西安交通大学
XI'AN JIAOTONG UNIVERSITY

人工智能围棋

曹晖 教授、郑晓东 博士、王超

电气工程学院

2025年2月

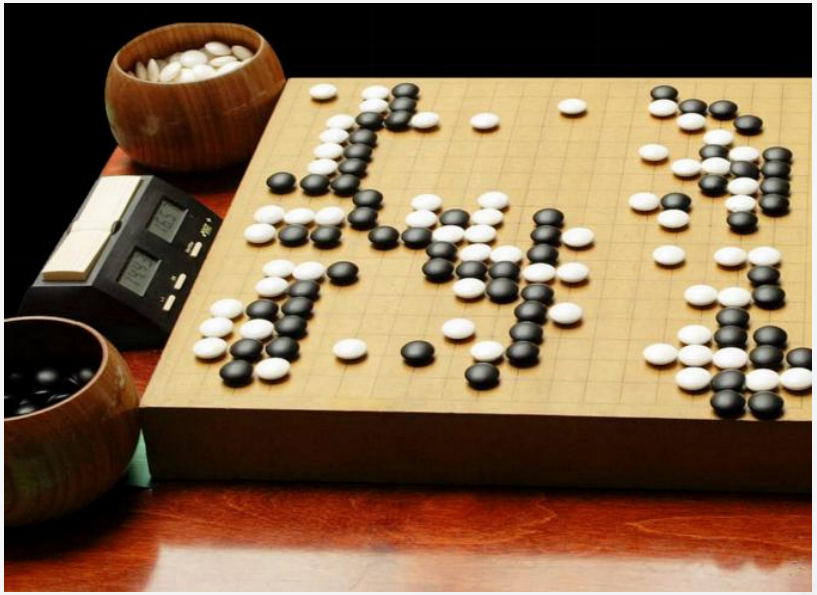
Contents Title

西安交通大学

XI'AN JIAOTONG UNIVERSITY

内容一：智能围棋项目简介

项目背景



围棋是一种策略棋类，使用格状棋盘及黑白二色棋子进行对弈。起源于中国，中国古时有“弈”、“碁（qí）”、“手谈”等多种称谓，属琴棋书画四艺。

西方称之为“Go”，是源自日语“碁”的发音。

进入新世纪，围棋的发展更是从桌面对战逐步演化到计算机网络对战，玩家可以通过计算机围棋程序与网络上的另一玩家进行对战游戏。

棋魂，寻找“神之一手”。

围棋和人工智能

近年来，人工智能技术随着计算机硬件的提升而得到快速发展，由人工智能型围棋程序代替人类玩家进行的围棋对战水平越来越高。对于人工智能围棋而言，大致可以将其发展分为三个阶段：

第一阶段：以模式识别和人工启发式算法为主，水平低于业余初段。

第二阶段：以蒙特卡洛搜索树算法为代表，水平最高达到业余5段。

第三阶段：以机器学习算法为突破，战胜人类职业棋手。

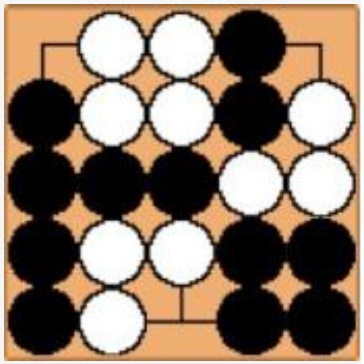
2017年5月，在中国乌镇围棋峰会上，谷歌的阿尔法围棋机器人与排名世界第一的世界围棋冠军柯洁对战，以3比0的总比分获胜。

围棋界公认阿尔法围棋的棋力已经超过人类职业围棋顶尖水平，在GoRatings网站公布的世界职业围棋排名中，其等级分曾超过排名人类第一的中国棋手柯洁。

项目内容

在本次项目中，你将基于一些人工智能算法开发自己的人工智能围棋程序。通过使用**多种最优搜索或强化学习的算法**在一个缩小版的围棋游戏中进行对战。由于围棋的复杂度过高（状态空间复杂度约为$3^{361}\approx10^{172}$），个人计算机难以实现优秀的围棋程序，因此在本次项目中使用的**围棋棋盘尺寸缩小为5x5**（正常围棋尺寸为19x19）。

你的目标**是实现你的智能围棋程序，通过使用不同的搜索算法，实现最优的围棋策略，打败不同难度的机器人程序和你的竞争对手们**，本次竞赛将根据你的算法表现以及竞赛成绩进行打分，你打败的对手越多，程序表现越好，则将取得更好的成绩，加油!



		X	X	
		O	X	
		O		
	O			

规则简介

围棋是一种非常抽象的棋盘策略游戏，每一局有两个对战选手，分别执黑子和白子。对战双方的目标是围掉更多的对方棋手的棋子，在棋盘上占领更多的交叉点则获胜。围棋的基本概念非常简单：

- 对战玩家：围棋由两个玩家对战，分为黑棋棋手和白棋棋手
- 棋盘：围棋游戏的主战场，由横纵穿插的方格网线组成。标准的棋盘是19x19大小，但在本项目中，棋盘的大小调整为5x5，以便适应个人电脑能力。
- 点：围棋棋盘上各个交叉点称之为点，棋盘上除了网格线的交叉点，还有棋盘边缘和四角顶点，这些点是棋手落子的地方。
- 黑白棋子：黑棋棋手执黑子，白棋棋手执白子

本项目中的游戏规则相比于真实围棋游戏稍有变动，规则如下，也很简单：

- 游戏开始于一个空棋盘，棋盘大小为5x5，最大步数为5x5-1=24步
- 两个棋手轮流落子，直到一方胜利
- 棋手可以选择任意未落子的空点进行落子（除了被”打劫”规则或“气”规则限制的位置）
- 在游戏开始后，除非被对方棋手捕获的棋子（“提子”）可以被清理出棋盘，其余已经落下的黑白棋子不可以再变动。

规则简介

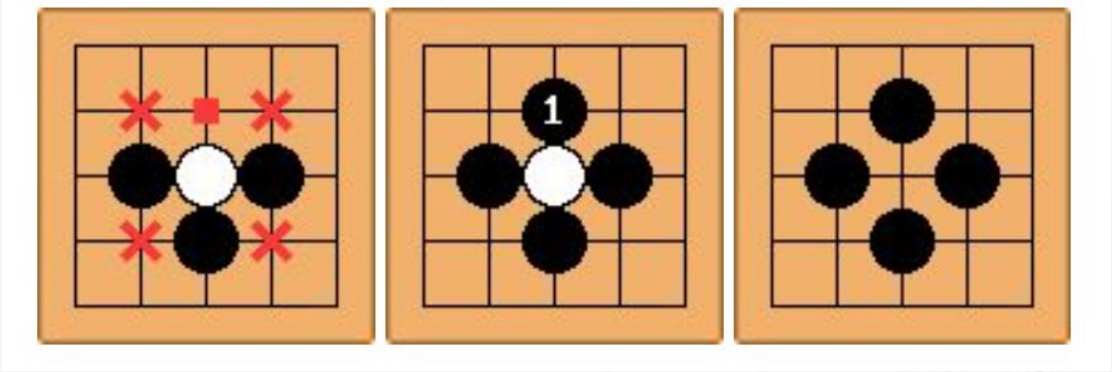
规则1：围棋的“气”

下棋时，对弈双方各执一种颜色的棋子，轮流将一枚棋子放置于交叉点上。**与棋子直线相连的空白交叉点叫做气。**当这些气都被对方棋子占据后，该棋子就没有了“气”，要被从棋盘上提掉。如果棋子的相邻（仅上下左右）直线交叉点上有了同色的棋子，则这两个棋子被叫做相连的。任意多个棋子可以以此方式联成一体，连成一体的棋子的气的数目是所有组成这块棋的单个棋子气数之和。如果这些气都被异色棋子占领，这块棋子就要被一起提掉，即被移出棋盘。

基于围棋的气规则，对战玩家被禁止投掷“自杀“棋子，即玩家不能够在没有气的点上落子抑或落无气之子，除非这样做时可以立即将对手的棋子捕获移除，空出点位形成气。

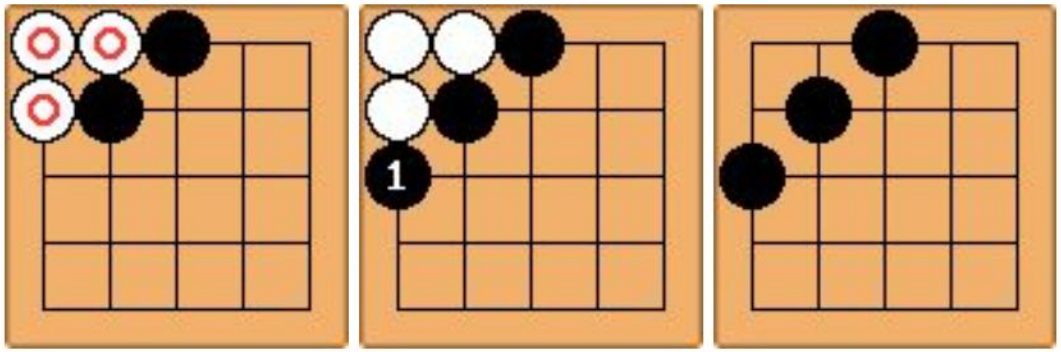
规则简介

例1、单棋子捕获：黑棋1落子后，白棋没有气，被提掉。



白棋再无法下到黑棋中间。黑棋用4个子，占住了5个点。

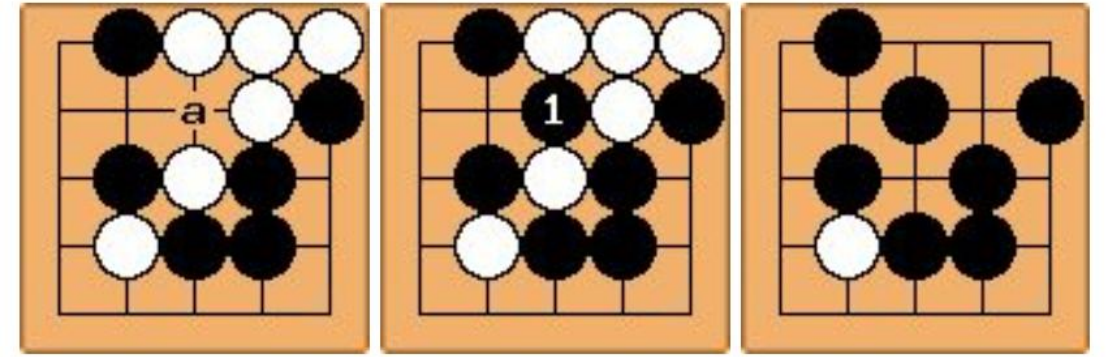
例2、边缘捕获：黑棋1落子后，整片白棋没有气，被提掉。



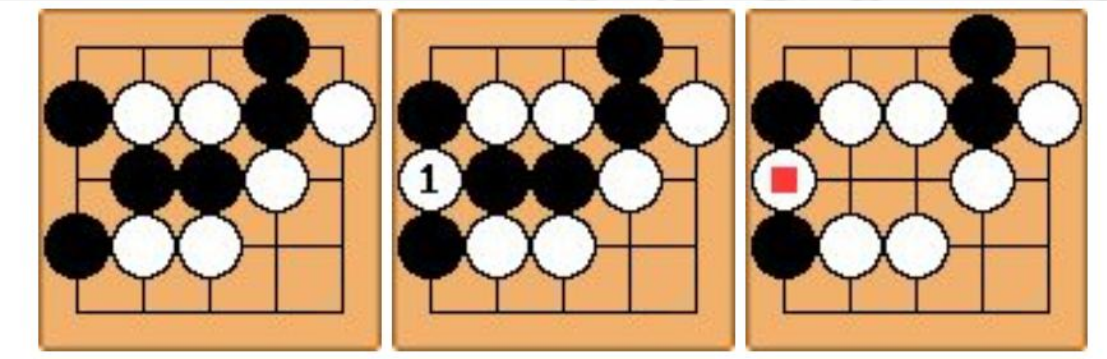
白棋再无法下到黑棋中间。黑棋用3个子，占住了6个点。

规则简介

例3、黑棋1落子后，同时两组白棋没有气，被提掉。



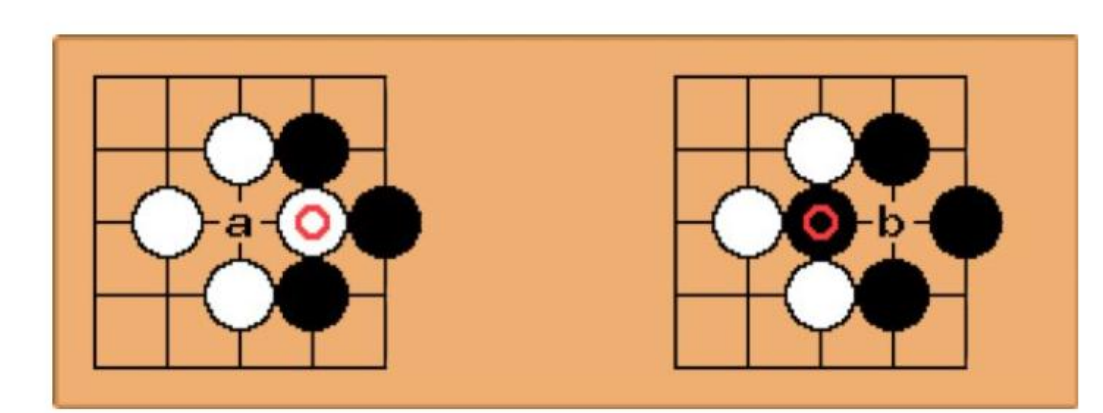
例4、白棋1的落子两个黑棋棋子被捕获，并由于黑棋棋子的移除，使得落子的白棋拥有的气，这种情况的落子是被允许的。



你想想：黑棋是否能再把白棋1提掉?

规则简介

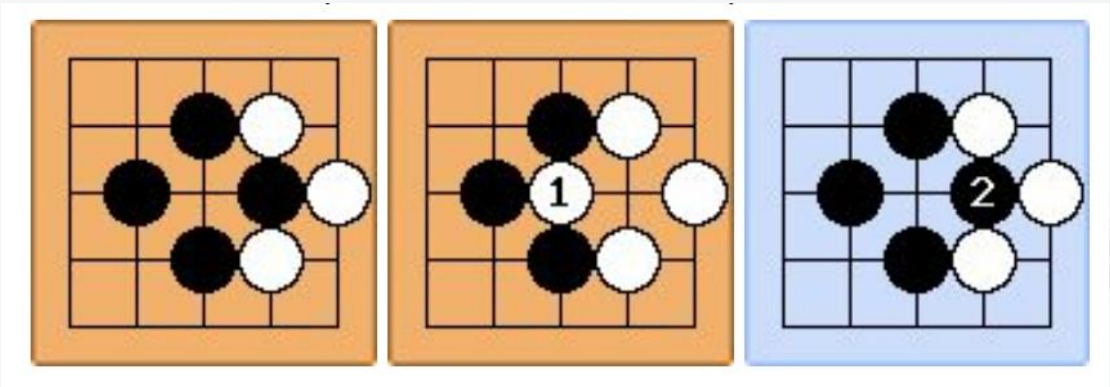
规则2：“打劫”规则



如上图所示左侧棋盘中，黑棋可以通过落子到a处实现对b位置的白棋（带红圈的白棋）的提掉。此时，由于白棋棋子被捕获，a处的黑棋获得了气；同理，在白棋落子时，可以通过落子到b处实现对a处黑子的捕获（右侧棋盘），使得棋盘从右侧局面回到左侧，如此循环往复，使得整个棋盘不能分出胜负，这种情况在围棋中被称之为“劫“。

规则简介

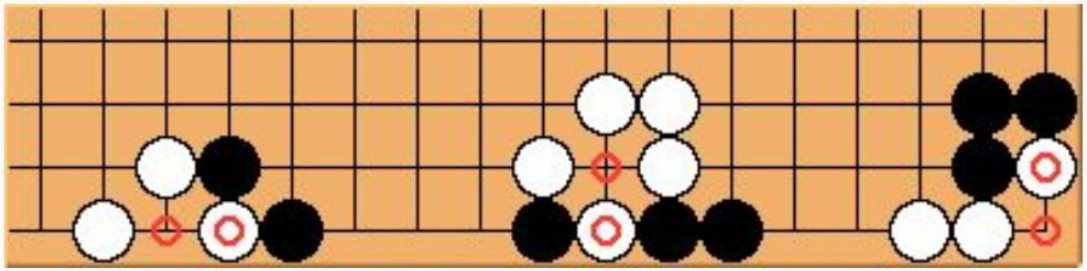
针对“劫”这种特殊情况，打劫规则规定，在棋局中一旦出现了“劫“，即一方玩家提掉了劫子后，另一方玩家不能够立刻选择再次提掉劫子。至少等一手。



在上图左图形成的劫局面上，黑子被**白子1**提掉后，由于“打劫规则“的存在，**黑子2**不能够立刻落子再次提掉**白子1**，黑棋棋手在本轮落子时必须选择在其他地方落子，等待下一轮落子时才可以再次选择捕捉白子1（如果劫依然存在的话）。

规则简介

劫不仅可以存在于棋盘的中央区域，也可以存在于棋盘的边缘，如下图所示：



规则简介

规则3：“贴目”规则

因为黑棋有第一步的优势，所以判给白棋一定的**补偿得分**，即**贴目**；在比赛结束时给了白棋一个补偿分数。 在本项目中棋盘大小为5x5，白色玩家的贴目设置为5/2 = 2.5。

规则4：“放弃落子”规则

棋手可以选择在特殊情况下**放弃**该轮落子的权利，在本游戏中即放弃掉当前一步落子。玩家可以选择任何一步进行放弃落子，以获取潜在的某些优势。

规则5：游戏结束条件

当一轮棋局满足下列四种情况中任意一种时即视为游戏结束：

1.当一方玩家**落子时间超时**时游戏结束，该轮游戏被判负。

2.当一方玩家**非法落子**时游戏结束（非法落子、自杀落子、违反打劫规则、违反气规则等）。

3.当双方玩家**同时放弃落子**游戏结束。

4.当该局游戏过程达到**最大步数**时游戏结束。

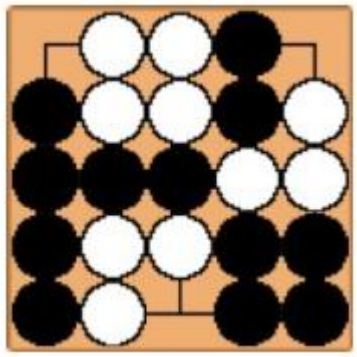
规则6：获胜条件

在真实围棋中有很多种情况可以使一方棋手获胜，但是在本项目中，我们基于下列几种情况得分作为最终的获胜条件：

- 占位得分：一方玩家的占位得分是该玩家在棋盘上所占据的点的个数。
- 最终得分：黑棋玩家的最终得分即为其占位得分，白棋玩家的最终得分由“占位得分“+”贴目“分数组成。
- 获胜判定：
  1. 如果一方玩家落下非法棋子（投掷“自杀”棋子，违背“打劫”规则被判负）
  2. 如果在一盘棋局中达到了最大步数，或者双方玩家都选择放弃落子，则最终得分较高一方获胜。

规则简介

如下图所示，黑子占位得分10分，白子占位得分10分，但白子获得贴目得分2.5分，故白子最终得分为12.5分，白子获胜。



特殊说明，为适应本次项目目的，本游戏规则相比于正式围棋规则有有所简化，比如在正式围棋中得分的计算方式为棋子数量加上围绕一方棋子的空点数量，在本游戏中省略该计分方式使得游戏在缩小版本的棋盘上更加公平合理，感兴趣的同学可以在本次项目后研究真实版围棋，并修改程序，使得游戏更具挑战性。

这里可能存在bug，例如上图中，如果双方都继续下棋.....

我们提供的围棋程序可能不完美，但是我们的目的是设计一个“AI”选手。

很多时候任务都是在很多约束下完成的。适应它，改造它。

西安交通大学
XI'AN JIAOTONG UNIVERSITY

内容二：项目内容和要求

项目评分

在本次项目中，各位同学的程序会与其他同学的程序进行对战，同时，也会对一个“智能对手”进行对战，胜率越高则得分越高。

提供三个“AI选手”，“智能程度”排序：player2 > player1 > random

第六周周三下午15：00前，各位同学提交程序，老师统一判定：

1. 挑战random agent，对战10局，胜率在70%以上者得到基础分60分。
2. 挑战player1 agent，对战10局，每获胜一局得1分，最多可得10分。
3. 挑战player2 agent，对战10局，每获胜一局得1分，最多可得10分。
4. 根据前面得分，组成8~10组循环对战，组内前三名加10分、5分、3分。

第七周课上公布组内冠军，第八周课上进行决赛，允许赛前小组优化程序。

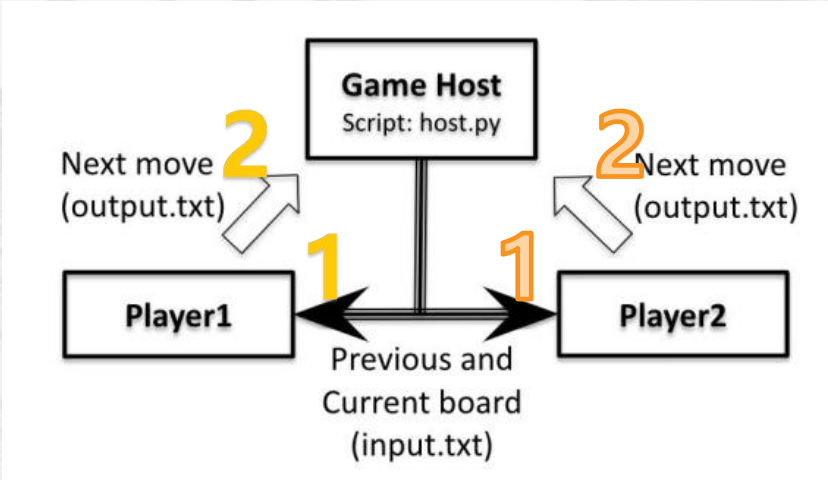
5. 组间循环对战，前三名的全组成员分别加10、7、4分，其余选手各加2分。
6. 现场展示讨论加分。比赛完成后，自愿进行实现逻辑解析。（70分左右的同学，可加分3~5分）
7. 其他加分，优化题目框架程序（比赛流程和规则）、UI界面，最多加分10分（如果是多人贡献，所有人统一打7折）。时间够的话，最后上课展示，大家匿名打分；时间不够，我来掌握打分。Ps.最终得分映射到21~35分的区间。

项目框架

本次智能围棋程序由一个主程序host.py进行对两个智能下棋程序的调用，使得两个程序可以轮番落子。主持程序将会跟踪游戏的进程，从每一方下棋程序中轮番获取双方下一步的落子，判断双方下一步的落子是否遵循游戏规则，并且将会从棋盘上去掉被捕获的棋子以及最终判定获胜方。

需要注意的是，双方下棋程序必须准确的按照规定的格式输出自己下一步棋子的位置，并保存在一个命名为output.txt的文件中交付给host.py程序。总体而言，每一方程序需要做的事情很简单，首先从host.py处获取当前对方落子后的棋盘状态，这个状态由一个input.txt文件保存，然后输出自己下一步落子后的棋盘状态给host.py，即输出一个output.txt文件给host.py。

程序架构示意图：



游戏参数

本项目程序中默认使用下列参数规定：

- 在棋盘上，0代表空点，1代表黑棋，2代表白棋
- 可视化输出中，X代表黑棋，O代表白棋
- 下棋顺序为黑先白后，所有游戏中总是如此
- 棋盘大小为5x5
- 每局游戏最大步数为5x5-1=24步
- 对白棋的贴目分数为5/2=2.5

	j=0	j=1	j=2	j=3	j=4
i=0	0	0	1	1	0
i=i	0	0	2	1	0
i=2	0	0	2	0	0
i=3	0	2	0	0	0
i=4	0	0	0	0	0

		X	X	
		O	X	
		O		
	O			

棋盘布局、棋盘落子状态示意图

```
Black makes move...
----------
XX   X
OOXOO
 OOOX
 OOXX
 O
----------

White makes move...
----------
  O X
OO OO
 OOOX
 OOXX
 O
----------
```

程序可视化输出示例

棋盘将使用以0为第一个坐标的坐标系，即左上角第一个位置坐标为(0，0)。（0，4）为右上角的坐标点，（4，0）为左下角坐标点，（4，4）为右下角坐标点。在棋局程序中，使用1代表黑棋，2代表白棋，0代表空点。在打印输出中，X为黑棋，O为白棋（host.py程序将会打印输出棋局的每一步）。

预设对手

针对本次项目，将会提供不同的简单机器人对手来和你的机器人进行对战，这些机器人对手的难易程度不同，你打败他们的胜率将决定你的基础成绩。这些机器人对手包括：

- Random palyer：随机机器人，在满足要求的空点随机落子
- Alpha-beta player：使用minimax算法进行下棋的机器人

在你完全打败这些AI机器人后，你可以和你的同学进行对战，你可以选择任何算法来优化你的程序，将它变得更加强大。

你可以选择使用一些机器学习算法来训练一个智能型的AI程序，比如使用Qlearning算法通过训练一个agent来进行对战，但前提是输入输出必须符合游戏规则，更加重要的是这些程序代码必须由你自己编写，不能够从其他地方获取，抑或是调用第三方已经编写好的训练模型和算法（辅助程序除外），一旦发现代码有抄袭行为，将会被视为作弊，失去所有35分。

你的代码

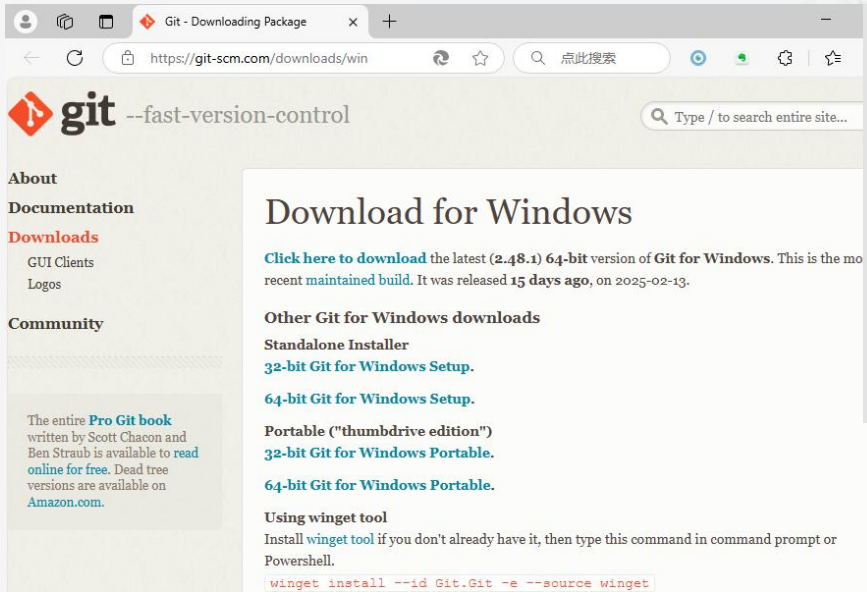
在本次项目中，你自己的程序应该命名为：my_palyer.py，使用python3开发。在游戏过程中，host.py将会循环判断游戏是否结束，如果没有结束，就执行下面的动作：

- 交替更改当前下棋的玩家；
- 清理掉当前目录中的input.txt文件和output.txt文件；
- 新建input.txt文件，输出上一步棋盘状态和当前棋盘状态；
- 调用本步棋手的程序，本步棋手程序应该读取input.txt并产生一个output.txt文件；
- 验证新的output.txt，检测是否有格式错误或者其所记录的新一步落子点是否违反游戏规则，如果有格式错误或者所落棋子违反了游戏规则，则将本局该棋手判负；
- 检查是否所有游戏棋局结束，如果游戏结束则输出获胜选手。

代码规范

为了帮助你更好的开始编程，本次项目将会把host.py程序和一个random.py程序给你，还有一个脚本程序build.sh程序也会一并发给你，你可以复制一个random.py并将其中一个重新命名为my_player.py，然后使用命令行提示符窗口在当前目录下运行 sh build.sh进行测试。对于使用MacOS的同学，可以直接使用其自带的terminal命令行进行测试。此外，通过调整build.sh中找到 “play_time=4”，可以调整测试的局数。

Git Bash 是一个轻量级的 Bash 环境，适合运行简单的脚本。下载并安装 Git for Windows，安装完成后，打开 Git Bash。

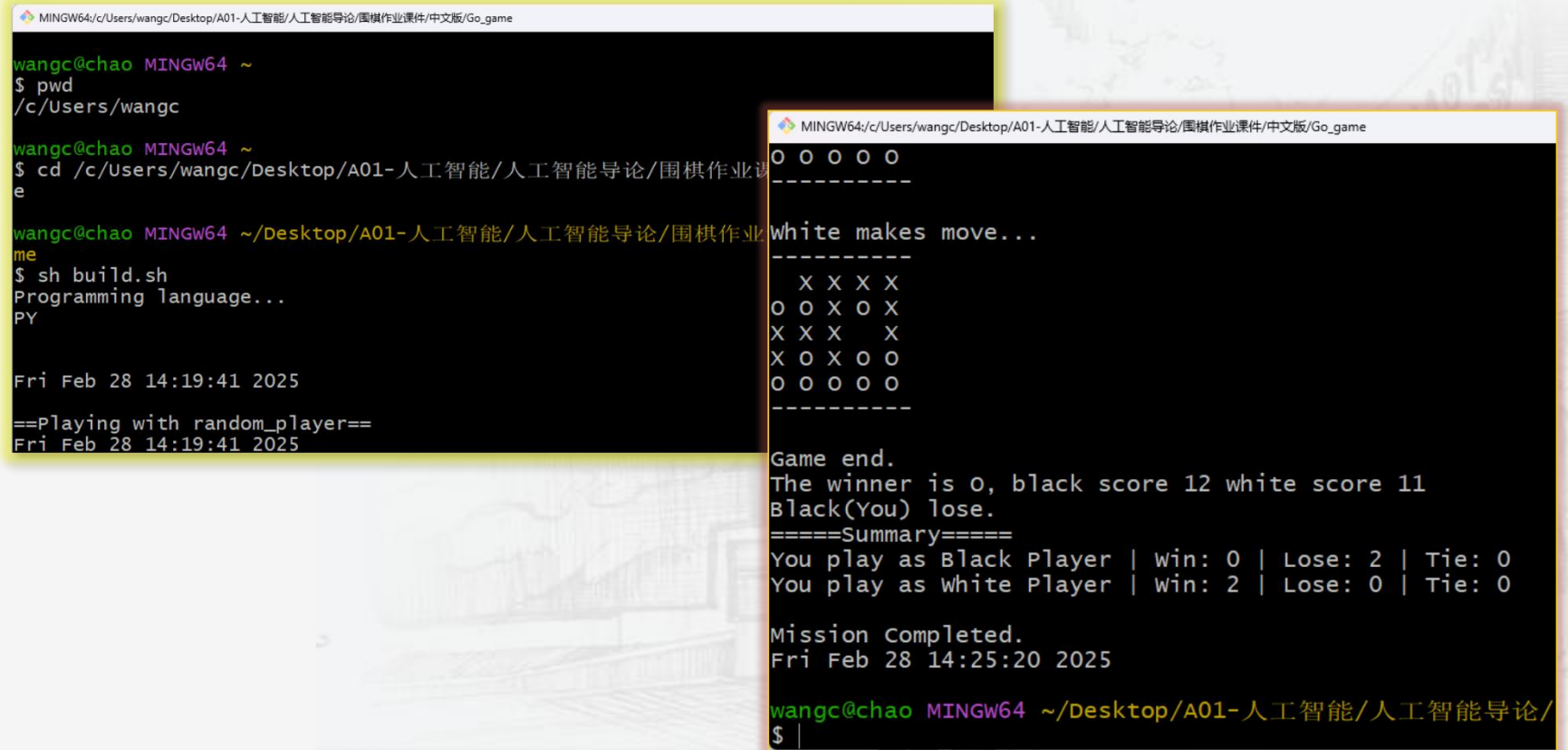


代码规范

在终端中，切换到对应目录，注意目录路径表示方式和windows的路径方式不一样，采用斜线分割，例如：

- /c/Users/wangc
- ~/Desktop/A01-人工智能/人工智能导论/围棋项目课件/中文版/Go_game

执行 build.sh，可以看到host.py调用两个player对战的胜负结果。



输入输出格式

每一步棋子落子位置的输入输出格式对于完成本次项目至关重要，你必须严格按照本节要求的输入输出格式进行落子位置的输出，否则你的程序将不可能赢得任何一场比赛。

- 输入input.txt：你的程序必须从当前目录下读取一个input.txt文件来获取棋盘状态，其包括以下几条格式。

```
========input.txt========
2
00110
00210
00200
02000
00000
00110
00210
00200
02010
00000
========================
```

不包含最上面的==input.txt==和最下面的===

1. 第1行：一个数值“1”或者数值“2”来表明本局游戏你的棋子的颜色（黑棋=1，白棋=2）
2. 第2-6行：对于前一个状态的棋盘的描述，由5行数字组成，每一行由5个数字组成，这个状态表明的是你的上一步下过后的棋盘状态，其中黑棋=1，白棋=2，空点=0
3. 第7-11行：对于当前棋盘状态的描述，由5行数字组成，每一行由5个数字组成，这个状态表明的是你的对手刚刚一步下过后的状态，其中黑棋=1，白棋=2，空点=0
4. 在最开始的时候，第2-11行input.txt文件的默认数值为0

输入输出格式

- 输出output.txt文件：为了使host.py知道你这一步的落子位置，你需要在你的程序中创建一个output.txt文件到当前目录下，其格式包含以下几条：

  1、输出位置由两个数字及一个逗号组成，比如

```
========output.txt=======
2,3
=========================
```

  如果你要放弃当前步落子，只需输出大写的PASS:

```
========output.txt=======
PASS
=========================
```

  注意，文件中不包含例子中的====output.txt====和例子中的最后一行====，输出文件中只有一行。

提示和帮助

- 请使用python开发你的脚本，文件应命名为my_player.py。
- 只需提交你的程序文件，即my_player.py，不接受其他文件。
- 在本次作业中，为了更好地帮助你完成作业，将会下发build.sh、host.py、以及一个random.py程序，帮你理解整个项目框架，但最终输入输出需要满足规则要求，且最终评分会在助教版本地host.py中进行。
- 除了以上几个程序外，还有一些帮助程序帮助你测试试验你自己的智能围棋程序，分别为：

  1.Write.py：输出output.txt的程序

  2.Read.py：读取input.txt程序

  3.Clean.py：清理，删除记录的程序

  4.Cresult.txt：游戏对战时生成的记录
- 程序打包下发，其中包括上述程序脚本文件以及一个init文件夹，init文件夹中包含input.txt，不要修改这个文件夹结构，直接解压并进行开发。
- 为了更好的组织最终的屏蔽，请控保证每一步落子的时间在10秒以内，超出10秒的限制会被助教版本的host.py直接判负。

西安交通大学
XI'AN JIAOTONG UNIVERSITY

内容三：提示和帮助

提示和帮助

- 请使用python开发你的脚本，文件应命名为my_player.py。
- 只需提交你的程序文件，即my_player.py，不接受其他文件。
- 在本次作业中，为了更好地帮助你完成作业，将会下发build.sh、host.py、以及一个random.py程序，帮你理解整个项目框架，但最终输入输出需要满足规则要求，且最终评分会在助教版本地host.py中进行。
- 除了以上几个程序外，还有一些帮助程序帮助你测试试验你自己的智能围棋程序，分别为：

  1.Write.py：输出output.txt的程序

  2.Read.py：读取input.txt程序

  3.Clean.py：清理，删除记录的程序

  4.Cresult.txt：游戏对战时生成的记录
- 程序打包下发，其中包括上述程序脚本文件以及一个init文件夹，init文件夹中包含input.txt，不要修改这个文件夹结构，直接解压并进行开发。
- 为了更好的组织最终的屏蔽，请控保证每一步落子的时间在10秒以内，超出10秒的限制会被助教版本的host.py直接判负。

build.sh

好的！我会重新分析你的代码，并生成一个更简化的 **Mermaid 流程图**，专注于核心逻辑，颗粒度更粗，类似于你提供的示例风格。

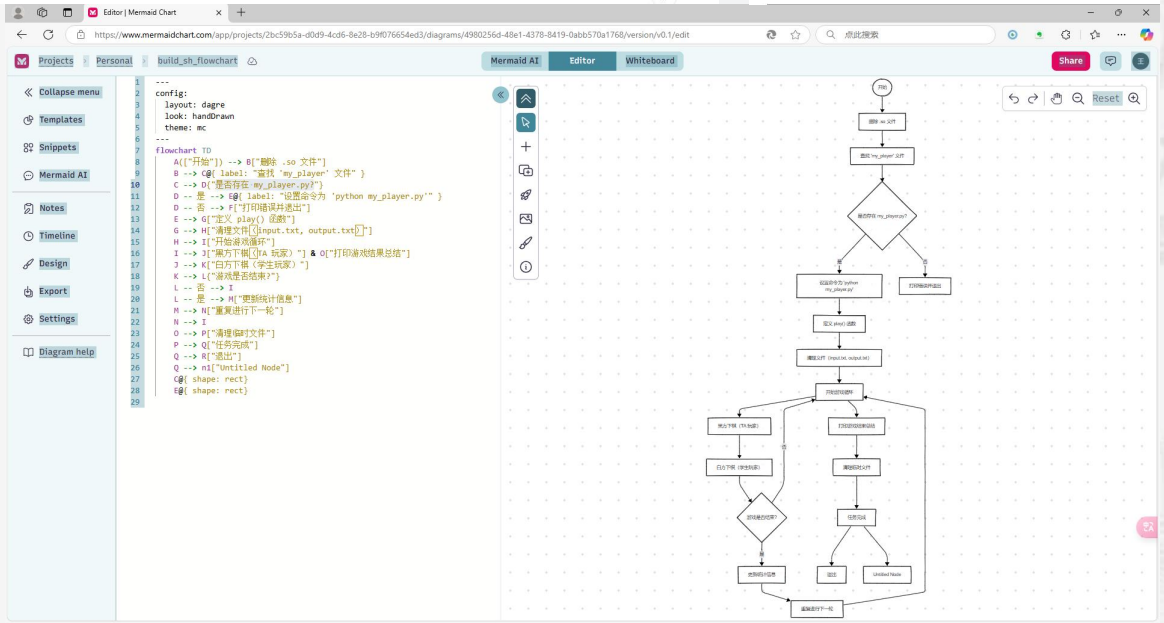
简化后的 Mermaid 流程图

mermaid 复制

```
---
config:
  layout: fixed
  look: classic
---
flowchart TD
    A(["开始"]) --> B["删除 .so 文件"]
    B --> C["查找 'my_player' 文件"]
    C --> D{"是否存在 my_player.py?"}
    D -- 是 --> E["设置命令为 'python my_player.py'"]
    D -- 否 --> F["打印错误并退出"]
    E --> G["定义 play() 函数"]
    G --> H["清理文件（input.txt, output.txt）"]
    H --> I["开始游戏循环"]
    I --> J["黑方下棋（TA 玩家）"]
    J --> K["白方下棋（学生玩家）"]
    K --> L{"游戏是否结束?"}
    L -- 否 --> I
    L -- 是 --> M["更新统计信息"]
    M --> N["重复进行下一轮"]
    N --> I
    I --> O["打印游戏结果总结"]
    O --> P["清理临时文件"]
    P --> Q["任务完成"]
    Q --> R["退出"]
```

核心逻辑说明

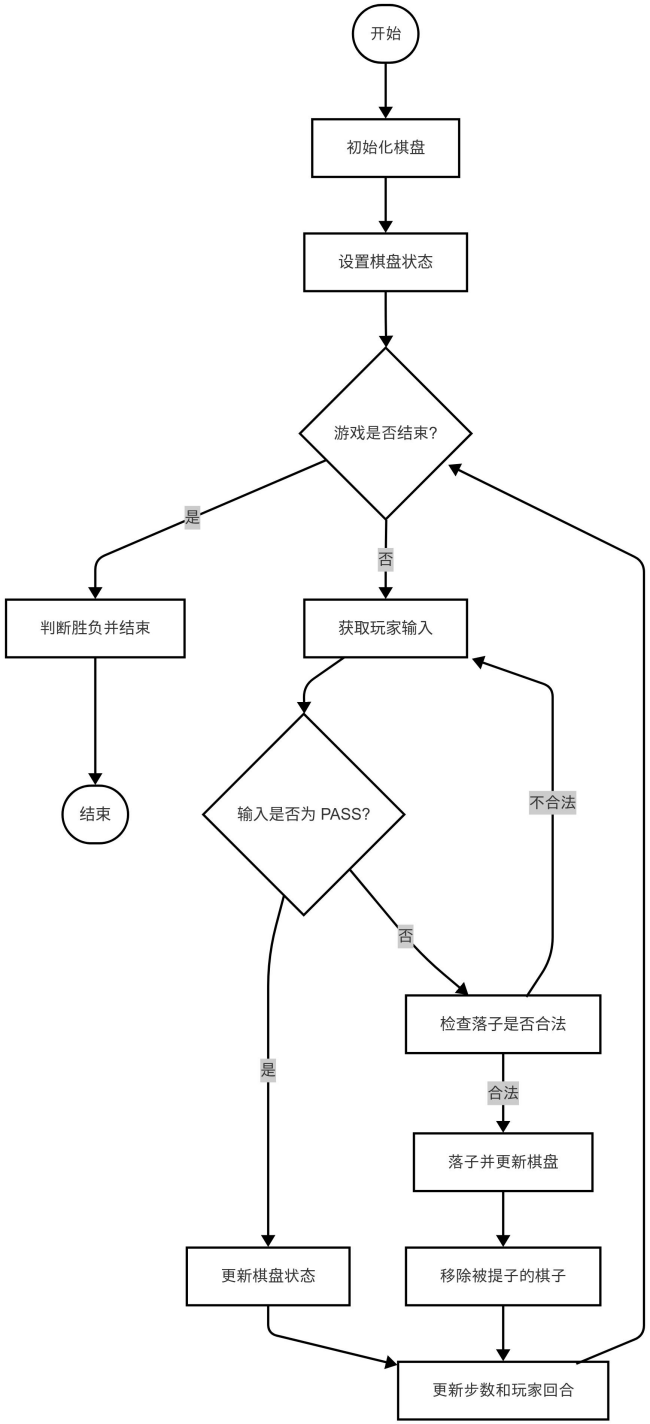
1. **开始**：脚本启动。
2. **删除 .so 文件**：清理当前目录下的 `.so` 文件。
3. **查找 'my_player' 文件**：检查是否存在 `my_player.py`。
4. **是否存在 my_player.py?**：
   - 如果存在，设置命令为 `python my_player.py`。
   - 如果不存在，打印错误信息并退出。
5. **定义 play() 函数**：定义游戏逻辑。
6. **清理文件**：删除 `input.txt` 和 `output.txt`。
7. **开始游戏循环**：进入游戏主循环。
8. **黑方下棋（TA 玩家）**：TA 玩家执黑下棋。
9. **白方下棋（学生玩家）**：学生玩家执白下棋。
10. **游戏是否结束?**：
    - 如果未结束，继续游戏循环。
    - 如果结束，更新统计信息。
11. **更新统计信息**：记录胜负或平局。
12. **重复进行下一轮**：继续下一轮游戏。
13. **打印游戏结果总结**：输出对战结果。
14. **清理临时文件**：删除临时文件。
15. **任务完成**：脚本结束。



让DeepSeek帮你读程序

build做的事情：

- 找到你的my_player.py
- 定义对战函数play（）
  - 主流程：调用两个player返回比赛结果。
- 定义对战轮次：play_time=4
- 循环执行：
  - TA版本执黑，调play
  - 统计结果
  - 学生版本执黑，调play
  - 统计结果
- 输出结果



host. py

开始：初始化棋盘和游戏状态。

设置棋盘状态：根据输入设置棋盘。

游戏是否结束：检查游戏是否达到结束条件。

获取玩家输入：input.txt

获取当前玩家的落子位置或 PASS。

输入是否为 PASS?：如果是，直接更新棋盘状态。

检查落子是否合法：

检查落子是否符合规则。

落子并更新棋盘，如果合法，更新棋盘状态。

移除被提子的棋子，移除被提子的棋子。

更新步数和玩家回合，更新步数并切换玩家回合。

判断胜负并结束，如果游戏结束，判断胜负并退出

random_player

定义 RandomPlayer 类： （你应该定义自己的xxx类）

__init__ 方法：初始化玩家类型为 random，表示这是一个随机玩家。

get_input 方法：

遍历棋盘的每一个位置，检查是否是一个合法的落子位置。

如果当前位置合法，将其加入 possible_placements 列表。

(注意这里调用了go定义的检查是否合法的函数)

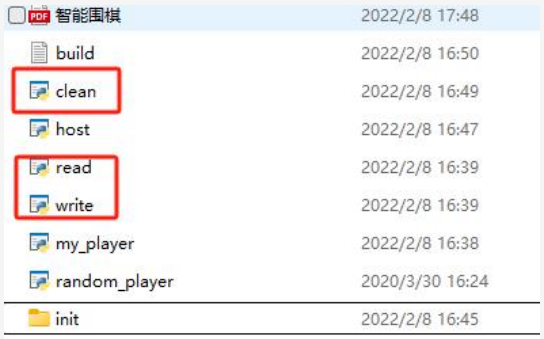
如果没有合法的落子位置，返回 "PASS"。

否则，从合法的落子位置中随机选择一个并返回。

主流程：

1. 调用 readInput(N) 读取当前玩家类型 (piece_type) 和棋盘状态 (previous_board, board)。
2. 初始化棋盘：创建 GO 实例，并调用 set_board 方法设置棋盘状态。
3. 初始化玩家：创建 RandomPlayer 实例。
4. 获取落子位置：调用 player.get_input() 获取随机玩家的落子位置或 "PASS"。
5. 调用 writeOutput(action) 将结果写入输出文件。

还剩什么



辅助代码：

- read.py
- write.py
- clean.py

这个项目和我们的课程有什么联系？

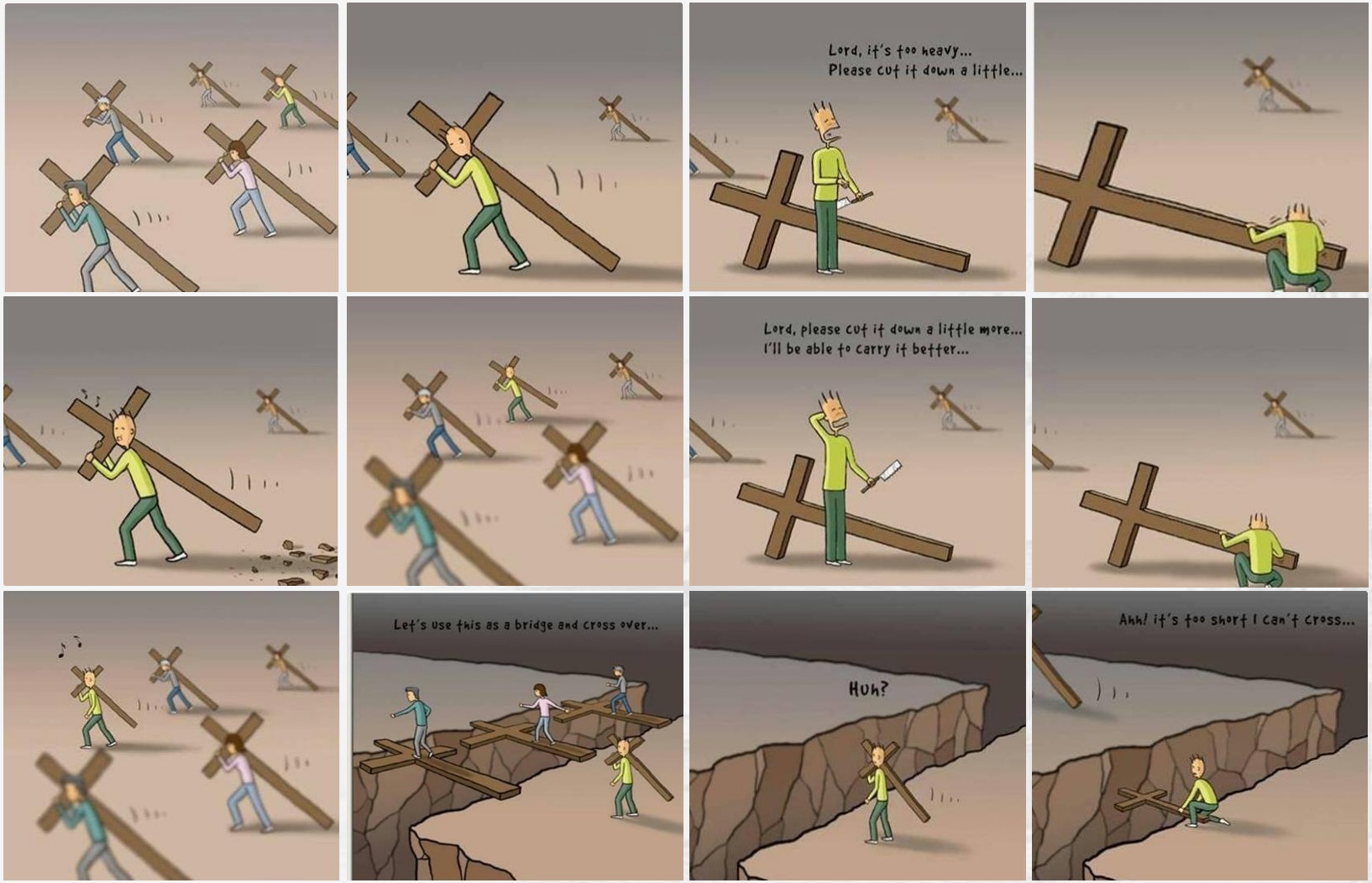
1、传统的思路：“搜索解空间”，找出最优解。
2、机器学习的方法，例如：Qlearning
3、神经网络？如何定义网络输出的目标？

Q-learning 是一种经典的 强化学习（Reinforcement Learning, RL）算法，用于解决 马尔可夫决策过程（Markov Decision Process, MDP）问题。它的目标是让智能体（Agent）通过与环境的交互，学习到一个最优策略，从而最大化累积奖励。

建议大家不要着急选择一个方法方案就开始做。建议先至少沉淀一周，我们留一周时间思考，下周会讲解搜索的概念和一般方法，还希望留下一些时间，大家套路这个思路。

AI的作用

- AI和人的关系：AI应该能成就人
- 借助AI提升自己的效率，重点在自己获得的成长，帮助人类进步



西安交通大学
XI'AN JIAOTONG UNIVERSITY

谢谢大家