

RLChina第一次习题课

题目要求:

具体实现流程

步骤1: 进入游戏板块

步骤2: 准备代码

步骤3: 提交

查看结果

RLChina第一次习题课

题目要求:

作业的目的: 就是让大家熟悉一下及第平台的基本使用

作业方式: 通过给推箱子小游戏, 提交一个随机动作的算法代码

作业要求及流程: 其实细心的小伙伴可以看到第一次作业就是及第平台**赶考**板块的第一个小实验。



操作流程如下:

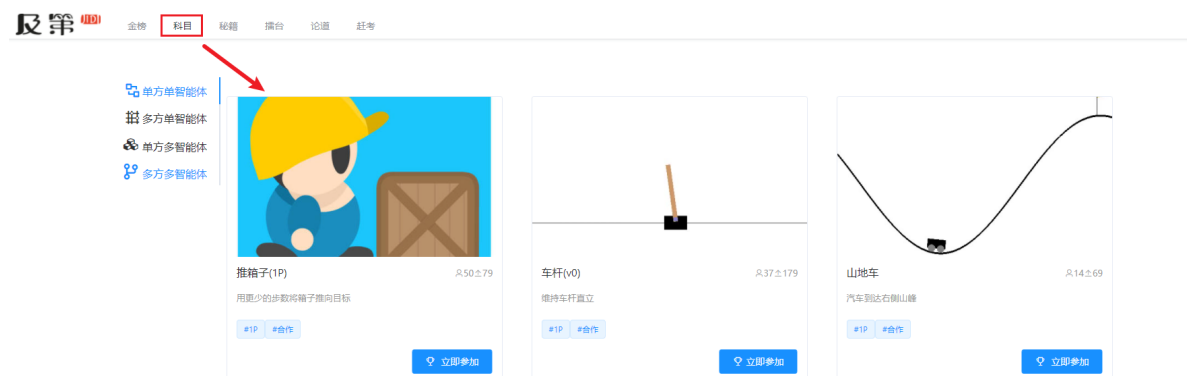
1. 注册平台 -> 登录平台 -> 进入**科目**, 找到**推箱子 (1p)**
2. 点击**提交说明** -> 在提交说明的帖子中打开**random** -> 复制代码并保存为**submission.py**
3. 在**推箱子 (1P)** 中, 点击**提交** -> 填写算法名称, 上传**submission.py**, 点击**提交** -> 稍作等待 (10min内), 您可在**金榜 - 推箱子 (1P)** 中找到自己的排名啦

具体实现流程

给大家截图一步步看一下

步骤1: 进入游戏板块

注册登录不用说, 接下来进入科目, 找到推箱子, 进入



步骤2：准备代码

简介

推箱子是一个经典的古老游戏。
在一个狭小的仓库中，要求把木箱放到指定的位置，稍不小心就会出现箱子无法移动或者通道被堵住的情况，所以需要巧妙的利用有限的空间和通道，合理安排移动的次数和位置，才能顺利地完成任務。



规则

- 玩家在 $n \times m$ 的网格中操纵一个人物，玩家只能控制人物的前进方向(上、下、左和右)从而移动箱子。
- 在设定的游戏地图里，箱子只能推不能拉。一人将全部箱子在指定步数内移动到相应的目的地。
- 如果玩家位置超过地图范围则不移动。
- 当游戏达到指定步数或箱子全部被移动到目的地时，游戏结束。
- 观测为一个字典，其中的键为`state_map`、`chess_player_idx`、`board_width`和`board_height`。`state_map`对应的值为 8×8 的矩阵，表示当前地图的状态，其中的值为0至 $n_{player} + 3$ 的整数，此处 $n_{player} = 1$ ，0表示没有物体，1表示围墙，2表示目标点，3表示箱子，4至 $n_{player} + 3$ 表示人物；`player_idx`的值为当前玩家的序号；`board_width`的值为地图的宽；`board_height`的值为地图的高。
- 动作空间为长度为 n_{action_dim} 的列表，其中 $n_{action_dim} = 1$ ，每个元素为Gym当中的Discrete类（Discrete Link），[Discrete(4)]
- 奖励reward为推到目的地的箱子个数。

提交说明

详见论道

训练说明

详见论道

开始

提交 - 推箱子 (1p)

排行榜

查看所有

名次	用户名	总排名
1	HenryLau	-48.00
2	eigebi	-49.00
3	Jidi_random	-49.00
4	赵士杰	-49.20
5	woshiliyuewen	-49.20

打开git上的案例：

<https://github.com/jidi.ai/lib/blob/master/examples/demo/random/submission.py>

提交说明 Submit Instructions

2021-08-03 10:53:55

项目仓库 Git Repository

<https://github.com/jidi.ai/lib>

平台提交说明

- 填写算法名称或描述，选择提交环境
- 上传一个或多个文件。
 - 其中必须包含一个运行文件，运行文件需包含`my_controller`函数的一个`submission.py`文件。
 - 附属文件支持`.pth`、`.py`类型文件。大小不超过100M，个数不超过5个。

注意：每次提交会覆盖之前的文件。

提交样例

<https://github.com/jidi.ai/lib/blob/master/examples/demo/random/submission.py>

Submit Instructions

- Fill in the algorithm name or description, and select the submission environment
- Upload one or more files.
 - It must include a `submission.py` file with the `my_controller` function.
 - The attached file supports `.pth` and `.py` type files. The size does not exceed 100M, and the number does not exceed 5.

Note: Each submission will overwrite the previous file.

Example Submission

<https://github.com/jidi.ai/lib/blob/master/examples/demo/random/submission.py>

建议大家在github后加上1s,可以快速查阅项目代码

<https://github1s.com/jidi.ai/lib/blob/master/examples/demo/random/submission.py>

```
14 def my_controller(observation, action_space, is_act_continuous=False):
15     agent_action = []
16     for i in range(len(action_space)):
17         action_ = sample_single_dim(action_space[i], is_act_continuous)
18         agent_action.append(action_)
19     return agent_action
20
21
22 def sample_single_dim(action_space_list_each, is_act_continuous):
23     each = []
24     if is_act_continuous:
25         each = action_space_list_each.sample()
26     else:
27         if action_space_list_each.__class__.__name__ == "Discrete":
28             each = [0] * action_space_list_each.n
29             idx = action_space_list_each.sample()
30             each[idx] = 1
31         elif action_space_list_each.__class__.__name__ == "MultiDiscreteParticle":
32             each = []
33             nvec = action_space_list_each.high - action_space_list_each.low + 1
34             sample_indexes = action_space_list_each.sample()
35
36             for i in range(len(nvec)):
37                 dim = nvec[i]
38                 new_action = [0] * dim
39                 index = sample_indexes[i]
40                 new_action[index] = 1
41                 each.extend(new_action)
42     return each
43
44
45 def sample(action_space_list_each, is_act_continuous):
46     player = []
47     if is_act_continuous:
48         for j in range(len(action_space_list_each)):
49             each = action_space_list_each[j].sample()
50             player.append(each)
```

然后在自己电脑本地新建一个python文件，命名为submission.py

将上面的代码复制粘贴到该文件中，保存

```
1 # -*- coding:utf-8 -*-
2 # Time : 2021/5/31 下午4:14
3 # Author: Yahui Cui
4
5 """
6 # ===== Important =====
7 Notes:
8 1. this agent is random agent , which can fit any env in Jidi platform.
9 2. if you want to load .pth file, please follow the instruction here:
10 https://github.com/jidi.ai/ai_lib/blob/master/examples/demo
11 """
12
13
14 def my_controller(observation, action_space, is_act_continuous=False):
15     agent_action = []
16     for i in range(len(action_space)):
17         action_ = sample_single_dim(action_space[i], is_act_continuous)
18         agent_action.append(action_)
19     return agent_action
20
21
22 def sample_single_dim(action_space_list_each, is_act_continuous):
23     each = []
24     if is_act_continuous:
25         each = action_space_list_each.sample()
26     else:
27         if action_space_list_each.__class__.__name__ == "Discrete":
28             each = [0] * action_space_list_each.n
```

PS: 如果github打不开的同学，可以直接复制这里的代码

```
1 def my_controller(observation, action_space, is_act_continuous=False):
2     agent_action = []
3     for i in range(len(action_space)):
```

```

4         action_ = sample_single_dim(action_space[i], is_act_continuous)
5         agent_action.append(action_)
6     return agent_action
7
8 def sample_single_dim(action_space_list_each, is_act_continuous):
9     each = []
10    if is_act_continuous:
11        each = action_space_list_each.sample()
12    else:
13        if action_space_list_each.__class__.__name__ == "Discrete":
14            each = [0] * action_space_list_each.n
15            idx = action_space_list_each.sample()
16            each[idx] = 1
17        elif action_space_list_each.__class__.__name__ ==
"MultiDiscreteParticle":
18            each = []
19            nvec = action_space_list_each.high - action_space_list_each.low
+ 1
20            sample_indexes = action_space_list_each.sample()
21
22            for i in range(len(nvec)):
23                dim = nvec[i]
24                new_action = [0] * dim
25                index = sample_indexes[i]
26                new_action[index] = 1
27                each.extend(new_action)
28    return each
29
30 def sample(action_space_list_each, is_act_continuous):
31     player = []
32     if is_act_continuous:
33         for j in range(len(action_space_list_each)):
34             each = action_space_list_each[j].sample()
35             player.append(each)
36     else:
37         player = []
38         for j in range(len(action_space_list_each)):
39             # each = [0] * action_space_list_each[j]
40             # idx = np.random.randint(action_space_list_each[j])
41             if action_space_list_each[j].__class__.__name__ == "Discrete":
42                 each = [0] * action_space_list_each[j].n
43                 idx = action_space_list_each[j].sample()
44                 each[idx] = 1
45                 player.append(each)
46             elif action_space_list_each[j].__class__.__name__ ==
"MultiDiscreteParticle":
47                 each = []
48                 nvec = action_space_list_each[j].high
49                 sample_indexes = action_space_list_each[j].sample()
50
51                 for i in range(len(nvec)):
52                     dim = nvec[i] + 1
53                     new_action = [0] * dim
54                     index = sample_indexes[i]
55                     new_action[index] = 1
56                     each.extend(new_action)
57                 player.append(each)
58     return player

```

步骤3：提交

再次进入推箱子板块，点击提交

及第 AI

金榜 科目 秘籍 擂台 论道 赶考

简介

推箱子是一个经典的古老游戏。在一个狭小的仓库中，要求把木箱放到指定的位置，稍不小心就会出现箱子无法移动或者通道被堵住的惨况，所以需要巧妙的利用有限的空间和通道，合理安排移动的次序和位置，才能顺利的完成任务。

规则

- 玩家在 $n \times m$ 的网格中操纵一个人物，玩家只能控制人物的前进方向(上、下、左和右)从而移动箱子。
- 在设定的游戏地图里，箱子只能推不可以拉。一人将全部箱子在指定步数内移动到相应的目的地。
- 如果玩家位置超过地图范围则不移动。
- 当游戏达到指定步数或箱子全部被移动到目的地时，游戏结束。
- 观测为一个字典，其中的键为"state_map"、"chess_player_idx"、"board_width"和"board_height"，"state_map"对应的值为 8×8 的矩阵，表示当前地图的状态，其中的值为0至 $n_{player} + 3$ 的整数，此处 $n_{player} = 1$ ，0表示没有物体，1表示围墙，2表示目标点，3表示箱子，4至 $n_{player} + 3$ 表示人物；"player_idx"的值为当前玩家的序号；"board_width"的值为地图的宽；"board_height"的值为地图的高。
- 动作空间为长度为 n_{action_dim} 的列表，其中 $n_{action_dim} = 1$ ，每个元素为Gym当中的Discrete类（Discrete Link），[Discrete(4)]
- 奖励reward为推到目的地的箱子个数。

提交说明

详见论道

训练说明

详见论道

开始

提交 - 推箱子 (1p)

排行榜

名次	用户名	总排名
1	HenryLau	-48.00
2	eigebi	-49.00
3	Jidi_random	-49.00
4	赵士杰	-49.20
5	woshiluyewen	-49.20

会有提示接口更新，先不用管。点击确定

信息 上传文件 提交

* 算法名称 算法名称

下一步

提示

接口有更新，请确认https://github.com/jidian/ai_lib后重新提交

确定

输入算法名称(随便一个名字)，点击下一步

信息 上传文件 提交

* 算法名称 random_test

下一步

请注意

针对于相同环境的提交，系统会自动覆盖之前提交的文件

点击上传文件

这里注意

1. 只能上传 .py 文件和.pth(模型文件)
2. 上传的文件中必须要有 submission.py 文件
3. 文件的总个数不能超过6个

⚠️ 请注意
针对于相同环境的提交，系统会自动覆盖之前提交的文件

信息

上传文件

提交

* 算法名称

random_test

运行文件

上传您的主文件及附属文件

您需上传一个名为submission包含my_controller函数的.py文件，及最多5个.pth、.py附属文件

提交

然后选中上面建立的submission.py文件，上传

信息

上传文件

提交

* 算法名称

random_test

运行文件

2.7 KB

submission.py

移除

提交

点击提交即可

结束~~ 撒花~~

查看结果

10min左右后，在个人中心的提交列表即可看到自己的提交信息

参与排行

提交列表

我的对局

我的竞赛

推箱子(1P)

提交序号 1110 - random

提交于 2021-08-06 23:24:43

在参与排行中，可以看到自己的提交通过信息



后面的操作可以显示对局等等

在金榜下, 推箱子(1p), 可以看到自己的排名

The screenshot shows the '金榜' (Leaderboard) page for '推箱子(1P)'. A red arrow points from the '金榜' tab to the '推箱子(1P)' dropdown menu. The table shows a list of users and their scores.

名次	用户名	积分	时间	回放
1	HenryLau	-48.00	2021-07-10 10:10:23	🔍
2	eigebi	-49.00	2021-06-22 14:25:12	🔍
3	Jidi_random	-49.00	2021-07-27 14:38:22	🔍
4	Tony_Xiao	-49.00	2021-08-09 17:03:41	🔍
5	vivian	-49.20	2021-08-09 16:37:09	🔍
6	邵威	-49.20	2021-08-06 14:52:48	🔍
7	woshiliguwen	-49.20	2021-08-05 15:01:57	🔍
8	twiple	-49.20	2021-08-06 18:31:59	🔍
9	HandsomeWu	-49.20	2021-08-06 23:24:43	🔍

然后 就可以去看看代码如何实现的。想想办法去设计 优化自己的AI算法啦~

恭喜出村~~

注: 由于推箱子游戏是走一步如果游戏没赢就要减一, 所以成绩是负值很正常(想要不是负值就要搞一些更加聪明的算法上去~~)

相关链接:

及第平台: <http://www.jidi.ai>

及第repo: https://github1s.com/jidi.ai/ai_lib/blob/master/examples/demo/random/submission.py

补充链接:

贪吃蛇3V3 Intro: <https://mp.weixin.qq.com/s/AM72QokvXmdsRjeFd-EbfA>

贪吃蛇3V3本地部署及提交: <https://mp.weixin.qq.com/s/AM72QokvXmdsRjeFd-EbfA>

作者: HandsomeWu(公众号同步)

有问题直接在群里询问 ~~ 冲冲冲

也可以关注公众号: **RLCN** 在后台提问: