**DeepMinic项目在Ubuntu系统复现的过程记录**

原文网址：<https://github.com/xbpeng/DeepMimic>

写在前面：（新增操作使用绿色标识，修改的操作使用红色标识）

**0. 安装依赖，在Ubuntu终端分别输入以下命令**：

sudo apt install libgl1-mesa-dev libx11-dev libxrandr-dev libxi-dev ；

sudo apt install mesa-utils ；

sudo apt install clang ；

sudo apt install cmake

**1. 安装bullet 2.88**

从原文提供的网址下载安装包指定版本（.tar.gz格式），解压缩；

手动将build\_cmake\_pybullet\_double.sh文件中USE\_DOUBLE\_PRECISION=ON改为OFF；

安装python开发包，解决PYTHON\_INCLUDE系统变量问题，命令行：sudo apt-get install python-dev；

命令行：./build\_cmake\_pybullet\_double.sh，运行后显示编译完成completed；

命令行：cd build\_cmake；sudo make install

**2. 安装Eigen 3.3.7**

从原文提供的网址下载安装包指定版本（.tar.gz格式），解压缩；

终端命令行：mkdir build；cd build；cmake ..；sudo make install

**3. 安装OpenGL**（Windows系统可能自带，但Ubuntu系统本例还是再安装了）

命令行：sudo apt-get update；

sudo apt-get install libglu1-mesa-dev freeglut3-dev mesa-common-dev。

（参考： https://github.com/zhaolongkzz/DeepMimic\_configuration#2-bullet）

**4. freeglut 3.0.0**

从原文提供的网址下载安装包指定版本（.tar.gz格式），解压缩；

命令行：cmake . ；make；sudo make install

**5. glew 2.1.0**

从原文提供的网址下载安装包指定版本（.tar.gz格式，不要下载成win），解压缩；

命令行：

sudo apt-get install build-essential libxmu-dev libxi-dev libgl-dev

make

sudo make install

make clean

**6. swig 4.0.0**

从原文提供的网址下载安装包指定版本（.tar.gz格式），解压缩；

命令行：./configure --without-pcre；make；sudo make install

添加swig路径到环境变量，参考自zhaolongkzz：

SWIG\_PATH=/usr/local/share/swig/4.0.0

PATH=$PATH:$SWIG\_PATH

**7. MPI**

命令行：sudo apt install libopenmpi-dev

**8. python**

命令行输入 python3，查看Ubuntu自带python版本，Ubuntu18自带版本3.6.9

（慎用3.7以上的，因为只有3.7以下的TensorFlow版本才有本文使用的1.13.1）

**9. PyOpenGL**

命令行：pip3 install PyOpenGL PyOpenGL\_accelerate （python3对应pip3，开始可能会要求安装pip3）

**10. Tensorflow 1.13.1**

命令行：pip3 install tensorflow==1.13.1

（如前所言，只有python3.7以下支持1.13.1的TensorFlow，更高版本的tensorflow未经测试）

**11. MPI4Py**

命令行：pip3 install mpi4py

**12. 修改makefile参数**

首选：用zhaolongkzz提供的makefile文件覆盖原有的makefile，将DeepMimicCore/util/MathUtil.cpp中的数值由0.001—>0.1，然后再make python

<https://github.com/zhaolongkzz/DeepMimic_configuration/blob/master/DeepMimicCore/Makefile>

次之：

EIGEN\_DIR：usr/local/include/eigen3

BULLET\_INC\_DIR：document/deepminicdependency/bullet3-2.88/src （源文件解压位置）

# 从手动安装的python上建立的虚拟环境不可行，需要anaconda虚拟环境或python原环境。

PYTHON\_INC：usr/include/python3.6m

PYTHON\_LIB：usr/lib/python3.6

**13. build wrapper** 定位到DeepMinicCore子文件夹

命令行：make python （不是make python3）

**14. 复制libGLEW**，否则运行demo的时候会报该文件缺失

命令行：sudo cp “usr/lib64/libGLEW.so” “usr/lib” ；

sudo cp “usr/lib64/libGLEW.so.2.1” “usr/lib” ；

sudo cp “usr/lib64/libGLEW.so.2.1.0” “usr/lib” ；

（直接CRTL+V权限不够）

**15. 运行demo**，回到主文件夹下

命令行：python3 DeepMimic.py --arg\_file args/run\_humanoid3d\_spinkick\_args.txt （python3不是python）

**DeepMinic项目在Windows系统复现的过程记录**

参考博客：

<https://note.youdao.com/ynoteshare1/index.html?id=097cb5f90e53bc8d8b662439f3cb4285&type=note>

注意要点：

0. 预先下载（压缩包zip格式）

软件：Anaconda（推荐3.5.2版本，python3.7以下），Visual Studio, Cmake

DeepMimic：<https://github.com/xbpeng/DeepMimic>，

Bullet2.87.zip：[https://github.com/bulletphysics/bullet3/releases](https://github.com/bulletphysics/bullet3/releases%20%20Eigen3.3.7)

Eigen3.3.7：<http://www.eigen.tuxfamily.org/index.php?title=Main_Page>

Feeglut3.3.0: <http://freeglut.sourceforge.net/>

Glew2.1.0: <http://glew.sourceforge.net/> （glew可以使用编译好的二进制文件）

Swig4.0.0: <http://www.swig.org/>

MSMPI: <https://docs.microsoft.com/en-us/message-passing-interface/microsoft-mpi>

1. 在计算机环境变量中添加PYTHON\_INCLUDE, PYTHON\_LIB, SWIG\_DIR变量，在系统变量Path中添加值：C:\windows\system32， C:\windows\sysWoW64。同时将glew32.dll, glfw3.dll的32位和64位版本分别在以上两个文件夹保存。（glfw3.dll额外下载的）。安装MSMPI（两个包）

2. 在Anaconda Prompt中安装包：

Pip install tensorflow==1.13.1;

Pip install PyOpenGL==3.1.0;

Pip install PyOpenGL-accelerate==3.1.0

Pip install mpi4py

（直接安装不指定版本号会报错，如不能初始化glutInit()等）

3. freeglut自行编译（release/debug, x64），编译后的freeglut.dll和freeglutd.dll都放入C:\windows\system32， C:\windows\sysWoW64各一份。

4. bullet编译，本文使用3-2.87版本，上述博客建议修改build\_visual\_studio\_vr\_pybullet\_double.bat中python的安装路径后双击安装（bullet的官方repo也这么说），但这样安装后的库文件是在原始bullet中一个叫bin的子文件夹里（.bat文件中也是这样设计的），但这样始终报错。

本文采用的方式是先用cmake编译（可以打开Cmakelists文件查看双精度选项，但应该已经是OFF了，所以可以直接cmake，指定编译后的文件夹与bullet源文件在同一上层文件夹下），cmake中configure选64位，然后generate，open project（打开的应该是BULLET\_PHYSICS.sln文件），里面BulletDynamics，BulletCollision，以及LinearMath三个项目文件的预定义中也是默认没有双精度。直接build完成后（release, x64），生成的库文件在bullet3build（cmake编译文件夹）\lib\Release中，也没有多余后缀，可以直接使用。

这步出错可能会出现LNK2001: unresolved external symbol

Issue75: <https://github.com/xbpeng/DeepMimic/issues/75>

5. 打开DeepMimicCore.sln，在additional include 和additional library中配置依赖，bullet src是指原文件的src,不是编译后的。如果之前没有复制freeglut和glew的头文件到VS安装目录，此处需要在项目配置中指定。我这里还额外指定了他们的库文件。

在release\_swig，x64下编译DeepMimicCore，一定注意之前刚才的项目配置也是在release\_swig，x64下

6. 运行

方法一（命令行）：打开Anaconda Prompt并定位到DeepMimic目录，输入参数，如：

python DeepMimic.py --arg\_file args/run\_humanoid3d\_spinkick\_args.txt

方法二（图形界面）：在DeepMimic目录中打开DeepMimic.pyproj文件并修改python版本参数Global|PythonCore|3.5，本例中将3.5改为3.6，打开DeepMimic.sln，打开项目属性，设置启动文件为DeepMimic.py，Script Arguments如--arg\_file args/run\_humanoid3d\_spinkick\_args.txt。运行，如果DeepMimicCore编译成功，但此处报错缺少dll，则是第一步中的有些dll未被放入文件夹。

方法三（图形界面）：不打开DeepMimic.sln，也可以直接打开DeepMimic.pyproj，前提是按照方法二修改了参数，并且在VS上安装对python的支持。