

\*\*\*\*\*\* YOUR ROUND \*\*\*\*\*\*

Please enter the position you want place

Such as: 1,2 means you want to place a piece in the 1st row and 2nd column

3 2

Successful placement

-----

O|X|

-----

X|X|O

-----

O|O|

-----

\*\*\*\*\*\* AI ROUND \*\*\*\*\*\*

hh,You wish

-----

O|X|

-----

X|X|O

-----

O|O|X

-----

\*\*\*\*\*\* YOUR ROUND \*\*\*\*\*\*

Please enter the position you want place

Such as: 1,2 means you want to place a piece in the 1st row and 2nd column

1 3

Successful placement

-----

O|X|O

-----

X|X|O

-----

O|O|X

-----

It's a tie!

实验分析：

首先让玩家选择玩家或者AI谁先手，

然后进入循环

每次下棋之前，先通过函数判断游戏是否结束，如果结束，break跳出循环

如果未结束，进入AI 回合或玩家回合

AI回合通过递归，循环迭代，通过MAX-MIN极大极小算法，找到最有可能赢得下棋位置，进行下棋。

4.实验中存在的问题及解决方法

存在的问题：

启发函数的设计方式可能存在多种方式。

解决方法：

通过上网查找相关资料，以及了解相关数学公式推导，采用更合理的启发函数。

5.代码实现（单独文档）

实验二：

实验题目

1.实验目的及要求

2.实验内容

3.实验结果及分析

4.实验中存在的问题及解决方法

5.代码实现（单独文档）