

# 深度强化学习环境搭建说明文档

## 目录

<b>1. 安装 VirtualBox.....</b>	<b>2</b>
1.1 下载.....	2
1.2 安装.....	2
<b>2. 虚拟机安装.....</b>	<b>2</b>
2.1 新建虚拟机.....	2
2.2 安装虚拟机.....	8
<b>3 系统搭建.....</b>	<b>8</b>
3.1 安装 OpenGL.....	8
3.2 安装 VNC.....	8
3.3 安装 mujoco.....	9
3.4 安装 Gym.....	12
3.5 安装 universe.....	12
3.6 安装 docker.....	13
3.7 universe 测试.....	13
3.8 安装 retro.....	14
3.9 安装 SC2.....	15

# 1. 安装 VirtualBox

## 1.1 下载

下载地址: <https://www.virtualbox.org/>

## 1.2 安装

此处为正常的软件安装过程, 在这里就不再赘述

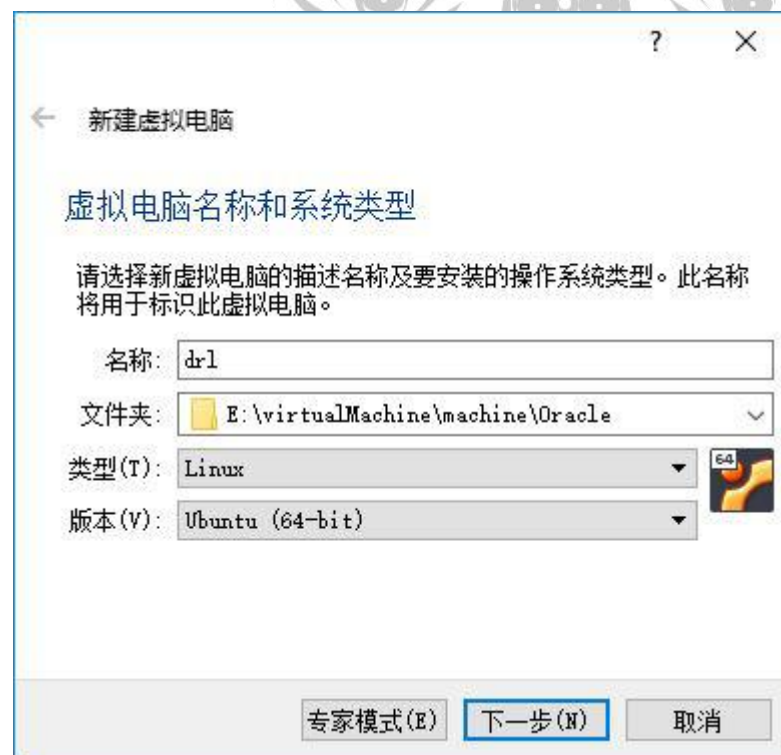
# 2. 虚拟机安装

## 2.1 新建虚拟机

首先, 点击新建虚拟机



弹出对话框



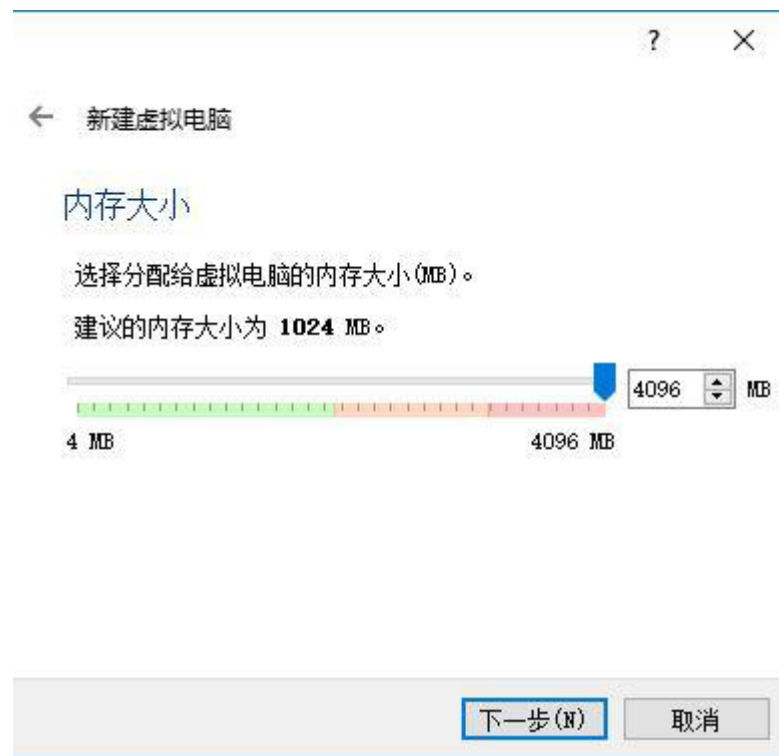
名称: 安装的虚拟机名称, 这个可以根据自己的喜好自由定义

文件夹：这个是虚拟机安装的位置，注意尽量选择存储空间较大的位置

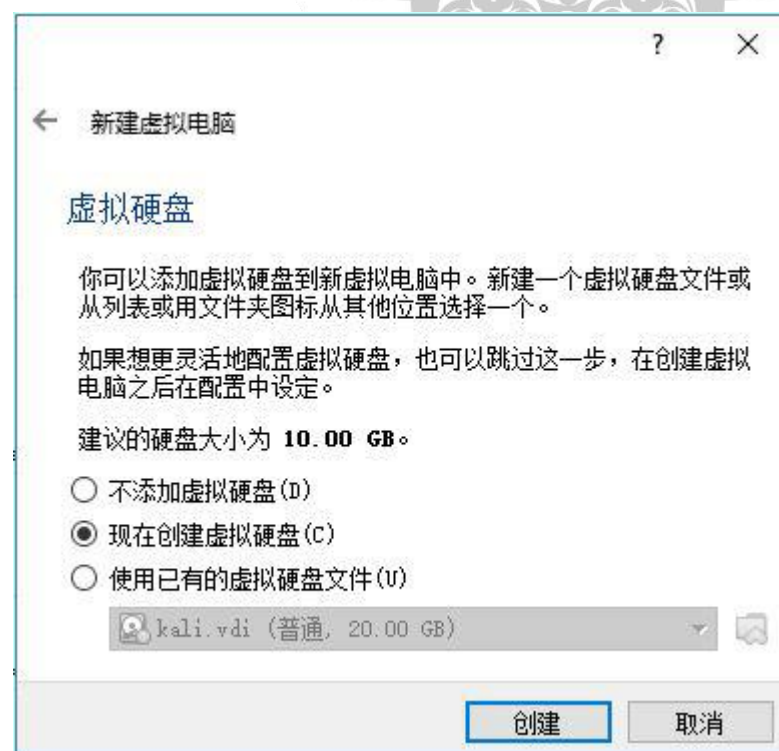
类型：根据自己的镜像文件进行选择，在这里选择 Linux

版本：根据自己的镜像文件进行选择，在这里选择 Ubuntu(64-bit)

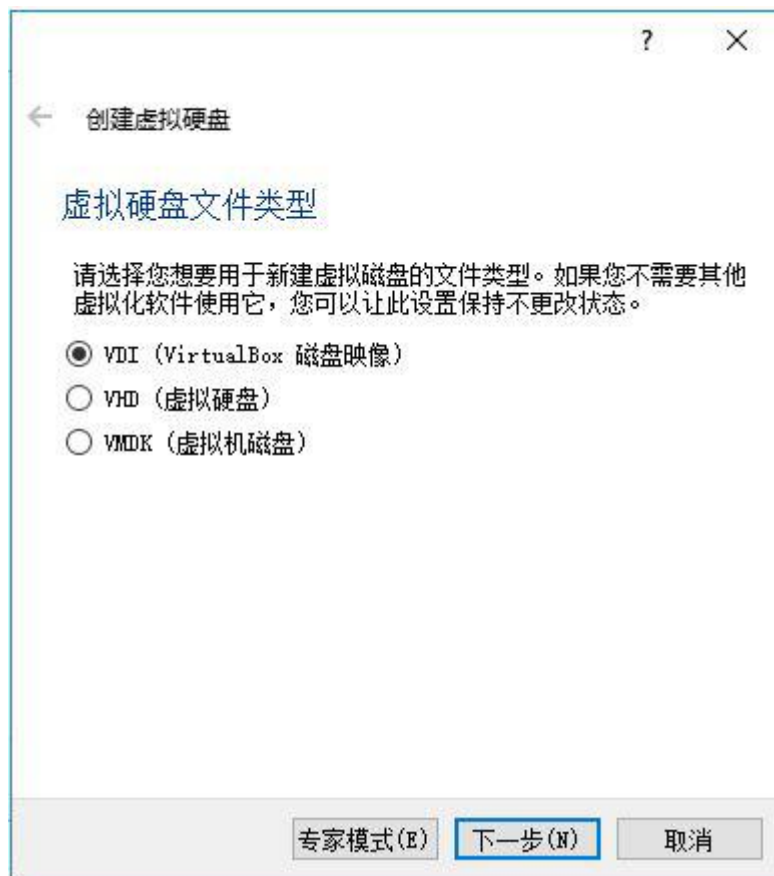
参数选择完成后点击下一步：



这里的内存大小请根据自己的系统进行选择，建议越大越好，但是不要超过主机系统的内存，以免影响系统的正常的运行，内存大小最好不要低于 2G，参数设置完成后点击下一步。



这里需要为虚拟机创建一个虚拟硬盘，如果您已经存在，可以选择使用已有的磁盘文件，在这里建议选择默认即可，点击创建。



默认即可，选择下一步。



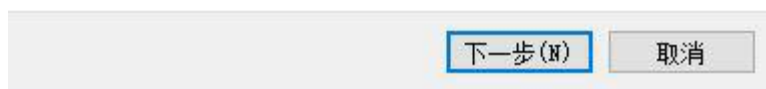
## 存储在物理硬盘上

请选择新建虚拟硬盘文件是应该为其使用而分配(动态分配)，还是应该创建完全分配(固定分配)。

**动态分配**的虚拟磁盘只是逐渐占用物理硬盘的空间（直至达到**分配的大小**），不过当其内部空间不用时不会自动缩减占用的物理硬盘空间。

**固定大小**的虚拟磁盘文件可能在某些系统中要花很长时间来创建，但它往往使用起来较快。

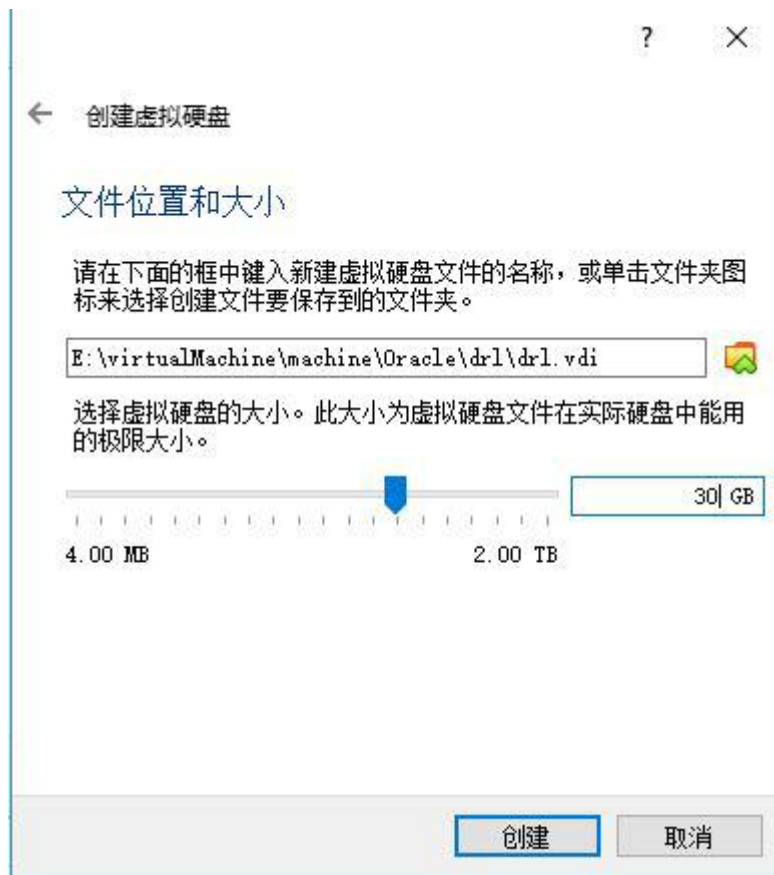
- ☒ 动态分配(D)  
☐ 固定大小(F)



动态大小：虚拟磁盘的大小是根据你的虚拟机的实际使用不断的变化，你用的越多，虚拟磁盘的实际大小就会越大，上限为你设置的磁盘大小，这种方式较为灵活，但是需要注意的是，磁盘大小只会增大，不会应为你改变虚拟机使用的空间大小而减小。

固定大小：虚拟磁盘的大小固定，不会变化，一旦创建，就会在您的主系统上占用相应的空间，这种方式不够灵活，不推荐。

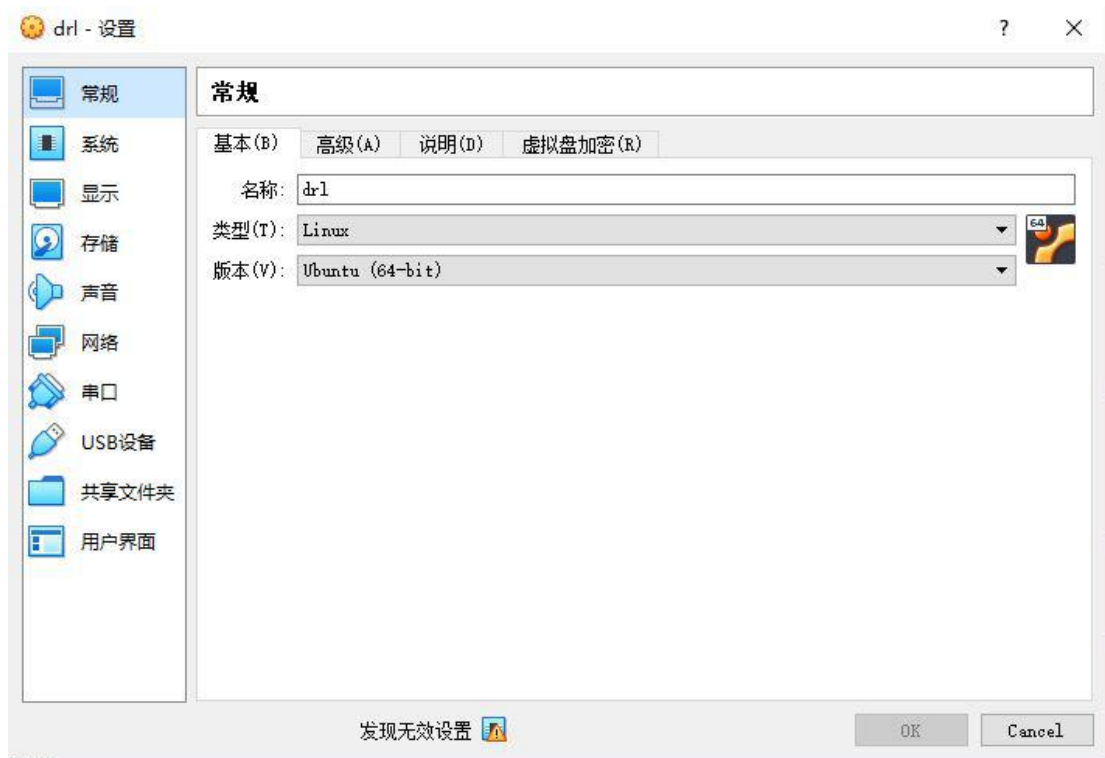
选择默认，下一步



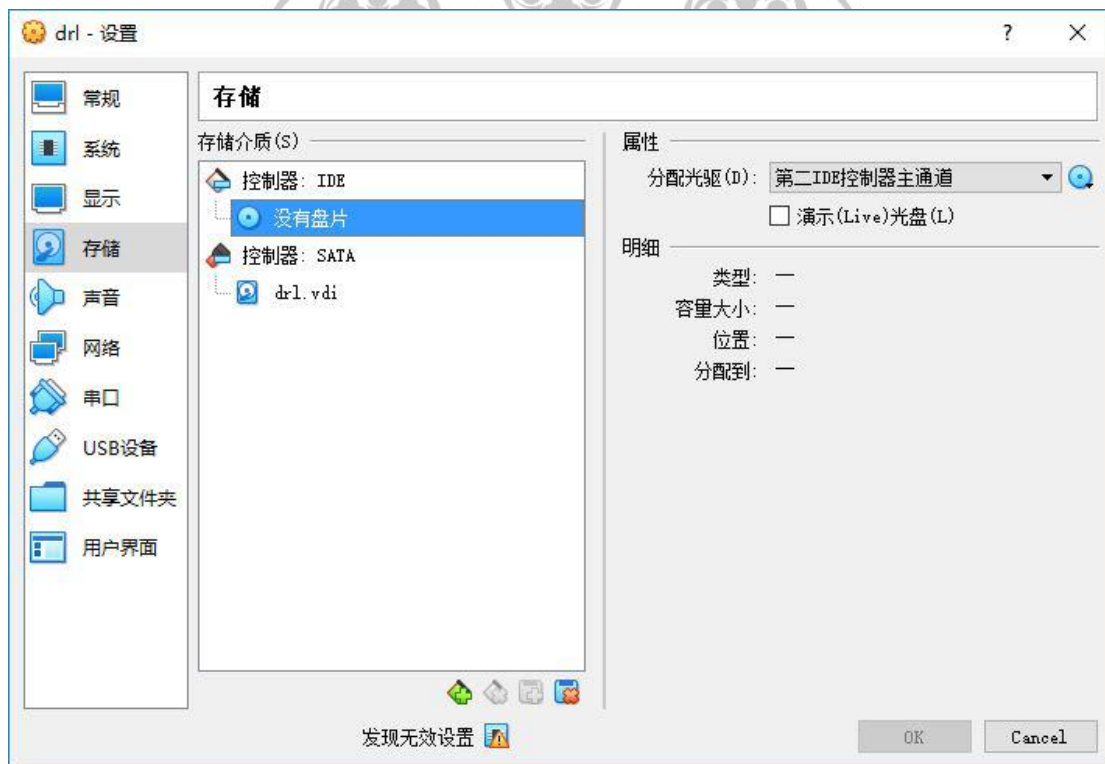
文件的位置：这里会自动帮您选择与之前虚拟机的安装目录一样的位置，不建议更改，如果您的空间有限，您可以更改  
文件的大小：根据自己的系统空间进行选择  
设置完成后，点击创建



创建完成后点击设置



点击存储



点击没有盘片，在属性--分配光驱，点击小碟片图标，选择您已下载的镜像文件

## 2.2 安装虚拟机



点击启动，开始虚拟机的安装

## 3 系统搭建

### 3.1 安装 OpenGL

打开链接：

[https://en.wikibooks.org/wiki/OpenGL\\_Programming/Installation/Linux](https://en.wikibooks.org/wiki/OpenGL_Programming/Installation/Linux)

执行对应的命令：

```
sudo apt-get install build-essential libgl1-mesa-dev
```

```
sudo apt-get install libglew-dev libSDL2-dev
```

```
libSDL2-image-dev libglm-dev libfreetype6-dev
```

检查安装是否成功：

```
glxinfo | grep OpenGL
```

如果提示 glxinfo 未安装：

```
sudo apt-get install glxinfo
```

### 3.2 安装 VNC

这里选择安装 tigerVNC

打开下面链接下载安装文件：

<http://tigervnc.bphinz.com/nightly/>

选择对应的版本下载：

Packages for Ubuntu 16.04 LTS			
filename	bytes	timestamp	md5sum
amd64			
<a href="#">tigervnc-java_1.9.80+20190603git3b6dec9f-1ubuntu1_amd64.deb</a>	5620	2019-06-03 22:51:40	caef19f4eafb90f4a12a6ff1c8296eb8
<a href="#">tigervncserver_1.9.80+20190603git3b6dec9f-1ubuntu1_amd64.deb</a>	1215896	2019-06-03 22:51:40	b5569c5dc8483206b9f5de086ec892d1
<a href="#">tigervncserver-dbg_1.9.80+20190603git3b6dec9f-1ubuntu1_amd64.ddeb</a>	4460720	2019-06-03 22:51:33	c9de68025b9b26a4f41ea03d5d828863
<a href="#">xtigervncviewer_1.9.80+20190603git3b6dec9f-1ubuntu1_amd64.deb</a>	201406	2019-06-03 22:51:40	e99babc23fcbde4d8c1376cfe44488e
<a href="#">xtigervncviewer-dbg_1.9.80+20190603git3b6dec9f-1ubuntu1_amd64.ddeb</a>	621066	2019-06-03 22:51:35	e536dfe537c9f9434086550a77f34121



下载完成后进入下载文件所在的目录，打开终端，执行下面的指令：

```
dpkg -i tigervncserver_1.9.80+20190603git3b6dec9f-1ubuntu1_amd64.deb
```

执行时会出现软件包未安装的情况，执行安装命令 `apt-get install` 缺失包在安装时会出现需要执行新的指令，这个提示在最后几行，复制命令执行即可安装完成后，再执行如下命令：

```
dpkg -i xtigervncviewer_1.9.80+20190603git3b6dec9f-1ubuntu1_amd64.deb
```

安装同样会出现上面的问题，解决方式一样

安装完成后，执行 `tigervncviewer` 指令，查看是否弹出 `tiger` 界面窗口，若弹出，则表示安装成功。

### 3.3 安装 mujoco

打开下面的链接：

<https://www.roboti.us/license.html>

如果您是学生且有学校的邮箱(edu)，您可以有一年的使用权限，如果你不是，您只有 30 天的使用权限。

对于非学生：

**MuJoCo Trial License: 30 days**

We invite you to register for a free trial of MuJoCo. Trials are limited to one per user per year. After registration you will receive an email with your activation key and license text. The activation key will be locked to your Computer id. Download the 'getid' executable corresponding to your platform (using the links below) and run it to obtain your Computer id.



Within the 30-day trial period you may request up to 3 activation keys for different computers or operating systems. All keys will expire on the same date. If you accidentally submit the same Computer id again, we will re-send the activation key without counting it towards the limit.

先点击 `linux` 下载一个文件，然后进入对应的目录，打开终端，执行如下指令：

```
chmod a+x getid_linux
```

```
./getid_linux
```

然后复制 `computer id` 密文

填写上面的表单，点击我同意

然后点击 `submit` 提交，之后打开邮箱会收到一份邮件，包含两个文件，下载到本地。

对于学生：



填写上面表单，获取文件

完成上面的操作后，打开下面的链接安装 `mujoco`

<https://github.com/openai/mujoco-py/blob/master/Dockerfile>

```
RUN apt-get update -q \
    && DEBIAN_FRONTEND=noninteractive apt-get install -y \
    curl \
    git \
    libgl1-mesa-dev \
    libgl1-mesa-glx \
    libglew-dev \
    libosmesa6-dev \
    software-properties-common \
    net-tools \
    unzip \
    vim \
    virtualenv \
    wget \
    xpra \
    xserver-xorg-dev \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

RUN DEBIAN_FRONTEND=noninteractive add-apt-repository --yes ppa:deadsnakes/ppa && apt-get update
RUN DEBIAN_FRONTEND=noninteractive apt-get install --yes python3.6-dev python3.6 python3-pip
RUN virtualenv --python=python3.6 env

RUN rm /usr/bin/python
RUN ln -s /env/bin/python3.6 /usr/bin/python
RUN ln -s /env/bin/pip3.6 /usr/bin/pip
RUN ln -s /env/bin/pytest /usr/bin/pytest
```

依此执行上面的代码，注意红色字体部分：

RUN ----->不输入

ENV----->不执行

```
1 # We need the CUDA base dockerfile to enable GPU rendering
2 # on hosts with GPUs.
3 # The image below is a pinned version of nvidia/cuda:9.1-cudnn7-devel-ubuntu16.04 (from Jan 2018)
4 # If updating the base image, be sure to test on GPU since it has broken in the past.
5 FROM nvidia/cuda@sha256:4df157f2afde1cb6077a191104ab134ed4b2fd62927f27b69d788e8e79a45fa1
6
7 RUN apt-get update -q \
8     && DEBIAN_FRONTEND=noninteractive apt-get install -y \
9     curl \
10    git \
11    libgl1-mesa-dev \
12    libgl1-mesa-glx \
13    libglew-dev \
14    libosmesa6-dev \
15    software-properties-common \
16    net-tools \
17    unzip \
18    vim \
19    virtualenv \
20    wget \
21    xpra \
22    xserver-xorg-dev \
23    && apt-get clean \
24    && rm -rf /var/lib/apt/lists/*
25
26 RUN DEBIAN_FRONTEND=noninteractive add-apt-repository --yes ppa:deadsnakes/ppa && apt-get update
27 RUN DEBIAN_FRONTEND=noninteractive apt-get install --yes python3.6-dev python3.6 python3-pip
28 RUN virtualenv --python=python3.6 env
29
```

复制上面的代码执行

完成后，执行下面的代码：

```
DEBIAN_FRONTEND=noninteractive add-apt-repository --yes
```

```
ppa:deadsnakes/ppa && apt-get update
```

```
DEBIAN_FRONTEND=noninteractive apt-get install --yes
```

```
python3.6-dev python3.6 python3-pip
```

```
cd /
```

```
virtualenv --python=python3.6 env
```

```
rm /usr/bin/python
```

```
ln -s /env/bin/python3.6 /usr/bin/python
```

```
ln -s /env/bin/pip3.6 /usr/bin/pip
```

```
ln -s /env/bin/pytest /usr/bin/pytest
```

```
RUN curl -o /usr/local/bin/patchelf https://s3-us-west-2.amazonaws.com/openai-sci-artifacts/manual-builds/patchelf_0.9_amd64.elf \
&& chmod +x /usr/local/bin/patchelf
```

复制上面的代码执行

复制下面的代码执行

```
RUN mkdir -p /root/.mujoco \
&& wget https://www.roboti.us/download/mujoco200_linux.zip -O mujoco.zip \
&& unzip mujoco.zip -d /root/.mujoco \
&& mv /root/.mujoco/mujoco200_linux /root/.mujoco/mujoco200 \
&& rm mujoco.zip
```

跳转到您刚在邮件中文件下载所在的目录

cd 邮件下载文件所在的目录

执行：

```
cp mjkey.txt /root/.mujoco/
```

```
gedit ~/.bashrc
```

在文件的末尾添加：

```
export LD_LIBRARY_PATH=/root/.mujoco/mujoco200/bin:${LD_LIBRARY_PATH}
```

```
export LD_LIBRARY_PATH /usr/local/nvidia/lib64:${LD_LIBRARY_PATH}
```

保存退出：

执行：

```
git clone https://github.com/openai/mujoco-py.git
```

```
cd mujoco-py
```

```
cp vendor/Xdummy /usr/local/bin/Xdummy
```

```
chmod +x /usr/local/bin/Xdummy
```

```
pip install --no-cache-dir -r requirements.txt
```

```
pip install --no-cache-dir -r requirements.dev.txt
```

```
python setup.py install
```

至此 mujoco 已安装完成

### 3.4 安装 Gym

```
git clone https://github.com/openai/gym.git
```

```
cd gym
```

```
apt-get install -y python-pyglut python3-opengl zlib1g-dev libjpeg-dev patchelf
```

```
\cmake swig libboost-all-dev libstdc++6-dev libosmesa6-dev xvfb ffmpeg
```

```
pip install -e '[all]'
```

### 3.5 安装 universe

```
pip install numpy
```

```
sudo apt-get install golang libjpeg-turbo8-dev make
```

```
git clone https://github.com/openai/universe.git
```

```
cd universe
```

```
pip install -e .
```

## 3.6 安装 docker

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

```
sudo apt-get remove docker docker-engine docker.io containerd runc
sudo apt-get update
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo apt-key fingerprint 0EBFCD88
sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
sudo docker run hello-world
```

## 3.7 universe 测试

```
cd universe
docker build -t universe .
docker run --privileged --rm -e DOCKER_NET_HOST=172.17.0.1 -v
/var/run/docker.sock:/var/run/docker.sock universe pytest
touch universe.py
gedit universe.py
```

复制下面的代码到文件中：

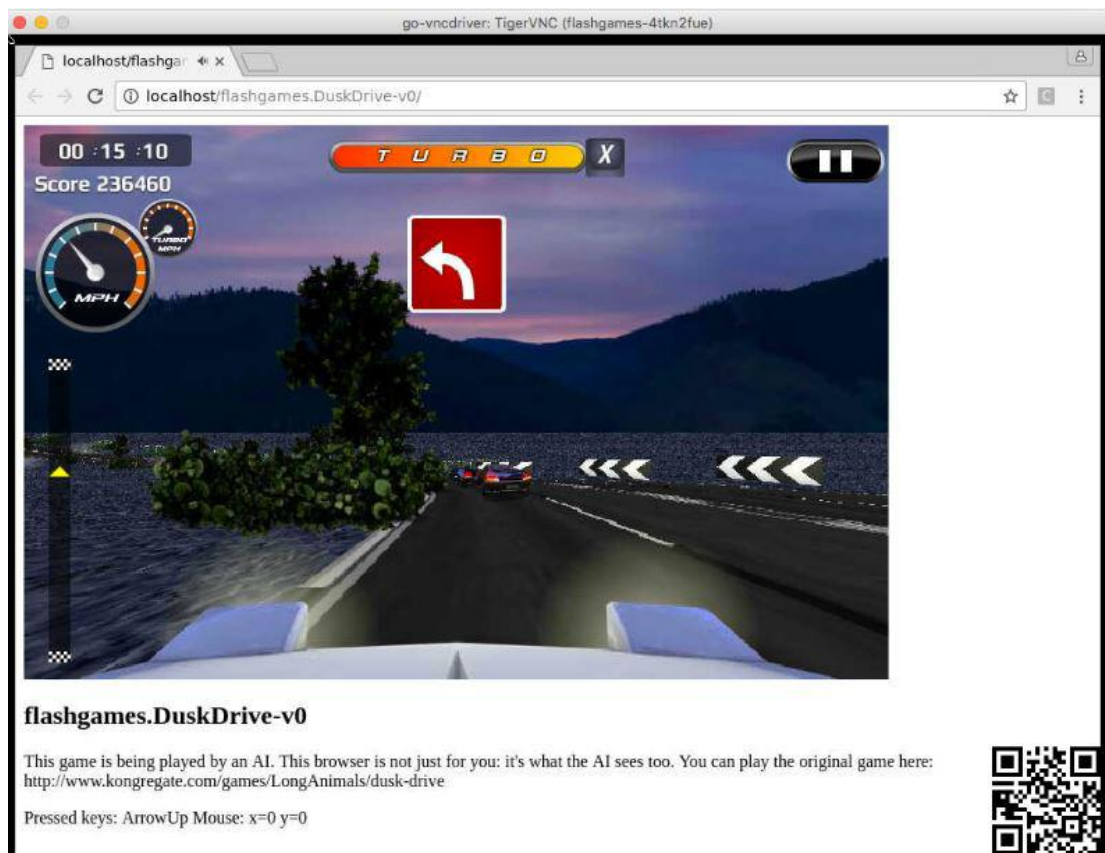
```
import gymimport universe # register the universe environments

env = gym.make('flashgames.DuskDrive-v0')
env.configure(remotes=1) # automatically creates a local docker container
observation_n = env.reset()
while True:
    action_n = [[('KeyEvent', 'ArrowUp', True)] for ob in observation_n] # your
agent here
    observation_n, reward_n, done_n, info = env.step(action_n)
    env.render()
```

保存后执行：

python universe.py

会看到下面的界面：



至此 universe 环境搭建完成。

### 3.8 安装 retro

```
pip install gym-retro
```



README.rst

This repository has been deprecated in favor of the Retro (<https://github.com/openai/retro>) library. See our Retro Contest (<https://blog.openai.com/retro-contest>) blog post for details.

[universe](#)

目前，universe 已经废弃，所以建议安装 retro

### 3.9 安装 SC2

```
git clone https://github.com/deepmind/pysc2.git
pip install pysc2/
```

下载：

<http://blzdistsc2-a.akamaihd.net/Linux/SC2.4.6.0.67926.zip>

<http://blzdistsc2-a.akamaihd.net/MapPacks/Ladder2019Season1.zip>

<http://blzdistsc2-a.akamaihd.net/MapPacks/Melee.zip>

```
sudo su
```

```
mv SC2.4.6.0.67926.zip ~
```

```
mv Ladder2019Season1.zip ~
```

```
mv Melee.zip ~
```

```
sudo ~
```

```
unzip SC2.4.6.0.67926.zip
```

```
unzip Ladder2019Season1.zip
```

```
unzip Melee.zip
```

解压密码：iagreetotheeula

```
cd StarCraft II
```

```
mkdir Maps
```

```
cd ..
```

```
cp -R Melee StarCraft II/Maps
```

```
python -m pysc2.bin.agent --map Simple64
```

可以看到下面的界面，则表明已成功：



至此，所有的环境搭建已经完成。

参考：

<https://github.com/openai/universe>  
<https://github.com/openai/mujoco-py>  
<https://github.com/openai/retro>  
<https://github.com/openai/gym>  
<https://github.com/deepmind/pysec2>