**人智第一次大作业说明**

**选择下面的三个任务中的一个来完成。**

注意：所有Project需要提交以下两份材料：

1. 源代码和可执行程序；可执行程序应保证能在以下环境之一运行：Ubuntu14.04、Win 8、Win 10、OS X EI Caption，报告中应清楚描述源代码需要的编译环境，并能在必要的时候在助教面前复现编译的过程。
2. 实验报告。注意：报告需要详细描述自己的UI设计，如果助教根本就不会使用你开发的软件，那么无法给你分数。在保证说明清楚的情况下，搜索算法部分的报告力求简洁

**任务一 MazeSearch**

# 搜索作业说明

本次作业提供了一个迷宫环境的接口，同学们只能每次获取每一个位置的后续节点信息，而没有全局的图，需要设计搜索算法，在迷宫中行走，同时实现如下一个或三个目标（取决于任务一还是任务二）：

1. 到达目标位置

2. 在行走过程中经过尽可能多的得分（某些地方的得分可能是负）

3. 对于周边环境的query次数尽可能少

## 环境说明

环境会读取文本layout文件，layout文件设计如下，例如small.lay文件（word对于空格的排版不太一样，迷宫具体见small.lay文件）

-1 -10 2 1 -3

%%%%%%%%%%%%%%%%%%%%

% 0 % G% 0 %

% %% %%1 %% 2%% %

% % % 1% % %

% %% % %%%%%% % %% %

% P 2 %

%%%%%%%%%%%%%%%%%%%%

假设迷宫大小为height x width，那么layout文件包含height+1行，第一行包含 2+n个数字，

第一个数字代表搜索过程中撞墙所收到的得分，一定为负

第二个数字代表搜索过程中尝试越出迷宫区域所收到的得分，一定为负

剩下n个数字，对应了地图中可能出现(０,...,n-1)的这些位置会分别给出的得分，

可正可负

layout文件剩下height行为对迷宫的描述，其中可能包括字符：

％: 墙，只要不是墙，其他都是可行走区域

P: 初始位置

G: 目标位置

0,...,n-1: 可以得到不同得分的位置，其他可行走位置得分为0

## 环境接口

同学们\*\*只允许使用如下GameState中的三个接口\*\*，助教会检查你的代码，若发现直接调用了GameState中的layout信息，本次作业只给提交分（大概比所有同学作业最低分还要低若干分）:

1. GameState.get\_current\_state():

得到当前所在位置的坐标

2. GameState.query\_successor(state):

给出从state出发，进行4个action分别会到达的（位置，得分，是否到达目标地点）

注意query次数是有限制的，不会超过query\_limit

3. Gamestate.step(action\_list):

按照action\_list顺序执行，返回到达的位置和当前得分，每个位置的得分只会在第一次到达时有效，之后得分变为0

## 作业任务

1. 环境中只包含目标位置，不计其他任何得分，算法只需要保证搜索到目标位置。同学们需要实现最简单的宽度优先搜索算法，深度优先搜索算法，等费用搜索算法，并且分析比较各算法的性能。当然，虽然没有给目标位置，同学们也可以考虑如何使用A\*算法和爬山算法，设计有理酌情加分。

2. 计入特别位置的得分以及撞墙越界等负分，同学们设计搜索算法，在达成上面三个目标间进行平衡，注意目标1是一定要达成的，目标2和目标3是存在矛盾的，即query更多次数会帮助搜索算法了解环境从而有机会获得更多的得分，但是相应地，目标3的得分也会降低。

## 提交形式与得分

提交代码与报告，注意助教会使用不同的layout文件进行测试，报告中需要\*\*说明如何运行你的代码进行搜索，layout文件名如何传入等等\*\*

满分10分：报告2分，接口清晰且程序运行无障碍2分，任务一4分，任务二2分，另外，如果有其他工作量（需在报告中说明，例如设计图形化界面）或者思路创新，酌情加0.5-1分

任务一主要考察同学对于不同搜索算法的实现和对比，任务二需要大家设计特别的搜索方式，最后的得分会综合评估大家对于搜索算法的设计思路以及三个目标：是否达到目标，路程上获得的得分，对于环境的query次数

## 注：

1. 本作业参考了UC Berkeley CS188 Intro to AI的pacman project

2. 本次作业提供的是python接口，python是个上手很简单的语言，建议大家使用，如果大家使用C++也可以调用python接口

3. 之所以设计成只能调用接口而不能看到全局的图，是因为这种与环境的交互形式更具有普遍性，机器学习中的RL（reinforcement learning）算法一般也都是遵循这样的假设。

**任务二 ShapeSearch**

# 任务说明

|  |  |  |
| --- | --- | --- |
| 目标图像 | 模版 | 图中对应形状 |
|  |  |  |
|  |  |  |

在本任务中，给定目标图像和模板，学生需要通过编写搜索算法，在给定图像中找到与模板对应的形状

## 输入输出及接口说明

1目标图像、模板和图中对应形状均以二值图（大小为32\*32）的形式表示，图中的形状由模版经过一定的缩放、旋转和仿射变换得到；

2目标图像、模板和图中对应形状储存在task2／data／中，命名规则如下，目标图像为query\_i\_j.txt，对应模版为template\_i.txt，得到对应形状的ground truth为gt\_i\_j.txt，在文件里面的内容各自对应一幅二值图，同一行的值用空格隔开；

3 同学们编写的程序应实现以下接口：输入目标图像和模版的文件名（文件格式按照给定的样例，为32\*32的二值图），把通过搜索找到的对应形状输出到task2/output文件下（如果目标图像名为query\_i\_j.txt，则输出的对应形状文件名应为output\_i\_j.txt），样例中的gt\_i\_j.txt用来给同学们测试自己算法的性能；

4 以上文件在task2/data中均有同名的png文件，供同学们参考。

##　提交形式与得分

提交代码与报告，报告中说明如何运行你的代码进行搜索，目标图像和模板的文件名如何传入等等

满分10分：报告2分，接口清晰且程序运行无障碍2分，能在大部分的样例（共三类形状，30个样本）中都能取得较好的效果得4分。

另外除提供给同学的样例外还有额外的50个测试样本（原来三类30个，全新的一类形状20个，只能确定是闭合的凸多边形）。在所有样本上计算F1-score：

假设实际的形状共有N个点，程序找到的形状有M个点，实际的形状与程 序找到的形状能匹配上的有n的点，则：

precision = n / M

recall = n /N

F1-score = 2 \* precision \* recall / (precision + recall)

在额外的测试样本上以F1-score为标准对选择本任务的同学的算法进行排序，得分低到高为0～2分。

另外，如果有其他工作量（需在报告中说明，例如设计图形化界面）或者思路创新，酌情加0.5-1分。

**任务三 自选**

自己设计一个利用搜索解决的问题，自己实现搜索算法以及必要的界面。题目选择的时候可以参考网上已有的小游戏，题目需要征得老师或者助教的同意之后才可以做。请大家不要做下面这些游戏：梵塔问题，食人生番问题，八数码难题（重排九宫），八皇后问题，国际棋盘跳马，独立钻石问题，夹子棋（黑白棋），捉放曹，一笔画问题，五子棋，跳棋，计算24，2048，十滴水，四子棋，数独。