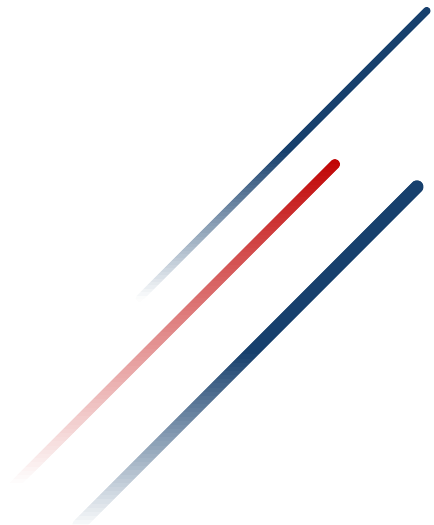


人工智能

实验1-搜索策略pacman



实验安排

- ◆ 实验共10学时，3个实验项目，总成绩30分（30%）
 - ◆ 每个实验评分点包括：代码、报告、考勤

实验内容	分值	学时
实验1：搜索	12分	4
实验2：深度学习	12分	4
实验3：强化学习	6分	2

- ◆ 指导书gitee地址：<https://gitee.com/hitsz-cslab/AI>
- ◆ 课程冲突同学可私聊实验老师换到其他班级



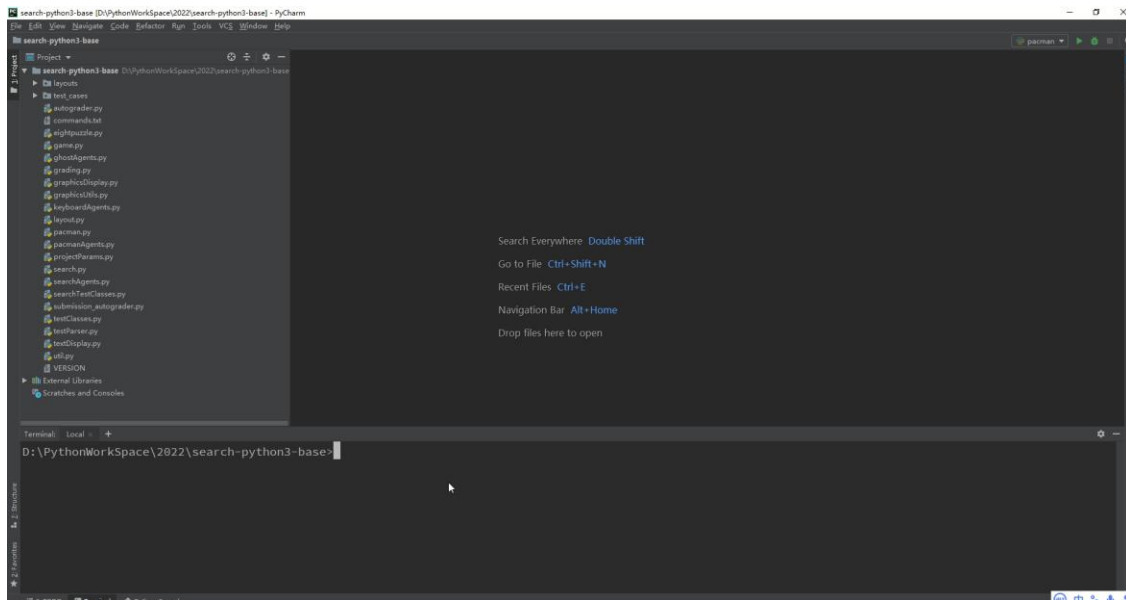
实验内容

使用搜索算法，编写代码解决吃豆人（Pacman）游戏中定义的问题1-8。

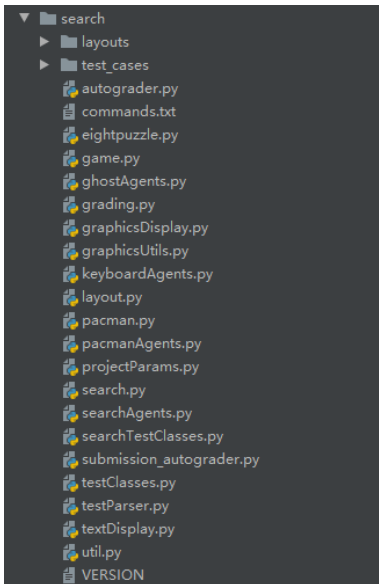


Pacman

Pacman是加州大学伯克利分校开源的人工智能实验项目
运行python pacman.py 进入交互模式



文件说明



需要阅读代码的文件

文件	主要功能
pacman.py	吃豆人游戏的主程序
game.py	吃豆人游戏的运行逻辑
util.py	搜索策略可以用到的数据结构

需要完善代码的文件

文件	主要功能
search.py	待实现的搜索算法
searchAgents.py	待实现的智能体和相关Problem



Pacman代码阅读

- ❑ 通过pdb等工具单步调试查看代码的运行过程
 - 找到函数的入口
 - 边运行边阅读源代码
- ❑ pacman项目较复杂，很多中间逻辑可以略过
 - 从searchAgents.py开始：
 - ❑ 重点SearchAgent、PositionSearchProblem两个类



Python学习

- python基础需自主学习，课程所有实验使用python完成
 - 版本要求为python3.6及以上
- 学习重点：基本的数据类型、函数、类、模块、包导入
 - [python官方教程中文版](#)

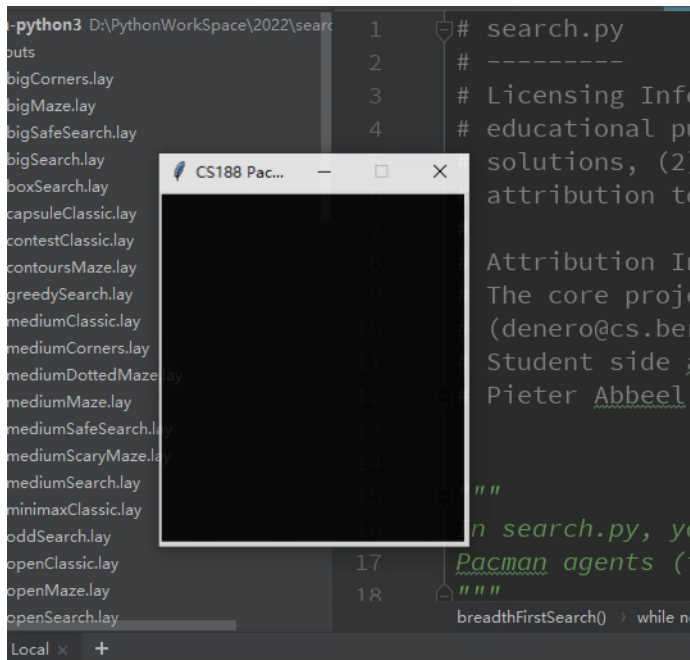


找到特定位置的豆子：问题1-4

❑ 问题1-4中地图中只有1个目标豆子

❑ 实现深度搜索算法后，运行以下命令：

```
python pacman.py -l tinyMaze -p SearchAgent
```



找到特定位置的豆子：问题1-4

Q1: 应用dfs找到一个特定的位置的豆

```
python pacman.py -l mediumMaze -p SearchAgent
```

```
python pacman.py -l bigMaze -z .5 -p SearchAgent
```

Q2: 应用bfs找到一个特定的位置的豆

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
```

```
python pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5
```

Q3: 应用代价一致算法找到一个特定的位置的豆

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
```

Q4: 应用A* 算法找到一个特定的位置的豆子

```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a
```

```
fn=astar,heuristic=manhattanHeuristic
```



常用的命令参数

指定使用的agent类型。默认是1，比1大则是放大，比1则是缩小。

```
python pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5
```

迷宫选项参数，从layouts/目录下加载指定类型地图

传递给agent的参数值。以字符串的形式，如果有多个取值以逗号分隔



自动测评

□ autograder.py

- 会依次测试每个问题的完成情况
- 也可-q 指定某个问题的测试
- 要求使用util.py文件中提供的数据结构

```
D:\PythonWorkSpace\2022\search-python3>python autograder.py
autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib; see the
  import imp
Starting on 3-29 at 17:10:40

Question q1
=====
*** PASS: test_cases\q1\graph_backtrack.test
***   solution:      ['1:A->C', '0:C->G']
***  expanded_states: ['A', 'D', 'C']
*** PASS: test_cases\q1\graph_bfs_vs_dfs.test
***   solution:      ['2:A->D', '0:D->G']
***  expanded_states: ['A', 'D']
*** PASS: test_cases\q1\graph_infinite.test
***   solution:      ['0:A->B', '1:B->C', '1:C->G']
***  expanded_states: ['A', 'B', 'C']
*** PASS: test_cases\q1\graph_manypaths.test
***   solution:      ['2:A->B2', '0:B2->C', '0:C->D', '2:D->E2', '0:E2->F', '0:F->G']
***  expanded_states: ['A', 'B2', 'C', 'D', 'E2', 'F']
*** PASS: test_cases\q1\pacman_1.test
***   pacman layout:      mediumMaze

Finished at 17:11:00

Provisional grades
=====
Question q1: 3/3
Question q2: 3/3
Question q3: 3/3
Question q4: 3/3
Question q5: 3/3
Question q6: 3/3
Question q7: 5/4
Question q8: 3/3
-----
Total: 26/25

Your grades are NOT yet registered. To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project
```



统一的图搜索框架

```
function GRAPH-SEARCH(problem, frontier) return a solution or failure
  reached ← an empty set
  frontier ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), frontier)
  while not IS-EMPTY(frontier) do
    node ← POP(frontier)
    if problem.IS-GOAL(node.STATE) then return node
    end if
    if node.STATE is not in reached then
      add node.STATE in reached
      for each child-node in EXPAND(problem, node) do
        frontier ← INSERT(child-node, frontier)
      end for
    end if
  end while
  return failure
```

reached对应closed表
frontier对应open表



找到所有的角落：问题5-6

Q5：找到所有的角落——基于BFS的角落问题（完成CornersProblem）

```
python pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
```

```
python pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
```

Q6：找到所有的角落——基于A*的角落问题（构建启发函数）

```
python pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5
```

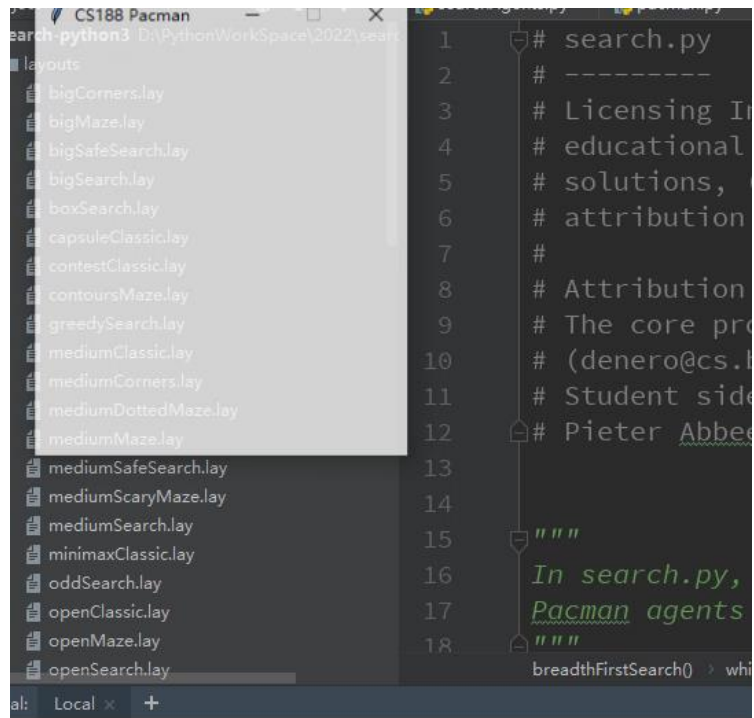
Number of nodes expanded	Grade
more than 2000	0/3
at most 2000	1/3
at most 1600	2/3
at most 1200	3/3

不同的启发函数效率不同，扩展的节点数不同

请尝试两种不同的启发函数！！！！



找到所有的角落：问题5-6



```
1 # search.py
2 # -----
3 # Licensing Information:
4 # This code is released under a Creative Commons
5 # Attribution 4.0 International license.
6 # You can find a copy at https://creativecommons.org/licenses/by/4.0/
7 #
8 # Attribution: This code was adapted from the Pacman project
9 # The core project is licensed under a Creative Commons
10 # Attribution 4.0 International license.
11 # Student side of the license: https://creativecommons.org/licenses/by/4.0/
12 # Pieter Abbeel
13
14 """
15 In search.py,
16 Pacman agents
17 """
18
19 def breadthFirstSearch():
20     while True:
```

吃掉所有的豆子：问题7-8

Q7：吃掉所有的豆子——食物搜索问题（A*算法，构造启发函数）

```
python pacman.py -l trickySearch -p AStarFoodSearchAgent
```

Number of nodes expanded	Grade
more than 15000	1/4
at most 15000	2/4
at most 12000	3/4
at most 9000	4/4 (full credit; medium)
at most 7000	5/4 (optional extra credit; hard)

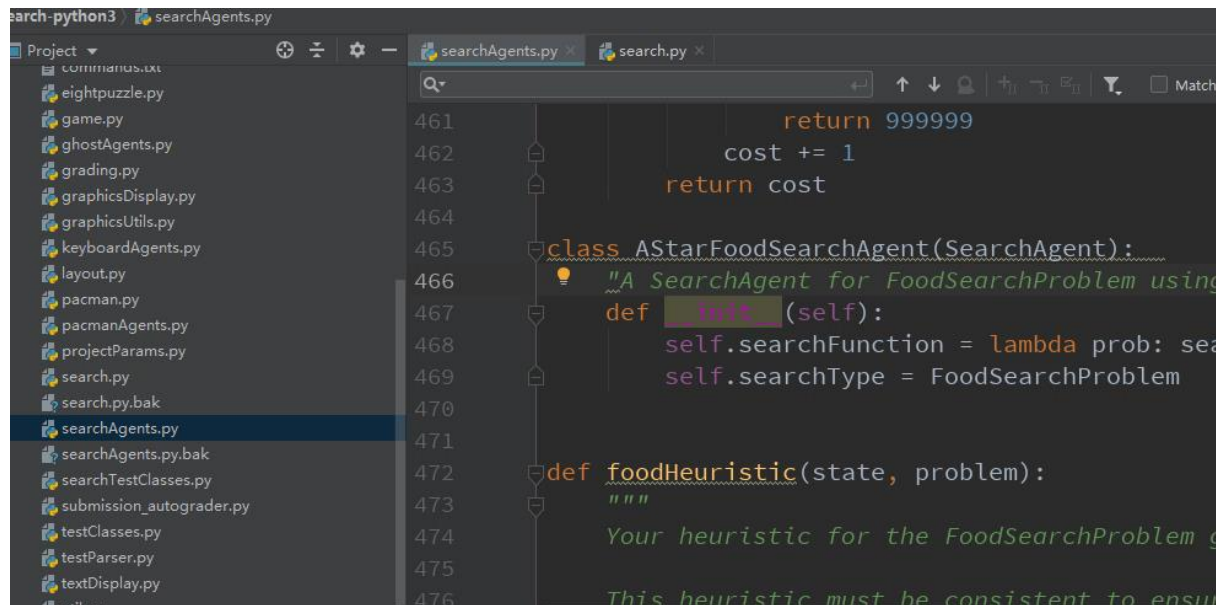
Q8：次最优搜索——任意食物搜索问题（优先吃最近的豆子加快搜索）

```
python pacman.py -l trickySearch -p ClosestDotSearchAgent （跟Q7同样的测试地图）
```

```
python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5
```



吃掉所有的豆子：问题7-8



```
search-python3 > searchAgents.py
Project
├── commands.txt
├── eightpuzzle.py
├── game.py
├── ghostAgents.py
├── grading.py
├── graphicsDisplay.py
├── graphicsUtils.py
├── keyboardAgents.py
├── layout.py
├── pacman.py
├── pacmanAgents.py
├── projectParams.py
├── search.py
├── search.py.bak
├── searchAgents.py
├── searchAgents.py.bak
├── searchTestClasses.py
├── submission_autograder.py
├── testClasses.py
├── testParser.py
└── textDisplay.py

461         return 999999
462         cost += 1
463         return cost
464
465 class AStarFoodSearchAgent(SearchAgent):
466     """A SearchAgent for FoodSearchProblem using
467     def __init__(self):
468         self.searchFunction = lambda prob: search
469         self.searchType = FoodSearchProblem
470
471
472 def foodHeuristic(state, problem):
473     """
474     Your heuristic for the FoodSearchProblem g
475
476     This heuristic must be consistent to ensu
```



实验报告提交

- ◆ 代码只需要提交search.py、searchAgents.py
- ◆ 报告内容及格式要求见实验报告模板，导出成pdf与代码打包提交
- ◆ 提交截止时间第13周周日，具体见作业提交系统

