Build and Installing CARLA on Linux

Refer to: https://carla.readthedocs.io/en/latest/build_linux/

-1 system check

Requirements

System specifics

- Ubuntu 18.04. CARLA provides support for previous Ubuntu versions up to 16.04. However proper compilers are needed for UE to work properly. The required dependencies for both Ubuntu 18.04 and previous versions are listed below. Make sure to install the ones corresponding to the system.
- 30GB disk space. Installing all the software needed and CARLA itself will require quite a lot of space, especially Unreal Engine. Make sure to have around 30/50GB of free disk space.
- An adequate GPU. CARLA aims for realistic simulations, so the server needs at least a 4GB GPU. A dedicated GPU is highly recommended for machine learning.
- Two TCP ports and good internet connection. 2000 and 2001 by default. Be sure neither the firewall nor any other application block these.

-2 Dependencies

CARLA needs many dependencies to run. Some of them are built automatically during this process, such as Boost.Python. Others are binaries that should be installed before starting the build (cmake, clang, different versions of Python and much more). In order to do so, run the commands below in a terminal window.

```
sudo apt-get update &&
sudo apt-get install wget software-properties-common &&
sudo add-apt-repository ppa:ubuntu-toolchain-r/test &&
wget -O - https://apt.llvm.org/llvm-snapshot.gpg.key|sudo apt-key add - &&
sudo apt-add-repository "deb http://apt.llvm.org/xenial/ llvm-toolchain-xenial-8 main" &&
sudo apt-get update
```

Type this into your command line first (bear in mind, it is one line, not multiple lines):

sudo apt-get update && sudo apt-get install wget software-properties-common && sudo add-apt-repository ppa:ubuntu-toolchain-r/test && wget -O - https://apt.llvm.org/llvm-snapshot.gpg.key|sudo apt-key add - && sudo apt-add-repository "deb http://apt.llvm.org/xenial/ llvm-toolchain-xenial-8 main" && sudo apt-get update

choose Yes or type ENTER when you encounter questions

Important

The following commands differ depending on the Ubuntu version. While the only change is `libpng16-dev` becoming `libpng-dev`, the full set of commands is here twice to ease the copy.

Ubuntu 18.04.

```
sudo apt-get install build-essential clang-8 lld-8 g++-7 cmake ninja-build libvulkan1 python py
pip2 install --user setuptools &&
pip3 install --user -Iv setuptools==47.3.1 &&
pip2 install --user distro &&
pip3 install --user distro
```

Previous Ubuntu versions.

```
sudo apt-get install build-essential clang-8 lld-8 g++-7 cmake ninja-build libvulkan1 python py
pip2 install --user setuptools &&
pip3 install --user -Iv setuptools==47.3.1 &&
pip2 install --user distro &&
pip3 install --user distro
```

Type this into your command line:

sudo apt-get install build-essential clang-8 lld-8 g++-7 cmake ninja-build libvulkan1 python python-pip python-dev python3-dev python3-pip libpng-dev libtiff5-dev libjpeg-dev tzdata sed curl unzip autoconf libtool rsync libxml2-dev libxerces-c-dev

pip2 install --user setuptools &&
pip3 install --user -Iv setuptools==47.3.1 &&
pip2 install --user distro &&
pip3 install --user distro

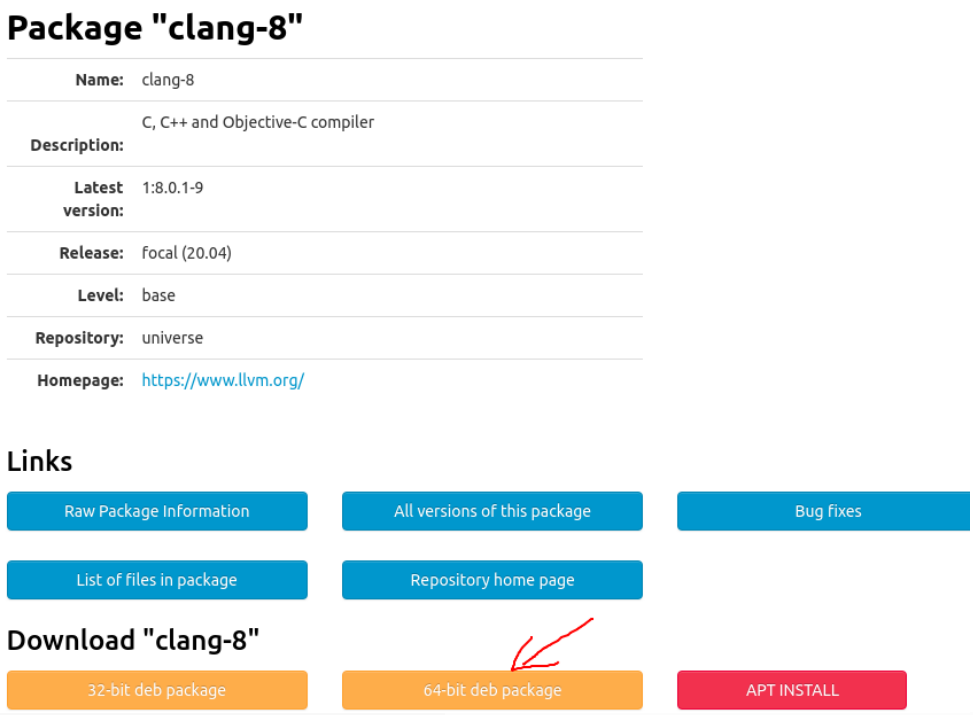
Break into sub-steps(in my case):

autoconf

build-essential

clang-8 (clang-10)

https://www.ubuntuupdates.org/package/core/focal/universe/base/clang-8



sudo apt install ./clang-8_8.0.1-9_amd64.deb

```
(base) lineojcd@pop-os:~/Downloads$ sudo apt install ./clang-8_8.0.1-9_amd64.de
b
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'clang-8' instead of './clang-8_8.0.1-9_amd64.deb'
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
 clang-8 : Depends: libclang-common-8-dev (= 1:8.0.1-9) but 1:8.0.1+svn369350-1
~exp1~20200112113617.82 is to be installed
           Recommends: libomp-8-dev but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
```

sudo apt install llvm-8-dev=1:8.0.1-9
sudo apt install libllvm8=1:8.0.1-9
sudo apt install libclang-common-8-dev=1:8.0.1-9
Be careful with clang-8, you have to make sure the dependencies version is met also.

sudo apt install libomp-8-dev

cmake

curl
g++-7 (g++-9)

```
g++-7/focal,now 7.5.0-6ubuntu2 amd64 [installed]
g++-9/focal,now 9.3.0-10ubuntu2 amd64 [installed,automatic]
```

libpng-dev
libtiff5-dev
libjpeg-dev
libvulkan1
libtool
libxml2-dev
libxerces-c-dev
lld-8
-A install libffi6 https://stackoverflow.com/questions/61875869/ubuntu-20-04-upgrade-python-missing-libffi-so-6

I am using Xubuntu 20.04 and recompiling the python version 3.7 did not work for me.

The way I solved this was to download the 19.10 version of the package from here:
http://mirrors.kernel.org/ubuntu/pool/main/libf/libffi/libffi6_3.2.1-8_amd64.deb

and then installing it

```
sudo apt install ./libffi6_3.2.1-8_amd64.deb
```

This will unpack the `libffi.so.6` and `libffi.so.6.0.4` files to `/usr/lib/x86_64-linux-gnu/`. The `libffi.so.6` file is just a link to `libffi.so.6.0.4` in the same directory.

As far as I could see this does not overwrite any files so should be safe.

Hopefully this helps someone as well.

-B sudo apt-get install libllvm8
-C sudo apt-get install lld-8
ninja-build
python
python-pip
How to Install Python Pip on Ubuntu 20.04
https://linuxize.com/post/how-to-install-pip-on-ubuntu-20.04/#installing-pip-for-python-2

python-dev
python3-dev
python3-pip
rsync
sed
tzdata
unzip

Next:
pip2 install --user setuptools &&
pip3 install --user -Iv setuptools==47.3.1 &&
pip2 install --user distro
pip3 install --user distro

To avoid compatibility issues between Unreal Engine and the CARLA dependencies, it is recommended to use the same compiler version and C++ runtime library to compile everything. The CARLA team uses clang-8 and LLVM's libc++. Change the default clang version to compile Unreal Engine and the CARLA dependencies.

```
sudo update-alternatives --install /usr/bin/clang++ clang++ /usr/lib/llvm-8/bin/clang++ 180 &&
sudo update-alternatives --install /usr/bin/clang clang /usr/lib/llvm-8/bin/clang 180
```

```
(base) lineojcd@pop-os:~/Downloads$ sudo update-alternatives --install /usr/bin/
clang++ clang++ /usr/lib/llvm-8/bin/clang++ 180 && sudo update-alternatives --in
stall /usr/bin/clang clang /usr/lib/llvm-8/bin/clang 180
update-alternatives: using /usr/lib/llvm-8/bin/clang++ to provide /usr/bin/clang
++ (clang++) in auto mode
update-alternatives: using /usr/lib/llvm-8/bin/clang to provide /usr/bin/clang (
clang) in auto mode
(base) lineojcd@pop-os:~/Downloads$
```

-3 install git

If you have done this step, just skip it.

-4 Download Unreal

The current version of CARLA runs on Unreal Engine 4.24 only. The path is irrelevant, but for the sake of this tutorial, installation will be done under **~/UnrealEngine_4.24**. If the path chosen differs, remember to change it accordingly when running the commands on terminal.

> **Note**
>
> Alternatively, there is this guide to build UE on Linux. When consulting it, remember that CARLA will need the **4.24 release**, not the latest.

Clone the content for Unreal Engine 4.24 in a local computer.

```
git clone --depth=1 -b 4.24 https://github.com/EpicGames/UnrealEngine.git ~/UnrealEngine_4.24
```

git clone --depth=1 -b 4.24 https://github.com/EpicGames/UnrealEngine.git ~/UnrealEngine_4.24

Get into said folder. Remember, this is the path where UE4.24 has been cloned.

```
cd ~/UnrealEngine_4.24
```

Get a patch for Unreal Engine. The patch fixes some Vulkan visualization issues that may occur when changing the map. Download and install it with the following commands.

```
wget https://carla-releases.s3.eu-west-3.amazonaws.com/Linux/UE_Patch/430667-13636743-patch.txt ~/430667-13636743-patch.txt
```

```
patch --strip=4 < ~/430667-13636743-patch.txt
```

Make the build.

```
./Setup.sh && ./GenerateProjectFiles.sh && make
```

when it ask you： *Register Unreal Engine file types* I clicked Yes

when you see this means the building finish properly.

```
**********  SUCCESS ***************
Setup successful.
Attempting to set up UE4 pretty printers for gdb (existing UE4Printers.py, if an
y, will be overwritten)...
        updated UE4Printers.py
        no ~/.gdbinit file found - creating a new one.

Setting up Unreal Engine 4 project files...

Fixing inconsistent case in filenames.
Setting up Mono
Generating data for project indexing... 100%
Generating data for project indexing... 100%
Writing project files... 100%
Generating data for project indexing... 100%
Generating data for project indexing... 100%
Generating data for project indexing... 100%
Generating data for project indexing... 100%
Writing project files... 100%
bash "/home/lineojcd/UnrealEngine_4.24/Engine/Build/BatchFiles/Linux/Build.sh" C
rashReportClient Linux Shipping
Fixing inconsistent case in filenames.
Setting up Mono
Building CrashReportClient...
```

and the unreal is building now

```
[67/228] Link (ld) libUnrealFrontend-AudioPlatformConfiguration.so
[68/228] Link (ld) libUnrealFrontend-DesktopPlatform.so
[69/228] Compile Module.JsonUtilities.cpp
[70/228] Compile Module.JsonUtilities.gen.cpp
[71/228] Compile UnrealEdMessages.init.gen.cpp
[72/228] Compile UnrealEdMessagesModule.cpp
[73/228] Compile FileServerMessages.gen.cpp
[74/228] Link (ld) libUnrealFrontend-SlateCore.so
[75/228] Compile AssetEditorMessages.gen.cpp
[76/228] Compile Module.MessageLog.cpp
[77/228] Compile Module.RenderCore.cpp
[78/228] Link (ld) libUnrealFrontend-Sockets.so
```

Finish building

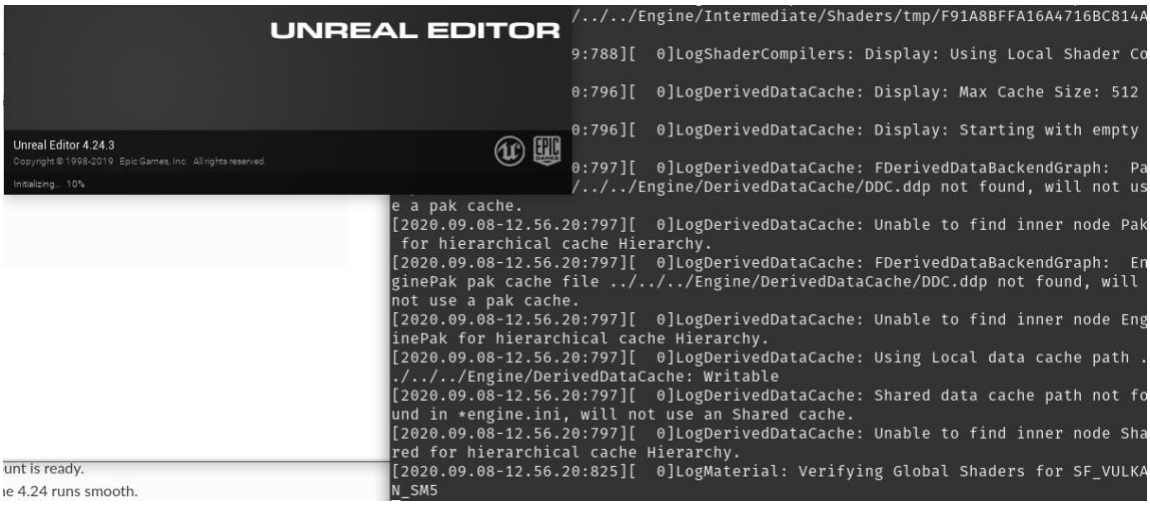
```
98/102] Link (ld) libUnrealInsights-SlateReflecto
99/102] Link (ld) libUnrealInsights-TraceInsights
100/102] Link (ld) libUnrealInsights-AppFramework
101/102] Link (lld) UnrealInsights
102/102] UnrealBuildTool.exe UnrealInsights.targe
otal time in Local executor: 147.90 seconds
otal execution time: 149.68 seconds
base) lineojcd@pop-os:~/UnrealEngine_4.24$
```

-4 Check Unreal

Unreal Engine should be installed in the system. Run `Engine/Binaries/Linux/UE4Editor.sh` to open the Editor and check it out.

```
cd ~/UnrealEngine_4.24/Engine/Binaries/Linux && ./UE4Editor
```

cd ~/UnrealEngine_4.24/Engine/Binaries/Linux
./UE4Editor



Start it with a specific project

https://github.com/EpicGames/UnrealEngine/blob/release/Engine/Build/BatchFiles/Linux/README.md

```
cd Engine/Binaries/Linux/
./UE4Editor "~/Documents/Unreal Projects/MyProject/MyProject.uproject"
```

You can also append -game if you want to run the project as a game (you can also do that from the running editor).

-5 Build Carla

> **Note**
>
> Downloading aria2 with `sudo apt-get install aria2` will speed up the following commands.

The official repository of the project. Either download and extract it or clone it using the following command line.

```
git clone https://github.com/carla-simulator/carla
```

Now the latest content for the project, known as `master` branch in the repository, has been copied in local.

> **Note**
>
> The `master` branch contains the latest fixes and features. Stable code is inside the `stable` and previous CARLA versions have their own branch. Always remember to check the current branch in git with the command `git branch`.

git clone https://github.com/carla-simulator/carla

My folder structure:

Get assets

Only the assets package is yet to be donwloaded. These are stored separately to make the repository a bit lighter. CARLA cannot be built without the assets. There is a script that downloads and extracts the latest content version. The package is >3GB, so downloading it may take some time.

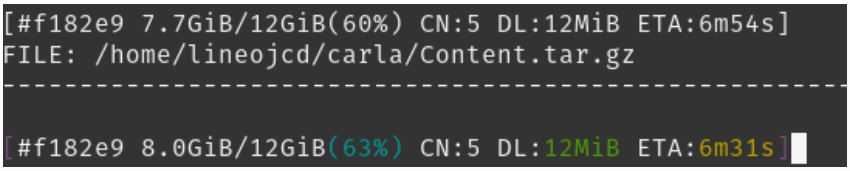
Get into the root carla folder. The path should correspond with the repository just cloned:

```
cd ~/carla
```

Run the script to get the assets.

```
./Update.sh
```

cd ~/carla
./Update.sh



Set the environment variable

This is necessary for CARLA to find the Unreal Engine 4.24 installation folder.

export UE4_ROOT=~/UnrealEngine_4.24

The variable should be added to `~/.bashrc` or `~/.profile` to be set persistently session-wide.
Otherwise, it will only be accessible from the current shell.

I use:

echo 'export UE4_ROOT=~/UnrealEngine_4.24' >> ~/.bashrc

```
export PATH=/usr/local/cuda-10.2/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-10.2/lib64${LD_LIBRARY_PATH:+:${LD_LIBRA
Y_PATH}}
export UE4_ROOT=~/UnrealEngine_4.24
(base) lineojcd@pop-os:~/carla$ cat ~/.bashrc
```

source ~/.bashrc

make CARLA

The last step is to finally build CARLA. There are different make commands to build the different modules. All of them run in the root CARLA folder.

Warning

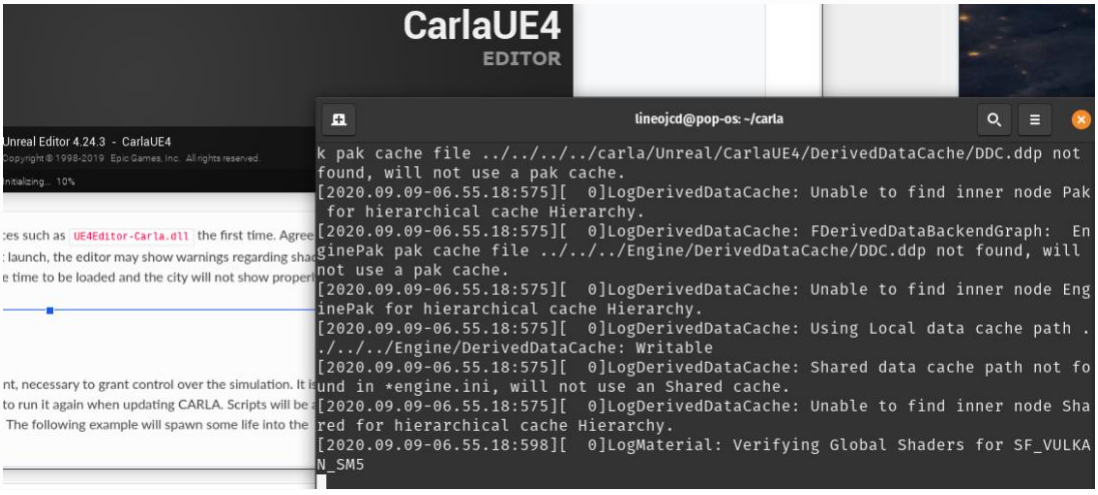
Make sure to run `make launch` to prepare the server and `make PythonAPI` for the client.
Alternatively `make LibCarla` will prepare the CARLA library to be imported anywhere.

- **make launch** compiles the server simulator and launches Unreal Engine. Press **Play** to start the spectator view and close the editor window to exit. Camera can be moved with `WASD` keys and rotated by clicking the scene while moving the mouse around.

```
make launch
```

The project may ask to build other instances such as `UE4Editor-Carla.dll` the first time. Agree in order to open the project. During the first launch, the editor may show warnings regarding shaders and mesh distance fields. These take some time to be loaded and the city will not show properly until then.

make launch



- **make PythonAPI** compiles the API client, necessary to grant control over the simulation. It is only needed the first time. Remember to run it again when updating CARLA. Scripts will be able to run after this command is executed. The following example will spawn some life into the town.

```
make PythonAPI && cd PythonAPI/examples && python3 spawn_npc.py
```

make PythonAPI

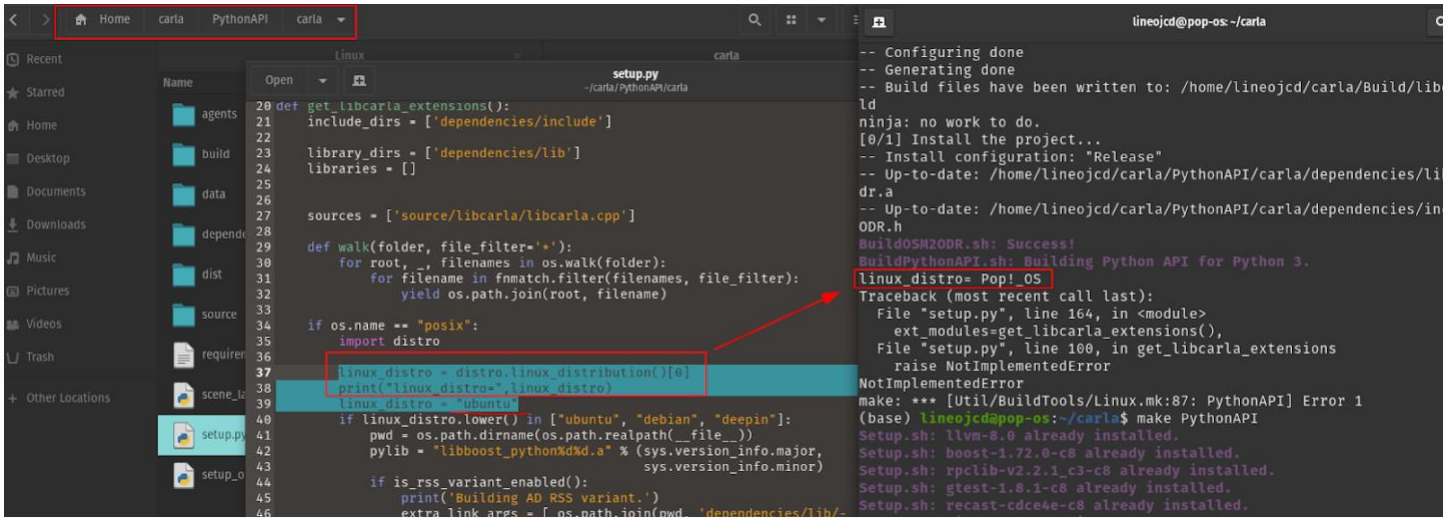
If you are using other linux system or a system based on ubuntu,it might cause errors:

```
BuildPythonAPI.sh: Building Python API for Python 3.
Traceback (most recent call last):
  File "setup.py", line 163, in <module>
    ext_modules=get_libcarla_extensions(),
  File "setup.py", line 99, in get_libcarla_extensions
    raise NotImplementedError
NotImplementedError
make: *** [Util/BuildTools/Linux.mk:87: PythonAPI] Error 1
(base) lineojcd@pop-os:~/carla$ make PythonAPI
```

How to solve?

Mintlinux 19.3, Carla 0.9.7, Unrealengine 4.22, 'Make PythonAPI' failed #2344
https://github.com/carla-simulator/carla/issues/2344

This is because: your system distro is not in ["ubuntu", "debian", "deepin"]. In my system, the distro is **Pop!_OS**, simply manually specify it.
linux_distro = "ubuntu"



Done!

```
carla.__pycache__.libcarla.cpython-38: module references __file__
creating dist
creating 'dist/carla-0.9.10-py3.8-linux-x86_64.egg' and adding 'build
linux-x86_64/egg' to it
removing 'build/bdist.linux-x86_64/egg' (and everything under it)
BuildPythonAPI.sh: Success!
(base) lineojcd@pop-os:~/carla$
```

cd PythonAPI/examples && python3 spawn_npc.py

Important

If the simulation is running at very low FPS rates, go to `Edit/Editor preferences/Performance` in the UE editor and disable **Use less CPU when in background**.

Now CARLA is ready to go. Here is a brief summary of the most useful `make` commands available.

Command	Description
`make help`	Prints all available commands.
`make launch`	Launches CARLA server in Editor window.
`make PythonAPI`	Builds the CARLA client.
`make package`	Builds CARLA and creates a packaged version for distribution.
`make clean`	Deletes all the binaries and temporals generated by the build system.
`make rebuild`	make clean and make launch both in one command.

how to run Carla with a Python client

CARLA Tutorial 00 - Getting Started

https://www.youtube.com/watch?v=AaJekfFR1KQ