

代码原始出处为openai的github项目：[Multi-Agent Particle Environment](#)

在阅读此文档之前，可以先阅读官网的README.md文档，里面有对各个代码文件的系统性介绍

课程所用的环境为simple_tag.py子环境，并进行了一定程度的修改，主要改动为是在openai的原始环境基础上，给simple_tag环境添加了边界墙，3个固定障碍物，1个打卡点。

注意：这是一个多智能体环境，但是追击者的策略已经给定，所以实际上是单智能体问题。

1.环境配置

1.1 环境所需要的依赖

根据实操后，需要安装以下依赖：**python3.6**及以上（在我电脑上安装的是python3.6，低于python3.6可能会报错），**gym 0.10.5**，**numpy**，**scipy**。

推荐使用anaconda创建虚拟环境，在虚拟环境中配置。具体如何创建可直接baidu

算法框架：tensorflow (例如1.14,cpu版本即可，有条件可以安装gpu)或者pytorch。根据需要安装，因为环境文件并没有用到这些框架。

安装均可在终端用pip的形式安装：

```
pip install gym==0.10.5
```

```
pip install tensorflow==1.14 #根据自己的需求决定是否安装
```

```
pip install numpy
```

```
pip install scipy
```

推荐使用**ubuntu**系统，18.04及以上都行，windows亦可，但有些电脑可能不能成功运行环境。

1.2 安装环境

OpenAI将该环境封装成了库，可以直接通过import形式输出环境。在此之前我们需要安装该环境。

首先切换到环境**multiagent-envs-ML**主目录下（你会看到有一个**setup.py**文件），在此目录下打开终端，运行以下代码安装：

```
pip install -e .
```

注意：号前面有空格。

安装完成后最终会提示：

```
Running setup.py develop for multiagent
Successfully installed multiagent-0.0.1
```

2.代码运行

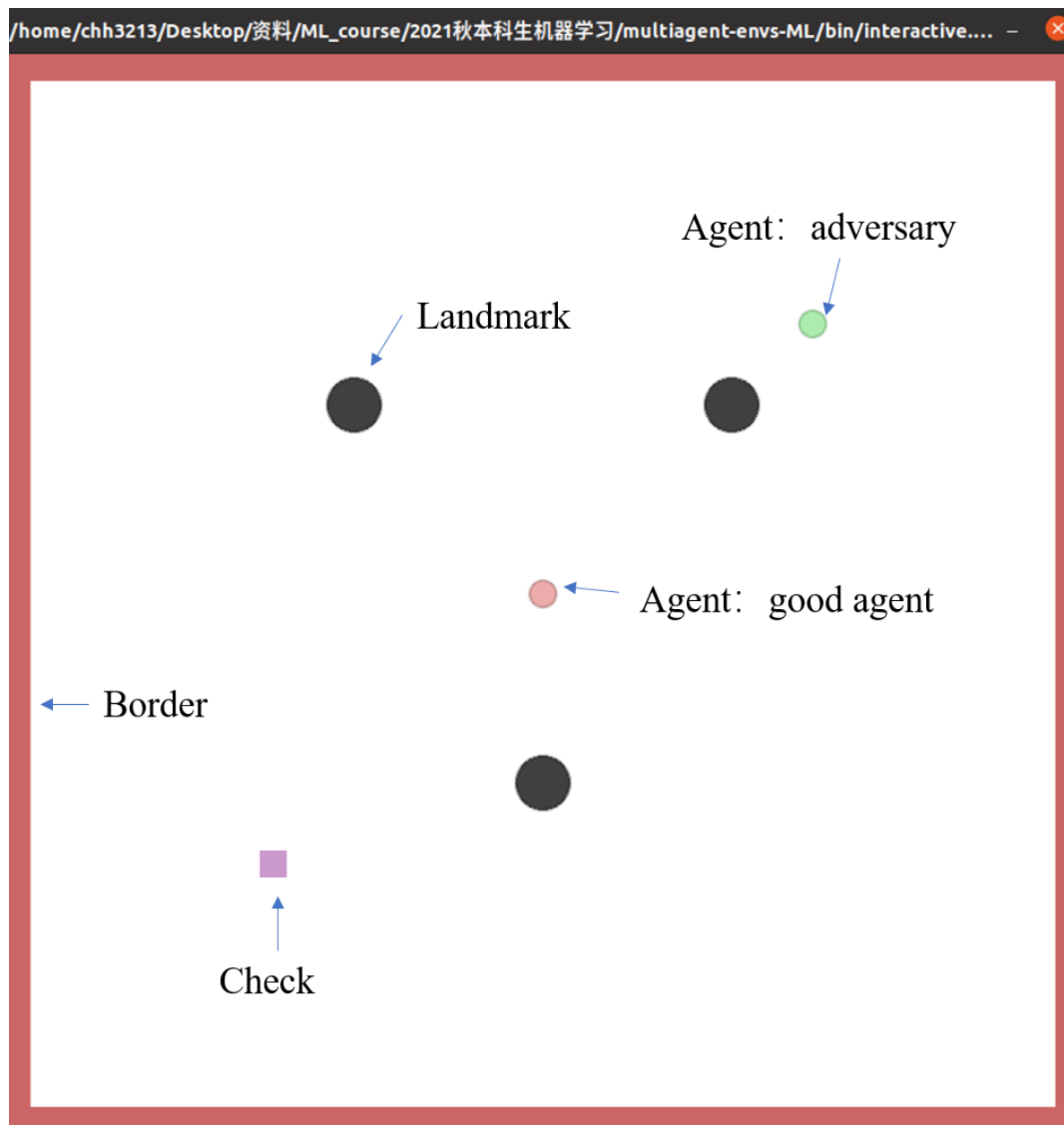
安装成功后，正常情况下代码运行应该不会报错。运行步骤如下：

cd到 `/multiagent-envs-ML/bin` 文件夹下，直接在终端运行

```
python interactive.py
```

即可。运行成功的话，你会看到有两个一样的环境窗口，这是因为有两个智能体（图中的adversary和good agent），可以分别在各自窗口通过键盘键盘的上下左右键控制其中一个智能体。

注意：**interactive.py**文件是做**演示使用**，大家可以把它当作主函数文件进行修改编写，或者自己重新编写算法和环境的主函数文件。其中追击者的策略已经给定，所以一开始运行interactive.py文件时，红色那个智能体会追着绿色智能体跑。



3.环境代码部分介绍

1. 该部分可以参考这篇博客的内容，是对openai环境及其算法的介绍：[深度解析OPENAI-MADDPG](#)
2. 修改的环境部分主要是增加了边界类**Border**，打卡点类**Check**。
3. 最重要的环境文件是 `core.py`，`simple_tag.py`，`environment.py` 这三个文件。这三个文件的大致调用关系是：**core**被**simple_tag**调用，**simple_tag**被**environment**调用。

4. **core**文件中主要声明了环境中出现的各个实体：agent, border, landmark, check等。
5. **simple_tag**文件主要设置实体的参数（初始位置、加速度等），reward设置等。
6. **environment**文件是强化学习算法中的经典环境接口（step,reset, render等）。你可以按照给定任务重写 `environment.py` 文件中的 `self._set_action` 函数。

4.任务说明

因为完成的任务包括：逃跑者到达目标点、避障、逃跑。因此判断游戏结束的条件为：逃跑者被抓到、碰到障碍物、到达目标点、到达指定时间。

为降低难度，在这里提前告诉大家，代码中给定的目标点，追击者，逃避者的位置就是答辩展示时的坐标设置。

注意：用什么算法取决于你自己，只要能够实现任务需求，你想干嘛都行，其他并没有约束。