Carla Lab

Team: Virtual Fast and Keyboard Furious

Carla has adapted to a server-client interface system in which the simulation is all handled server-side and the client is used to give commands to the server that spawns a car or sensor and follows these actions and stuff like that.

Server-side needs more computation than the client-side and Ideally, you can run the Server side on a great system and run the client from any moderate system with a network connection.

One problem to this is that there will be a delay depending on the ping of your network which may or may not be appropriate for the project you might be working on.

Carla Installation:

1. There are 2 ways to get Carla working on your system.
    a. Get the prebuilt version: That has unity and source code complied and put in a package.
    b. Build it from Source: Linux Build (There are options for Windows Build as well)

2. Building from Source
    a. Requirements:
        i. Unity
        ii. GPU system
        iii. Minimum 30 Gb of disk space

    b. Unity Installation
        i. To get source code to install unity
            1. Register on Epic
            2. Go to Accounts → Connections → Connect GitHub to Epic account
            3. That will give access to the private repository of Unity.
        ii. Clone the Repository and follow the instructions.
        iii. Set “export UE4_ROOT=~/UnrealEngine_4.24”

    c. Installing Server Side
        i. Git clone the repo.
        ii. Follow steps from Documentation.
        iii. This will build and install Carla and also compile the environment in Unity and takes a while to finish.

d. Installing Client-Side for Carla 0.9.9(Carla Client is referred to as Python API):
    i. Before proceeding to make PythonAPI, make sure you have: Python 2.7 and Python 3.5
    ii. Run the command “make PythonAPI”
    iii. Carefully observe what is the output and make sure it is successful for both python 2.7 and 2.5
    iv. Even if you are just gonna use either 3.5 or 2.7 installation needs to be done with both the requirements met or else the client side will not build correctly.
    v. Once it has all successfully built and installed export python path to the egg file created by make python API command in order for the client-side to work.
    vi. Run examples files to make sure everything is working.

e. Installing Ros Bridge:
    i. Follow steps on ROS Bridge Page.
    ii. There are two options:
        1. User
        2. Source (Developer)
    iii. Choose the appropriate option according to your project and install it.

3. Pitfalls:
    a. Python API not building egg files:
        i. Check Python version and go through the output on the command line and see where it failed
        ii. Even if it fails for one python it does not give an error that installation was aborted rather after that every time you try running the command again it will say installation is complete which is not the case.
        iii. Go through the output find the error and then satisfy that and do “make clean”
        iv. This will remove files from the previous make and then try again.

    b. Import Carla error:
        i. Check the python version you are using and check which python path is given.
        ii. Make sure the same version python path is provided.

    c. Ros bridge carla_msgs not found:
        i. Go to the git repository
        ii. Find the package carla_msgs
        iii. Go to ros-Carla-msgs repository
        iv. Git clone and change name to carla_msgs

Carla Lane Finding

TODO