上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

启发搜索与对抗搜索算法实践

马超

2025年3月22日

启发搜索算法实践 — 搜索算法在吃豆人中的应用

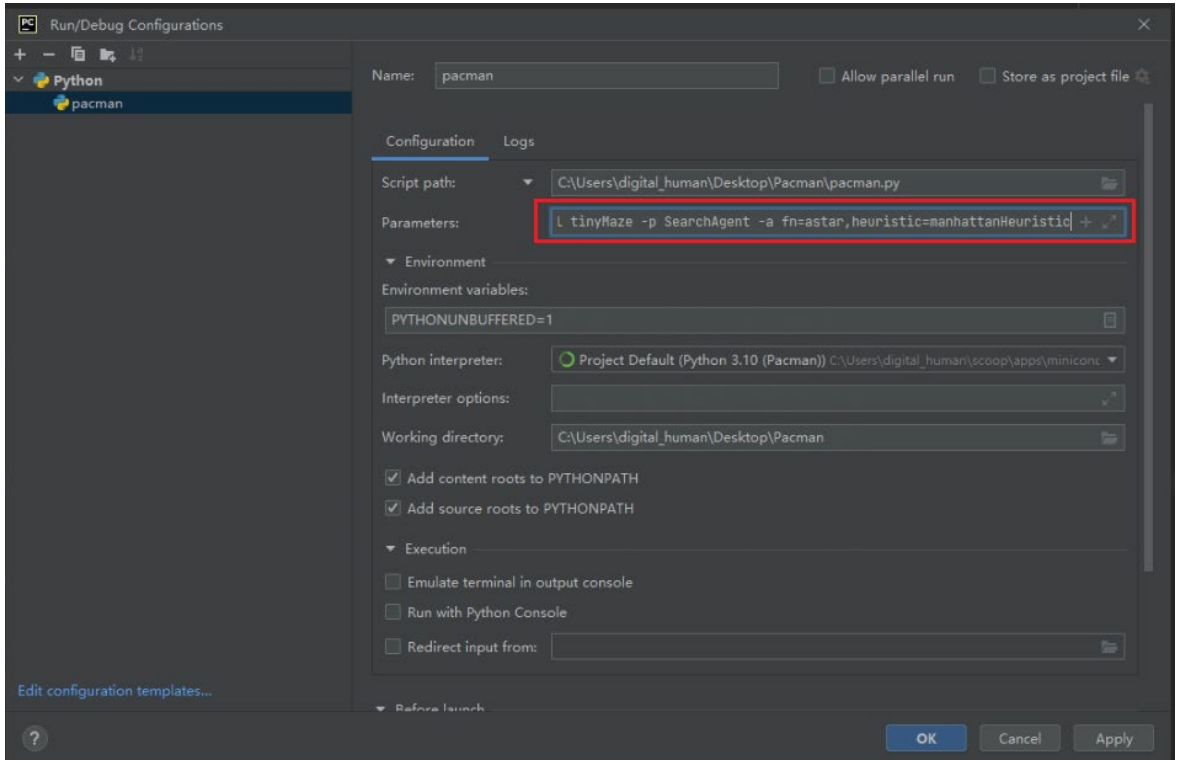
- 1.在代码中TODO部分补全search.py中启发搜索算法的代码
  - aStarSearch

```
181  def aStarSearch(problem, heuristic=nullHeuristic):
182      """Search the node that has the lowest combined cost and heuristic first."""
183      "*** YOUR CODE HERE ***"
184      closed = set()
185      fringe = util.Stack() # TODO util.Queue() or util.PriorityQueue()
186      node = {
187          'position': problem.getStartState(),
188          'path': [],
189          'cost': 0
190      }
191
192      # value of the heuristic function is heuristic(node['position'], problem)
193      fringe.push(node, node['cost'] + heuristic(node['position'], problem))
```

启发搜索算法实践 — 搜索算法在吃豆人中的应用

▪ 2. 代码测试

测试方法：在Parameters处: -l tinyMaze -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic or euclideanHeuristic

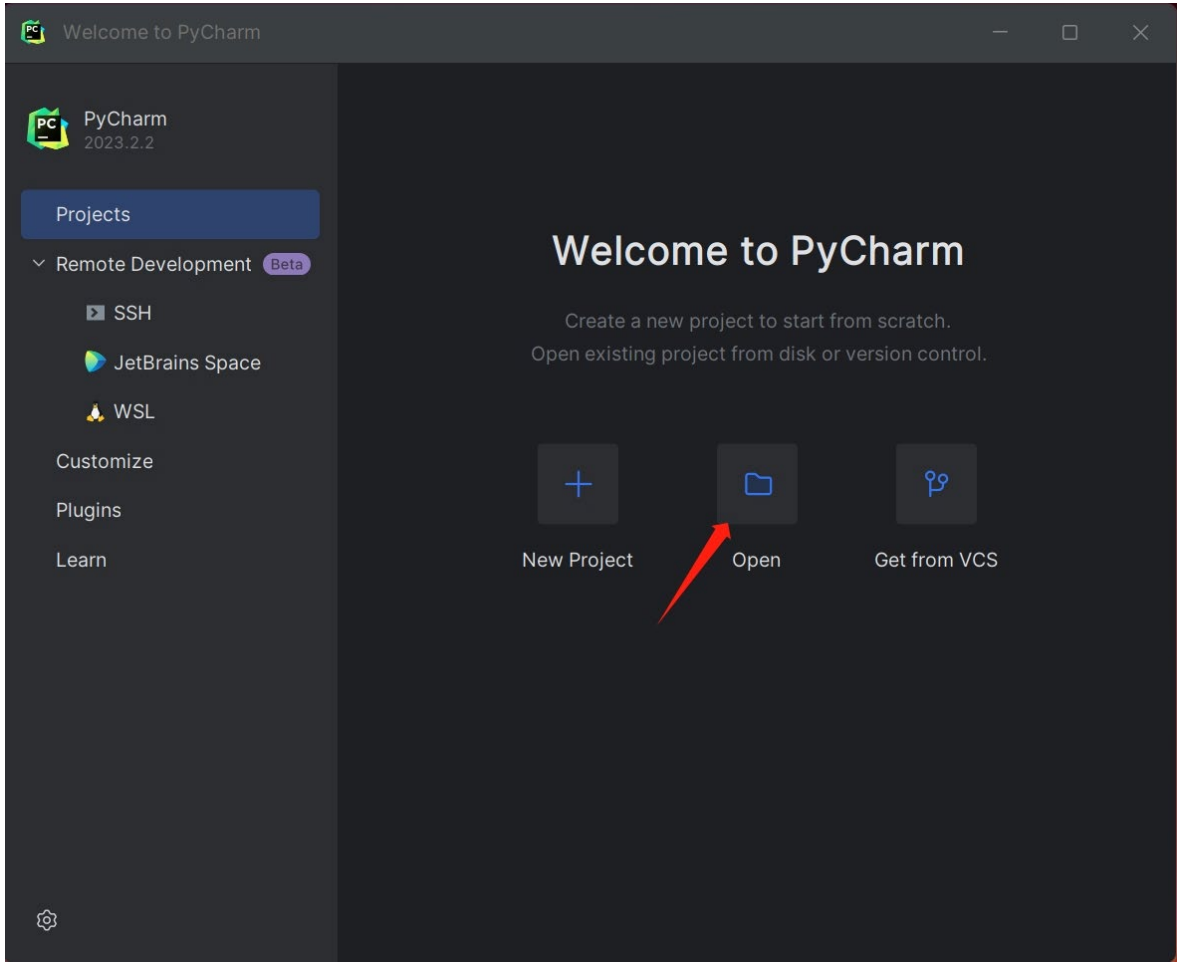


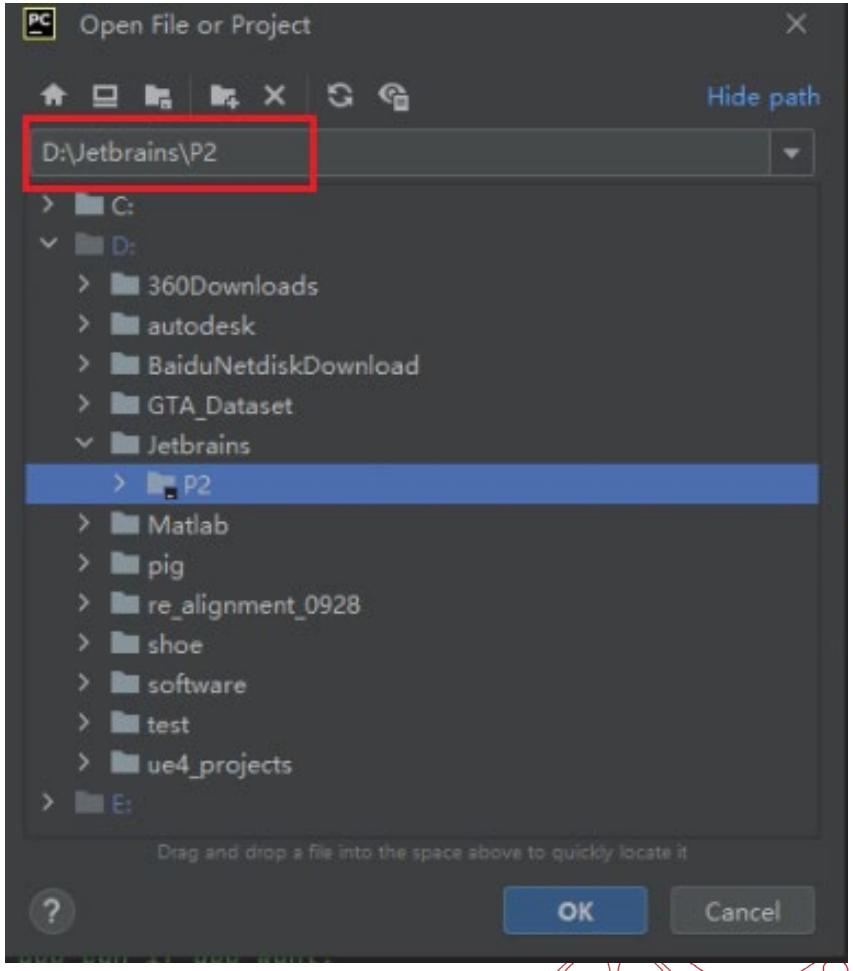
启发搜索算法实践 — 搜索算法在吃豆人中的应用

- 2. 代码测试
  - 测试时可以将 –l 之后的tinyMaze换成mediumMaze或者bigMaze来测试不同的地图，在使用bigMaze时可加上：-z 0.5来显示完整的地图，并使用--frameTime 0来加快Pacman的移动速度。
  - 将 –l 后的 tinyMaze 替换为 mediumMaze 和 bigMaze，测试三种搜索算法在三种不同的地图上的搜索表现
  - 要求：记录实验中的搜索到路径的长度“total cost”以及搜索的节点数“nodes expanded”，将结果记录在result.docx
  - 注：在测试 bigMaze 时，可加上 -z 0.5 来显示完整的地图

对抗搜索算法实践 — 搜索算法在吃豆人中的应用

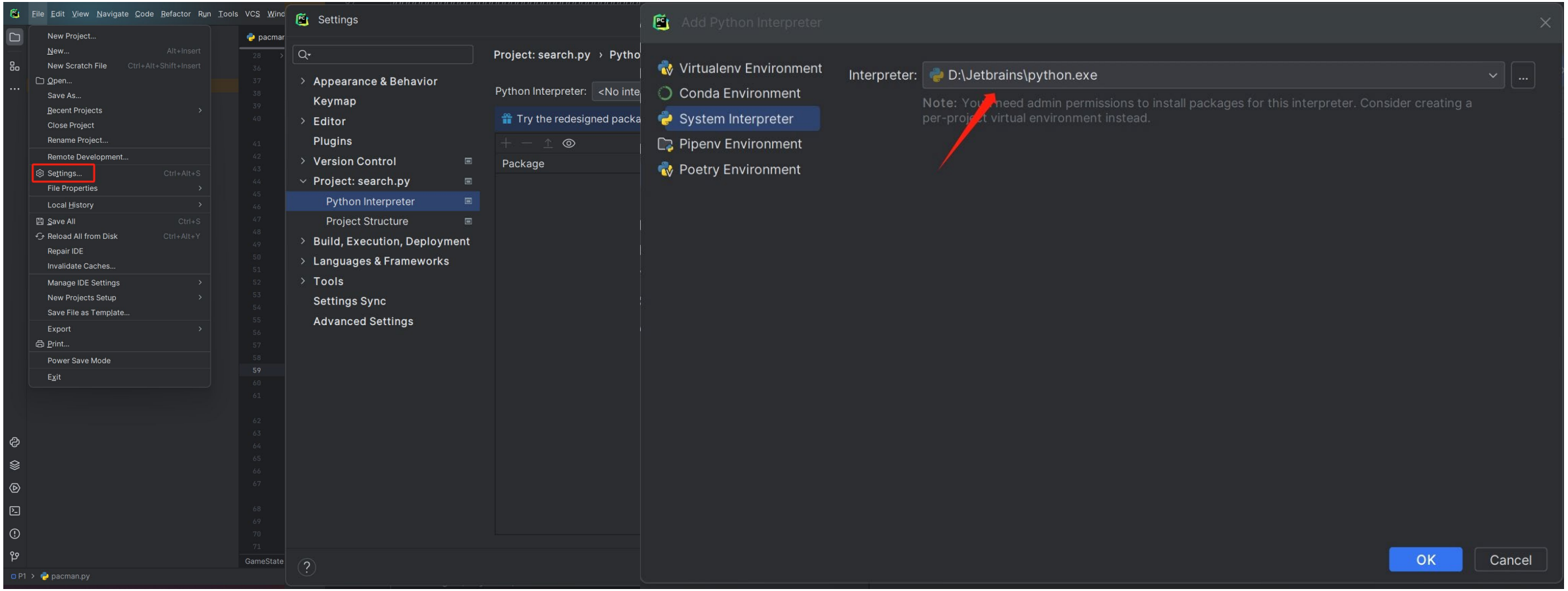
- 下载代码库P2，解压，放到 D:\Jetbrains\P2 路径
- 使用PyCharm打开D:\Jetbrains\P2文件夹，并点击pacman.py文件





对抗搜索算法实践 — 搜索算法在吃豆人中的应用

- 配置python环境，点击右上角设置，选择安装的python环境



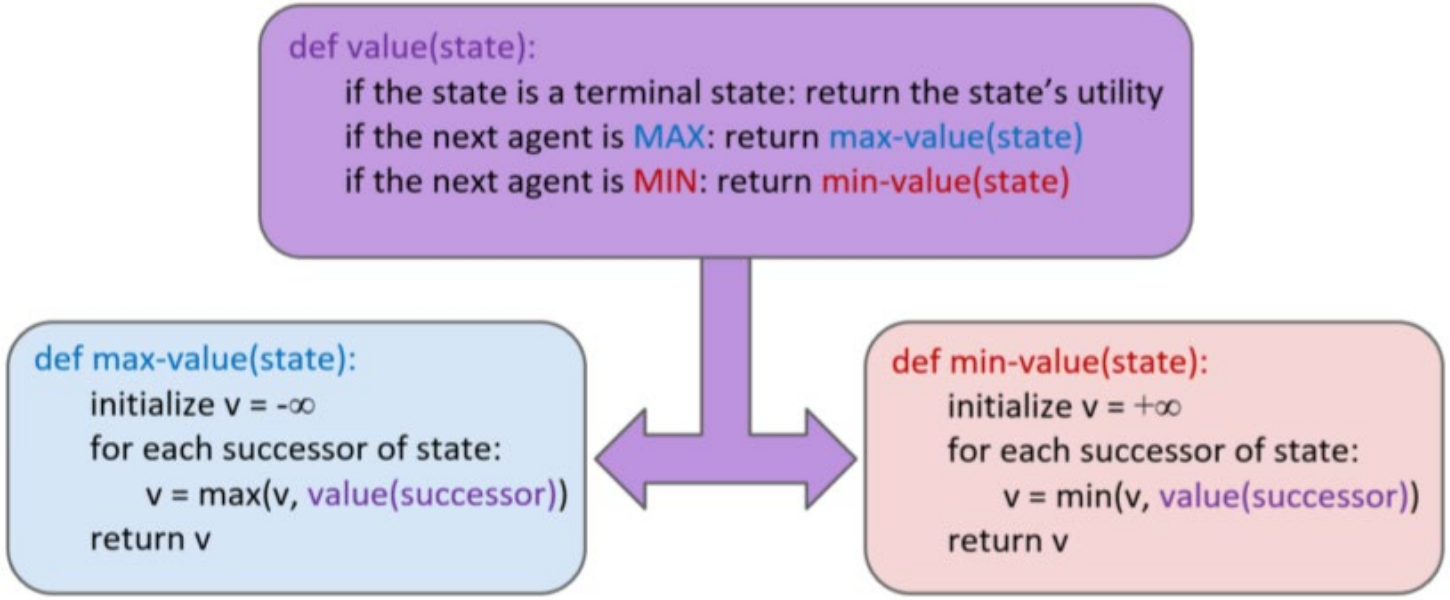
对抗搜索算法实践 — 搜索算法在吃豆人中的应用

- 1.在代码中TODO部分补全multiAgents.py中的函数
  - MinimaxAgent 类的 maxValue 和 minValue 函数
  - AlphaBetaAgent 类的 maxValue 和 minValue 函数

```
42    def maxValue(self, gameState, depth):
43        if gameState.isWin() or gameState.isLose():
44            return self.evaluationFunction(gameState), None
45        v = - math.inf
46        actionBest = None
47        for action in gameState.getLegalActions(self.index):
48            successorGameState = gameState.generateSuccessor(self.index, action)
49            value = None # TODO self.minValue(successorGameState, self.index + 1, depth) or self.maxValue(successorGameState, depth + 1)[0]
50            if value > v:
51                v = value
52                actionBest = action
53        return v, actionBest
54
55    def minValue(self, gameState, agentIndex, depth):
56        if gameState.isWin() or gameState.isLose():
57            return self.evaluationFunction(gameState)
58        v = math.inf
59        for action in gameState.getLegalActions(agentIndex):
60            successorGameState = gameState.generateSuccessor(agentIndex, action)
61            if agentIndex < gameState.getNumAgents() - 1:
62                value = self.minValue(successorGameState, agentIndex + 1, depth)
63            else:
64                if depth < self.depth:
65                    value = None # TODO self.minValue(successorGameState, agentIndex + 1, depth) or self.maxValue(successorGameState, depth + 1)[0]
66                else:
67                    value = self.evaluationFunction(successorGameState)
68            v = min(v, value)
69        return v
```

对抗搜索算法实践 — 搜索算法在吃豆人中的应用

- 2.代码测试
  - MinimaxAgent测试时在 Parameters 处写入：-p MinimaxAgent -l minimaxClassic -a depth=4
  - 注意：深度为 4 的Minimax 搜索并不能保证获胜，但大致胜率为 60%，可以在 Parameters后加上 -n 10 来测试连续 10 轮的结果。



对抗搜索算法实践 — 搜索算法在吃豆人中的应用

- 2.代码测试
  - AlphaBetaAgent测试时在 Parameters 处写入：-p AlphaBetaAgent -a depth=3 -l smallClassic
  - 注意补全v和α，β大小关系判断，此处可以对比使用MinimaxAgent 和 AlphaBetaAgent 在深度同为 3 时的运行速度

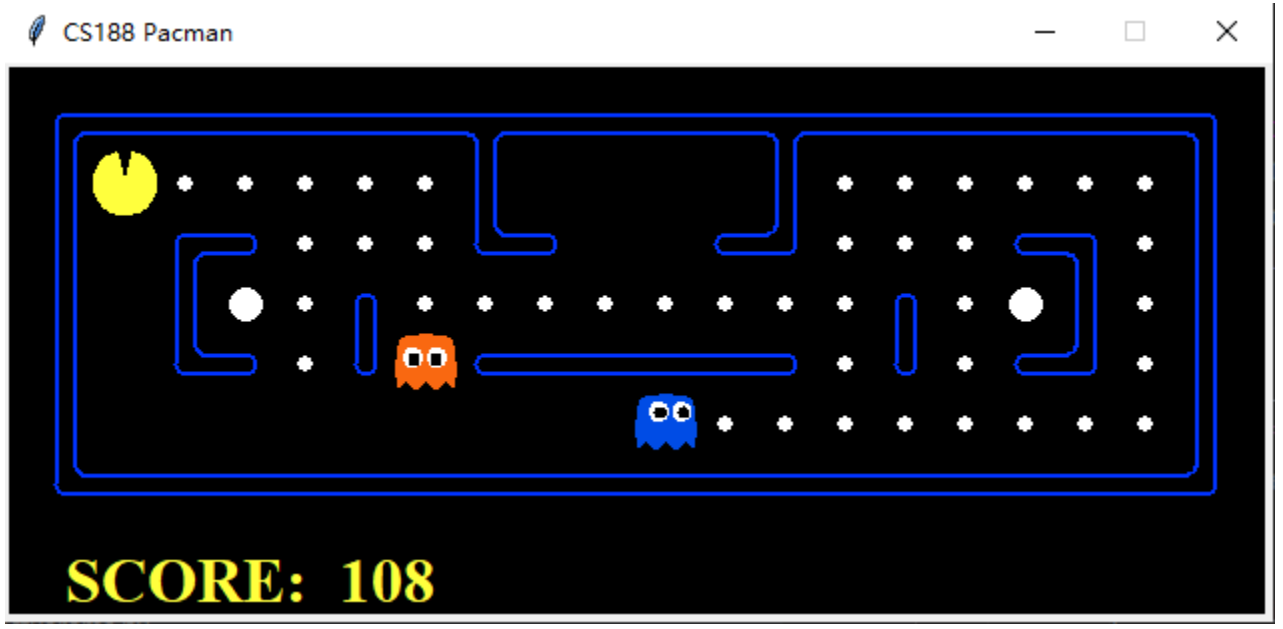
α: MAX's best option on path to root
β: MIN's best option on path to root

```
def max-value(state, α, β):
    initialize v = -∞
    for each successor of state:
        v = max(v, value(successor, α, β))
        if v ≥ β return v
        α = max(α, v)
    return v
```

```
def min-value(state , α, β):
    initialize v = +∞
    for each successor of state:
        v = min(v, value(successor, α, β))
        if v ≤ α return v
        β = min(β, v)
    return v
```

对抗搜索算法实践 — 搜索算法在吃豆人中的应用

- 2. 代码测试
  - 要求：将运行程序界面截图，将结果记录在result.docx



上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

人工智能研究院
Artificial Intelligence Institute

谢 谢！

饮 水 思 源　　爱 国 荣 校