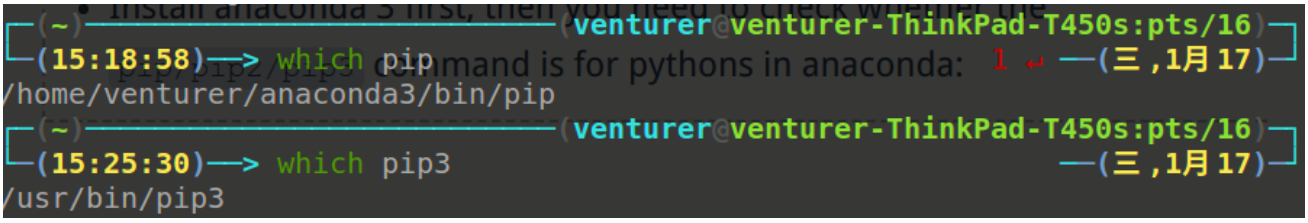
Install OpenAI gym and roboschool on Ubuntu 16.04 via Anaconda 3

- **Author : Yunqiu Xu**
- Reference:
  - https://github.com/openai/gym
  - http://blog.csdn.net/jinzhuojun/article/details/78508203
  - https://www.jianshu.com/p/350553547c09
  - https://www.jianshu.com/p/b9f14b8b1bab
- (Updated on 2018/01/21) Install new environment

---

1. Preparation

- Install anaconda 3 first, then you need to check whether the `pip/pip2/pip3` command is for pythons in anaconda:



- We can see that `pip3` is still for build-in python3, so we need to change its interpreter:
  - Open pip3 file
  - Change the first line to python3 in anaconda
  - You can check changes by running `pip3 list`

2. Install gym

- Install system packages:

```
sudo apt-get install -y python-numpy python-dev cmake zlib1g-dev
libjpeg-dev xvfb libav-tools xorg-dev python-opengl libboost-all-dev
```

```
libsdl2-dev swig
```

- Install gym:

```
pip3 install -e '.[all]'
```

- You can try to run "LunarLand" to check whether Box2D is installed correctly, if not, try following codes:

```
pip3 uninstall Box2D-kengz
git clone https://github.com/pybox2d/pybox2d.git
cd pybox2d
python3 setup.py clean
python3 setup.py install
```

3. Install roboschool

- Create a new folder "robot" in anaconda, then put roboschool in it:

```
mkdir /home/venturer/anaconda3/lib/python3.6/site-packages/robot
cd /home/venturer/anaconda3/lib/python3.6/site-packages/robot
git clone http://github.com/openai/roboschool.git
```

- Add environment variable in .zshrc

```
ROBOSCHOOL_PATH=/home/venturer/anaconda3/lib/python3.6/site-packages/robot/roboschool
export PATH="$PATH:ROBOSCHOOL_PATH"
```

- Make sure following packages are installed correctly:

```
sudo apt-get install cmake ffmpeg pkg-config qtbase5-dev libqt5opengl5-dev libassimp-dev libpython3.5-dev libboost-python-dev libtinyxml-dev
```

- Install bullet3

```
git clone https://github.com/olegklimov/bullet3 -b roboschool_self_collision
mkdir bullet3/build
```

```
cd    bullet3/build
cmake -DBUILD_SHARED_LIBS=ON -DUSE_DOUBLE_PRECISION=1 -DCMAKE_INSTALL_PREFIX:PATH=$ROBOSCHOOL_PATH/roboschool/cpp-household/bullet_local_install -DBUILD_CPU_DEMOS=OFF -DBUILD_BULLET2_DEMOS=OFF -DBUILD_EXTRAS=OFF -DBUILD_UNIT_TESTS=OFF -DBUILD_CLSOCKET=OFF -DBUILD_ENET=OFF -DBUILD_OPENGL3_DEMOS=OFF ..
make -j4
make install
cd ../..
```

- Install roboschool

```
pip3 install -e $ROBOSCHOOL_PATH
```

- Check whether roboschool is installed correctly:
  - Run `pip3 list`, you can see roboschool installed
  - Import roboschool in ipython or python3
- Run demos:

```
python3 $ROBOSCHOOL_PATH/agent_zoo/RoboschoolWalker2d_v1_2017jul.py
```

4. Install New Environment

- Here I will take SuperMario for instance: https://github.com/ppaquette/gym-super-mario
- Before installing you should follow the instruction to meet the requirments:
  - gym: **0.7.4**, if you use 0.9.4 gym-pull will not be installed successfully. Insteresting, after installing gym-pull, gym will be upgraded to 0.9.4 automatically :D
  - gym-pull
- Install new env via gym-pull: only required once, envs will be loaded with import gym_pull afterwards

```
import gym
import gym_pull
gym_pull.pull('github.com/ppaquette/gym-super-mario')
```

- Then copy the new env to gym/envs

```
git clone https://github.com/ppaquette/gym-super-mario
# copy gym-super-mario/ppaquette_gym_super_mario to your gym/envs
```

- Modify gym/envs/__init__.py

```
register(
     id='SuperMarioBros-1-1-v0',
     entry_point='gym.envs.ppaquette_gym_super_mario:MetaSuperMarioBrosEnv',
)
```

- Modify /gym/scoreboard/__init__.py

```
add_group(
     id='ppaquette_gym_super_mario',
     name='ppaquette_gym_super_mario',
     description='super_mario'
)
# mario bros
add_task(
     id='ppaquette/SuperMarioBros-1-1-v0',
     group='ppaquette_gym_super_mario',
     summary="SuperMarioBros-1-1-v0"
)
```

- Test new env:

```
import gym
env = gym.make('SuperMarioBros-1-1-v0')
observation = env.reset()
for _ in range(1000):
    env.render()
    action = env.action_space.sample() # your agent here (this takes random actions)
    observation, reward, done, info = env.step(action)
```