Building Carla from Source and adding a tractor-semi trailer combination to Carla

When Carla is downloaded, we generally obtain a pre-compiled (compiled by Developers) version of Carla. However, if we want to add something new to the Carla, for example, a map, vehicle etc., there is a need to recompile the Carla with this new entity. In this case, the Carla needs to be built from the source. Once the Carla is built, the compiled version can be distributed. This compiled version of Carla will have the new entity.

This report shows the process of adding a tractor-semitrailer combination to Carla which is not present in the distributed pre-compiled version of Carla. Before starting with the process of adding the new vehicle to Carla, make sure you have downloaded the pre-compiled version of Carla from https://github.com/carla-simulator/carla/releases/tag/0.9.14/. Ubuntu 23.04 – Lunar (Other versions are also okay!) will be used throughout the process. Many people, as they say in forums, have faced issues with Windows to compile Carla. Hence, if possible, avoid Windows. It is important to note that Carla's developers work with versions until Ubuntu 20.04 and Python 3.8. We will use Python 3.7, but it will be shown how to make it work for higher versions of Ubuntu.

It is common practice to create a virtual environment and install packages in the virtual environment. This ensures that the installed packages remain organized and can be altered without affecting system libraries. However, while compiling, DO NOT use a virtual environment anywhere. The first part of the compilation works fine but the second part which involves Unreal will throw an error if you use a virtual environment. The same problem happens whenever 'sudo' is used for installing packages in some steps during the compilation. So, DO NOT use 'sudo' unless mentioned. Although Unreal gives access to source files, it does not give administrator access which is why errors will pop out later if 'sudo' is used casually. Nevertheless, in some situations, 'sudo' will be used. That will be explicitly mentioned in this document when to use it.

Now, we start the process of adding a tractor-semitrailer to Carla. The steps are followed from: https://carla.readthedocs.io/en/0.9.14/build_linux/ with some modifications. Only the modifications are shown here, follow the actual documentation for other parts.

Installing Prerequisites:

1. First, we will go through the software requirements. (THIS STEP IS THE MOST IMPORTANT STEP. IF YOU DO IT INCORRECTLY, IT WILL NOT THROW ERROR AT THIS POINT. BUT IT WILL THROW A LOT OF ERRORS DURING THE LAST STEPS OF COMPILATION. SO, MAKE SURE TO DO IT CORRECTLY)

   Commands:

```
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-get update
```

```
sudo apt-add-repository "deb http://apt.llvm.org/focal/ llvm-toolchain-lunar main"
sudo apt-get install build-essential clang-13 lld-13 g++-9 cmake ninja-build libvulkan1
python-is-python3 python-dev-is-python3 python3-dev python3-pip libpng-dev libtiff5-dev
libjpeg-dev tzdata sed curl unzip autoconf libtool rsync libxml2-dev git
sudo update-alternatives --install /usr/bin/clang++ clang++ /usr/lib/llvm-13/bin/clang++
180 &&
sudo update-alternatives --install /usr/bin/clang clang /usr/lib/llvm-13/bin/clang 180
```

Understand what these commands are doing:
These commands are installing necessary dependencies which are compatible with both Carla and Unreal Engine.

Disclaimer for other Ubuntu versions:
Be careful while copying these commands. Some of the interpreters like clang, lld, g++ have different versions than what is specified on Carla's website. That is because they use a different Ubuntu version. For Ubuntu 23.04 (lunar), we used clang-13, lld-13, g++ -9 . You can find the corresponding packages from https://packages.ubuntu.com/search?keywords=ubuntu-software that suits your Ubuntu version. You should also change the repository address if you are using different ubuntu version. For example, if you are using focal, you should use this "deb http://apt.llvm.org/focal/ llvm-toolchain-focal main".

2. This is a step that is not mentioned on the website, but you must do for successful compilation. Install python3.7 and make sure it is in the following path (/usr/local/bin/python3.7). Python3.7 is what developers work with. So, you need to have it. For installation, I downloaded the python3.7 package and configured it manually. You will never use python3.7 but the source code will use python3.7 for compilation. Again, a virtual environment is not a good idea. When I installed python3.7 using a virtual environment, the compiler could not find it and always used python3.11 (the default version on my PC) for compiling which then threw an error.

3. Install Python dependencies as given in the Carla website.

Now, you are done with installing all the prerequisites. If you have done all the things correctly, you won't have any issues later. If you have not done it properly, you will not be able to compile later. So, even though you did not get any error, just cross-check if you have done things correctly.

Installing Unreal Engine:

I will not go through it. The process is straightforward. Just follow the instructions on the Carla website. You have to have an unreal engine's account and you should get the developer access. It may take some time but is not a difficult process.

Build Carla:

1. We already have a pre-compiled version of Carla. No need to clone it again.
2. Download the latest assets using the following command in Carla root. The root location is generally at ~/Carla/CARLA_0.9.14/carla.

   ```
   ./Update.sh
   ```
3. Set the unreal engine variable for all sessions, i.e. add `export UE4_ROOT=~/UnrealEngine_4.26` to the bash as mentioned on the website.
4. The previous step will ensure that unreal is included in the compilation path. However, for some reason, it does not work. Hence, we need to add this manually in two files. The first file is 'BuildCarlaUE4.sh' which is located at ~/Carla/CARLA_0.9.14/carla/util/BuildTools. Open the file and add `UE4_ROOT=~/UnrealEngine_4.26` in the first line. Do the same thing with another file named 'Package.sh' which is in the same location. The first file is needed to compile Python API. The second file is needed to package when the vehicle is successfully added.
5. Now follow the steps on the Carla website and compile the Python API client with Python3.7. Subsequently compile the server using 'make launch', which will take some time to launch Unreal. There will be some warnings, but you can ignore them.

If you have performed all the steps carefully, you should be able to launch Unity. For the first time, it takes some time to load the shaders. If Unreal is not being launched, then you might have got some error when you did 'make launch'. These errors are generally related to the prerequisites you installed in the first step. There were some errors when I tried to launch (I don't remember the errors though), but they are easy to fix.

Adding a vehicle to Carla:

Until now, we have not added the tractor-semi trailer combination. The next steps are focused on how to add the vehicle to unreal. If you want to add any other vehicle, just follow the steps here: https://carla.readthedocs.io/en/0.9.14/tuto_A_add_vehicle/ . Go through these videos if you want to create your vehicle in Carla:

1. https://www.youtube.com/watch?v=JwJplj92QoU&t=3414s
2. https://www.youtube.com/watch?v=mHgCuJc_Zh0
3. https://www.youtube.com/watch?v=c5IkFuJNMXE&t=12s
4. https://www.youtube.com/watch?v=mJufrK7RkeI&t=599s

Now the idea is to add a 3D model of the tractor-semitrailer to the Unreal engine where we have successfully launched the Carla. Once we do that, create a new compiled version of the Carla which will have our vehicle.

Blender is generally used to create 3D models of vehicles. Here, you can change the dimensions of the vehicle. But once the '.uasset' files are imported from Blender, you cannot change any dimension of the vehicle. You can scale the components in the vehicle though. I will not go through how to create models in a blender since it is already explained in the videos above.

The tractor-semitrailer 3d models are created by https://github.com/frankeng/CarlaSemiTruckTrailer. Follow the instructions on his page to add the files in the Blueprint library. However, do not use these files, as they have issues with connecting the tractor with the semi-trailer and reversing as well. Replace these files with the files given to you. All these issues have been fixed. Just to summarize what you must do:

1. Get the right files for the Blueprint folder and Static folder and paste them in the designated location.
   a. Paste the DAFTruck and Trailer folders (BluePrint) in this location Unreal/CarlaUE4/Content/Carla/Blueprints/Vehicles
   b. Paste the DAFTruck and Trailer folders (Static) in this location Unreal/CarlaUE4/Content/Carla/Static/Vehicles/4Wheeled

   Create the static folder in the mentioned location if it is not available.

   PS: Go to the content browser in Unreal and open the trailer which is in BluePrint folder. Cross-check that there are 6 functions. Out of these 6, two functions named 'couple tractor and trailer' and 'SetReverseGear' should be there. If they are not, you do not have the right version.

2. Get the manual_controlSemiTrailer.py and paste it into Python API examples. You can run it and it will launch the vehicle in pygame. When you run this file, you will use your default Python version. Make sure all the necessary packages are installed with this Python version. Here you can use a virtual environment as it is not a part of the compilation.

Creating new binaries:

Now everything is successfully added. You can create a new compiled version of the Carla which includes this vehicle. You can continue without creating a compiled version. But the problem is that you cannot have this vehicle in Carla simulator. Because the Carla simulator is not yet built from source. Remember this simulator is the developer's compiled version that you downloaded. However, you can run it in Unreal. You must do 'make launch' in Carla root whenever you want to launch the vehicle Unreal. Also, this cannot be distributed.

If you don't want this hassle, there are two ways to create a compiled version:

1. Go to File/Package Project/Build Target/CarlaUE4.
2. 'make project' in Carla root will also package the project. One of the errors in this process involves setting up the environment variable for Unreal in 'Package.sh'. But we had already done this when we were building Carla. The other errors that I don't remember are quite easy to fix.

Irrespective of what is used to create the compiled version, it takes a considerable amount of time (for me it took 15 hrs) to make the package. When you go for the second option, the project is packaged in a zip folder within 'Dist' in the Carla root.