

Carla Installation

AKIS

7. Februar 2023

Diese Anleitung beschreibt die Installation des CARLA-Simulators auf Linux- (Abschnitt 2) und Windows-Systemen (Abschnitt 3). CARLA ist in einer Server-Client-Architektur aufgebaut, wobei der Server die Simulation ausführt und die Szene rendert, während der Client für die Interaktion zwischen dem autonomen Agenten und dem Server über Sockets verantwortlich ist. Zur Installation von Carla müssen sowohl der Client als auch der Server installiert werden.

1 Anforderungen

Die folgenden Voraussetzungen sollten vor der Installation von CARLA erfüllt sein:

- **Systemanforderungen.** CARLA ist für Windows- und Linux-Systeme verfügbar.
- **Eine angemessene GPU.** CARLA zielt auf realistische Simulationen ab, daher benötigt der Server mindestens einen 6-GB-Grafikprozessor, obwohl 8 GB empfohlen werden, insbesondere wenn es um maschinelles Lernen geht.
- **Festplattenplatz.** CARLA wird etwa 20 GB Speicherplatz benötigen.
- **Python.** Python ist die wichtigste Skriptsprache in CARLA. Unter Linux unterstützt CARLA Python 2.7 und Python 3.7. Unter Windows wird nur Python 3.7 unterstützt.
- **Pip.** Einige Installationsmethoden der CARLA-Client-Bibliothek erfordern pip oder pip3 (je nach Python-Version) Version 20.3 oder höher.

2 Linux Installation

Das Debian-Paket ist sowohl für Ubuntu 18.04 als auch für Ubuntu 20.04 verfügbar.

2.1 Install the Carla Server on Linux

- Richten Sie das Debian-Repository im System ein (jeder der beiden folgenden Befehle muss in einer einzigen Zeile ausgeführt werden):

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1AF1527DE64CB8D9
```

```
$ sudo add-apt-repository "deb [arch=amd64] http://dist.carla.org/carla(lsb_release -sc) main"
```

- Aktualisieren Sie die Paketlisten und installieren Sie den Carla-Server:

```
$ sudo apt-get update
$ sudo apt-get install carla-simulator
```

- Importieren Sie zusätzliche Assets:

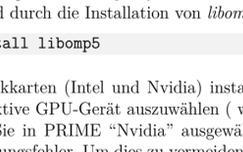
```
$ cd /opt/carla-simulator
$ ./ImportAssets.sh
```

- Testen Sie Ihre Serverinstallation:

Sie können den Carla-Server mit dem folgenden Befehl starten:

```
$ ./CarlaUE4.sh
```

Es öffnet sich ein Fenster mit einem Blick über eine Stadt. Dies ist die Zuschauerperspektive. Um durch die Stadt zu fliegen, benutzen Sie die Maus und die Tasten **[W]** **[A]** **[D]** **[S]**, wobei Sie die rechte Maustaste gedrückt halten, um die Richtung zu steuern.



Dies ist der Serversimulator, der jetzt läuft und darauf wartet, dass sich ein Client verbindet und mit der Welt interagiert.

Troubleshooting

Beim Betreiben des Servers können einige Probleme auftreten. Diese Probleme werden im Folgenden erläutert:

- Sie erhalten die Fehlermeldung: "error while loading shared libraries: libomp.so.5: cannot open shared object file: No such file or directory". Dieser Fehler wird durch die Installation von libomp5 behoben:

```
$ sudo apt install libomp5
```

- Wenn Sie 2 Grafikkarten (Intel und Nvidia) installiert haben, neigt Carla dazu, das erste aktive GPU-Gerät auszuwählen (welches oft die Intel GPU ist), auch wenn Sie in PRIME "Nvidia" ausgewählt haben. Dies führt zu einem Segmentierungsfehler. Um dies zu vermeiden, müssen Sie Carla sagen, dass es die Nvidia-GPU bevorzugen soll:

```
$ ./CarlaUE4.sh -preferNvidia
```

- Wenn Sie Carla auf einem Remote-Rechner installiert haben, schlägt die Übertragung der Carla-Server-GUI fehl. Um dieses Problem zu lösen, müssen Sie Carla im Off-Screen-Modus starten. In diesem Modus arbeitet die Unreal Engine wie gewohnt, das Rendering wird berechnet, aber es ist keine Anzeige verfügbar. GPU-basierte Sensoren liefern wie gewohnt Daten.

```
$ ./CarlaUE4.sh -RenderOffScreen
```

2.2 Carla-Client unter Linux installieren

Die Carla-Client-API ist in Python implementiert.

2.2.1 Erstellen einer Conda-Umgebung (optional)

Es wird empfohlen, die CARLA-Client-Bibliothek in einer virtuellen conda-Umgebung zu installieren, um Paketkonflikte mit anderen Projekten zu vermeiden. Beachten Sie zum Beispiel, dass Carla Python 3.7 (oder Python 2.7) benötigt. Wenn Sie mit der Python-Distribution Anaconda nicht vertraut sind, finden Sie eine Anleitung zur Installation unter Linux im Anhang 4.2.

Angenommen, Sie haben Anaconda installiert, erstellen Sie eine virtuelle Umgebung (z.B. mit dem Namen carla) mit Python 3.7 und aktivieren Sie es:

```
$ conda create -n carla python=3.7
$ conda activate carla
```

2.2.2 Überprüfen Sie Ihre Pip-Version

- Pip3 benötigt Version 20.3 oder höher. Überprüfen Sie Ihre pip3 Version:

```
$ pip3 -V
```

Falls erforderlich, aktualisieren Sie pip3:

```
$ pip3 install --upgrade pip
```

- Installieren Sie die Python-Abhängigkeiten:

```
$ pip3 install pygame opencv-python
$ conda install numpy matplotlib
```

2.2.3 Installieren Sie den Client

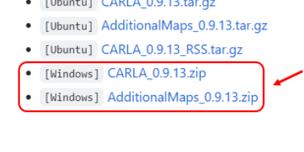
Sie sind nun bereit, den carla Client in der erstellten virtuellen Umgebung zu installieren.

CARLA stellt .whl-Dateien für verschiedene Python-Versionen zur Verfügung. Diese finden Sie unter PythonAPI/carla/dist/. Es gibt eine Datei pro unterstützter Python-Version, die durch den Dateinamen gekennzeichnet ist. Installieren Sie die Datei, die Ihrer Carla-Server- und Python-Version entspricht (0.9.13 & 3.7 zum Zeitpunkt der Erstellung dieses Dokuments):

```
$ cd /opt/carla-simulator/PythonAPI/carla/dist
$ pip3 install carla-0.9.13-cp37-cp37m-manylinux_2_27_x86_64.whl
```

2.3 Testen Sie Ihre komplette Linux-Installation

Sie können Ihre komplette Installation (Client & Server) testen, indem Sie das Skript `demo_run_carla.py` ausführen, das mit dieser Installationsanleitung geliefert wird. Es öffnet sich ein OpenCV-Fenster, in dem ein Auto herumfährt. Beachten Sie, dass Sie dieses Skript in Ihrer Carla-Conda-Umgebung ausführen müssen und der Server gestartet sein muss (über ./CarlaUE4.sh).



3 Windows Installation

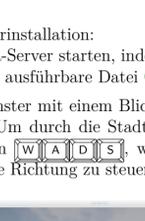
3.1 Carla Server für Windows herunterladen

- Laden Sie die neueste Version (0.9.13 zum Zeitpunkt der Erstellung dieses Dokuments) des carla-Paketes sowie die AdditionalMaps von [hier](#) herunter, wie in der Abbildung unten gezeigt:



- Entpacken Sie das Carla-Paket (CARLA_0.9.13) in einen Ordner Ihrer Wahl.

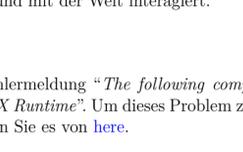
- Extrahieren Sie den Inhalt von AdditionalMaps_0.9.13 direkt in den carla-Root-Ordner, d. h. in den Ordner, in dem sich andere carla-Module wie CarlaUE4, Engine usw. befinden. Die sich daraus ergebende Struktur ist wie in der Abbildung unten aus:



- Testen Sie Ihre Serverinstallation:

Sie können den Carla-Server starten, indem Sie auf die in der obigen Abbildung rot eingekreiste ausführbare Datei `CarlaUE4.exe` doppelklicken.

Es öffnet sich ein Fenster mit einem Blick über eine Stadt. Dies ist die Zuschauerperspektive. Um durch die Stadt herumzufahren, benutzen Sie die Maus und die Tasten **[W]** **[A]** **[D]** **[S]**, wobei Sie die rechte Maustaste gedrückt halten, um die Richtung zu steuern.



Dies ist der Serversimulator, der jetzt läuft und darauf wartet, dass sich ein Client verbindet und mit der Welt interagiert.

Troubleshooting

Sie bekommen eine Fehlermeldung "The following component(s) are required to run this program DirectX Runtime". Um dieses Problem zu lösen, laden Sie directX herunter und installieren Sie es von [here](#).

3.2 Installieren Sie den Carla-Client für Windows

- Create conda virtual environment (Optional but recommended):

Erstellen Sie eine virtuelle Umgebung von conda (optional, aber empfohlen): Es wird empfohlen, die CARLA-Client-Bibliothek in einer virtuellen conda-Umgebung zu installieren, um Paketkonflikte mit Ihren anderen Projekten zu vermeiden. Beachten Sie zum Beispiel, dass Carla Python 3.7 (oder Python 2.7) benötigt.

Wenn Sie mit der Python-Distribution Anaconda nicht vertraut sind, finden Sie eine Anleitung zur Installation unter Windows im Anhang 4.1.

Angenommen, Sie haben Anaconda installiert, erstellen Sie eine virtuelle Umgebung (z. B. mit dem Namen carla) mit Python 3.7 und aktivieren Sie es:

```
$ conda create -n carla python=3.7
$ conda activate carla
```

- Installieren Sie die Python-Abhängigkeiten:

```
$ pip3 install pygame opencv-python
$ conda install numpy matplotlib
```

- Installieren Sie den Client über die whl-Datei: CARLA stellt .whl-Dateien für verschiedene Python-Versionen zur Verfügung. Diese finden Sie unter PythonAPI/carla/dist/. Es gibt eine Datei pro unterstützter Python-Version, die durch den Dateinamen gekennzeichnet ist. Installieren Sie in der conda-Eingabeaufforderung zu dem Verzeichnis, in das Sie Carla extrahiert haben, dann zu PythonAPI/carla/dist/ und führen Sie den folgenden Befehl aus, um den Client zu installieren, der Ihrer Carla-Server- und Python-Version entspricht (0.9.13 & 3.7 zum Zeitpunkt des Schreibens):

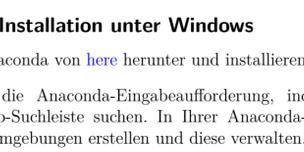
```
$ pip3 install carla-0.9.13-cp37-cp37m-win_amd64.whl
```

Das Verfahren ist in der folgenden Abbildung dargestellt:



3.3 Testen Sie Ihre komplette Windows-Installation

Sie können Ihre komplette Installation (Client & Server) testen, indem Sie das Skript `demo_run_carla.py` ausführen, das mit dieser Installationsanleitung geliefert wird. Es öffnet sich ein OpenCV-Fenster, in dem ein Auto herumfährt. Beachten Sie, dass Sie dieses Skript in Ihrer Carla-Conda-Umgebung ausführen müssen und der Server gestartet sein muss (über CarlaUE4.exe).

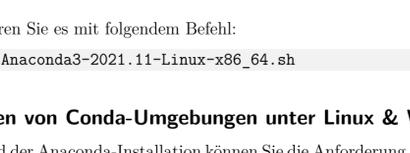


4 Appendix

4.1 Anaconda-Installation unter Windows

- Laden Sie anaconda von [here](#) herunter und installieren Sie es.

- Starten Sie die Anaconda-Eingabeaufforderung, indem Sie sie in der Windows-App-Suchleiste suchen. In Ihrer Anaconda-Eingabeaufforderung können Sie Umgebungen erstellen und diese verwalten.



4.2 Anaconda-Installation unter Linux

- Laden Sie die anaconda .deb-Datei von [hier](#).

- Installieren Sie es mit folgendem Befehl:

```
$ bash Anaconda3-2021.11-Linux-x86_64.sh
```

4.3 Erstellen von Conda-Umgebungen unter Linux & Windows

- Während der Anaconda-Installation können Sie die Anforderung akzeptieren, dass die Basisumgebung automatisch aktiviert werden soll, oder Sie können es manuell aktivieren über:

```
$ conda activate
```

- Um eine Umgebung mit einer bestimmten Python-Version (z. B. 3.7) zu installieren, gehen Sie wie folgt vor:

```
$ conda create -n myenv_name python=3.7
```

Für weitere Informationen über die conda-Umgebung siehe [here](#).