东　北　大　学

机器人原理及应用

课程名称：　机器人原理及应用

设计题目：　机器人下棋

指导教师：　赵姝颖、孙浩、王泓潇

小组成员：　张硕、张欣宇、姜滔

摘要

我们小组将从下棋机器人概况、Alpha Zero-Hex 算法介绍、汉语言版块实现、教学测四个方面讲述了本次《机器人下棋》课程设计的内容。本次设计通过汉语言软件与 Python 混合编程的方式，智能化改造了 AlphaGo Zero 模型，使用蒙特卡洛树搜索构建算法模型，采用深度优先算法编写海克斯棋游戏规则，并搭建了以汉语言为界面的功能成熟的海克斯棋完整小程序，并制作完整教学测，系统介绍了海克斯棋完整内容以及小程序使用手册。

关键词：**AlphaGo Zero 蒙特卡洛树搜索 深度优先算法 混合编程**

Our group will discuss the content of the course design for 'Robot Chess' from four aspects: an overview of chess robots, an introduction to the Alpha Zero Hex algorithm, implementation of the Chinese language section, and teaching testing. In this design, the AlphaGo Zero model is intelligently transformed by means of mixed programming of Chinese language software and Python, the algorithm model is built by using Monte Carlo tree search, the rules of the game of Hicks are written by using the depth first algorithm, and a fully functional small program of Hicks is built with the Chinese language as the interface, and a complete teaching test is made. The complete content of Hicks and the manual for using the small program are systematically introduced.

Keywords: **AlphaGo Zero、Monte Carlo tree search、depth first algorithm、hybrid programming**

第 1 章 概述

1.1 背景介绍

Hex 棋的起源可以追溯到 20 世纪 50 年代。当时，丹麦数学家 Piet Hein 和美国数学家 John Nash 独立发明了这个游戏，他们都是游戏理论研究的专家。它最初是在丹麦一家报纸上发表的。

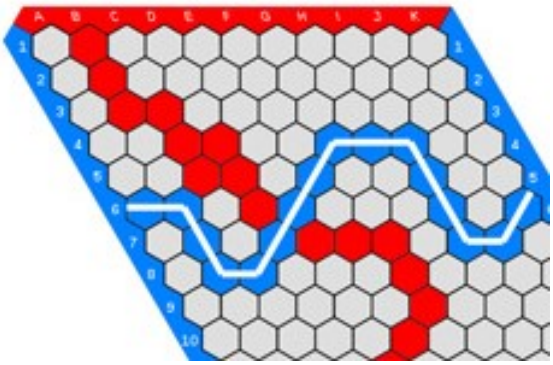
Hex 棋的游戏板是一个六边形格子，棋子通常是黑色和白色。每个玩家轮流将自己的棋子放置在空格子上，试图建立一条从一个边缘连接到另一个边缘的路径，同时阻止对手完成同样的任务。

因为这个游戏使用六边形的棋盘，它比其他传统的棋盘游戏（如国际象棋和五子棋）有更多种类的移动和策略，这使得游戏更加复杂。游戏的力量在于其十分简单的规则，但却有无数的策略和变化，因此 Hex 棋的深度和丰富性使得它成为了研究游戏理论的重要对象，也被广泛运用于人工智能和计算机科学的研究中。

此外，Hex 棋作为一种没有元素偏向和非准拟合（non-paraimitic）的完全信息博弈（two-player full-information game），也因其特殊的游戏特性（如每个玩家有不同的胜利目标）而成为学术界研究的热门对象。近年来，在 Hex 棋上的 AI 分析也逐渐成为了学者们关注的重点。

1.2 规则介绍

六贯棋是在六边形格的棋盘上玩的图版游戏，亦是数学游戏，通常使用 10 乘 10 或 11 乘 11 的菱形棋盘。 六贯棋由两个人一起玩，有两种颜色，通常是红、蓝或黑、白。四个边平行填上两方的颜色。双方轮流下，每次占领一处空白格，在空白格放上自己颜色的棋子（或填上自己的颜色）。最先将棋盘属于自己的颜色的边连成一线的一方为胜。



第 2 章 程序设计

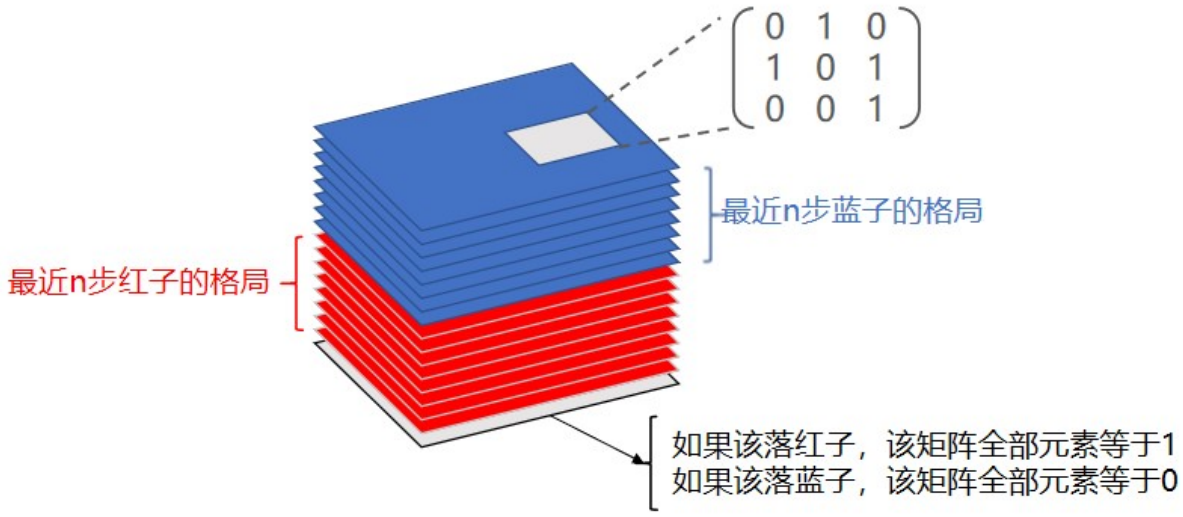
2.1 算法介绍

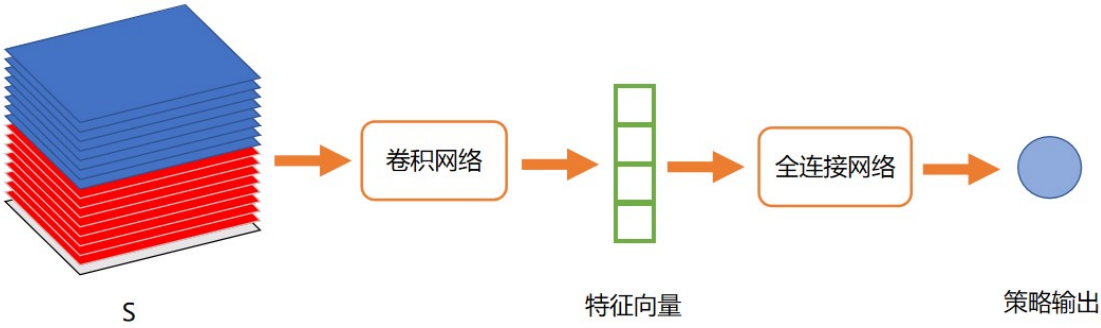
和人类下棋类似，AlphaGo 在做决策前，需要在“大脑里”做预判，确保几步以后很可能会占优势。如果只根据当前格局做判断，不往前看，是很难获胜的。

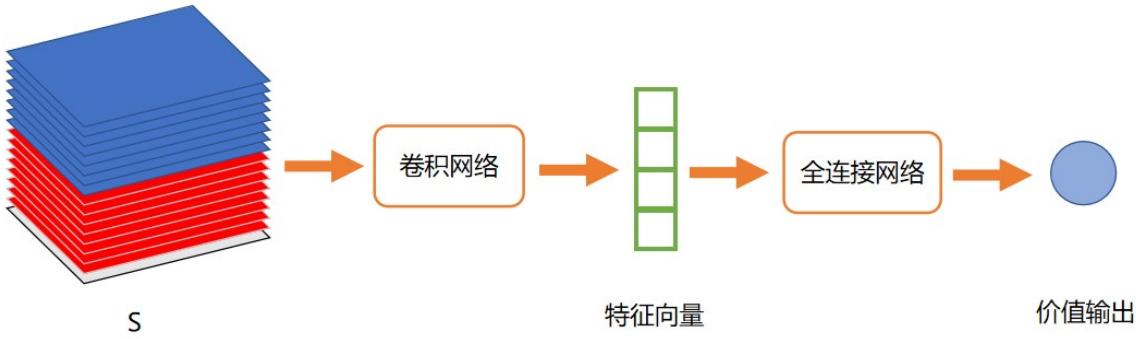
AlphaGo 下棋非常“暴力”：每走一步棋之前，它先在“脑海里”模拟几千、几万局， 它可以预知它每一种动作带来的后果，对手最有可能做出的反应都在 AlphaGo 的算计之内。

如何预判？

MCTS 的基本思想就是向前看，模拟未来可能发生的情况，从而找出当前最优的动作。AlphaGo 每走一步棋，都要用 MCTS 做成千上万次模拟，从而判断出哪个动作的胜算最大。







如何模型？

做模拟的基本思想如下：

假设当前有三种看起来很好的动作。每次模拟的时候从三种动作中选出一种，然后将一局游戏进行到底，从而知晓胜负。重复成千上万次模拟，统计一下每种动作的胜负频率，发现三种动作胜率分别是 48%、56%、52%。

那么 AlphaGo 应当执行第二种动作，因为它的胜算最大，此时第二步的移动即为算法真正下了下一步。

模型流程：

MCTS 的每一次模拟选出一个动作 a，执行这个动作，然后把一局游戏进行到底，用胜负来评价这个动作的好坏。

MCTS 的每一次模拟分为四个步骤：选择(Selection)、扩展(Expansion)、求值(Evaluation)、回溯(Backup)。

1、选择：

第一步——选择——的目的就是找出胜算较高的动作，只搜索这些好的动作，忽略掉其他的动作。

指标：

$$score(a) = Q(a) + \frac{\eta}{1+N(a)}\pi\ (a|s;\ \theta)$$

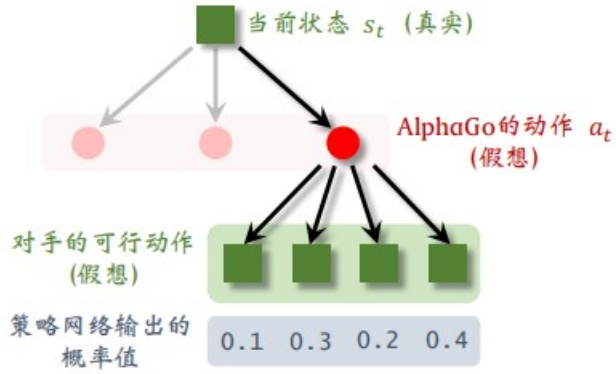
- η 是个需要调的超参数
- N(a)是动作a已经被访问过的次数。
- 初始的时候，对于所有的a，令N(a)←0。
- 动作a每被选中一次，我们就把N(a)加一：N(a)←N(a)+1。
- Q(a)是之前N(a)次模拟算出来的动作价值，主要由胜率和价值函数决定。
- Q(a)的初始值是0；
- 动作a每被选中一次，就会更新一次Q(a)；

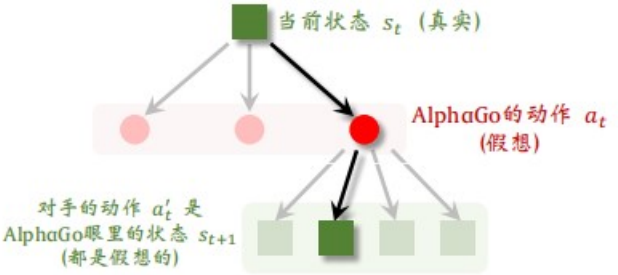
如何理解？

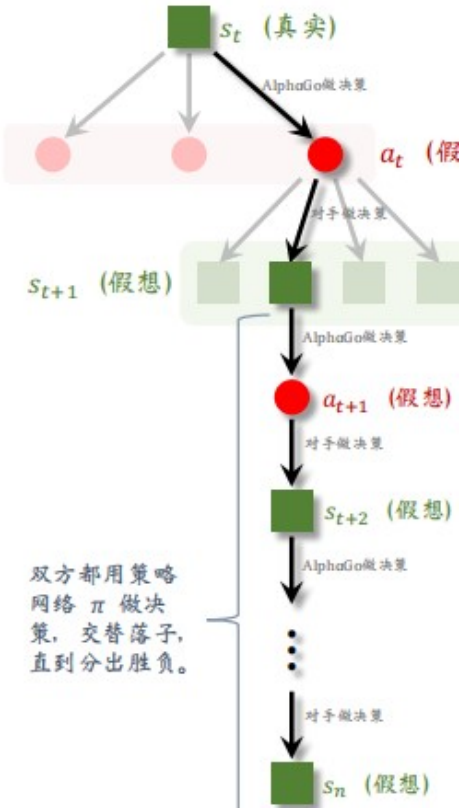
动作 a 选择次数低时，主要由策略网络起作用，鼓励探索次数少的选择。动作 a 选择次

数多时，主要由动作价值起作用，鼓励选择探索后胜率高的。

2、扩展：







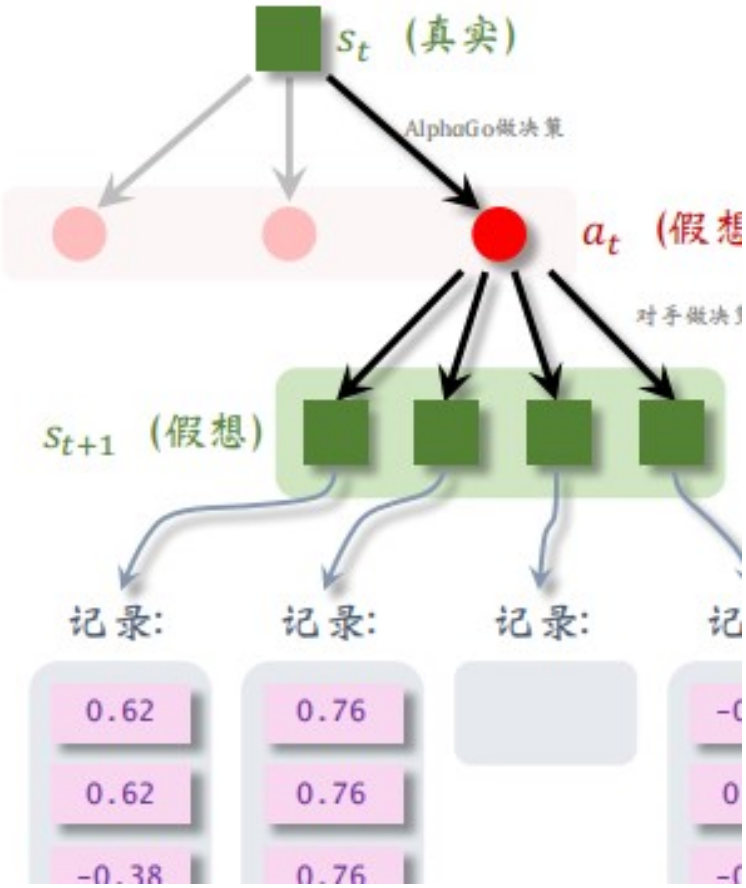
3、求值：（算出某一状态的价值）

AlphaGo 的解决方案是把奖励 r 与价值网络的输出 $v(s_{t+1};\boldsymbol{w})$取平均，作为对状态$s_{t+1}$的评价

$$V(s_{t+1}) \triangleq \frac{r + v(s_{t+1};}{2}$$

奖励 r 是模拟获得的胜负，是对s_{t+1}很可靠的评价，但是随机性太大。
价值网络的评估 $v(s_{t+1};\boldsymbol{w})$没有 r 可靠，但是价值网络更稳定、 随机性小。

4、回溯：



第 t 步的动作 at 下面有多个可能的状态（子节点），每个状态下面有若干条记录。把 $a\,t$ 下面所有的记录取平均，记作价值 $Q(a\,t)$，它可以反映出动作 $a\,t$ 的好坏。

第 t 步的动作 at 下面有多个可能的状态（子节点），每个状态下面有若干条记录。把 $a\,t$ 下面所有的记录取平均，记作价值 $Q(a\,t)$，它可以反映出动作 $a\,t$ 的好坏。

2.2 汉语言部分实现

一、棋盘、棋子显示。

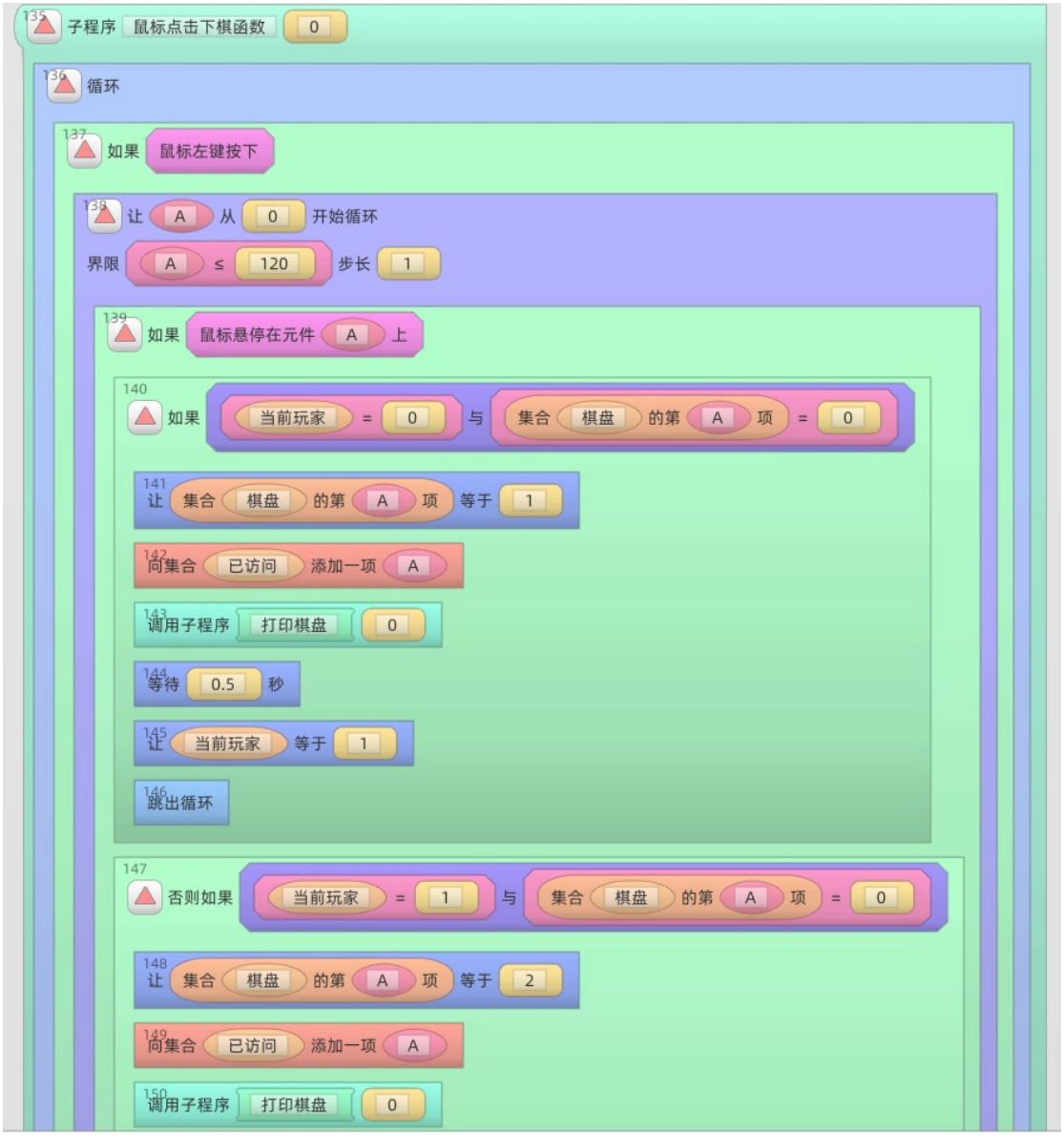
棋盘、棋子均使用图片元件显示，其中棋盘当作背景元件；棋子元件命名使用不同的编号进行区分(从左上到右下依次为0~120），采用循环结构在不同的位置生成棋子以匹配棋盘。

二、棋盘储存和显示

为了便于棋盘不同棋子颜色的储存，定义一个命名为棋盘的集合储存棋子的颜色（红、蓝或空（1、2或0）），集合的索引为棋子的编号，在每次某一方下完棋后依据该集合进行棋盘的更新。

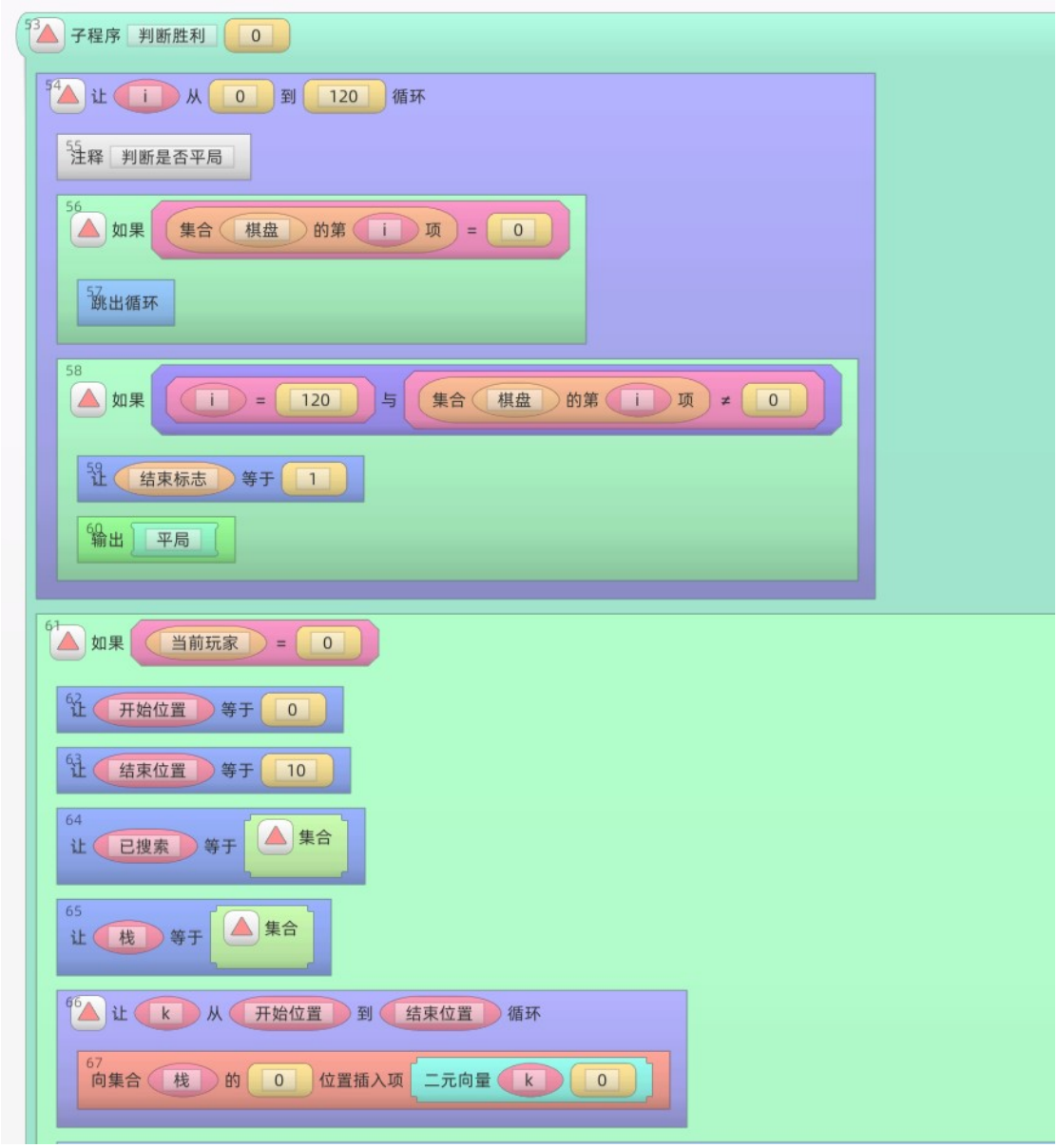
三、鼠标点击下棋实现

首先定义当前下棋的玩家，用全局变量当前玩家储存，每次有效下棋后进行玩家的替换。当主循环执行到需要玩家下棋时，程序开始循环检测，知道鼠标点击棋盘上的有效位置（当前点击位置棋子为“空”，且点击位置是棋子元件），然后改变棋盘集合进行棋盘显示的更新。



四、胜利检测

在这个位置采用的栈的思想。首先检查棋盘上是否还有可用位置，如果没有，输出平局。接着，轮流对两位玩家进行胜利检查。对于每个玩家，按照不同的起始位置（蓝色从左侧开始，红色从上侧开始），从起始位置出发，依次遍历所有相邻棋子，标记已访问过的位置。如果找到了棋盘另一侧的棋子，则该玩家获胜。该算法的实现原理是采用深度优先搜索，在每个节点处检查该节点是否为玩家棋子，如果是则标记为已访问过的节点，尝试从该节点往六个方向继续搜索，直到找到棋盘的边缘或者已访问过的节点，然后回溯到上一个节点继续搜索，直到找到一侧的棋子，或者所有节点都被访问过。



五、主程序循环

定义上一步时刻的棋盘状态，便于添加悔棋功能。

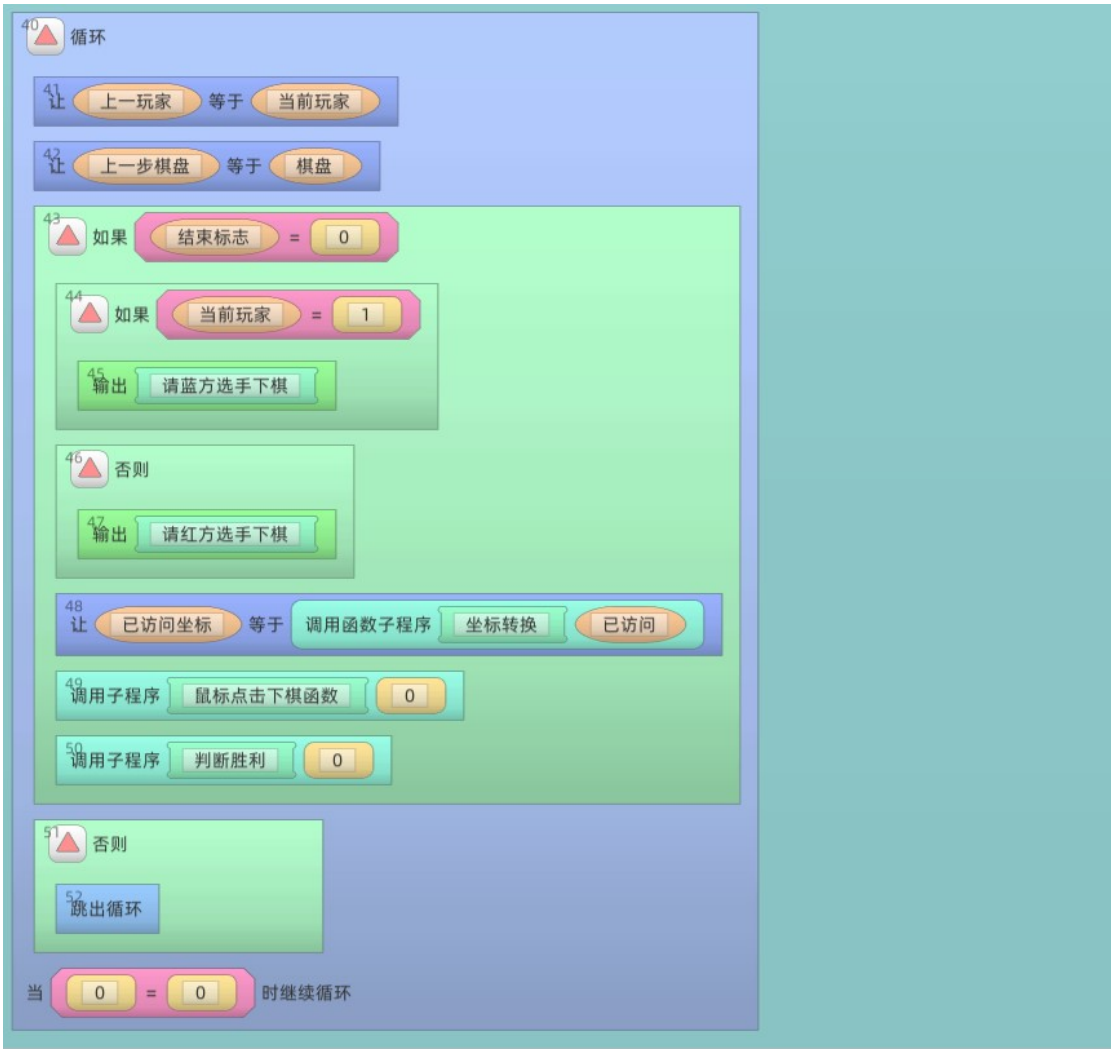
检测当前游戏是否结束，若未结束则输出提示当前下棋的玩家，玩家下玩后进行胜利检测，若有一方胜利，则游戏结束，输出胜利玩家的棋子颜色。

六、将要开发改善的功能

显示：当前的显示只是输出字符串，以后会做出一个界面，在界面上进行下棋玩家的提示。

美化，功能实现后会对游戏界面进行美化。

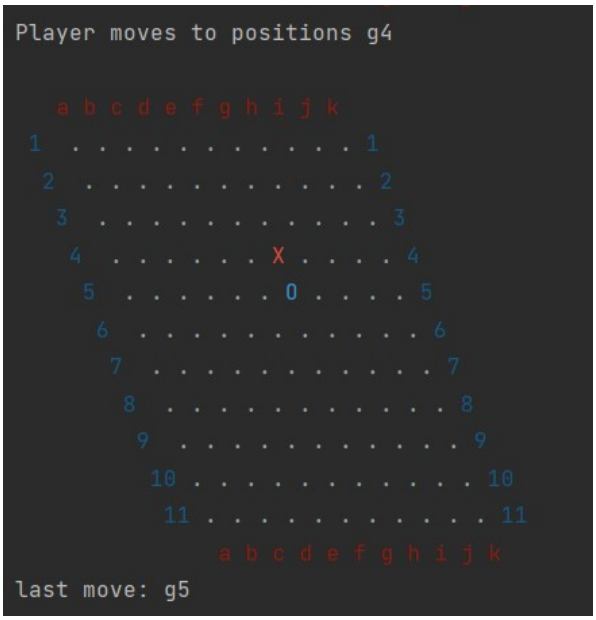
人机对战：通过调用外部python程序，进行人机对战的实现。



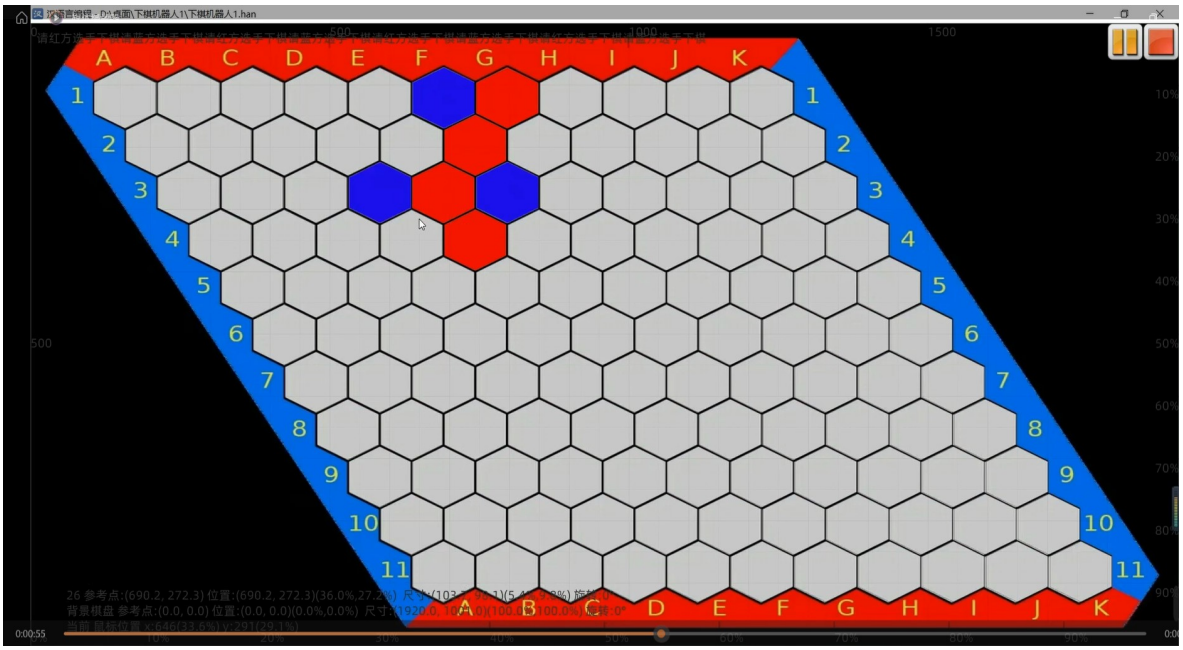
第3章 结果展示

3.1 项目进展

在我们的项目进展中，我们已经完成了海克斯棋的算法设计改造流程，并附完整的代码在附件中，算法可以实现成熟的人机对战模式，可以实现下棋中常规的先后手选择等功能，但模型的训练仍在进行中，目前训练效果较优，但仍未充分收敛。下图是 Python 界面下棋展示图。



在汉语言模块中，图形调用等功能十分方便，目前我们已经完成了海克斯棋的环境的搭载程序，可以实现基础的人人对战模式，同时，我们小组通过深度优先搜索算法在汉语言程序中也实现的胜负判定，此外，我们也借助汉语言通过遍历搜索的方式，验证了海克斯棋不可能出现平局的特性。程序展望图如下。



完整结果请看演示视频

3.2 未来展望

在我们小组设想是，我们结合蒙特卡洛树搜索的思路，智能化改造网络流行的 AlphaGo Zero 算法结构，并将其采用汉语言编程设计小程序的图形化界面，并配备成熟

的下棋程序功能。

从现在的进展来看，项目设想是完全可以实现的，在使用用汉语言编程软件制作海克斯棋的图形化界面时，我们巧妙地借用了文本文件作为媒介，来传递两组程序的输入输出情况，小组成员不禁思考，这是否也是一种混合编程的实现呢？

在教学测方面的进展中，我们小组目前已经制作完成了教学测程序的封面如下，我们计划从海克斯棋的起源、发展、游戏规则以及下棋策略等方面进行展示与教学，并通过人人对战、人机对战等模式，展示海克斯棋的独特魅力。

项目的进展过程让我们小组成员收获颇丰，也遇到了诸多困扰的问题，在本周的实验课向赵老师、孙老师充分请教之后，相信我们也能够解决大半，完备且成熟的海克斯棋小程序在向我们招手！

第 4 章 成员介绍

姓名	班级	学号	分工	分工占比
张硕	自动化 2003 班	20203473	在 python 中智能化改造 AlphaGo Zero 算法	33%
张欣宇	自动化 2003 班	20205499	在汉语言中编写人人、人机对战算法	33%
姜滔	自动化 2003 班	20205481	在汉语言中编写教、学、测软件	33%