|  |
| --- |
| **基于人工智能的井字棋小游戏** |
|  |
| **董潇宇 北京交通大学** |
| **蒋逸 北京交通大学** |
| **陈鲽晴 北京交通大学** |
|  |
|  |

Abstract

井字棋是生活中非常常见的一类小游戏，本文将介绍由python编程软件实现的基于敌对搜索算法的井字棋小游戏。该程序可以通过计算实现得到任意棋盘落子情况下的最佳落子点，设计了三种游戏结束形式（赢、输、平局），并且支持重复进行游戏，以及选择棋子类型，该程序成功实现了基于人工智能算法的井字棋小游戏所需要的所有功能。

关键字：井字棋、python、敌对搜索、人工智能

Introduction

本次课程中，小组被分配在学习ai的分组之中。在陈一帅老师以及郭宇春老师的指导下，小组着眼于搜索的相关算法并且由此对人工智能的相关只是产生了极大的兴趣。因此在小组的大作业中，小组决定利用所学在老师选定的框架下选择井字棋的ai游戏开发项目作为本次课程的大作业。

井字棋的ai游戏开发的大作业是在CS50ai的下辖项目。该项目需要在python语言环境下开展，且需要先行安装第三方库pygame。项目已经提前给出了相关的主题程序（runner.py）代码以及调用的函数集（tictactoe.py）的相关代码。而大作业的主要任务就是补充函数集代码中调用到的相关的函数。

值得注意的是，程序的编写开始前需要对主体函数对函数集的相关调用进行了解，明确函数的具体调用过程以及所期望得到的具体返回值，由此才能保证函数集能够满足主体函数的调用需求。

代码函数解析（一）

这一节主要是对井字棋实现代码的解析，分别对各自完成的代码部分进行讲解。由董潇宇负责完成。

基础九宫格生成函数

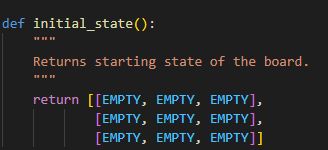


图1基础九宫格生成函数

创建一个3\*3矩阵，其中元素都为空，即创建棋盘

* 1. 判断玩家回合

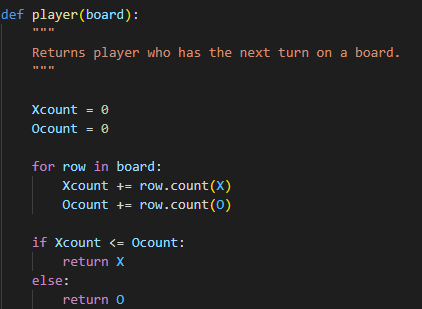


图2判断玩家回合

主要是对矩阵的每一行元素逐行进行累计，以此来判断当前回合为X棋子还是O棋子。

用到了count函数，统计字符串里某个字符出现的次数。

Ps:count函数还可以选择字符串索引的起始位置和结束位置。

* 1. 检索棋盘中的空位函数

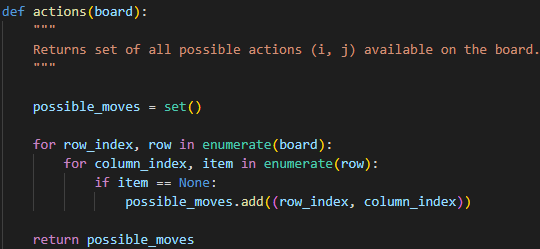


图3检索棋盘中的空位函数

主要对每行每列的元素进行判断，判断该位置是否为None，若为None则记下当前位置，表示该位置没有棋子。

用到了enumerate函数，能够更简便地访问到索引位置和对应的元素。

当前回合玩家落子函数

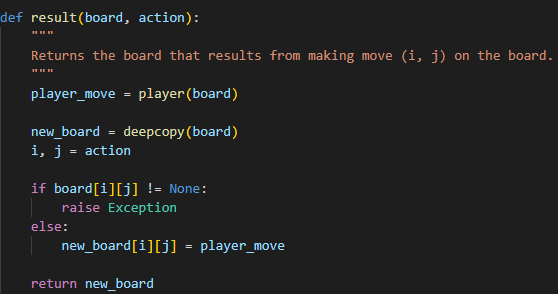


图4当前回合玩家落子函数

完成了将棋子下入棋盘的功能，先获得原棋盘，然后将棋盘复制，并获得当前玩家落子位置，与棋盘上对应位置比较，若棋盘上对应位置不为空，则不执行。若为空，将对应位置的元素赋值为玩家的移动，即为落子。

总结与收获

本次任务完成过程中我的任务较少，其主要时间和精力还是花费在python的学习上，因为之前没有过python的学习，所以大部分时间还是花费在了python的学习上。虽然之前学习过C语言，但是python的程序编写和C语言还是有较大的不同的。刚开始，我只是学习了python的基础部分，简单的过了一遍，但是在编程过程中，却出现了很多问题，很多编程语句不对，且学习的视频中较多的都是对于编程思路以及概念的讲解，对于程序编写还是介绍的较少，同时缺乏相应的训练，且编程环境与教学视频中的不同，所以刚开始遇到的问题还是很多的。所以由于并不清楚编写程序具体需要哪一部分的知识，所以还是对python进行了较为系统的学习。主要是对比学习，边学习边比较python和C语言编程的异同，避免编程过程中一些语句使用C语言中的语法很难检查出来的错误。

在编程中遇到的问题就是，起初不熟悉python的语法，虽然看视频进行了学习，但是没有经过训练，所以印象不深，导致编程的时候经常会出现语法错误。其次就是编程的时候由于是因为刚刚接触python还是不可避免的会出现一些问题，例如许多内置函数了解的不够多，还是需要查找资料和询问同学，来找到合适的解决办法。例如上述图片中的deepcopy函数就是问了同学之后才知道的，之前的编写就是创建一个新的数组，然后就直接用了赋值，结果老是出问题，在询问同学之后才知道需要用deepcopy这个语句。原因是因为深复制，是从输入变量完全复刻一个相同的变量，无论怎么改变新变量，原有变量的值都不会受到影响。与等号赋值不同，等号复制类似于贴标签，两者实质上是同一段内存。所以利用原先的方法还是两个相同的变量进行比较。

代码函数解析（二）

这一节主要针对棋局的情况进行判断，主要包含了对棋局是否结束的判断、对于棋局胜利方执子的判断以及对和棋的判断，同时对于每种情况都予以了相应的返回值。该部分的核心函数是winner函数，另外两个函数都是在winner函数的返回值的基础上进行的。该部分由蒋逸进行完成。

## **胜利者判决函数**

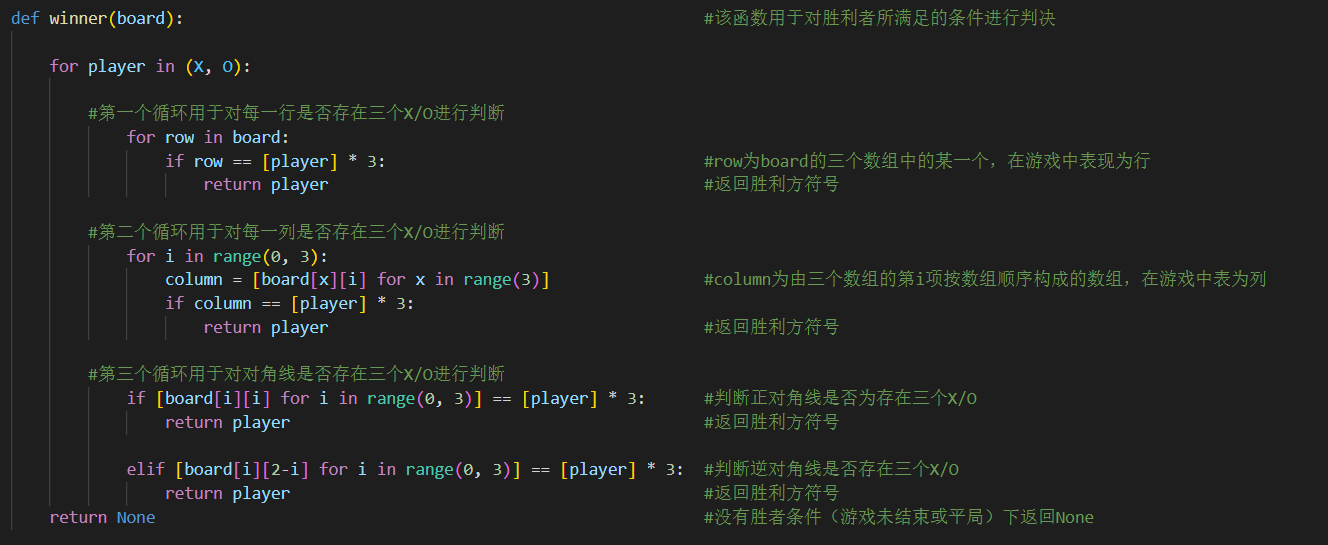


图5胜利者判决函数

实现了对于棋局胜利者的判定，同时会返回胜利者所执棋子。但是在平局或棋局未结束时会返回None的函数值。棋局的具体情况需要后续的函数进行判决。该函数只负责检验并返回胜利者。

## **对局结束判决函数**

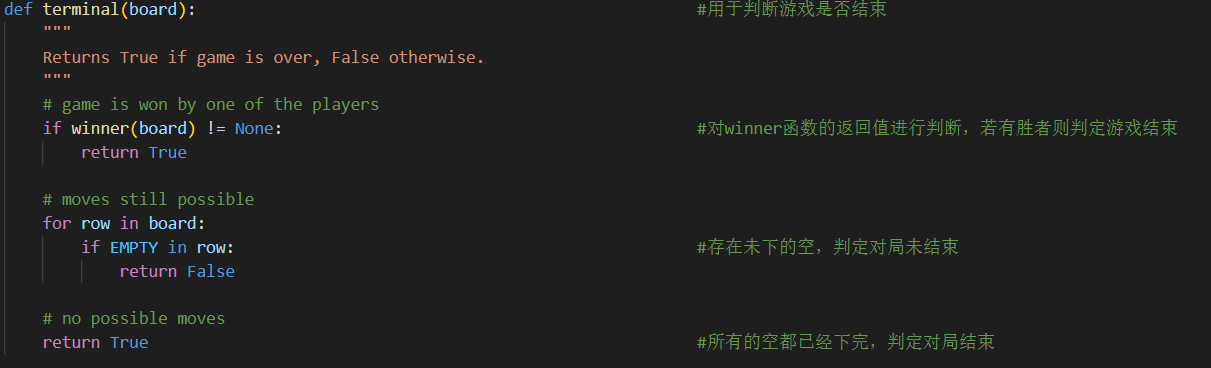


图6对局结束判决函数

此函数为对局结束判决函数，主要用于对棋局的结束情况进行判定。该函数需要调用到胜利者判决函数，对于胜利者判决函数的返回值进行判定。在确定有棋局胜利者或者检测到棋局中不再存在其余空可下时判定棋局结束，反之判定未结束。

## **Winner返回值判决函数**

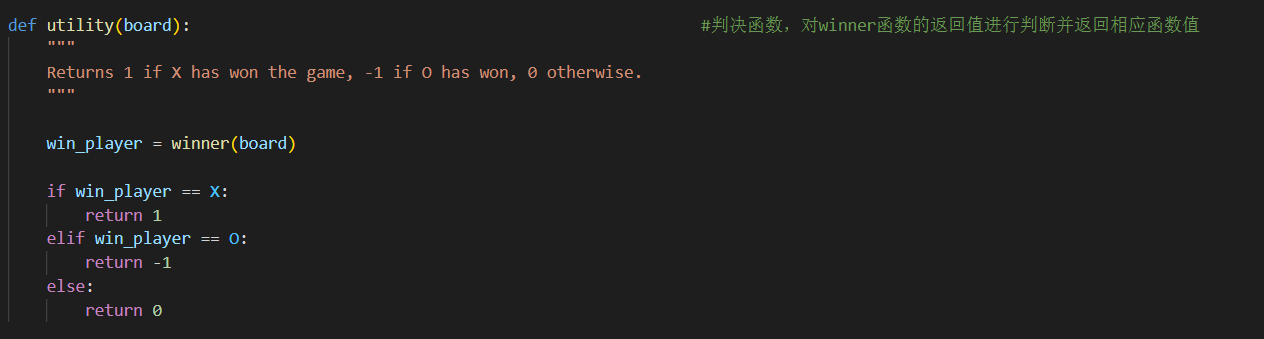


图7Winner返回值判决函数

此函数主要针对winner函数的返回值进行判决。对winner函数的返回值进行判定后将其以具体数字的形式进行返回，以便于后学函数的接受以及进行相关的判决。

## **总结与收获**

本次任务的过程中最大的困难在于对python的学习。由于课程的时间安排较为紧张，导致了对于python的学习进度较慢、对于程序的整体开发进度较慢。Python的使用中，存在与学习过的C语言的语法中较大的相似之处，这对于python的学习既是方便之处也是难点所在。同时学习的教材中，对于具体的开发的详细介绍较少，以相关的程序编写思路为主，而这对于编程实际操作的学习是不利的。本次程序开发过程中最大的难点在于对棋盘上各值的判决。而究其本质，其关键还是在python中对于数组的利用与修改。程序的总体开发难度不是很大，最后的成果也充分让人感受到了编程在一定程度上的趣味性与成就感。在程序的实际编写过程中还存在着诸多问题，例如由于疫情只能线上交流，队友之间的配合存在有一定的问题，项目的开发也存在有一定的混叠。所幸通过小组成员之间的反复耐心交流，项目得以按期完成。

代码函数解析（三）

陈鲽晴负责的该部分代码是敌对搜索的minimax函数，主要就是max\_value和min\_value两个子函数的双重递归：

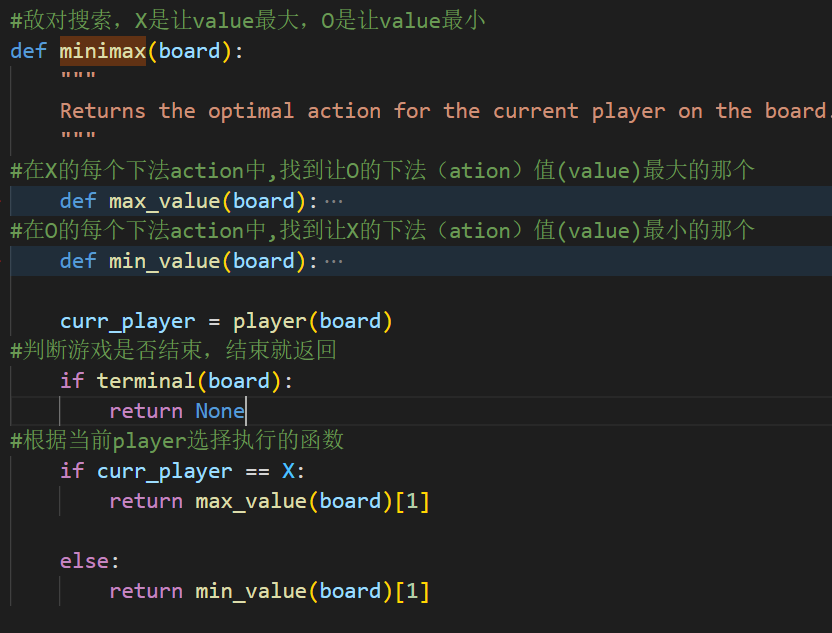


图8敌对搜索函数

X是让value最大，O是让value最小

根据老师上课给的伪代码编写的：

1. Max\_value（）函数：

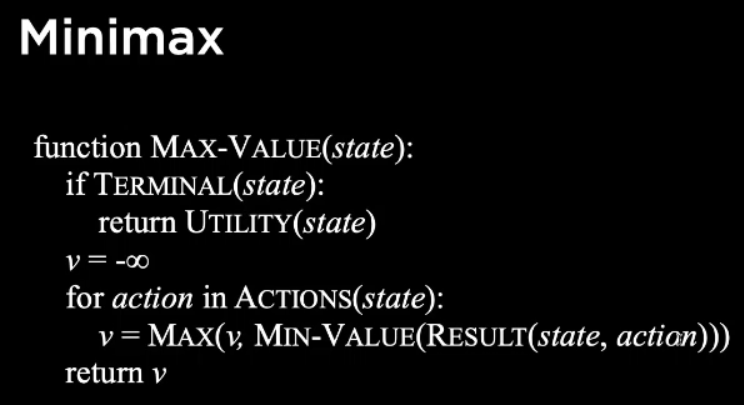


图9minimax函数伪代码

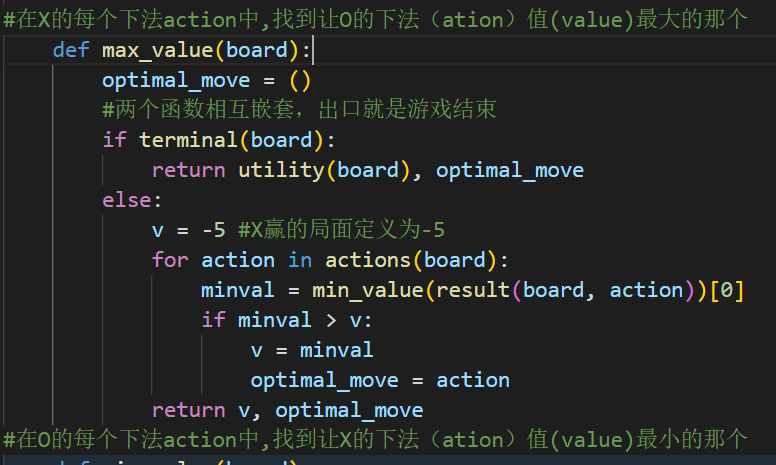


图10max\_value函数

先调用terminal()函数检查游戏是否结束，结束就返回效用函数utility()的返回值和最佳的走法。没结束就双重递归。对每个X的走法（action），O都可能有很多下一步走法，X会选择O的所有走法min\_value最大的那个。最后就返回最大值v和最佳走法option\_move。

1. min\_value（）函数：

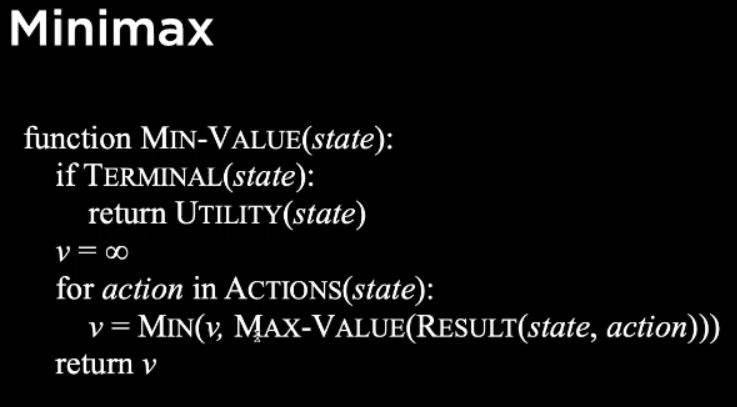


图11mini\_value函数伪代码

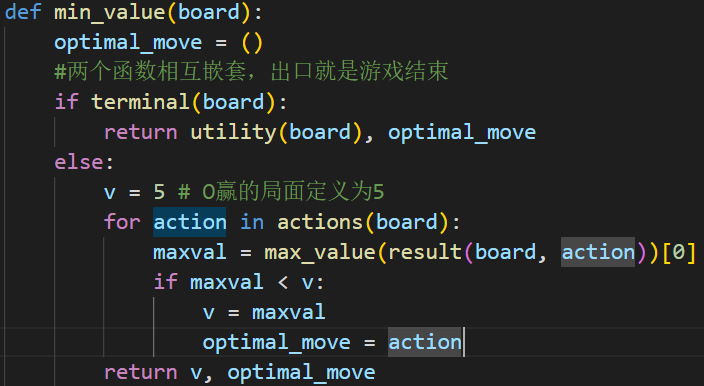


图12min\_value函数

同样对O的每个走法，X也可能有很多走法，O会选择X的所有走法max\_value最小的那个，于是就用到了双重递归。递归的终点就是游戏结束。

1. 总结与收获

这次主要负责敌对搜索（minimax）的代码部分。首先是学习python，从b站上找的播放量比较高的视频大概学了一遍，因为有C语言的基础，而且此项目不会涉及面向对象的部分，所以语法的理解还是比较快的。然后是敌对搜索算法的学习理解，因为老师上课讲过，理论上的敌对搜索（minimax）还是比较好理解的，但应用到实际项目中，比如本项目井字棋中，python语言实现上要用到双重递归，就很难理解，编写起来也比较困难。主要还是是对python语言的不熟悉，即使有老师给的伪代码，编写成python代码也花了很长时间。虽然代码上不是很理解，但是照着老师给的伪代码和上课讲的思路，还是勉强编写了出来，测试也行得通。其中遇到了很多困难，比如还是会习惯性以“;”结束每个语句导致运行不通过。还有就是和伙伴之间的沟通协作了，因为是一个项目，每个人都有不同的写代码习惯，加上疫情的关系联系不是很方便，统一变量名称等代码风格也挺费劲。经过大家的合作探讨，我们的项目最终还是勉强完成了，可能有很多瑕疵，但大家在完成这个项目的过程中都学到了很多，特别是对python语言的掌握。

程序运行图

初始界面，选择玩家，X先走，O后走



图13程序初始界面

选择X后，先走一步，接着ai会走下一步：

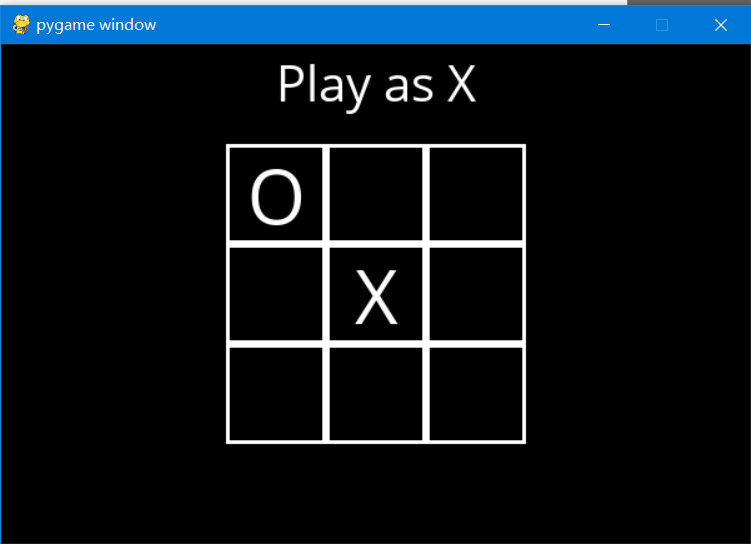


图14最初步棋局

直到对局结束：



图15最终棋局（电脑胜利）

项目总结

本次的井字棋项目是在老师讲解和给了大部分代码的基础上完成的，主要就是对9个函数的代码实现，最后成功实现了与电脑下井字棋的功能。

Git链接

https://github.com/dddxy123/dxy-cdq-jy-team/tree/main

B站视频演示链接

https://www.bilibili.com/video/BV1fB4y197ZQ?pop\_share=1