

# Realistic Car Controller Pro

Thank you for purchasing and using Realistic Car Controller Pro. This documentation will guide you on definition of the enter / exit system.

## Content

Install & Import BCG Shared Assets to The Project.....	2
How The Integration Works.....	2
How The System Works (Simple) .....	2
How To Make It Work.....	2
How The System Works (Detailed) .....	3
Common Mistakes.....	4

## Install & Import BCG Shared Assets to The Project

First, you'll need to import the latest **BCG Shared Assets** to your project. You can import it from the **Tools → BCG → RCC Pro → Welcome Window → Addons**. There will be two new scenes added to your scenes folder. **Enter Exit FPS** and **Enter Exit TPS** scenes.

## How The Integration Works

FPS and TPS controllers are using different cameras. Usually, camera of the FPS character is placed in the gameobject. But TPS characters have unattached and independent cameras, and they are not parented to the character. Therefore, **BCG\_EnterExitPlayer.cs** script has two options for use with FPS or TPS. If TPS is chosen, you must select the TPS camera in the scene. You can edit some settings from **BCG Shared Asset Settings** (**Tools → BCG → Shared Assets → Edit Settings**).

## How The System Works (Simple)

All enter exit RCCP vehicles must have "**BCG\_EnterExitVehicle**" component. And our FPS/TPS characters must have "**BCG\_EnterExitPlayer**" component. **BCG\_EnterExitManager** (Will be created automatically on the scene) will be observing all vehicles and characters in the scene. If vehicle has no driver, \*canControl of the vehicle will be disabled, and not registered as player vehicle. If our character has entered any vehicle, **BCG\_EnterExitManager** will register the target vehicle as player vehicle, enables the canControl of the vehicle. At that moment, our character will be parented to the vehicle and disabled entirely. Now, **BCG\_EnterExitVehicle** has a \*\*driver now. If player gets out from the vehicle, **BCG\_EnterExitManager** will be activating character controller, and transport it to "**Get Out Pos**" of the vehicle. And lastly, deregister the vehicle and disable \*canControl of the vehicle. Now, vehicle is not controllable, and it's no longer a player vehicle.

\* = canControl bool of the **RCCP\_CarController**. If it's enabled, the vehicle is controllable.

\*\* = driver variable of the **BCG\_EnterExitVehicle**.

## How To Make It Work

All you have to do is, add necessary scripts to the vehicles and to your character player. Get out position of the vehicles are created automatically in script if you don't select it in **BCG\_EnterExitVehicle**. You can replace get out positions.

For FPS characters, it will depend on your camera of the FPS player. For TPS player, it depends on your actual TPS Player, not camera.

- Character Controllers must have **BCG\_EnterExitPlayer**
- Vehicles must have **BCG\_EnterExitVehicle**
- **\_BCGEnterExitManager** will be created automatically in your scene.  
**\_BCGEnterExitManager** will be observing and managing all the process.

## How The System Works (Detailed)

Please read simple version first if you haven't read yet. Let's get it straight with more detailed version.

- 1- **BCG\_EnterExitVehicle** – must be attached to all vehicles.
- 2- **BCG\_EnterExitPlayer** – must be attached to the character.
- 3- **BCG\_EnterExitManager** – will be created automatically.

All the system is based on events. **BCG\_EnterExitPlayer** is firing an event when it's spawned, destroyed, gets in a vehicle, or gets out of the vehicle. **BCG\_EnterExitManager** will be listening to this event as well. So, manager knows your all your character events. All events are explained below;

1. public delegate void onBCGPlayerSpawned(BCG\_EnterExitPlayer player);
2. public delegate void onBCGPlayerDestroyed(BCG\_EnterExitPlayer player);
3. public delegate void onBCGPlayerEnteredAVehicle(BCG\_EnterExitPlayer player,  
BCG\_EnterExitVehicle vehicle);
4. public delegate void onBCGPlayerExitedFromAVehicle(BCG\_EnterExitPlayer player,  
BCG\_EnterExitVehicle vehicle);

You can listen any events on the **BCG\_EnterExitPlayer** and **BCG\_EnterExitVehicle**.

*For example:*

**BCG\_EnterExitManager** will be using **GetIn()** or **GetOut()** methods by listening these events. **GetIn()** method requires “**targetVehicle**”. **GetOut()** method doesn't require any parameters.

When our character standing next to the any vehicle, get in text will be enabled by **BCG\_EnterExitPlayer**. If player pushes the interaction button, **OnBCGPlayerEnteredAVehicle** event will be fired with **targetVehicle** data. As you well now, this event has been listening by **BCG\_EnterExitManager**. So, **BCG\_EnterExitManager** will be firing **GetIn(targetVehicle)** method to do the rest...

## Common Mistakes

- My character can't get in the vehicle, get in text doesn't appear.
- Be sure your character controller and vehicles have proper scripts.
- My character controller gets out from the vehicle at wrong location.
- Change the position of the “Get Out Pos”.
- I'm having multiple cameras and audiolisteners in the scene.
- Be sure your character controller and vehicles have proper scripts.
- I want to get in the vehicle without starting the engine.
- Disable the auto start engine option in the **BCG Shared Asset Settings** (**Tools** → **BCG** → **Shared Assets** → **Edit Settings**).