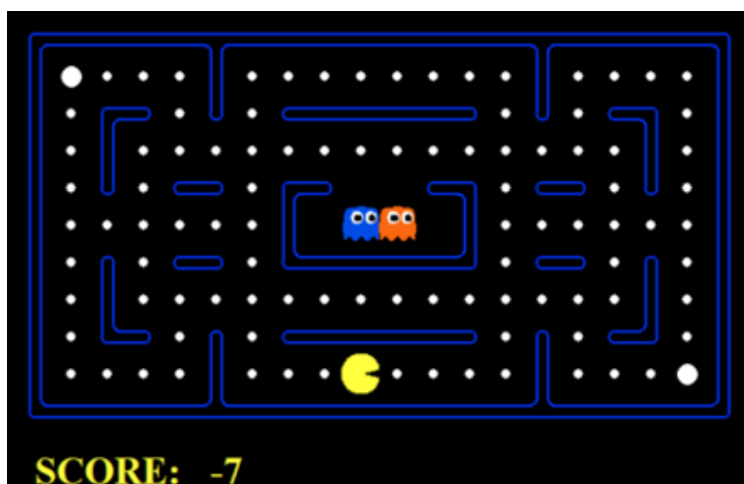


2022-HSEA-第一次作业

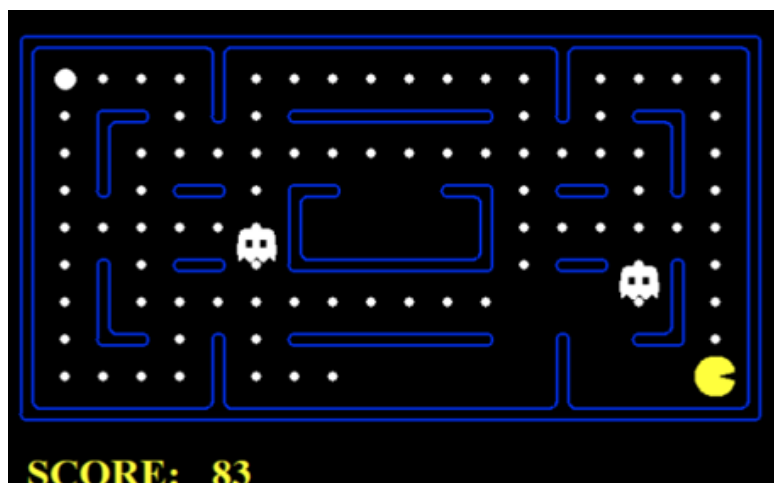
pacman游戏规则

黄色吃豆人是玩家控制的agent，可上下左右在迷宫中行走。

如下图所示，吃豆人吃掉小豆子和大豆子都可获得加分。吃豆人的目标是吃掉豆子获得分数，幽灵的目标是吃掉吃豆人。注意，随着时间消耗，你的得分也会减少。



另外，当吃豆人吃掉大豆子后随即会处于加强状态一段时间，此刻吃豆人可以反杀幽灵获得额外分数，界面如下图所示。游戏会在“吃豆人吃掉所有的豆子”或“吃豆人被幽灵吃掉”或“严重超时”的情况下终止。你需要在游戏规则下最大化agent的得分。



代码框架

[作业框架](#)，点击即可下载。

如何熟悉代码框架？

1.直观上：下载代码后，在正确的目录下，你可以通过在命令行输入“python pacman.py”试玩游戏。通过箭头控制吃豆人的移动，快速熟悉游戏要求。

2.查看关键代码

你需要编辑的文件: search.py、searchAgents.py (在后续作业要求处再介绍)

你需要了解的文件: pacman.py、game.py、util.py (代码中也有更具体的注释)

其他支撑文件

简单举例: 你可以在pacman.py的GameState类中获取有关游戏状态的信息,这有助于你代码的编写:

```
def generatePacmanSuccessor( self, action ):... 产生后继节点
```

```
def getPacmanState( self ):... 获取吃豆人状态
```

```
def getPacmanPosition( self ):... 获取吃豆人位置
```

```
def getGhostStates( self ):...
```

```
def getGhostState( self, agentIndex ):...
```

```
def getGhostPosition( self, agentIndex ):...
```

```
def getGhostPositions( self ):...
```

```
def getNumAgents( self ):...
```

```
def getScore( self ):... 获取得分
```

你也可以在pacman.py的runGames、readCommand等函数了解游戏调用的参数,例如游戏布局--layout/-l、界面大--zoom -z、代理的类型 --pacman -p等信息。

你也可以直接输入“python pacman.py -h”获取帮助。

```
Options:
-h, --help            show this help message and exit
-n GAMES, --numGames=GAMES
                       the number of GAMES to play [Default: 1] 游戏次数
-l LAYOUT_FILE, --layout=LAYOUT_FILE
                       the LAYOUT_FILE from which to load the map layout
                       [Default: originalClassic] 游戏地图
-p TYPE, --pacman=TYPE
                       the agent TYPE in the pacmanAgents module to use
                       [Default: KeyboardAgent] 游戏Agent
-t, --textGraphics   Display output as text only
-q, --quietTextGraphics
                       Generate minimal output and no graphics
-g TYPE, --ghosts=TYPE
                       the ghost agent TYPE in the ghostAgents module to use
                       [Default: RandomGhost]
-k NUMGHOSTS, --numghosts=NUMGHOSTS
                       The maximum number of ghosts to use [Default: 4]
-z ZOOM, --zoom=ZOOM Zoom the size of the graphics window [Default: 0.5] 界面大小
-f, --fixRandomSeed  Fixes the random seed to always play the same game
```

3.通过简单例子了解

运行“python pacman.py -l eg -p SearchAgent -a fn=tinyMazeSearch”指令，代码会生成layouts/eg.lay中的布局地图，并采用SearchAgent中的tinyMazeSearch算法进行游戏。

你可以通过这个简单例子了解代码之间的联系。

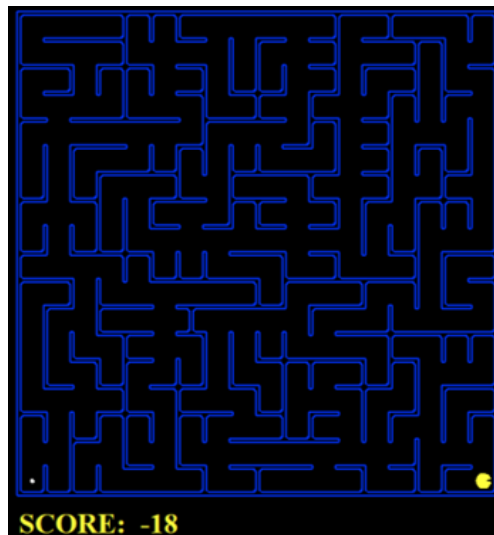
(注：这只是个让吃豆人进行移动的简单例子)

作业内容

Layouts文件夹中包含若干不同类型的地图

- **任务一(30)**: 在search.py中的空函数aStarSearch中实现A* 的图搜索代码和启发式函数代码。

aStarSearch输入中有一个参数为启发式函数，其默认值为 search.py 中的 nullHeuristic函数，这只是一个简单的启发式函数例子，你需要自行选择并编写效果更好的启发式函数，补充在myHeuristic中，并在报告中对两种启发式函数的效果进行分析。



我们将在bigMaze、openMaze、smallMaze三个布局上运行

```
“python pacman.py -l bigMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic”
```

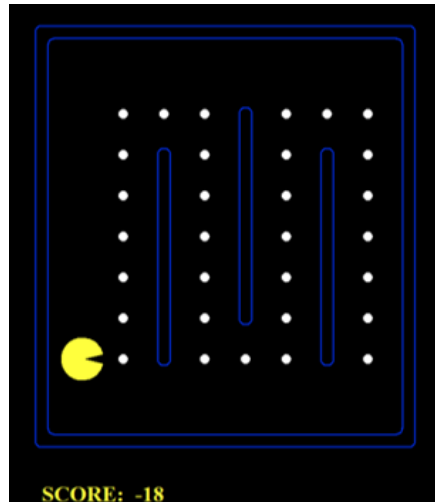
```
“python pacman.py -l openMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic”
```

```
“python pacman.py -l smallMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic”
```

获取算法的结果。

- **任务二(30)**: 基于任务一，当编写好aStarSearch函数后，现在我们将解决一个困难的搜索问题：在尽可能少的步骤中吃掉所有的豆子。为此，我们引入一个新的搜索问题定义来形式化食物清理问题，即，searchAgents.py 中的FoodSearchProblem。一个解决方案被定义为一条收集 Pacman 世界中所有食物的路径。你需要完成 searchAgents.py 中foodHeuristic函数的编写，来帮助吃豆人进行游戏。

我们期待你实现满足admissible和consistent的启发式函数，请在实验报告中对此进行阐述。如果你的启发式函数不满足此性质，我们同样也希望看到你对失败尝试的总结。



我们将在Search1、Search2、Search3三个布局上运行

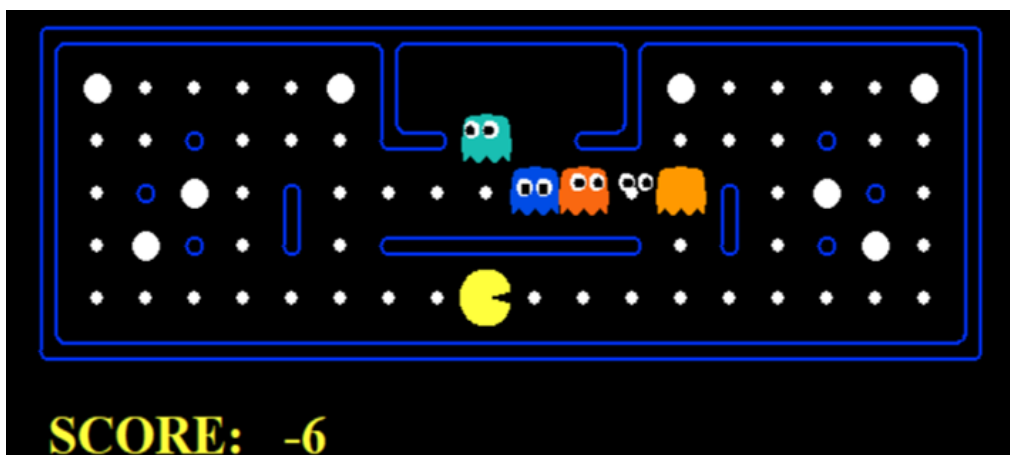
```
"python pacman.py -l Search1 -p AStarFoodSearchAgent"
```

```
"python pacman.py -l Search2 -p AStarFoodSearchAgent"
```

```
"python pacman.py -l Search3 -p AStarFoodSearchAgent"
```

获取算法的结果。我们也会通过你的启发式算法拓展的节点数目相比 $h=0$ 情况的减少量进行打分。

- **任务三(5)**: 任务二中的解决方案只取决于墙壁、普通食物（小豆子）和吃豆人的位置,没有考虑任何幽灵和能量食物（大豆子）的作用。任务三中，我们将在minimaxClassic、originalClassic、powerClassic三个有幽灵的布局上试运行任务二中的A*搜索算法。



```
"python pacman.py -l minimaxClassic -p AStarFoodSearchAgent -n 5"
```

```
"python pacman.py -l originalClassic -p AStarFoodSearchAgent -n 5"
```

```
"python pacman.py -l powerClassic -p AStarFoodSearchAgent -n 5"
```

获取你的算法在5次游戏下的平均值。（附加任务，算法在此布局上的效果不会要求很严格）

评分标准

作业的评分主要参考任务的完成情况（60%）以及报告的书写（40%）。

在报告中，你需要分别阐述每个任务的解决过程。实验报告应包括但不限于：任务叙述+解决方法+实验效果+必要分析。以及复现实验效果的操作说明。

特别提醒：一份逻辑清晰的实验报告和实验效果同等重要。

如果你的算法能取得好的效果，实验报告却潦草，最终也会影响你的分数。如果你无法实现效果好的算法，一份展现你努力尝试过程的报告也会弥补一些分数。

作业提交

- 你需要提交一份压缩文件，以“学号_姓名”的方式命名。文件中需要包含完整的项目代码和实验报告，在作业截止日期前发送到liudx@lamda.nju.edu.cn，邮件标题命名和压缩文件一致。
- 延期提交的折扣为-10/天，即每延迟一天，本次作业得分减10。你的作业流程可简单分为：熟悉代码、编程、修改、书写实验报告，请合理分配时间。
- 作业的说明和代码会发布在课程群/课程主页。

注：若发现结果造假和作业出现雷同的情况，会根据相关规定给予惩罚，详情请参考课程主页中“学术诚信”的相关内容。请同学们务必独立完成作业！