

# 2024课程设计选题

本学期的课程设计将会分为两个阶段:

## 上半学期

- **目标：**着重探讨**代码编写**和**策略搜索**，主要涵盖贪吃蛇、五子棋等游戏博弈问题。这个阶段的学习要求先掌握并实现基本的游戏规则，然后通过自己的实践和经验来设计搜索策略，让程序能够自动地进行游戏。
- **选题：**任选一题完成阶段一任务。
- **截止时间：**2023.4.14 20:00
- **展示时间：**2023.4.25
- **提交材料：**
  - 源代码，提交代码推荐包含可执行指令以及路径（不要使用绝对路径），或可执行文件，“一键执行”；
  - Readme文件，用来说明运行源代码所需要执行的命令，环境依赖等；
  - 课程设计报告，包含选题描述、方案设计、代码模块的功能划分与描述、实现效果等。推荐使用markdown文件，并在报告中体现每个功能的gif演示。推荐用精简的语言介绍代码结构，算法原理只需指出哪部分代码实现的是什么算法，不需要讲具体细节。

## 下半学期

- **目标：**结合课程所学的人工智能知识来解决实际任务。针对选题，利用所学知识编写AI来进行游戏，将所学知识应用于实际问题。
- **选题：**针对上半学期选题，完成阶段二任务。
- **截止时间：**2023.5.28 20:00
- **展示时间：**2023.6.6
- **提交材料：**
  - 源代码，提交代码推荐包含可执行指令以及路径（不要使用绝对路径），或可执行文件，“一键执行”；
  - Readme文件，用来说明运行源代码所需要执行的命令，环境依赖等；
  - 课程设计报告，包含选题描述、方案设计、代码模块的功能划分与描述、实现效果等。推荐使用markdown文件，并在报告中体现每个功能的gif演示。推荐用精简的语言介绍代码结构，算法原理只需指出哪部分代码实现的是什么算法，不需要讲具体细节。

注：允许自行拟定选题，但需满足以上课程要求，并与助教老师商量。

# 1. 掇蛋【赖哲剑】

## 背景介绍：

掇蛋是一种流行于中国江苏省的扑克牌游戏，通常由四个人玩两副牌，分为两队进行对抗。每位玩家初始获得27张牌，游戏目标是**尽快出完手中的牌**。掇蛋游戏中有多种牌型，如单张、对子、顺子、连对、炸弹等，具有丰富的策略和合作元素。

## 阶段一任务：

### 搭建环境：

我们将实现掇蛋游戏的**简化版**，游戏环境如下所述，并遵循特定的规则进行简化：

1. 游戏仅包括2个玩家（一名玩家和一名AI），每人随机发27张牌。
2. 不完全信息博弈，即玩家和AI的手牌互相不可见。
3. 简化牌型：不使用花色，仅考虑牌的点数，去除大小王【共  $(54 - 2) \times 2 = 104$  张牌】并去除部分复杂牌型，如同花顺、四大天王等。
4. 当前级数为2（即本局中2是除大小王最大的牌）

### 牌型大小和出牌规则（参考）：

- 牌点由大到小排列为：2、A、K、Q、J、10、9、8、7、6、5、4、3。
- 相同牌型之间的大小比较
  - a. 单张、对子、三同张、三连对、三同连张、顺子，以上六种牌型直接根据牌点确定大小。
  - b. 三带对：仅比较三同张牌点大小，不比较所带对子牌点大小。
  - c. 炸弹：张数多者大，张数相同则比较牌点大小。
- 不同牌型之间的大小比较
  - a. 炸弹 > 其他7种牌型（单张、对子、三同张、三带对、三连对、三同连张和顺子）
- 出牌规则
  - a. 若自己为下家则必须与上家出同种类型且更大；若为上家，则可以任意出牌
  - b. 先出的一方为上家，跟牌的一方为下家；若某一回合无法接牌，选择pass，则对方自动成为上家

### 牌型种类：7种（所有牌型不计花色）

1. 单张

$A > K > Q > J > 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3 > 2$

## 2. 对子

22 > AA > KK > ... > 33, 不计花色, 上家出33, 必须出44或以上

## 3. 顺子

连续的5张牌, 不计花色, 例如: 23456, 10JQKA。

## 4. 三连对

连续的3条对子, 例如: 77 88 99, 44 55 66

## 5. 三同张

3张一样的牌, 例如222, AAA, 666, 888等

## 6. 三带对

3张一样的牌带2张一样的牌, 例如222 33, AAA JJ等

## 7. 三同连张

两个连续的三同张, 例如 222 333, 777 888等

## 8. 炸弹

四张或四张以上牌点相同的牌, 如: 2222、33333、444444等。

## 实现目标：实现基于规则的游戏AI

你的程序需要实现玩家和机器对弈, 随机一方先出牌, 之后玩家和AI轮流出牌 (或pass, 即主动或被迫不出牌), 先出完牌的一方获胜。例如:

1. 每一次出牌, 先展示双方手牌
2. 玩家输入想出的牌序列, 例如AAA, 系统判断玩家持有这些牌, 是合理的牌型, 并且要么牌型可接, 要么玩家是上家, 才完成出牌操作
3. 否则提示玩家”无效出牌“, 玩家修改牌序列, 或选择pass
4. 使用基于规则的策略实现AI的自动打牌
5. (有能力选做) 可视化交互界面: 使用如Tkinter、PyQt等库实现GUI

## 实现建议

- 牌型识别: 首先, AI需要能够识别所有可能的牌型, 包括单张、对子、顺子、连对、三带、炸弹等。这要求程序能够分析手中的牌并确定所有可能的合法出牌选项。
- 出牌规则: 定义一套出牌规则, 让AI决定何时跟牌、何时主动出牌。例如, 如果AI是上家, 它可能会选择较弱的牌型出牌; 如果是下家, 它可能需要选择能够大于上家的牌型。
- 优先级设置: 为不同的牌型和出牌策略设置优先级。通常情况下, 炸弹等强力牌型应当作为最后的手段, 除非该出牌显著提高胜率。

## 阶段二任务：

### 实现目标：实现基于机器学习算法的游戏AI

1. 利用机器学习算法训练斗地主策略。
2. 探索和实践不同特征表示方法，提升模型性能。
3. （有能力选做）完善一些用户友好型功能，如：
  - 实时提示与帮助：在游戏过程中，为玩家提供实时的出牌提示和规则帮助。例如，当玩家轮到出牌时，系统可以提示当前可出的牌型范围或推荐出牌，帮助新手玩家更快熟悉游戏。
  - 调整游戏难度：提供不同难度级别的AI对手，玩家可以根据自己的水平选择合适的难度。随着玩家技术的提升，可以逐步挑战更高级别的AI，保持游戏的挑战性和趣味性。

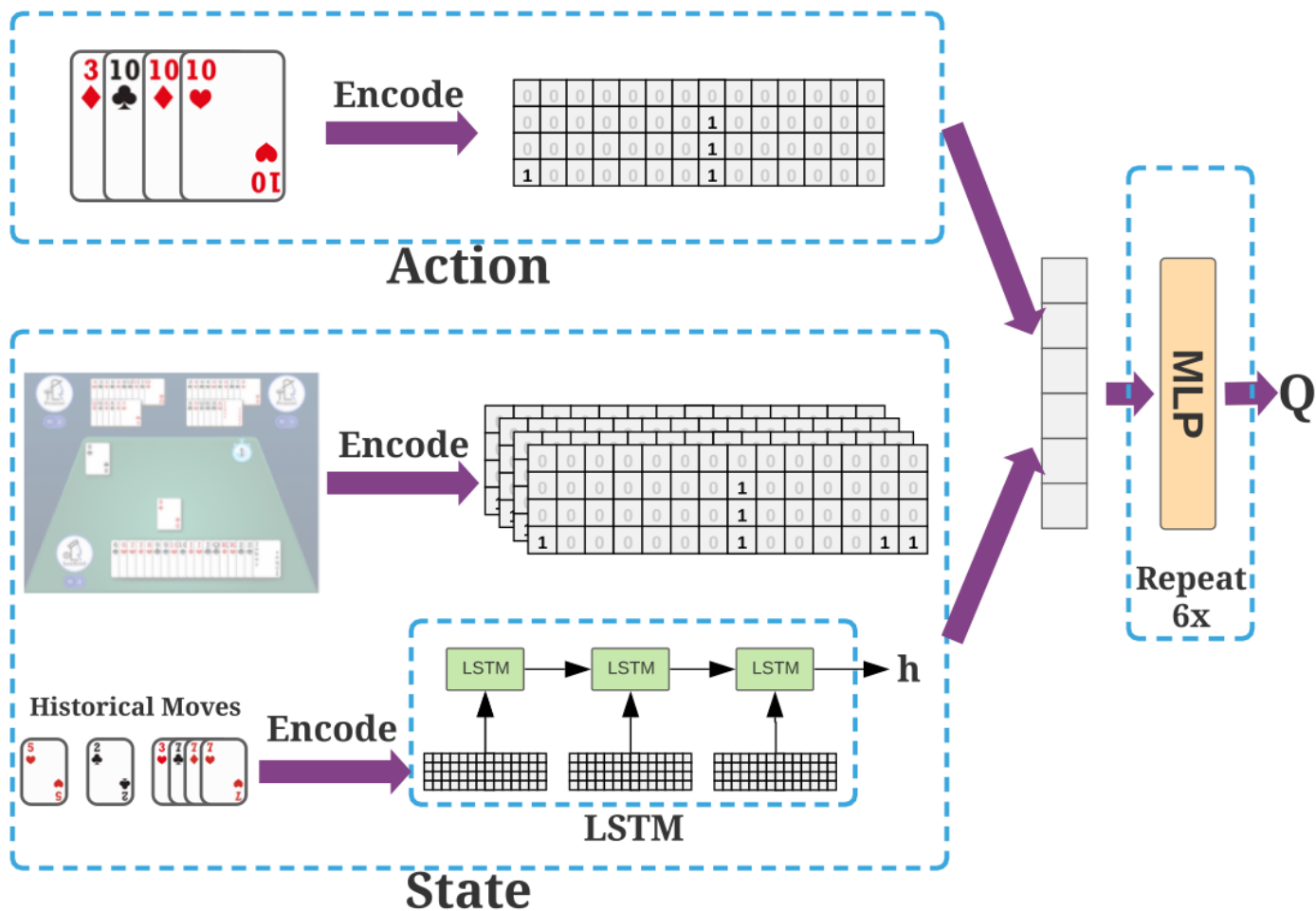
### 参考方案1：应用传统机器学习模型

1. 机器学习基础：深入掌握机器学习的基本概念，特别是监督学习算法如决策树和随机森林。
2. 定义问题：根据局势，预测下一步最优的牌型（单张、对子、顺子等）。这可以被看作一个分类问题，其中输入特征是当前的游戏状态（包括所有可见的手牌和已经出的牌），输出是一系列标签（分别映射阶段一所设计的不同牌型）。
3. 数据收集与处理：机器学习模型的训练需要大量数据。对于AI攒蛋，你需要收集游戏数据，数据可以来源于自己的对局，可以来源于网上已有的统计结果，也可以使用强化学习的模拟环境进行机器对弈。

另外还需要对获得的数据进行处理，步骤可能包括：

- 特征工程：从原始数据中提取有用的特征，如手牌的牌型组合的频率等。
  - 数据清洗：处理缺失值、异常值等。
4. 模型选择、训练和评估：
    - 根据问题的性质和数据的特点，选择适合的机器学习模型。对于分类问题，常见的模型包括：
      - **决策树**和**随机森林**等：直观且易于理解，适合处理有明显规则的分类问题。实现可参考[sklearn帮助文档](#)。
      - **Transformer-Encoder**：强大灵活，能够捕捉复杂的非线性关系，但需要大量数据和计算资源。实现可参考[transformers帮助文档](#)。
    - 评估模型性能，使用诸如准确率、召回率等指标，并通过交叉验证等方法进行模型的优化和调整。

### 参考方案2：强化学习



## 步骤

1. **强化学习基础**：掌握强化学习的基本概念和算法，如Q-learning、PPO等。
2. **环境搭建**：使用一个根据之前编写的规则，修改RL-Card工具包提供的游戏环境，这将作为强化学习模型的训练场所。
3. **数据准备和特征工程**：使用模拟的掇蛋游戏来生成数据并设计有效的特征来表示斗地主中的牌局状态。
4. **模型训练**：设计并实现强化学习模型，通过与环境交互学习最优策略。
5. **策略优化和测试**：对训练好的模型进行评估和优化，通过与人类玩家或其他AI策略的对抗测试，评估学习到的策略的实际效果。

除此之外，同学们可以探索将上半学期开发的基于规则的策略与下半学期通过强化学习训练出的策略进行集成的方案，以达到更好的效果。

## 参考材料

- 阶段一实现可参考：
  - **【Python实现掇蛋逻辑】** <https://github.com/LSTM-Kirigaya/egg-pancake/blob/master/AIAction.py>
  - **【斗地主牌型策略设计思路】** <https://blog.csdn.net/sm9sun/article/details/70787814>

- 【C++实现的规则斗地主】 <https://github.com/songbaoming/DouDiZhu>，主要参考 [player.cpp](#) 中的牌型设计方案
  - 阶段二实现可参考：
    - 【攒蛋出牌频率数据集】 <https://github.com/CavefireTech/EggInjectStatistics>
    - 【强化学习环境】 [卡牌游戏环境工具包](#)
    - 【使用决策树实现斗地主】 Determinization and information set Monte Carlo Tree Search for the card game Dou Di Zhu, CIG 2011 [论文参考](#)
    - 【强化学习实现斗地主】 Zha, Daochen et al. “DouZero: Mastering DouDizhu with Self-Play Deep Reinforcement Learning.” ICML (2021). [论文参考](#) [代码参考](#)
- 

## 2. 贪吃蛇大作战【张逸凯】

### 背景介绍

这是一款超好玩的休闲竞技游戏，不仅比拼手速，更考验你的策略！在贪吃蛇大作战的世界中，每个人在初始都化身为一条小蛇，通过不断努力变得越来越长，最终制霸一方！

- 游戏玩法
  - 控制方向键走位你的小蛇，吃掉地图上的“食物”，就会变长。
  - 随机出现追猎者！快躲开它。
  - 随机出现虫洞！穿越到另一端。
  - **详细要求：**
    - 游戏界面的大小是 32x32，蛇的初始长度为 5x1。
    - 两种玩法模式：
      - 单人模式仅一位玩家，游戏结束标志为玩家死亡。
      - \*双人模式两位玩家在一台机子上一起玩，游戏结束标志为一方玩家死亡。
    - “食物”大小为1x1：吃完后蛇头部分长度+1。
    - \*长按加速键，巧妙地走位。
    - \*随机出现另一条小蛇（追猎者），它在地图上随机移动，初始长度为 5x1，它也可以吃食物，并且可以穿过自己的身体。
    - \*随机出现虫洞，在地图上为两个 1x1 的洞，蛇从一个洞进去之后从另外一个出来。
    - 单人模式中：蛇头碰到障碍物、自己的蛇身以及追猎者就会死亡。

- \*双人模式中：蛇头碰到障碍物、别人的蛇身以及追猎者就会死亡。

## 阶段一任务

- 请实现未加星标\*的要求。
- 请实现简单的贪吃蛇智能体算法，使其尽量长，可以参考 BFS 或 A\* 搜索算法。

## 阶段二任务

- 请实现加星标\*的要求。
- 请邀请一位同学（非舍友）一起玩你设计的双人模式。
- 请尝试将一些关键时刻的游戏界面转化为自然语言，输入任意大语言模型（例如ChatGPT、通义千问等），问问它们的决策，Prompt的样式不限，例如：
  - 假设你是一个贪吃蛇玩家，现在蛇的位置是xxx，在xxx的位置有一个xxx，能带来的效果是xxx，在xxx有xxx ..... 请给出xxx的决策？

# 3. 五子棋对弈【李冠容】

## 背景介绍

五子棋起源于中国，是一种两人对弈的纯策略型棋类游戏。双方分别使用黑白两色的棋子，轮流落子在棋盘竖线与横线的交叉点上，先形成五子连珠者获胜（横线，竖线，斜线均可）。

## 阶段一任务：

- **搭建游戏环境，完成五子棋游戏的正常运行和胜负判断**
  - 在游戏开始时，随机指定黑棋或白棋先行，并且能够指定棋盘的大小
  - 按照玩家指定位置放置棋子，并且判断放置位置的合法性；如果放置位置不合法，输出提示并要求玩家重新选择放置位置
  - 判断游戏的输赢，并给出结果
  - 可视化的游戏界面，命令行即可
  - （在此之后的目标仅供学有余力的同学参考，不作为要求）
  - 使用Tkinter、pyQT等库实现GUI，可以实现鼠标点击落子，参考：[tkinter使用教程](#)、[tkinter官方文档](#)、[pyQT安装及使用](#)、[pyQT教程](#)、[pyQT6官方文档](#)
- **实现基于规则的简单对弈AI**



实现人机对弈的功能，玩家和AI随机一方先行，AI在尽可能封堵玩家的同时，尽量自己达到五子连珠目标。

以下实现的AI二者选一即可

- 实现基于规则的基础AI。基于五子棋的知识，设计规则指导AI的运行
- 实现基于BFS、或DFS的暴力搜索AI，尝试调节搜索深度，探索搜索式AI的能力与运行时间的关系

**提示：**

- 设计五子棋规则时，请考虑规则的完备性，即所有情况下，AI均会选择下棋的位置；以及规则的合理性，尽量不要让AI随机下棋。
- 为了简化设计，规则可以优先考虑封堵，仅在必要时考虑达成自己目标，如遇到双三、双四或长连（连续的三个以上四子）等必定取胜的情况
- 考虑的规则可以有：单三，单四，双三、双四或长连（连续的三个以上四子）等等
- 基于BFS，DFS等搜索方法时，由于搜索空间巨大，AI运行时间可以过长。可以考虑采用启发式规则的方式，减少搜索空间，例如贴合对方棋子的位置优于远离对方棋子位置、从可选的位置中随机选择几个进行进一步搜索（类似蒙特卡洛方法）

## 阶段二任务

**背景介绍-博弈树：**

博弈树（game tree）是一种特殊的根树，它是人工智能领域一个重要的研究课题。博弈树可以表示两名游戏参与者之间的一场博弈（游戏），他们交替行棋，试图获胜。树中的每个顶点都对应于棋盘的一个布局（configuration）。我们的目标是在每个顶点中对应的棋盘布局中，寻找最佳的落子点。

因此，我们需要一套对棋盘布局的量化评估方法，该评分方法需要满足：对于任一棋局，对黑方越有利，则评分越高；对白方越有利，评分越低（黑白房颠倒也可以）；同时要有估算未结束的棋局的能力。我们可以将这个函数称为价值评估函数（Critic）。

此外，我们在每一步寻找最佳落子点的过程中，也需要一个选择策略，例如贪婪策略等等。我们将这个策略称为策略选择函数（Actor）。

本阶段中，要求实现基于机器学习算法的五子棋AI，能够实时给出AI判断的胜率。

**步骤：**

- 实践不同的博弈树优化策略，例如 $\alpha$ - $\beta$ 剪枝、蒙特卡洛树搜索（MCTS）等等（任选其一即可）
- **可选方案1-机器学习算法：**

优化策略中采用的价值评估函数，可以任选其一实现。通常策略选择函数为贪婪策略，即落子到价值评估函数中评分最高的位置。

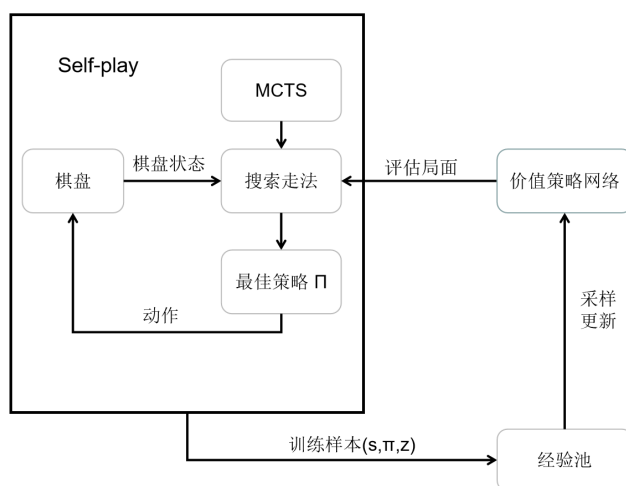
- 自己建立固定的价值评估函数，为不同的棋局给予不同的分数。例如：单三的分数高于单二；双三，双四等禁手（必定胜利）给予最高分数，不封堵对方的禁手给予最低分数等等。



- 通过棋谱数据（棋谱数据见最后的补充文件），训练回归算法（如logistic回归，支持向量回归，K近邻算法，卷积神经网络等等），实现价值评估函数。

### 建议：

- 棋谱数据需要进行清洗，去除其中的缺失值和异常值
- 如果自行训练价值评估函数，建议将棋盘大小限制为8x8或者10x10
- 如果需要进行梯度下降的训练过程，建议增加L2正则化项，以避免过拟合。
- 通过旋转和翻转，在极少代价的情况下扩增数据集。
- 可选方案2-强化学习算法（类似alpha zero）**，相对较难，建议学有余力的同学选择：



该方法类似于alpha zero，为蒙特卡洛树搜索（MCTS）与神经网络的结合。

### 步骤：

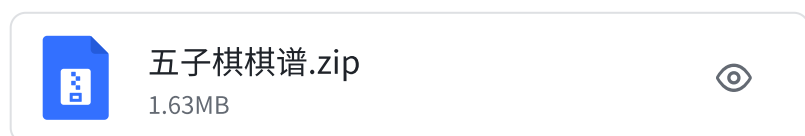
- 实现MCTS，作为运行算法的基础框架。
- 建立策略价值网络，即策略函数与价值函数共享部分网络参数。
- 将MCTS中的策略选择过程和价值评估过程替换为策略价值网络。
- 实现五子棋的自我对弈功能，用于收集五子棋对局数据。数据应当包含每一步执行时的棋盘状态、选择执行策略时每种策略的概率，以及最终的胜者。
- 自我对弈，每一轮更新策略价值网络后，均进行自我对弈采集新的数据（建议选择1-3场即可），然后进行下一轮的训练和更新。

### 建议：

- 建议策略价值网络实现为共享的三层卷积神经网络CNN，叠加独立策略线性层以及价值线性层组成。

- 如果训练强化学习，在训练阶段棋盘大小建议设置为6x6，以保证训练速度。如果同学的PC有较为强大的GPU，设置为8x8也可，不建议使用更大的棋盘大小。在实际使用时，可以尝试增大棋盘大小，验证AI的表现。
- 训练轮次可以通过观察模型的损失函数值确定，通常500轮左右的自我对弈已经能够得到足够好的结果。
- 可以从初始状态直接通过自我对弈训练策略价值网络（类似alpha zero），也可以将其在棋谱数据上经过训练之后，再通过自我对弈的数据训练（类似alpha go）。
- 通过旋转和翻转，在极少代价的情况下扩增数据集。

补充材料：



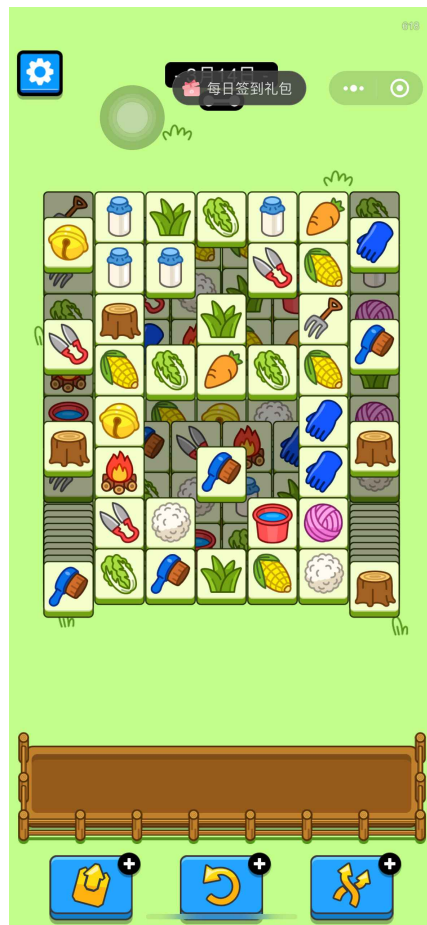
大家也可以自行寻找棋谱，在此提供一个可行的网站：

[棋谱库链接](#)、[棋谱标准说明书](#)

## 4. 羊了个羊【李冠容、赖哲剑】

### 背景介绍：

羊了个羊是一款在传统三消游戏基础上加以修改但是的新游戏。该游戏有两部分组成，分别为待处理卡牌区和固定容量的暂存区。玩家可以通过点击将卡牌区中的卡牌移动到暂存区中，如果暂存区中有三个完全一致的卡牌，则可以将其消除。卡牌区由多层组成，没有移开上层对应位置的卡牌时，无法移动下层的卡牌，游戏目标为消除卡牌区中的所有卡牌。

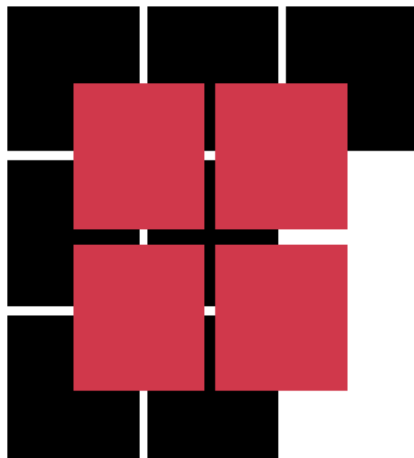


### 阶段一任务：

## 搭建环境

这一阶段我们将搭建简化版的羊了个羊游戏环境，游戏环境如下所示：

- 卡牌类别总数为10，分别用数字0-9表示。
- 具有卡牌区和待处理区，在每一步移动卡牌之后，都能够展示当前的卡牌区和暂存区。其中，卡牌区按照从上层到下层分层展示，不可见的卡牌可以表示为X，已经被移出卡牌区的卡牌可以表示为O。
- 不同层之间交错半个卡牌的距离排列，如图所示。



- 只有在下层卡牌对应的上方四个卡牌任意一个移开后，下方卡牌的数值才可见。
- 只有在下层卡牌对应的上方四个卡牌全部一个移开后，下方卡牌才可以移动。
- 可以自行定义卡牌区的层数，每一层卡牌的数量，以及暂存区容量
- 卡牌排列随机生成

### 实现基于贪心算法的AI

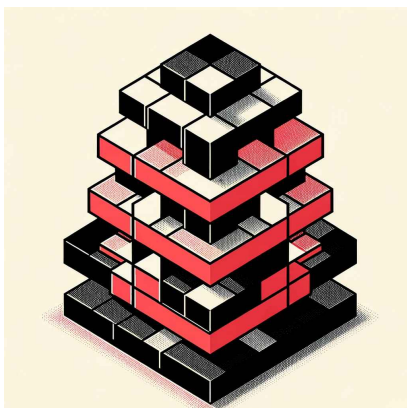
- 实现基于贪心算法的AI，解决上述羊了个羊问题。你的算法不应该知晓不可见的卡牌是什么
- 实现游戏运行过程的可视化，使用命令行实现即可
- （不做强制要求）使用Tkinter、pyQT等库实现GUI，参考：[tkinter使用教程](#)、[tkinter官方文档](#)、[pyQT安装及使用](#)、[pyQT教程](#)、[pyQT6官方文档](#)

### 提示：

请考虑任务无法完成的情况，即该任务无解的情况。

## 阶段二任务：

- 实现目标：实现基于机器学习算法的游戏AI（阐述具体的建议和要求）
  - 将阶段一中的卡牌层数深化，阶段一中卡牌只有两层，现在我们需要你针对卡牌含有若干层，如下面这个示意图所示



- 将表面你所可以看到的卡牌信息抽象成一些可以学习的特征（例如将每一张卡牌编码后，使用CNN网络学习上层卡牌与下层可见卡牌之间的关联信息）
- 强化学习算法，比如Q-learning或DQN，可以通过接收策略网络的输出作为输入，来学习每个动作的价值函数。
- 根据学习到的价值函数选择期望回报最大的动作，即选择哪张卡牌移动到暂存区。
  - 如果暂存区已有两张相同的卡牌，则优先移动同样数值的第三张卡牌以消除。
  - 如果暂存区未满足且没有可消除的组合，则选择对未来回报可能最大的卡牌进行移动。

（具体强化学习策略需要自己设计，但在羊了个羊游戏中一种往往有效的策略是有两张相同，则优先选择相同的第三卡牌，因此可能规则方法拥有更好的表现）

#### 参考材料：

- 【羊了个羊环境搭建参考Vue】 <https://github.com/chenxch/xlegex?tab=readme-ov-file>

## 5. 多API的学术系统【张业勤】

### 阶段一任务：学术问答

#### 背景介绍：

这个学术问答系统目标是为研究人员、学生和学术爱好者提供一个能够快速获取学术信息的工具。用户可以查询特定论文的细节、作者信息、引用量和参考文献，以及特定学者的详细信息和他们发表的论文。从头构建一个AI系统理解用户的所有查询是困难的，幸运地是，大语言模型LLM（Large Language Models）可以帮助我们做这件事。本课题要求通过整合大语言模型和学术API（如Semantic Scholar API可以提供学者和论文的详细信息），提供一个综合的、易于访问的学术信息查询平台。

#### 任务设计

##### 1. 查询处理阶段：

- 用户输入查询请求（论文详情、作者信息等），如“请问李飞飞最新发表的论文有哪些？”。
- 系统利用LLM理解查询的类型并确定需要调用的API。

##### 2. 多轮信息检索阶段：

- 根据查询类型，系统通过适当的API检索信息。
- 对于论文查询，系统需获取论文具体内容、作者信息、引用量和参考文献。
- 对于学者查询，系统需获取学者相关信息和其发表的论文列表。

##### 3. 信息整合与展示阶段：

- 系统通过LLM整合来自不同API的信息，综合回复用户的输入查询请求。

## 任务步骤

1. 首先需要根据API的功能和参数，设计输入大语言模型的指令。指令可以分为三个部分，例如：

- 任务介绍：请利用下列API回答下面的学术问题。
- 学术问题（用户查询）：请问李飞飞最新发表的论文有哪些？
- API介绍：包括解决该学术问题涉及的工具/函数的描述以及输入参数的描述，如：

```
{  
  "description": "获取论文的摘要，论文作者及其机构",  
  "name": "SearchPublication",  
  "parameters": {  
    "properties": {  
      "personName": {  
        "description": "学者姓名",  
        "type": "string"  
      }  
    }  
  }  
}
```

2. 将指令输入大语言模型，让大语言模型输出该学术问题所需的工具调用，如：

```
"function_call": {  
  "name": "SearchPublication",  
  "arguments": "{ \"personName\": \"Fei Fei Li\"}"  
}
```

3. 基于大模型的工具调用输出，调用第三方API，获得结果：

"李飛飛的著作包括：年份：2023，标题：VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models，引用量：8。年份：2023，标题：The ObjectFolder Benchmark: Multisensory Learning With Neural and Real Objects，期刊：Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)。"

#### 4. 融合第三方API结果，回复学术问题

"李飞飞在斯坦福大学目前发表了以下论文：1. \"VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models\"，发表于2023年，被引用8次。2. \"The ObjectFolder Benchmark: Multisensory Learning With Neural and Real Objects\"，发表于IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)会议。"

## 阶段二任务：学者研究兴趣挖掘

### 背景介绍：

在基础的学术问答功能上，阶段二的目标是通过深度分析学者的发表论文集，揭示其核心研究领域和兴趣点。该功能旨在为研究人员、学生和学术爱好者提供更为精准的学术导向和合作机会发现。

### 任务设计

#### 1. 查询处理阶段：

- 用户提交关于特定学者的查询请求，如查询其研究兴趣。
- 系统准确理解并定位到指定学者，确定适合的数据检索范围。

#### 2. 信息检索阶段：

- 依托于学者的名称或其他标识信息，系统通过连接至学术数据库API，检索该学者发表的所有论文。
- 收集论文的关键数据，如标题、摘要、关键词、发表年份等，为数据挖掘阶段做好准备。

#### 3. 数据挖掘阶段：

- 应用文本分析和聚类算法，如K-means或主题模型算法，对收集到的论文数据进行主题分析。
- 根据论文的关键词、摘要和主题分析结果，识别出学者研究兴趣的聚类中心，揭示其主要的研究领域和兴趣点。

### 任务步骤

1. 根据API的功能和参数要求，构建并提交指令给大语言模型，从而获得所需的工具调用语句。
2. 根据大语言模型提供的指令，激活适宜的第三方API，以获取目标学者的相关论文信息，为数据挖掘阶段做好准备。
3. 对收集到的论文信息应用数据挖掘算法，以科学地分析和识别学者的主要研究兴趣。

## 参考材料



# API介绍

## 1. API 类型介绍：

### a. 论文类：

- i. 论文具体内容
- ii. 论文作者信息
- iii. 论文引用量
- iv. 论文参考文献

### b. 学者类：

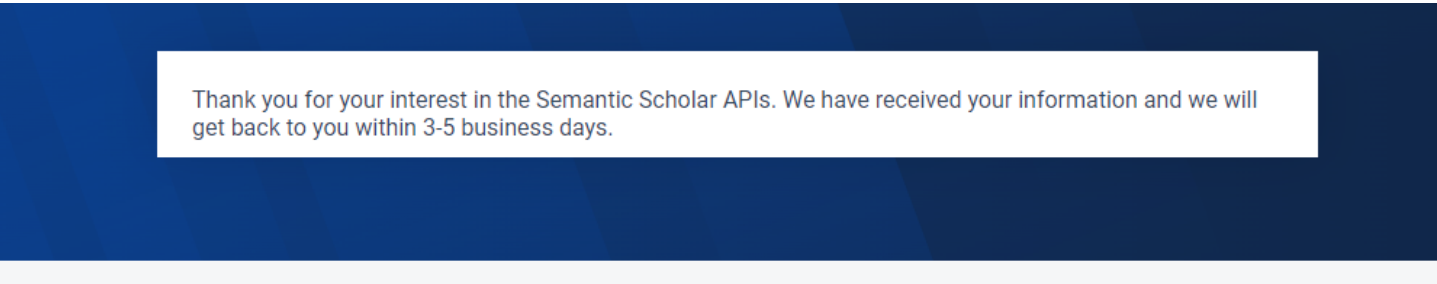
- i. 学者相关信息
- ii. 学者发表的论文

## 2. Semantic Scholar是一个可以查询论文和学者信息的网站，Semantic Scholar API调用样例：

```
1 import requests
2
3 # Define the API endpoint URL
4 url = 'https://api.semanticscholar.org/graph/v1/paper/search'
5
6 # More specific query parameter
7 query_params = {'query': 'quantum computing'}
8
9 # Directly define the API key (Reminder: Securely handle API keys in
  production environments)
10 api_key = 'your api key goes here' # Replace with the actual API key
11
12 # Define headers with API key
13 headers = {'x-api-key': api_key}
14
15 # Send the API request
16 response = requests.get(url, params=query_params, headers=headers)
17
18 # Check response status
19 if response.status_code == 200:
20     response_data = response.json()
21     # Process and print the response data as needed
22     print(response_data)
23 else:
24     print(f"Request failed with status code {response.status_code}:
      {response.text}")
25
```

Semantic scholar API key 申请地址：  
<https://www.semanticscholar.org/product/api#api-key-form>

Semantic scholar API 文档：  
<https://api.semanticscholar.org/api-docs>  
<https://www.semanticscholar.org/product/api/tutorial>



Api-key: BbAaGPZW8n9uIBJfsoEiXnepsnj75sj2wbyCbBp1

限时免费的大语言模型API：

计费管理

| 模型类别        | 模型名称                 | 计量计费说明   | 计量单价               | 免费额度   |
|-------------|----------------------|--|--------------------|--|
| 通义千问        | qwen-turbo           | 模型API调用时按照输入、输出token 总数进行计费：对于中文文本来说，1个token通常对应一个汉字。对于英文文本来说，1个token通常对应3至4个字母。                 | 0.008元/1000 tokens | 开通DashScope即获赠总计 2,000,000 tokens限时免费使用额度，有效期 180 天。   |
| 通义千问        | qwen-plus            | 模型API调用时按照输入、输出token 总数进行计费：对于中文文本来说，1个token通常对应一个汉字。对于英文文本来说，1个token通常对应3至4个字母。                 | 0.04元/1000 tokens  | 开通DashScope即获赠总计 1,000,000 tokens限时免费使用额度，有效期 180 天。   |
| 通义千问        | qwen-max             | qwen-max 模型调用时按照输入、输出token 总数进行计费。目前该模型API限时免费开放中。   | 限时免费               | 限时免费。免费期间限流规则：<br>1、调用频次 <= 60 QPM，每分钟不超过80次API调用；<br>2、Token消耗 <= 130,000 TPM，每分钟消耗的Token数目不超过130,000。  |
| 通义千问        | qwen-max-longcontext | qwen-max-longcontext 模型API调用限时免费中。   | 限时免费               | 限时免费。免费期间限流规则：<br>1、调用频次 <= 1 QPM，每分钟不超过1次API调用；<br>2、Token消耗 <= 300,000 TPM，每分钟消耗的Token数目不超过300,000。    |
| 通义千问-开源系列模型 | qwen-1.8b-chat       | 模型API调用时按照输入、输出token 总数进行计费：对于中文文本来说，1个token通常对应一个汉字。对于英文文本来说，1个token通常对应3至4个字母。                 | 限时免费               | 限时免费。免费期间限流规则：<br>1、调用频次 <= 60 QPM，每分钟不超过120次API调用；<br>2、Token消耗 <= 200,000 TPM，每分钟消耗的Token数目不超过200,000。 |
| 通义千问-开源系列模型 | qwen-7b-chat         | 模型API调用时按照输入、输出token 总数进行计费：对于中文文本来说，1个token通常对应一个汉字。对于英文文本来说，1个token通常对应3至4个字母。                 | 0.006元/1000 tokens | 开通DashScope即获赠总计 1,000,000 tokens 限时免费使用额度，有效期180天。  |
| 通义千问-开源系列模型 | qwen-14b-chat        | 模型API调用时按照输入、输出token 总数进行计费：对于中文文本来说，1个token通常对应一个汉字。对于英文文本来说，1个token通常对应3至4个字母。                 | 0.008元/1000 tokens | 开通DashScope即获赠总计 1,000,000 tokens 限时免费使用额度，有效期180天。  |
| 通义千问-开源系列模型 | qwen-72b-chat        | 模型API调用时按照输入、输出token 总数进行计费：对于中文文本来说，1个token通常对应一个汉字。对于英文文本来说，1个token通常对应3至4个字母。目前该模型API限时免费开放中。 | 限时免费               | 限时免费。免费期间限流规则：<br>1、调用频次 <= 80 QPM，每分钟不超过80次API调用；<br>2、Token消耗 <= 130,000 TPM，每分钟消耗的Token数目不超过130,000。  |

阿里通义  
<https://dashscope.console.aliyun.com/>

讯飞开放平台

平台首页

消息23

工单中心

财务

龙太子

< 返回

英文语音转写zyq

星火认知大模型

星火大模型V1.5

星火大模型V2.0

星火大模型V3.0

星火大模型V3.5

星火知识库

图片生成

图片理解

虚拟人合成

语音识别

语音合成

语音扩展

司法AI产品

人脸识别

实时用量

用量预警通知 已关闭 管理

今日已用token数

剩余token数

QPS

有效期至

立即购买

历史用量

2024-02-14 - 2024-03-14

导出Excel

暂无数据

服务接口认证信息

APPID

APISecret

APIKey

星火大模型V3.5 SDK

SDK名称

版本

操作

Android SDK

Linux SDK

Windows SDK

IOS SDK

Web

服务名称

类型

接口地址

操作

版权所有 © 科大讯飞股份有限公司 皖ICP备05001217号-71 皖公网安备 34019202000117号

科大讯飞

科大讯飞

iFLYOS

AI营销云

讯飞星火v3.5 10w token免费

<https://console.xfyun.cn/services/bm35>

讯飞开放平台

立即注册 微信扫码 手机快捷登录 账号密码登录 +86 获取验证码 未注册的手机号将自动注册。勾选即代表您同意并接受 服务协议 与 隐私政策 登录 帮助 Help 快速指引 视频教程 交流论坛 用户中心 User Center 乐享会员中心 语音云控制台 添加新应用 我的应...

相关知识

大语言模型简介：[https://zhuanlan.zhihu.com/p/675906726?utm\\_psn=1751628115309367296](https://zhuanlan.zhihu.com/p/675906726?utm_psn=1751628115309367296)

大语言模型调用API/工具简介：[https://zhuanlan.zhihu.com/p/672703922?utm\\_psn=1751630479218999296](https://zhuanlan.zhihu.com/p/672703922?utm_psn=1751630479218999296)

大语言模型调用API/工具样例：[https://github.com/THUDM/ChatGLM3/tree/main/tools\\_using\\_demo](https://github.com/THUDM/ChatGLM3/tree/main/tools_using_demo)