Importing Packages

First, download and import **Photon2**. Pass your **AppID** to Photon setup (explained below), and then import the Photon integration package for Realistic Car Controller. Import the integration package from the welcome window too. **Tools → BCG → RCC → Welcome Window → Addons**.

After importing Photon2 and integration packages to the project, you can test the Photon demo scenes. Two new photon demo scenes can be found in the demo scenes folder. Or you can open them via RCC's welcome window (**Tools → BCG → RCC → Welcome Window**).

There should be a new scene named "**RCC City Photon2**" in the **Demo Scenes** folder after importing the integration package. Same scene with regular city scene. The only difference is this scene has **RCC_PhotonDemo**. Connects to the photon server, enters lobby, creates or joins room, and spawns the vehicle.

How Photon Works

Well, it's quite easy. If you don't know anything about Photon, or even multiplayer based games, here are the events for how it works;

1 – First, you must connect to the server. In this case, our server is **Photon Cloud Server**.

2 – If your connection to server succeeded, you will be in lobby with callbacks. Now you can create your room, or join someone's room, or join to random room here.

3 – Congrats, now you are in an online multiplayer-based room in real-time.

4 - Each gameobject with the **PhotonView** component will be synchronized over the network. **RCC_PhotonSync** is syncing all vehicle inputs with **PhotonView** over the network.

Installing Photon

As soon as you have imported it to your project, it will ask for your **AppID**. You must register and create a new app id for your project. Pass it. And now you are ready to develop your real-time based multiplayer levels. It's free. And of course you are limited to 20 CCU.

Photon and RCC

Photon PUN2 has many simple methods in their API. It's extremely easy to understand. Let me explain how the demo scene works.

The city demo scene has a **RCC_PhotonDemo**. This script handles the multiplayer section of the scene. Uses these methods and callbacks (You can find all methods from Photon2's docs).

```
Photon.Pun.PhotonNetwork.ConnectUsingSettings ();
```

We are connecting to the server first. We can listen to which connection status we are on in **OnGUI()** method. Like this.

```
GUILayout.Label("State: " + Photon.Pun.PhotonNetwork.NetworkClientState.ToString());
```

Once the connection to the server is established, we must join the lobby.

```
Photon.Pun.PhotonNetwork.JoinLobby ();
```

As soon as we are connected to the lobby, we want to join a random room by.

```
Photon.Pun.PhotonNetwork.JoinRandomRoom();
```

If it fails, this means there are no any active other room. We are creating the new room by **Photon.Pun.PhotonNetwork.CreateRoom(null);** This method needs a room name. I didn't use it, because there are no any room lists in the demo.

I take a string that belongs to the player here. And set it by **Photon.Pun.PhotonNetwork.NickName = name;** Enabling / disabling few UI gameobjects depends on connection state. That's basically, how the demo scene works.

For vehicle sync, each vehicle has **PhotonView** and **RCC_PhotonSync**. These scripts are necessary for each network vehicle. **RCC_PhotonSync** is observed by **PhotonView**.

RCC_PhotonSync is synchronizing all control inputs, position, rotation, and rigid velocity smoothly. If the vehicle is our vehicle, it will broadcast our data to the server. If the vehicle is not our vehicle, it will receive all data from the server.

These vehicles are not instantiated or destroyed with regular ~~**GameObject.Instantiate**~~ or Destroy. You must do it with **PhotonNetwork.Instantiate** or **Destroy**. Unfortunately, it won't work with your prefab. It accepts only strings for your vehicle. That means, it will use **Resources** folder for accessing your vehicles. Your vehicle prefabs must be at **Resources** folder.

Photon Lobby

RCC_PhotonManager script handles multiplayer section of this scene. Connects to the photon server, enters lobby, lists current rooms, creates or joins rooms. You can find many commented lines in this script. **RCC_PhotonManager** and **RCC_PhotonDemo** scripts are not the same. Manager script handles lobby scene, and demo scene handles actual city demo scene where you can spawn your vehicles.