

**人工智能课程报告**

学 号： 20191001171

班级序号： 12

学生姓名： 滕德淋

指导教师： 叶亚琴

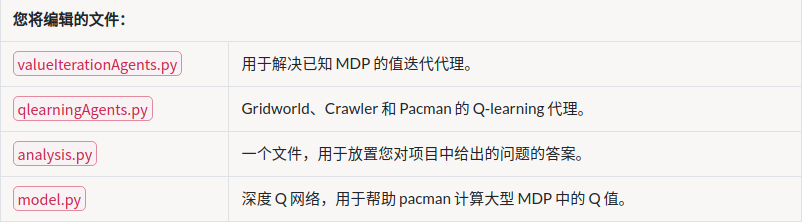
**中国地质大学地理与信息工程学院软件工程系**

**2021年 10 月**

# 一、Reinforcement Learning

## 1、实验描述

在这个项目中，我将实现 value iteration 和 Q-learning。将首先在 Gridworld（来自课堂）上测试的代理，然后将它们应用到模拟机器人控制器（Crawler）和 Pacman。







首先，在使用箭头键的手动控制模式下运行 Gridworld：

python gridworld.py -m

运行以下命令可以获得完整的选项列表：

python gridworld.py -h

默认代理随机移动

python gridworld.py -g MazeGrid

注意：Gridworld MDP 是这样的，必须首先进入预终止状态（GUI 中显示的双框），

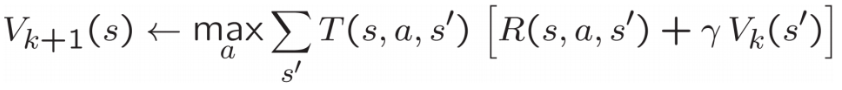
伴随图形输出的控制台输出（或对所有文本使用 -t）。与吃豆子一样，位置由(x,y)笛卡尔坐标表示，任何数组都由 索引[x][y]，“north”是增加 y 的方向，等等。默认情况下，大多数过渡将获得零奖励，尽管我们可以通过活着的人改变这一点奖励选项（-r）。

在对该项目有整体结构上的了解之后，开始对单个问题进行实现。

## 2、问题实现

### Question 1: Value Iteration

值迭代状态更新方程：



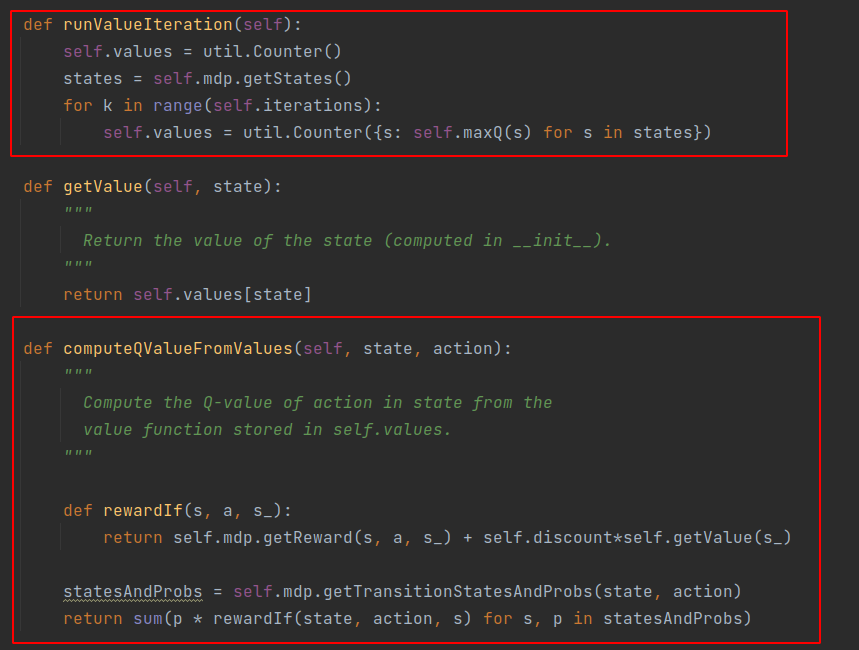
在valueIterationAgents.py中编写一个值迭代代理，在ValueIterationAgent中已部分实现，Value Iteration 代理是离线规划器，而不是强化学习代理，因此相关的训练选项是它-i在初始规划阶段应该运行的价值迭代迭代次数（选项）。ValueIterationAgent在构造函数时采用 MDP 并在构造函数返回之前运行指定迭代次数的值迭代。

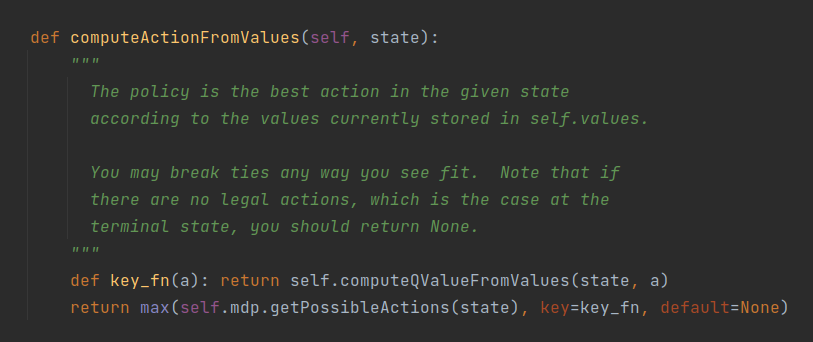
值迭代计算k最优值的步估计，V\_k。除了运行值迭代之外，还可以实现以下ValueIterationAgent使用方法V\_k：

computeActionFromValues(state)根据由 给出的价值函数计算最佳动作self.values

computeQValueFromValues(state, action)返回由 给出的值函数给出的（状态，动作）对的 Q 值self.values

这些数量都显示在 GUI 中：值是平方数，Q 值是四分之一平方数，策略是每个方块中的箭头。



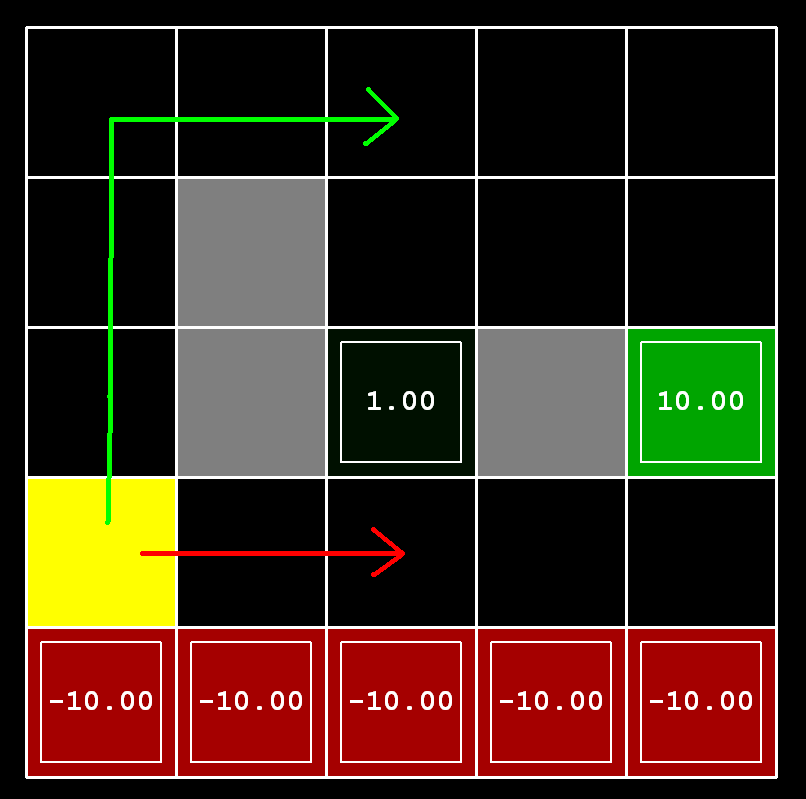


### Question 2: Policies

考虑DiscountGrid布局，如下所示。该网格有两个具有正收益的终端状态（在中间行），一个收益为 +1 的关闭退出和收益为 +10 的远程退出。网格的底行由具有负收益的终端状态组成（以红色显示）；这个“悬崖”区域中的每个州的收益都是 -10。起始状态是黄色方块。我们区分两种类型的路径：

（1）“冒险悬崖”并在网格底部行附近行驶的路径；这些路径较短，但有可能获得较大的负收益，并由下图中的红色箭头表示。

（2）“避开悬崖”并沿着网格的顶部边缘行进的路径。这些路径更长，但不太可能产生巨大的负收益。



在这个问题中，我们为 MDP 选择 discount, noise, and living 参数的设置，以生成几种不同类型的最佳策略。我们为每个部分设置的参数值应该具有这样的特性，如果我们的代理遵循其最佳策略而不受任何噪音的影响，它将表现出给定的行为。如果任何参数设置都没有实现特定行为，则通过返回字符串“NOT POSSIBLE”来断言该策略是不可能的。

以下是我们应该尝试生成的最佳策略类型：

(a) 更喜欢接近出口 (+1)，冒着悬崖的风险 (-10)

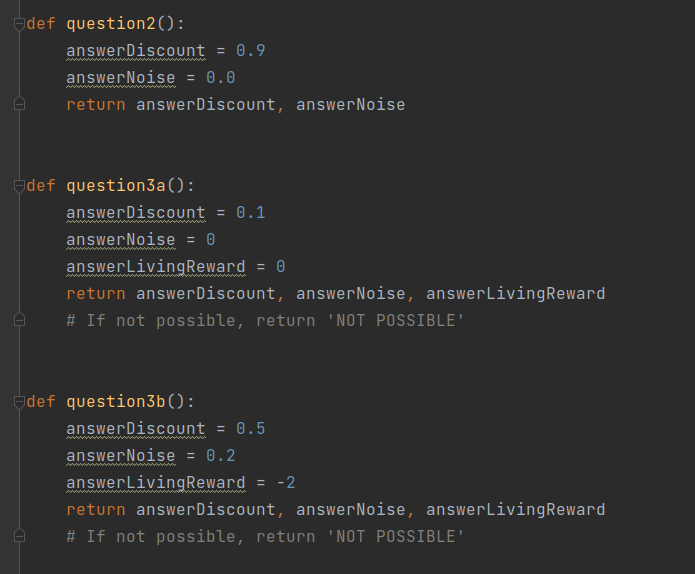
(b) 喜欢靠近出口 (+1)，但要避开悬崖 (-10)

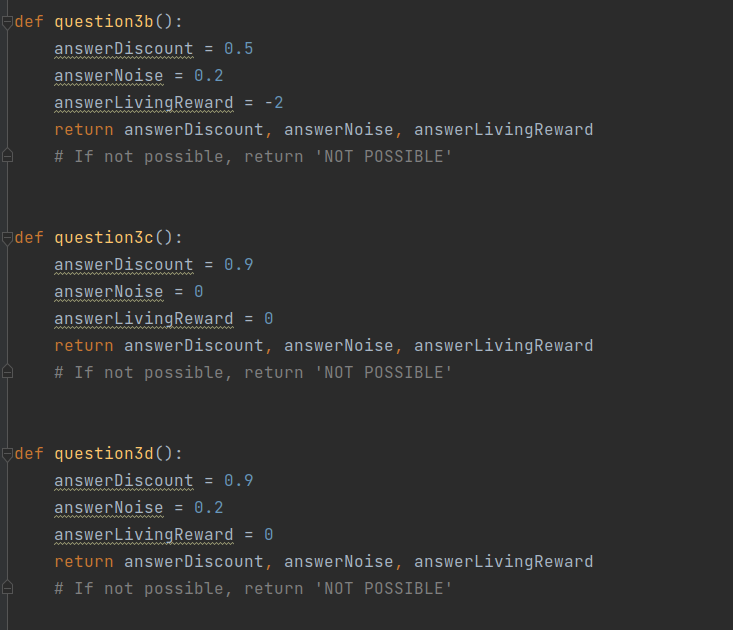
(c) 偏向远方出口（+10），冒着悬崖风险（-10）

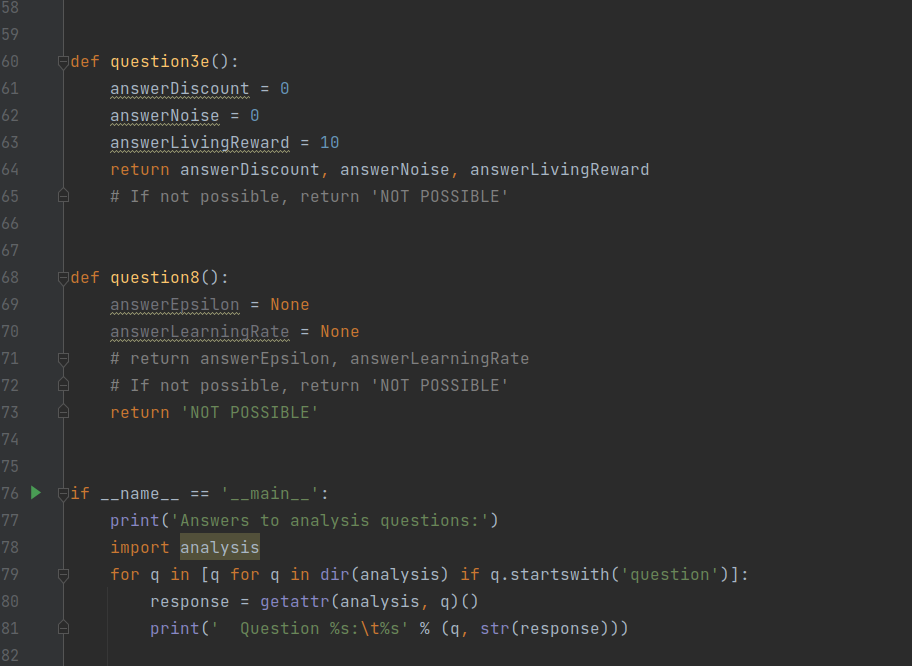
(d) 喜欢远处的出口（+10），避开悬崖（-10）

(e) 避开出口和悬崖（因此一集永远不应该终止）

具体代码：



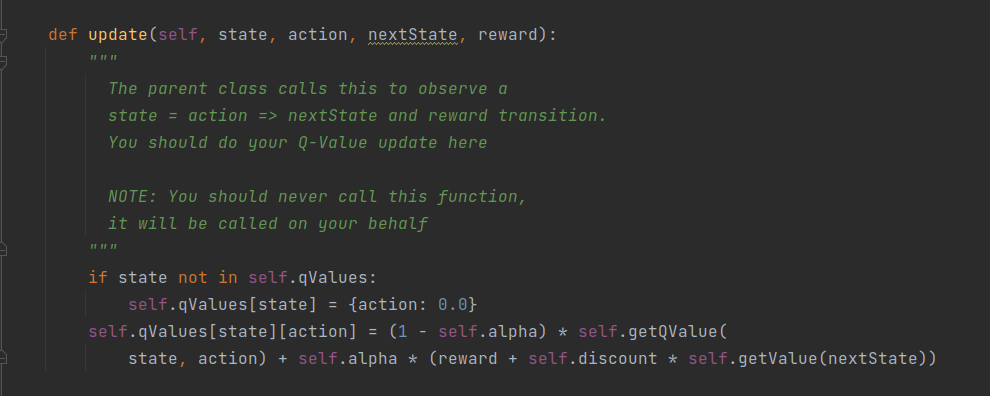


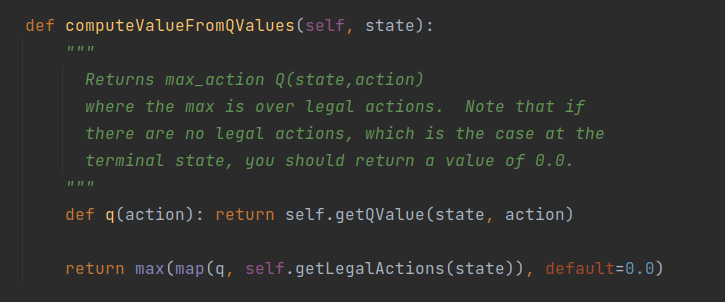


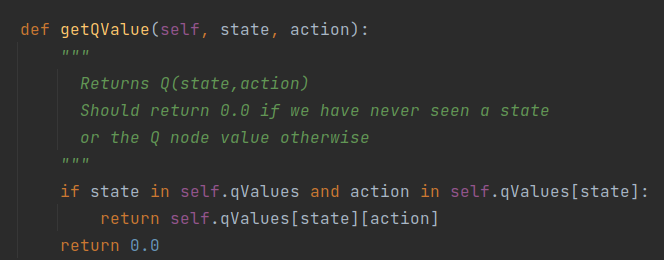
### Question 3: Q-Learning

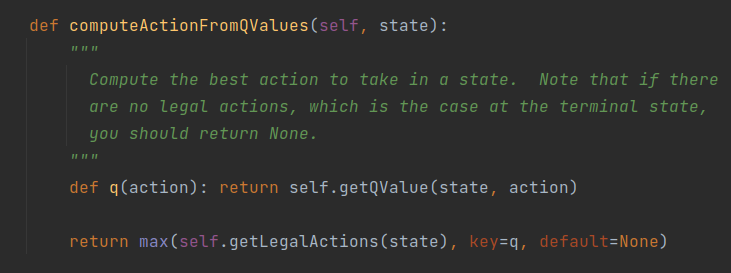
我们的价值迭代代理实际上并未从经验中学习。相反，它会在与真实环境交互之前考虑其 MDP 模型以得出完整的策略。当它确实与环境交互时，它只是遵循预先计算好的策略（例如，它成为一个反射代理）。这种区别在像 Gridword 这样的模拟环境中可能很微妙，但在真实 MDP 不可用的现实世界中非常重要。

我们现在将编写一个 Q-learning 代理，它在构造方面做的很少，而是通过它的update(state, action, nextState, reward)方法从与环境的交互中反复试验来学习。在 qlearningAgents.py 中指定了 Q-learner 的实现 QLearningAgent ，您可以使用选项 选择它'-a q'。对于这个问题，我们必须实现update，computeValueFromQValues，getQValue，和computeActionFromQValues方法。







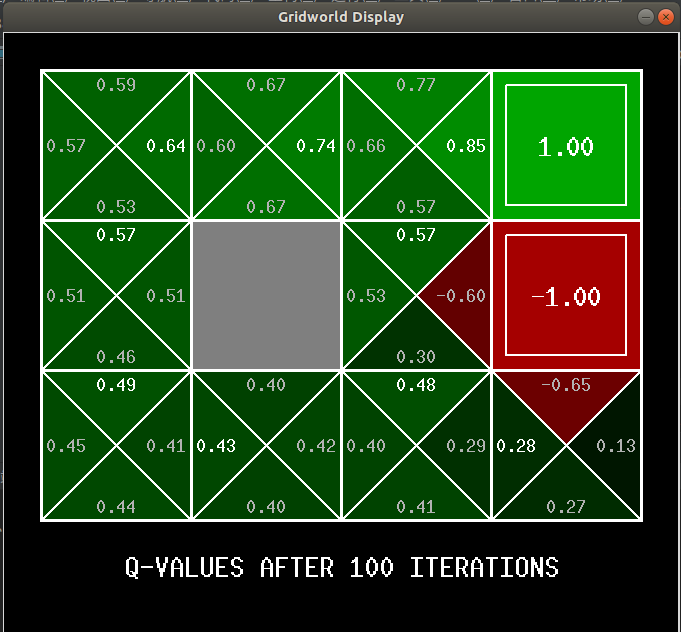


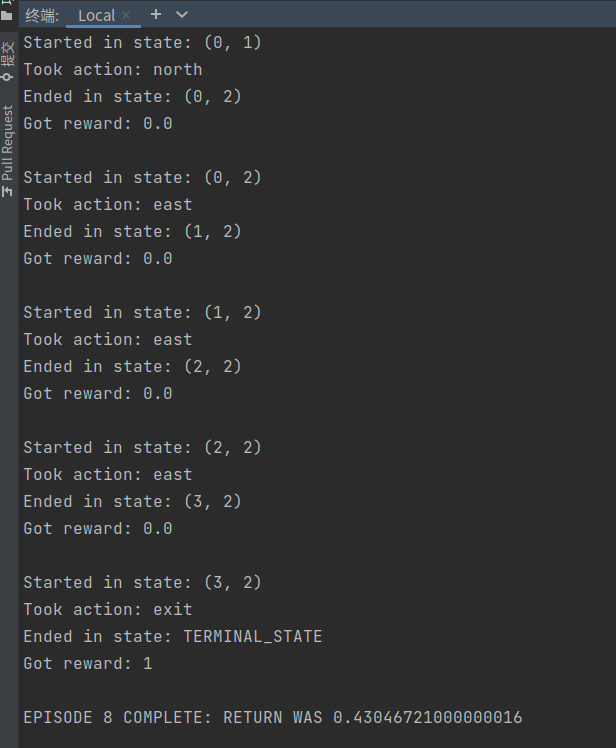
## 3、实验结果

### Question 1: Value Iteration

以下命令加载ValueIterationAgent，这将计算策略并执行 10 次。按一个键可在value、Q 值和模拟之间循环。我们会发现开始状态的值（V(start)我们可以从 GUI 中读取）和经验所得平均奖励（在 10 轮执行完成后打印）非常接近。

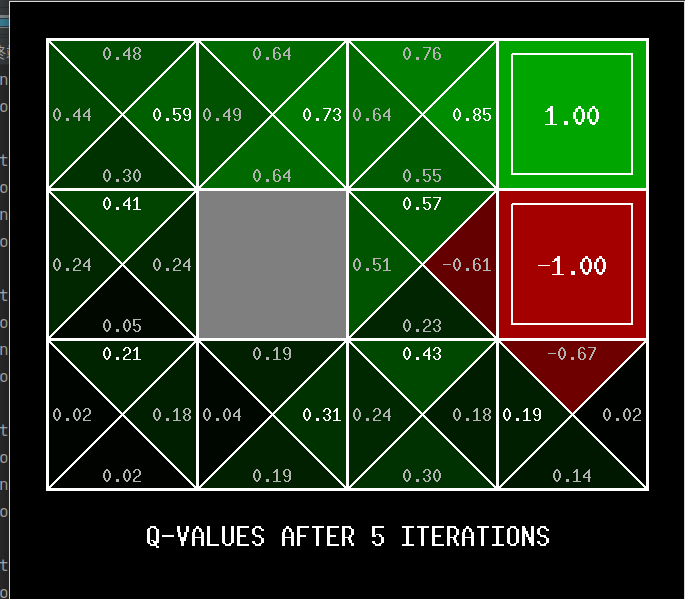
python gridworld.py -a value -i 100 -k 10

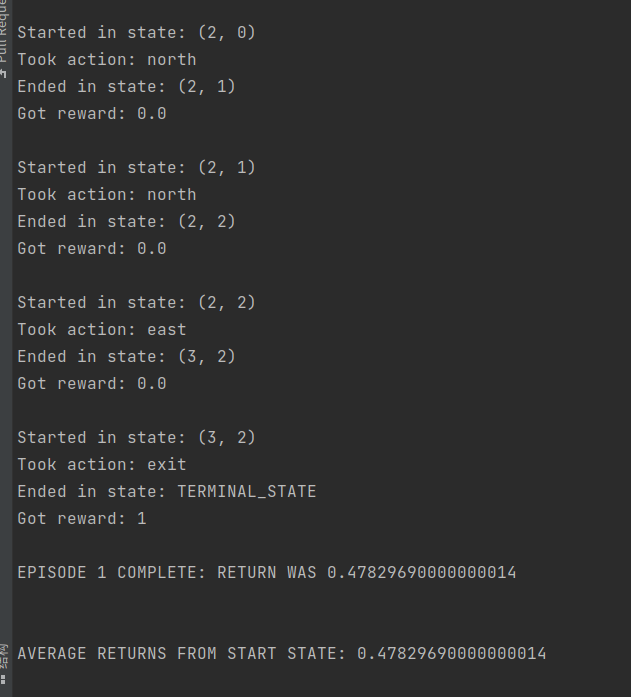




在默认的 BookGrid 上，运行 5 次迭代的值迭代应该给你这个输出：

python gridworld.py -a value -i 5





### Question 2: Policies

运行命令查看：

python gridworld.py -a value -i 100 -g BridgeGrid --discount 0.9 --noise 0.2

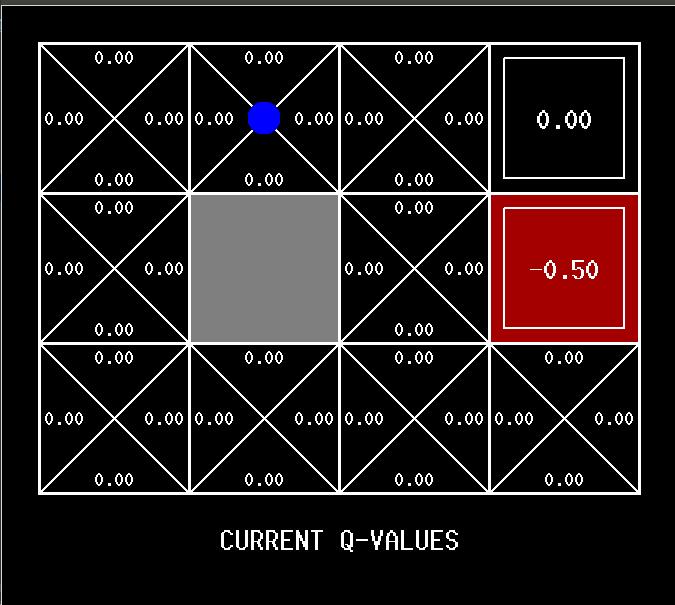


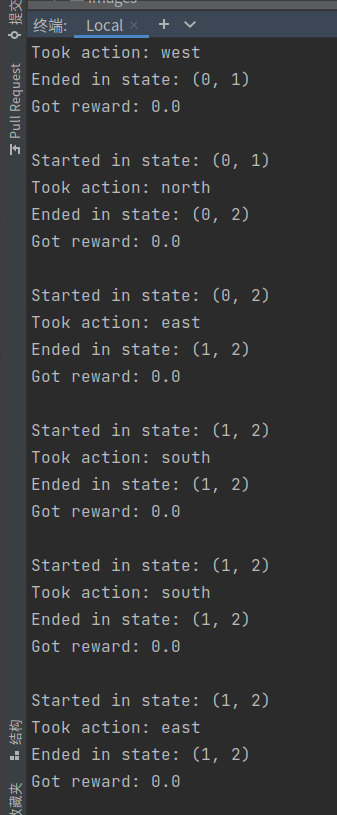


### Question 3: Q-Learning

随着 Q-learning 更新到位，您可以使用键盘在手动控制下观看 Q-learner 学习：

python gridworld.py -a q -k 5 -m





# 二、课程总结

## 1、课程内容总结

### 引言

人工智能发展历史、国内外研究现状

人工智能、机器学习、深度学习、数据分析等等概念的解释

### 搜索算法

Heuristic和A\*算法

LocalSearch算法

博弈论与对抗搜索

蒙特卡洛树搜索

### 知识表示

知识的概念及其分类

一阶逻辑

语义网络

本体

### 贝叶斯网络的分析与推理

Probability

Bayesi network 贝叶斯网络的构建

贝叶斯网络推理：变量消元

贝叶斯网络推理：D分离

### 自动推理模型

朴素贝叶斯网络

马尔科夫模型

隐马尔科夫模型

### 增强学习

MDP

Q-learning

## 2、课程建议

* 课程实习给的时间太少，也有可能是我们其他课作业太多，导致做实习作业的时间不太够，再加上难度比较大，导致实习效果不太好。
* 实习作业建议找几个完成的比较好的同学进行讲解。