

使用CoppeliaSim运行NAO机器人虚拟环境

作者：丁家恩

联系邮箱：labmemzzf@qq.com

2022/12/17

一：问题描述

NAO机器人官方在图形化工具 `Choregraphe` 中提供了强大的动作设计功能盒子 `Timeline`，见下图 1.1。`Timeline` 中设计完成的动作序列可以在 `Choregraphe` 中进行一定的仿真模拟，对于简单的、短时间的动作序列，这种仿真的效果是足够的；但是对于复杂的、长时间的动作序列，由于 `Choregraphe` 缺少完成的物理引擎，该仿真模拟无法满足仿真需求，例如，对于舞蹈动作的模拟。

对于 `Timeline` 中复杂动作序列的仿真模拟，通常的做法是将动作序列导出为代码，适配到第三方软件中运行仿真。该方法缺乏便利性与实时性，也提高了初学者的学习成本，故本文介绍一种连接 `Choregraphe` 与第三方仿真软件的方法，使得在 `Timeline` 盒子中的动作序列可以实时在第三方仿真软件中运行，见下图 1.2。

图1.1： `Choregraphe` 中具备强大功能的动作设计盒子 `Timeline`，图形化交互良好。

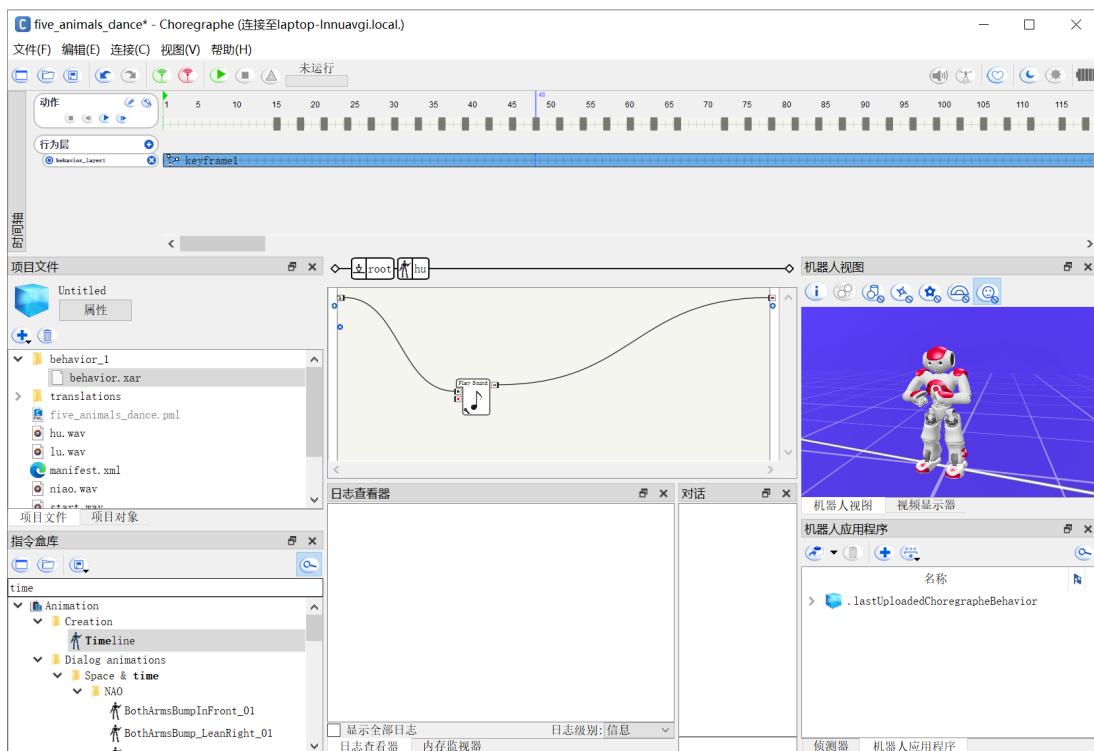
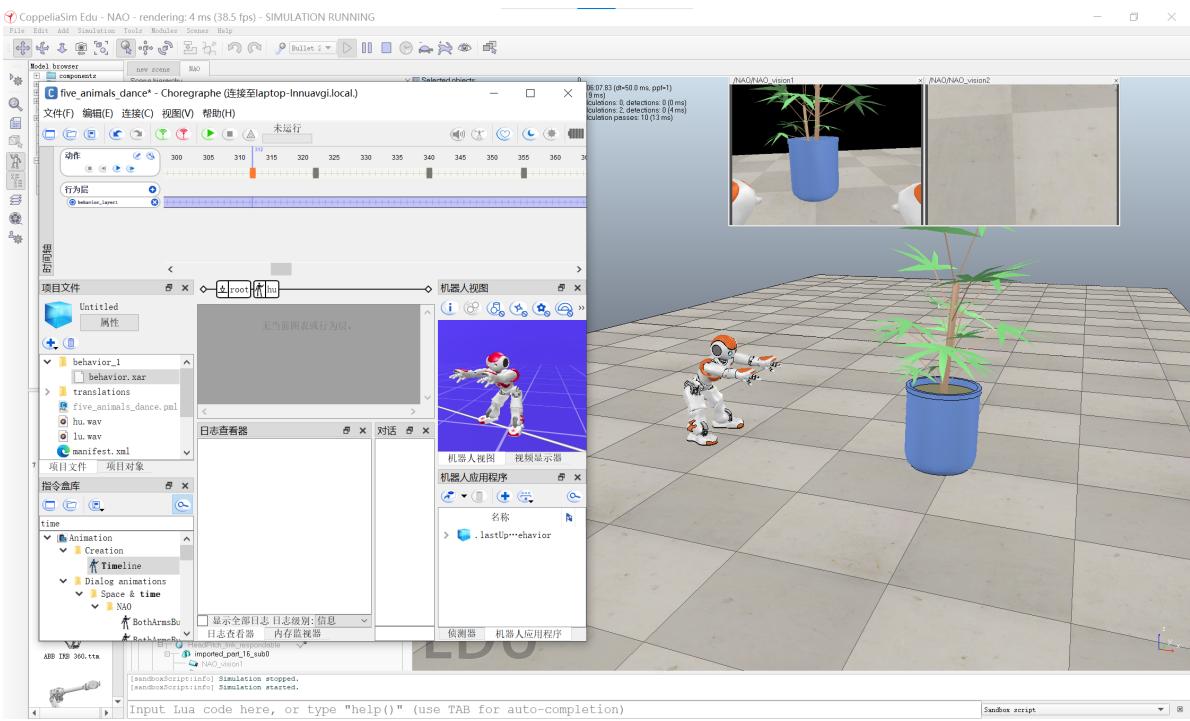


图2.2： `Choregraphe` 与第三方仿真软件 `CoppeliaSim` 连接，实时运行 `Timeline` 盒子中的动作序列。通过这种方式，可以很便利地完成复杂动作的设计与仿真过程。



二：环境依赖

详细的安装说明在最后章节讨论，本章节仅作简介。

- 本文将通过运行 `github` 开源代码的方式完成目标，因此，首先需要感谢项目原作者 [PierreJac](#) 对开源社区的贡献。开源项目地址页面：<https://github.com/PierreJac/Project-NAO-Control/tree/rework>。
- 另外，本文需要使用到仿真模拟软件 `coppeliasim`，原名 `V-REP`，官网地址：<https://www.coppearobotics.com/>。该模拟软件提供多个版本下载，其中，个人版和教育版免费使用。教育版声明如下：“May only be used by students, teachers, professors, schools and universities.” 本文将下载教育版使用。下载页面直达：<https://www.coppearobotics.com/downloads>。
- `NAO` 机器人其他所需软件资源，访问<https://www.aldebaran.com/en>主页获取，包括本文所需 `Choregraphe` 与 `Python NAOqi-SDK` 的下载安装，以及文档阅读等。软件资源下载页面直达：[http://www.aldebaran.com/fr/support/nao-6/downloads-softwares](https://www.aldebaran.com/fr/support/nao-6/downloads-softwares)。
- `Python NAOqi-SDK` 依赖 `Python2.7` 运行环境，访问官网<https://www.python.org/>。对应需要下载版本与 `Python NAOqi-SDK` 版本有关，请参考下文。下载页面直达：<https://www.python.org/downloads/>。

三：快速启动指南

1. NAO V6 环境（六代机器人，产品外壳颜色为灰黑色）

- 安装运行环境，并下载 `github` 项目代码。
- 解压缩 `github` 项目文件可得文件夹“`Project-NAO-Control-rework`”。
- 打开“`Project-NAO-Control-rework`”，对代码做出两处修改并保存：

```

# 第一处修改
# Scripts\single_nao_control.py 第35行到第38行
...
posture = 'StandZero'
print 'Posture Initialization : ' + posture
motionProxy.stiffnessInterpolation('Body', 1.0, 1.0)
postureProxy.goToPosture(posture,1.0,1.0)
...

# 将上面这段代码完全删除或者注释，改为

...
# posture = 'StandZero'
# print 'Posture Initialization : ' + posture
# motionProxy.stiffnessInterpolation('Body', 1.0, 1.0)
# postureProxy.goToPosture(posture,1.0,1.0)
...

```

```

# 第二处修改
# scripts\vision_sensor.py 第59行
...
clientID=vrep.simxStart('127.0.0.2',19999,True,True,5000,5)
...

# 将上面的代码修改为

...
clientID = vrep.simxStart('127.0.0.1', 19997, True, True, 5000, 5)
...

```

- 运行 coppeliasim Edu 并点击左上角菜单"Files->Open Scene"打开文件"Project-NAO-Control-rework\Vrep-Scene\NAO.ttt"
- 运行 choregraphe 2.8.6，依次点击左上角菜单"编辑->首先项->虚拟机器人->机器人模式->NAO H25 (V6) ->确定"，再点击左上角菜单"连接->连接至虚拟机器人"。见图3.3.1。

图3.3.1。

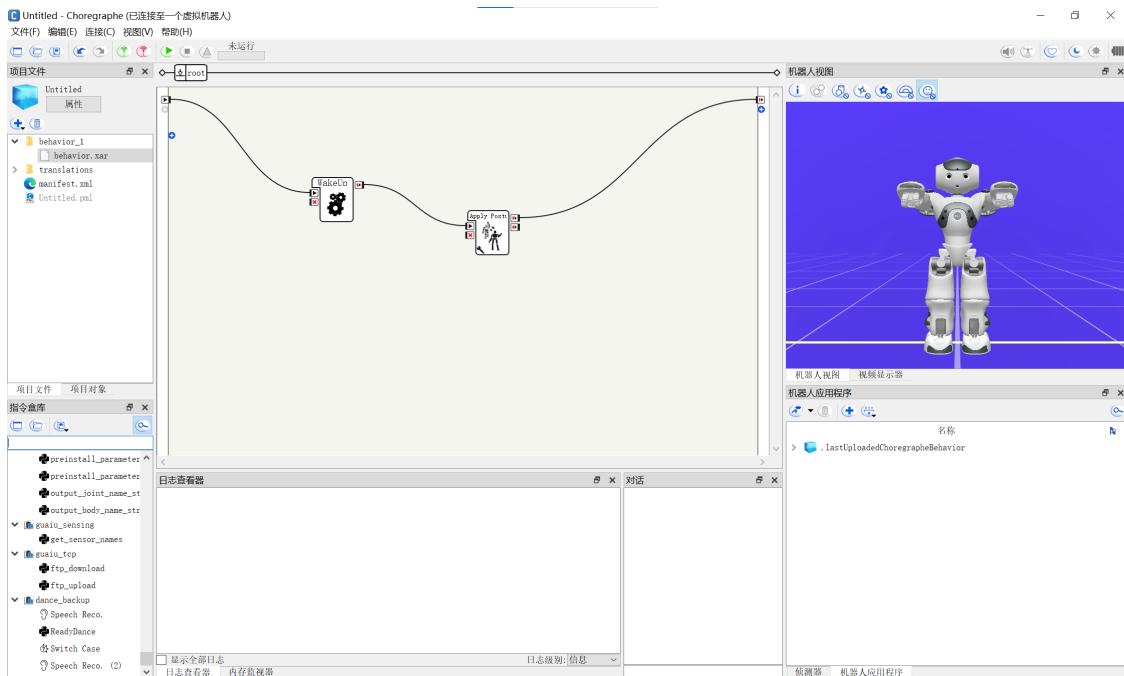


- 关闭虚拟机器人自主生活模式。点击 choregraphe 右上角爱心图标处选择开启或者关闭该模式。



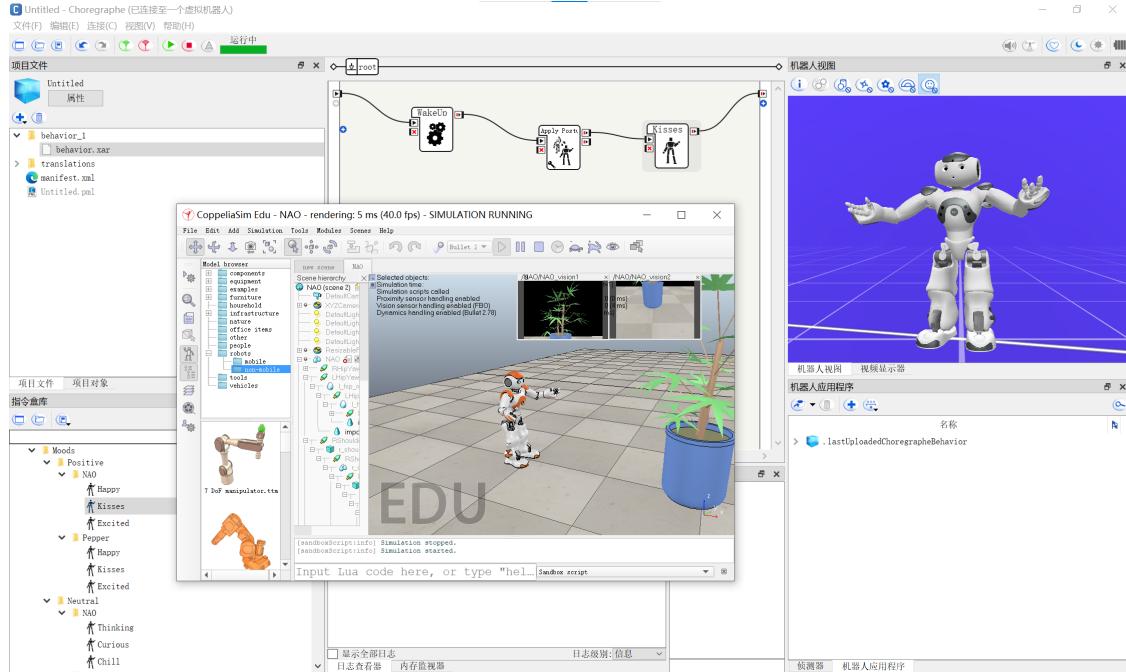
- 运行 coppeliasim 仿真场景，仿真机器人初始化。点击上方右箭头图标 开启仿真。
- 可选：虚拟机器人初始姿态与仿真机器人初始姿态同步化。不执行该选项可能导致 coppeliasim 中的仿真机器人在初次连接的过程中摔倒。在 choregraphe 拉取并依次连接盒子 "WakeUp -> Apply Posture (参数Name的值设置为'StandZero'，可点击盒子左下角扳手图标进行设置)"，并执行一次，如下图3.1.2。

图3.1.2。



- 使用 python2 环境运行"Project-NAO-Control-rework\scripts\single_nao_control.py"。
- 根据提示输入IP值与端口值。若是本地虚拟机器人，IP值输入"127.0.0.1"；端口值查看"编辑->首先项->虚拟机器人"，菜单栏下方"Naqi的状态"处显示当前虚拟机器人端口值。
- `choregraphe` 中运行的盒子将在 `coppeliasim` 仿真中同步实现效果。
- Enjoy! (见图3.1.3)

图3.1.3。



2. NAO V5 环境 (五代机器人，产品外壳颜色为蓝色或红色)

- 安装运行环境，并下载 `github` 项目代码。
- 解压缩 `github` 项目文件可得文件夹"Project-NAO-Control-rework"。
- 打开"Project-NAO-Control-rework"，对代码做出两处修改并保存：

```
# 第一处修改
# Scripts\single_nao_control.py 第35行到第38行
...
posture = 'Standzero'
print 'Posture Initialization : ' + posture
motionProxy.stiffnessInterpolation('Body', 1.0, 1.0)
postureProxy.goToPosture(posture,1.0,1.0)
...

# 将上面这段代码完全删除或者注释，改为
...
# posture = 'Standzero'
# print 'Posture Initialization : ' + posture
# motionProxy.stiffnessInterpolation('Body', 1.0, 1.0)
# postureProxy.goToPosture(posture,1.0,1.0)
```

```

# 第二处修改
# scripts\vision_sensor.py 第59行
...
clientID=vrep.simxStart('127.0.0.2',19999,True,True,5000,5)
...

# 将上面的代码修改为
...
clientID = vrep.simxStart('127.0.0.1', 19997, True, True, 5000, 5)
...

```

- 运行 coppeliasim Edu 并点击左上角菜单"Files->Open Scene"打开文件"Project-NAO-Control-rework\Vrep-Scene\NAO.ttt"
- 运行 choregraphe 2.1.4, 依次点击左上角菜单"编辑->首选项->虚拟机器人->机器人模式->NAO H25 (V50) ->确定", 再点击左上角菜单"连接->连接至虚拟机器人"。见图3.2.1。

图3.2.1。

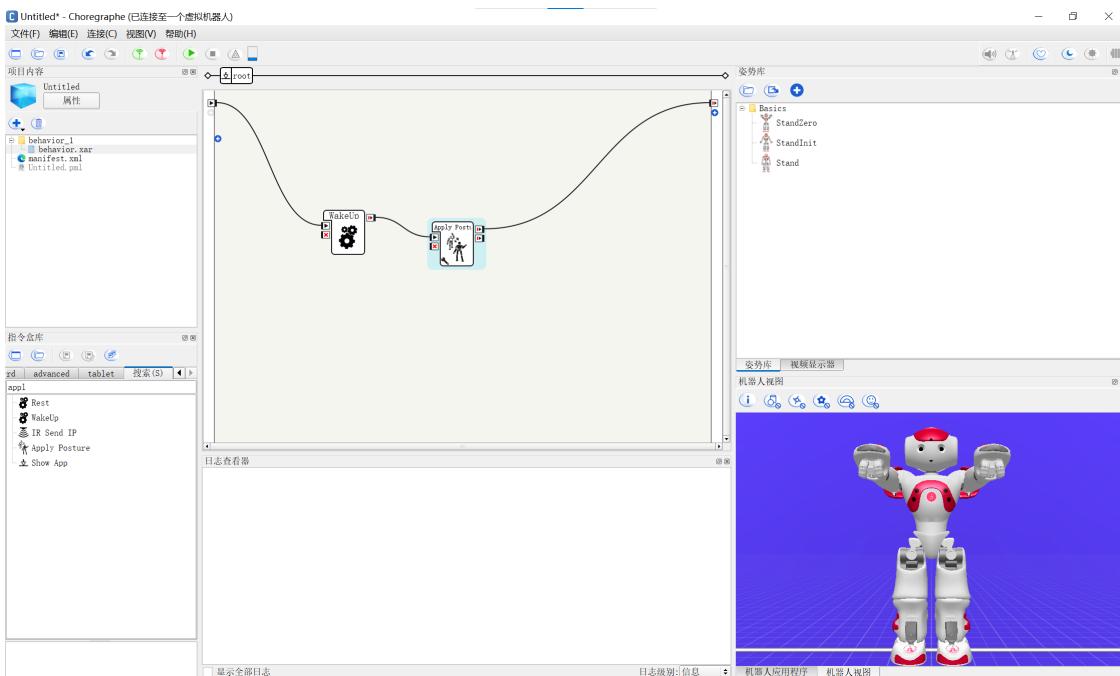


- 关闭虚拟机器人自主生活模式。点击 choregraphe 右上角爱心图标处选择开启或者关闭该模式。



- 运行 coppeliasim 仿真场景，仿真机器人初始化。点击上方右箭头图标 开启仿真。
- 可选：虚拟机器人初始姿态与仿真机器人初始姿态同步化。不执行该选项可能导致 coppeliasim 中的仿真机器人在初次连接的过程中摔倒。在 choregraphe 拉取并依次连接盒子"WakeUp -> Apply Posture (参数Name的值设置为'StandZero', 可点击盒子左下角扳手图标进行设置)"，并执行一次，如下图3.2.2。

图3.2.2。



- 使用 python2 环境运行"Project-NAO-Control-rework\scripts\single_nao_control.py"。

```
...\\Project-NAO-Control-rework\\scripts> python2 .\\single_nao_control.py
```

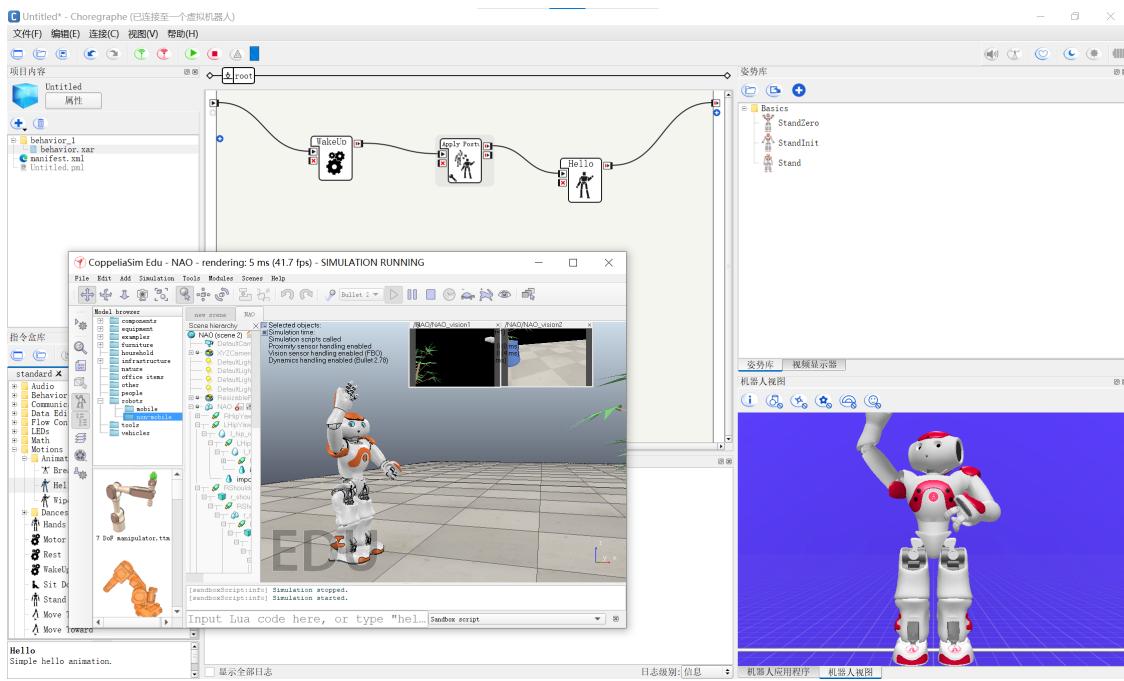
或者

```
...\\Project-NAO-Control-rework\\scripts> python .\\single_nao_control.py
```

取决于你的环境变量中启动 `python2` 程序的指令是"`python2`"还是"`python`"。

- 根据提示输入IP值与端口值。若是本地虚拟机器人，IP值输入"127.0.0.1"；端口值查看"编辑->首选项->虚拟机器人"，菜单栏下方"Naqi的状态"处显示当前虚拟机器人端口值。
- `Choregraphe` 中运行的盒子将在 `CoppeliaSim` 仿真中同步实现效果。
- 程序运行过程中 `CoppeliaSim` 中仿真机器人摔倒的话，可以点击上方正方形图标停止仿真再点击箭头图标重启仿真 ，然后按照上文步骤重新进行连接流程。
- 每次执行动作盒子前，请先确保机器人已经用"WakeUp"盒子唤醒。
- enjoy! (见图3.2.3)

图3.2.3。



四：环境安装说明

本文全程在 windows10 操作系统环境中演示进行。另外，对于有经验的读者，推荐使用 docker 软件进行环境的配置与管理，本文不做讨论。

下文将提供通用资源，以及两种硬件版本的 NAO 机器人环境安装推荐方案，理由如下文。

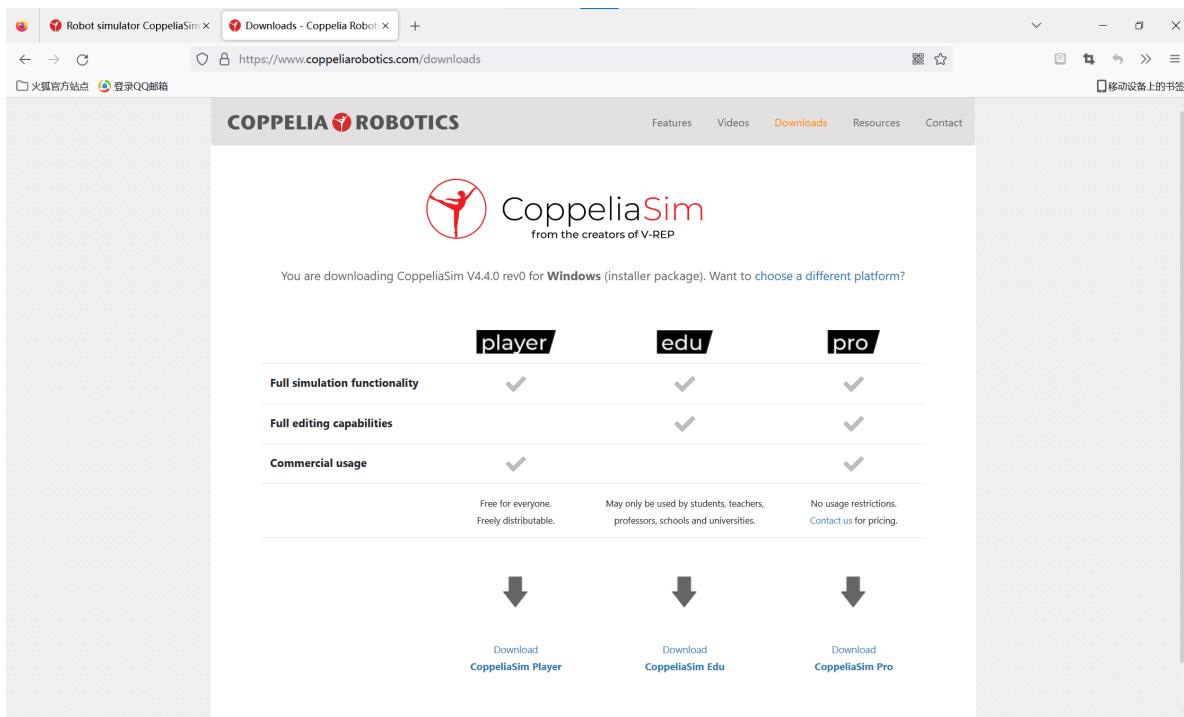
NAO 机器人的最新一代产品为 NAO V6，也即 NAO 六代机器人，产品外壳颜色为灰黑色，使用最新版本的 Choregraphe 与 Python NAOqi-SDK 软件。然而，目前仍有大量旧式的 NAO 机器人在使用中，例如 NAO V5 也即 NAO 五代机器人，产品外壳颜色为红色或蓝色；更重要的是，github 上的 NAO 相关代码大多不是在 NAO V6 环境中编写的，包括本文所使用到的项目，流行的仿真软件上的 NAO 模型也停留更旧式的产品版本上(甚至是 NAO V3 与 NAO V4，产品外壳颜色有诸如橙色等)。因此，仅仅讨论 NAO 六代机器人环境的话，可能会出现适配问题，故本文提供 NAO V6 与 NAO V5 两种机器人环境的安装推荐方案，以较少适配性问题。

1. CoppeliaSim 安装

首先需要下载最新版本的 coppeliasim 进行安装，目前为止最新的版本号为"Release 4.4.0"。选择最新版本的理由有二：一，安装更旧版本的软件会受到内置时间限制政策的影响，也即强制更新政策，坚持使用旧版本软件需要修改配置文件，可参考https://blog.csdn.net/sunrise_lk/article/details/124245609；二，新版本中 NAO 模型未有更新，与旧版本一致，使用旧版本没有意义。本文使用软件版本为"Release 4.4.0"。

访问<https://www.coppeliarobotics.com/downloads>进行软件下载。在 window10 平台上有一个 .exe 版本安装包下载，运行安装包按照提示安装即可。

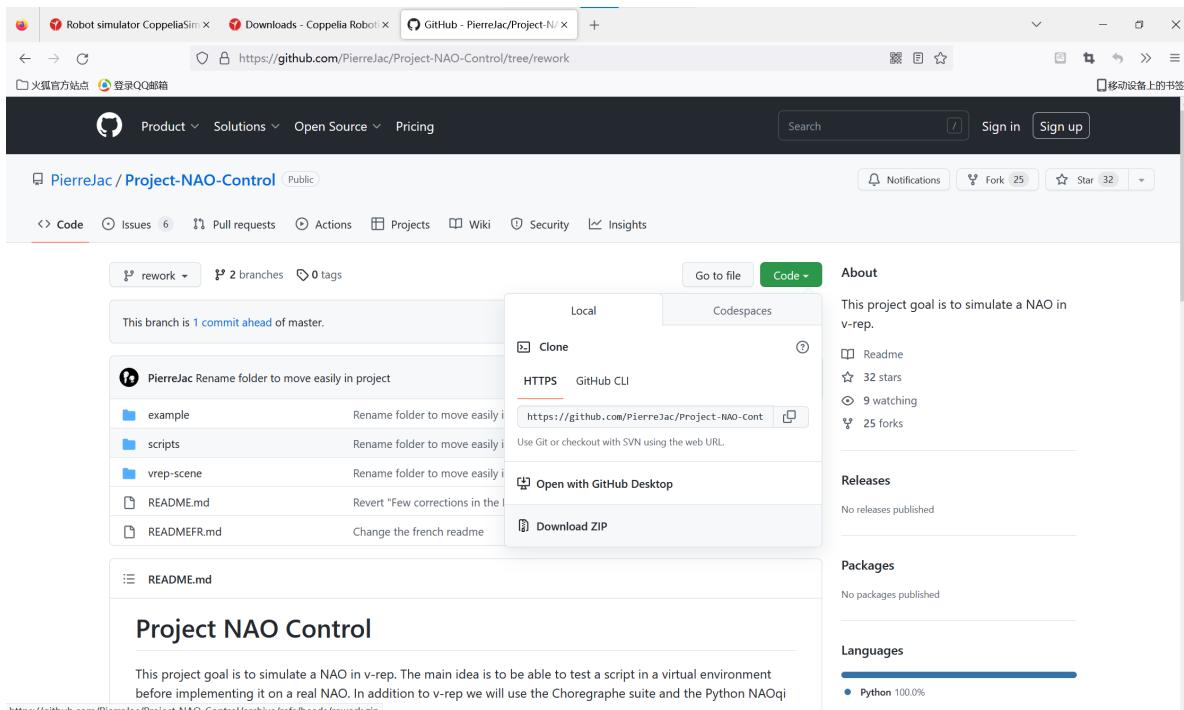
图3.1.1.a: CoppeliaSim 安装页面，选择教育版点击页面下方"Download"链接安装。需要注意页面上方"Want to choose a different platform? "处可以点击进行操作平台版本切换，下图中已切换至 windows 版本。



2. github项目源码拉取

访问开源项目地址：<https://github.com/PierreJac/Project-NAO-Control/tree/rework>，需要注意本地址已切换至"rework"版本，这是我们所需要的项目分支版本。点击页面中"Code"按钮，在下拉菜单中选择点击"Download ZIP"等待下载完成。下载的 .zip 文件需要解压，推荐使用解压软件 7-zip，软件主页为<https://www.7-zip.org/>，请读者自行下载安装或者使用其他的解压软件。

图3.1.2.a: github项目页面，点击"Code"按钮展开下拉菜单。注意项目分支已切换至"rework"版本，若未切换至该版本可点击下图"rework"按钮处（若为其他分支版本，按钮处显示"master"或其他）进行切换。



3. NAO V6 机器人环境安装

NAO V6 需要下载的 choregraphe 版本为"2.8.6 Setup"，需要下载的 Python NAOqi-SDK 版本为"2.1.4 Python 2.7 SDK Setup"（要先安装 Python2.7.13 运行环境）。

下载直链：

Choregraphe-windows-2.8.6-Setup:

<https://community-static.aldebaran.com/resources/2.8.6/choregraphe-suite-2.8.6.23-win64-vs2015-setup.exe>

naoqi-windows-Python2.7-SDK-2.1.4.13-Setup:

<https://community-static.aldebaran.com/resources/2.1.4.13/sdk-python/pynaoqi-2.1.4.13.win32.exe>

Python2.7.13:

<https://www.python.org/ftp/python/2.7.13/python-2.7.13.amd64.msi>

安装顺序为：Python2.7、Python NAOqi-SDK、Choregraphe。三种安装包均是图像化安装，读者按照提示依次安装即可。有安装困难的，请读者参考对应安装文档，并自行验证安装是否成功。

特别说明的是 NAO V6 的 Python NAOqi-SDK 是有支持 Python3 版本的，但是并不支持 windows 环境，有经验的读者可以尝试。另外，不选择更高版本的 Python NAOqi-SDK 版本的理由是更高的版本没有 .exe 安装包版本，仅有 .zip 版本或其他版本，有经验的、有更高需求的读者可以尝试，在本文中"2.1.4 Python 2.7 SDK Setup"版本已足够使用。

不推荐 NAO V6 使用更低版本的 Choregraphe，部分功能无法使用。

4. NAO V5 机器人环境安装

NAO V5 需要下载的 choregraphe 版本为"2.1.4 Setup"，需要下载的 Python NAOqi-SDK 版本为"2.1.4 Python 2.7 SDK Setup"（要先安装 Python2.7.13 运行环境）。

下载直链：

Choregraphe-windows-2.1.4-Setup:

<https://community-static.aldebaran.com/resources/2.1.4.13/choregraphe/choregraphe-suite-2.1.4.13-win32-setup.exe>

naoqi-windows-Python2.7-SDK-2.1.4.13-Setup:

<https://community-static.aldebaran.com/resources/2.1.4.13/sdk-python/pynaoqi-2.1.4.13.win32.exe>

Python2.7.13:

<https://www.python.org/ftp/python/2.7.13/python-2.7.13.amd64.msi>

安装顺序为：Python2.7.13、Python NAOqi-SDK、Choregraphe。三种安装包均是图像化安装，读者按照提示依次安装即可。有安装困难的，请读者参考对应安装文档，并自行验证安装是否成功。

不推荐 NAO V5 使用更高版本的 Choregraphe，部分功能无法使用。对于更旧产品版本的 NAO 机器人，可以尝试更旧版本的 Choregraphe 版本，推荐的版本是 1.14，但是这些版本的软件官网已经停止下载，需要读者自行查找。

五：更多

- 更多功能请阅读<https://github.com/PierreJac/Project-NAO-Control/tree/rework>页面说明文档，包括多机器人连接、视觉画面连接等。
- 同类优秀项目，推荐<https://github.com/cyberbotics/naoqisim/tree/master>。该项目使用 webots 作为第三方仿真运行平台。