Realistic Car Controller V3.53

First of all, thank you for purchasing and using Realistic Car Controller!

Contents

You can find more updated details on

http://www.bonecrackergames.com/realistic-car-controller

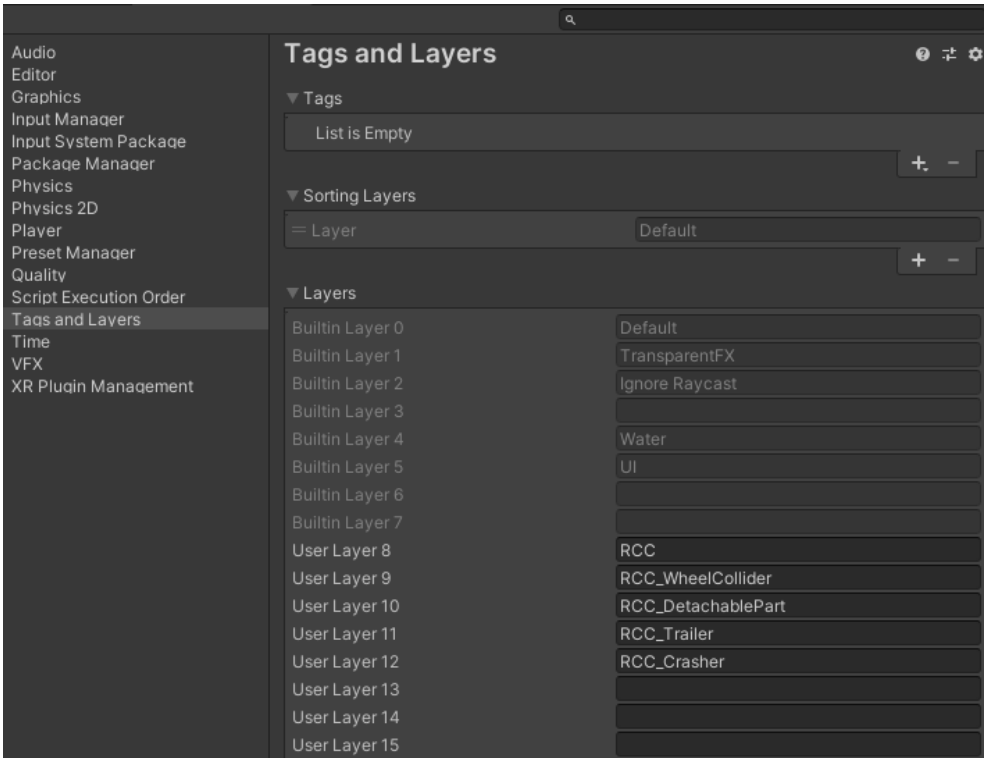
https://www.youtube.com/playlist?list=PLRXTqAVrLDpoW58lKf8XA1AWD6kDkoKb1

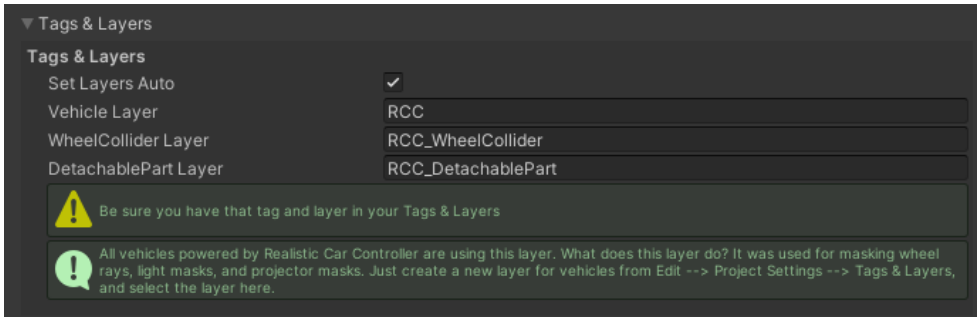
(You can zoom in with CTRL + ScrollUp for enlarge PDF pages)

First to Do!

Always backup your project before updating any asset or Unity Editor. Keep your own assets outside of the RealisticCarControllerV3 folder. Delete the entire folder, and import updated version.

RCC is using **LayerMask** for avoid unwanted raycast hits. Necessary layers are created automatically, but feel free to check them after the import. These layers must be selected in **RCC Settings**. Also you can import it from **Welcome Screen**, but it will overwrite your **Tags & Layers**.

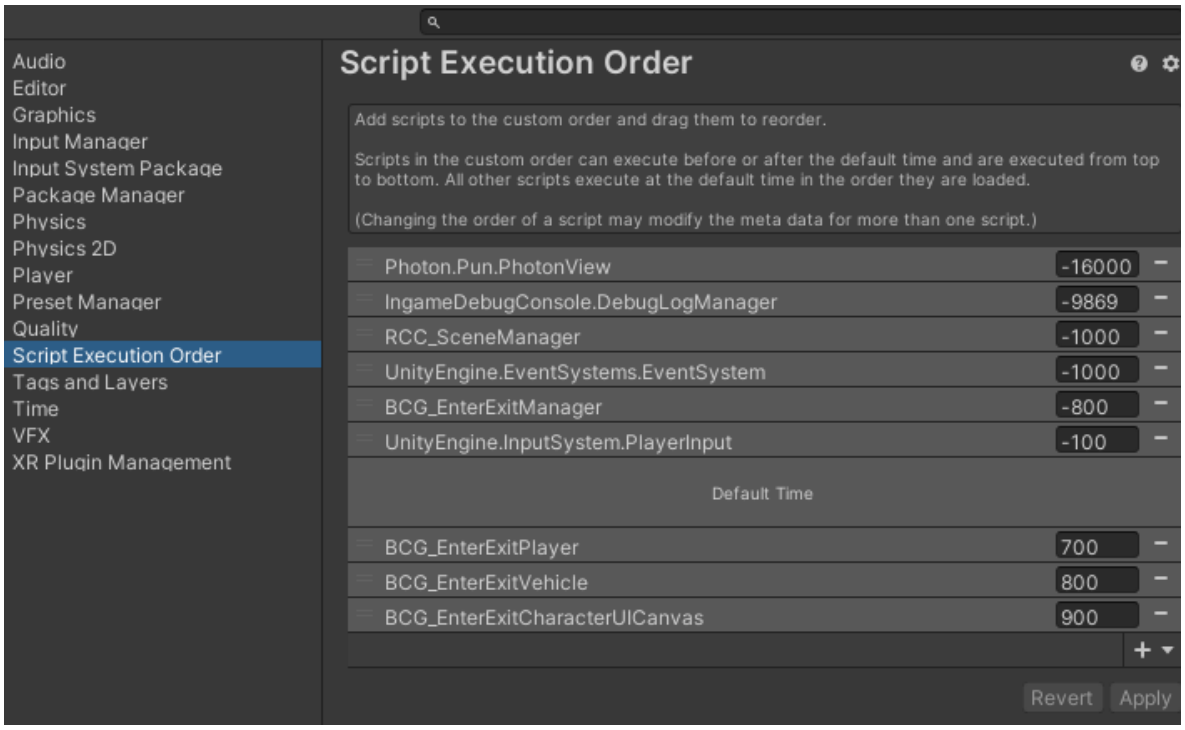




(Tools → BoneCracker Games → Realistic Car Controller → RCC Settings)

Script Execution Order

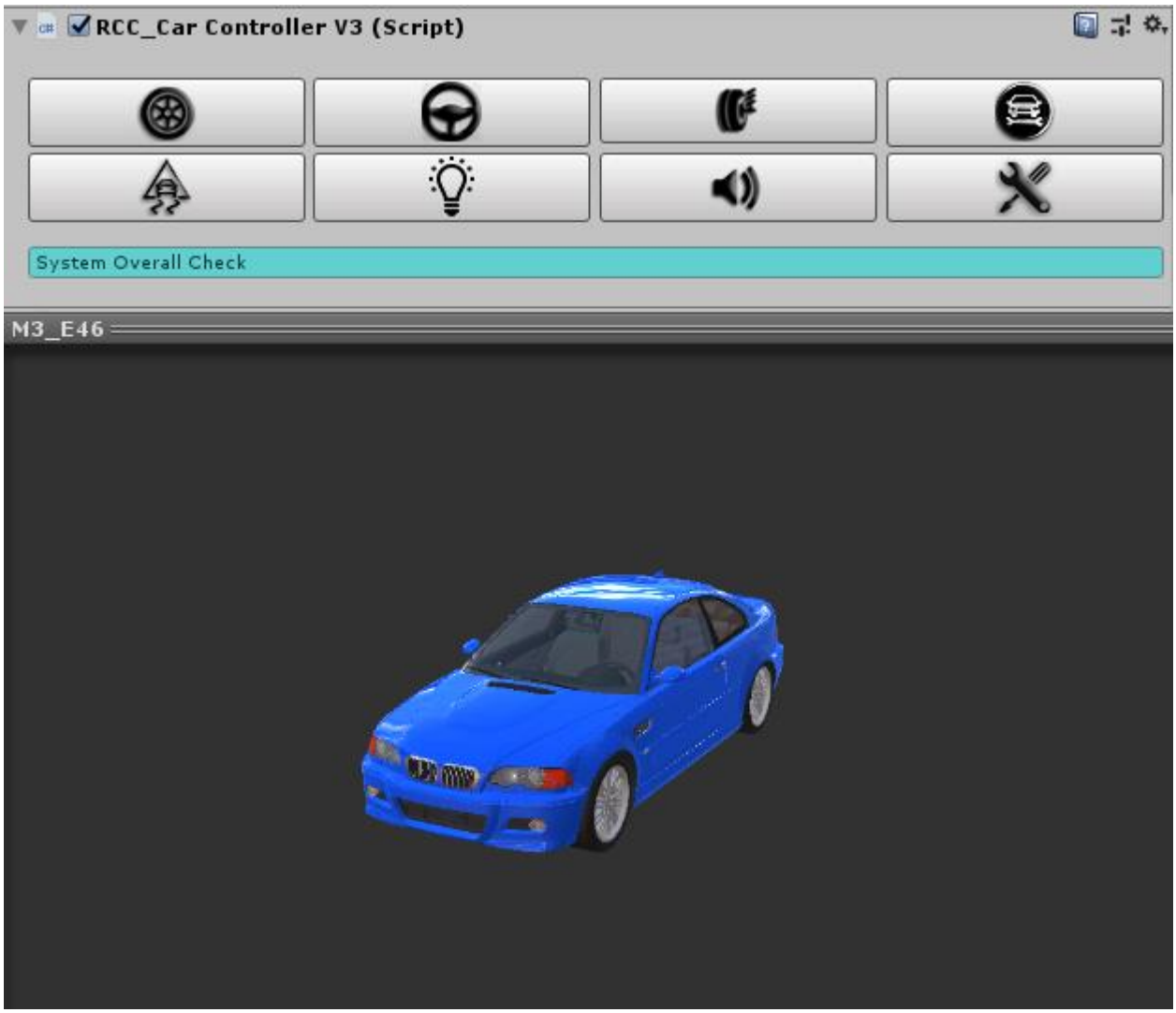
RCC is using **Script Execution Order** for avoid unexpected event conflicts. This should be imported successfully when RCC installed, and doesn't require any action. Just make sure you have this order. You can check it in **Edit → Project Settings → Script Execution Order**.



Overview

Each vehicle has it's own **RCC_CarControllerV3.cs** script. Each vehicle is responsible for own **RCC_CarControllerV3.cs**. All global shared settings are located in **RCC Settings** (**Tools → BCG → RCC → Edit Settings**). Lights, cameras, exhausts are addons and not required as an essential. Inputs are processed by **RCC_InputManager.cs** script. It will receive corresponding inputs from the selected device. **RCC_SceneManager.cs** is managing active player vehicle, other vehicles, AI vehicles, record/replay, UI canvases, etc… All other main topics can be found below.

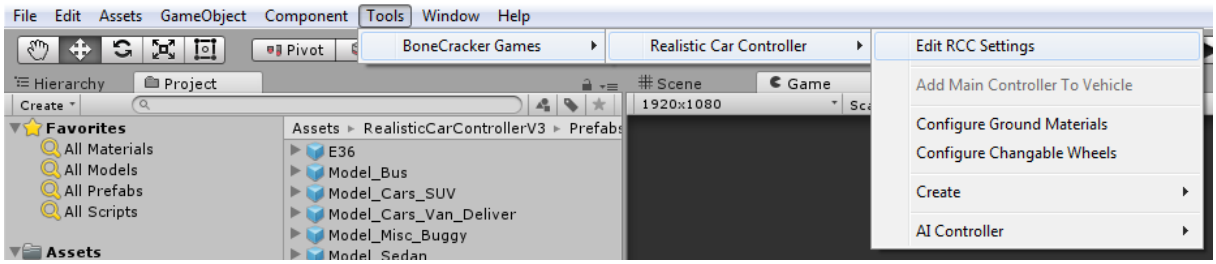
RCC_CarControllerV3.cs



8 Main Categories for easily and understandable creating / configurating vehicles.

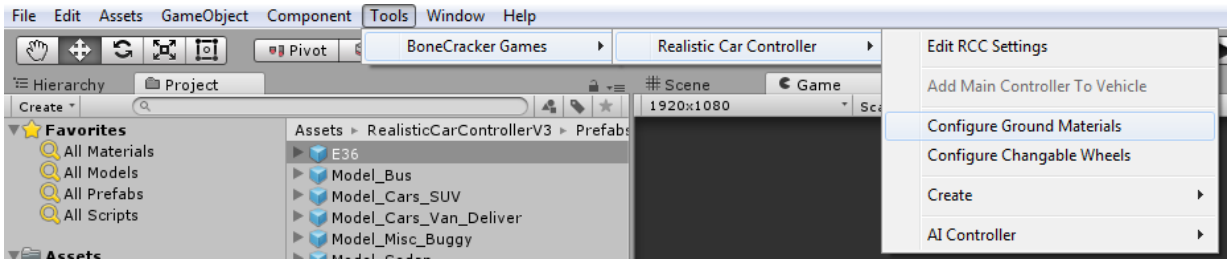
Wheels, **Steering**, **Suspensions**, **Mechanic Configuration**, **Stability**, **Lights**, **Sounds**, and **Damage**.

All vehicles are sharing global settings, sounds, configurations via **RCC Settings**.

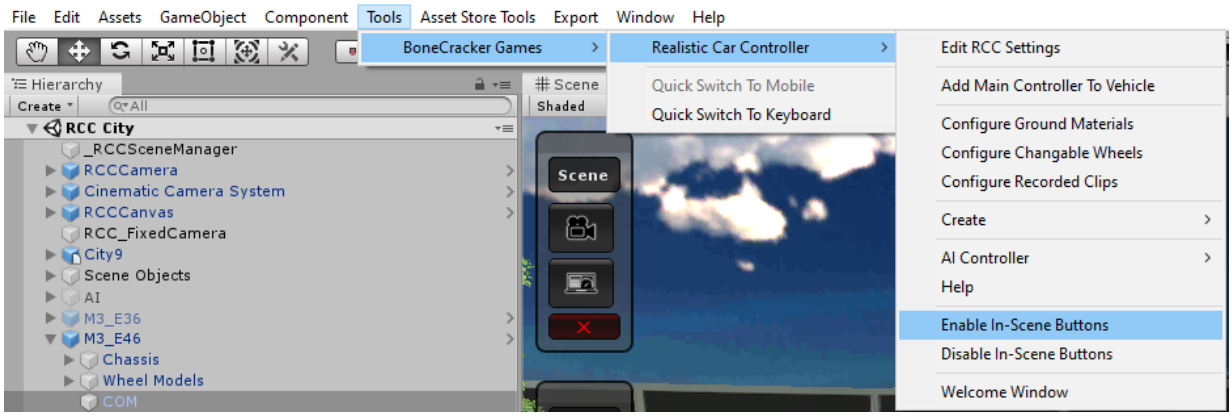


Creating new vehicles is explained in documentation named “**Realistic Car Controller V3.53 How to Create New Vehicles**.”

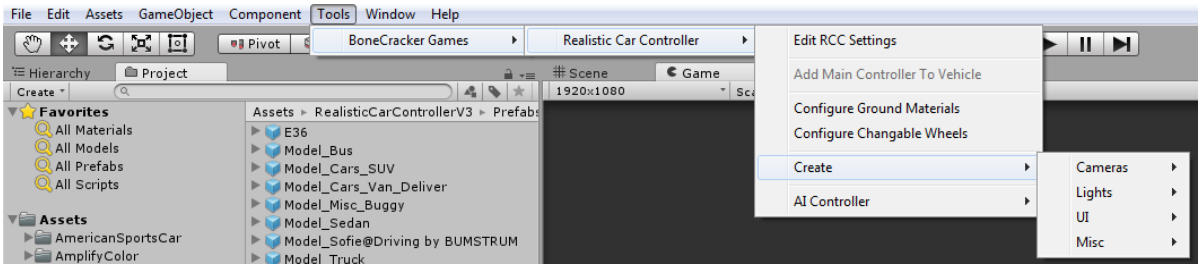
Changing ground materials physics, particles, sounds, etc in **Tools → BoneCracker Games → Realistic Car Controller → Configure Ground Materials**. (Detailed explanation in documentation named “**Realistic Car Controller V3.53 RCC_GroundMaterials**”)



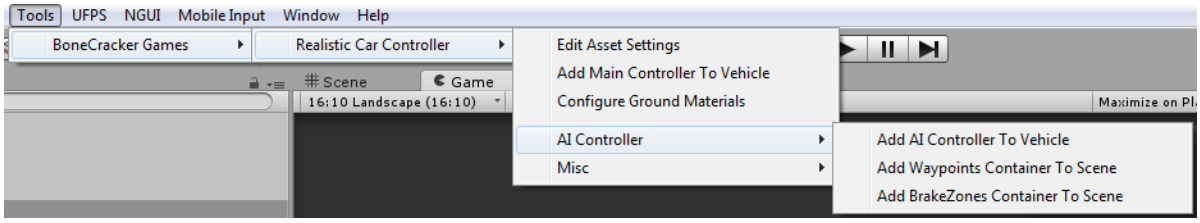
You may want to enable In-Scene buttons to create addons fastest way. **Tools → BoneCracker Games → Realistic Car Controller → Enable In-Scene Buttons**. (Detailed explanation in documentation named “**Realistic Car Controller V3.53 How To Create New Vehicles**”)



Creating lights, exhausts, mirrors, cameras, etc in **Tools → BoneCracker Games → Realistic Car Controller → Create.** (Detailed explanation in documentation named “**Realistic Car Controller V3.53 How To Create New Vehicles**”)

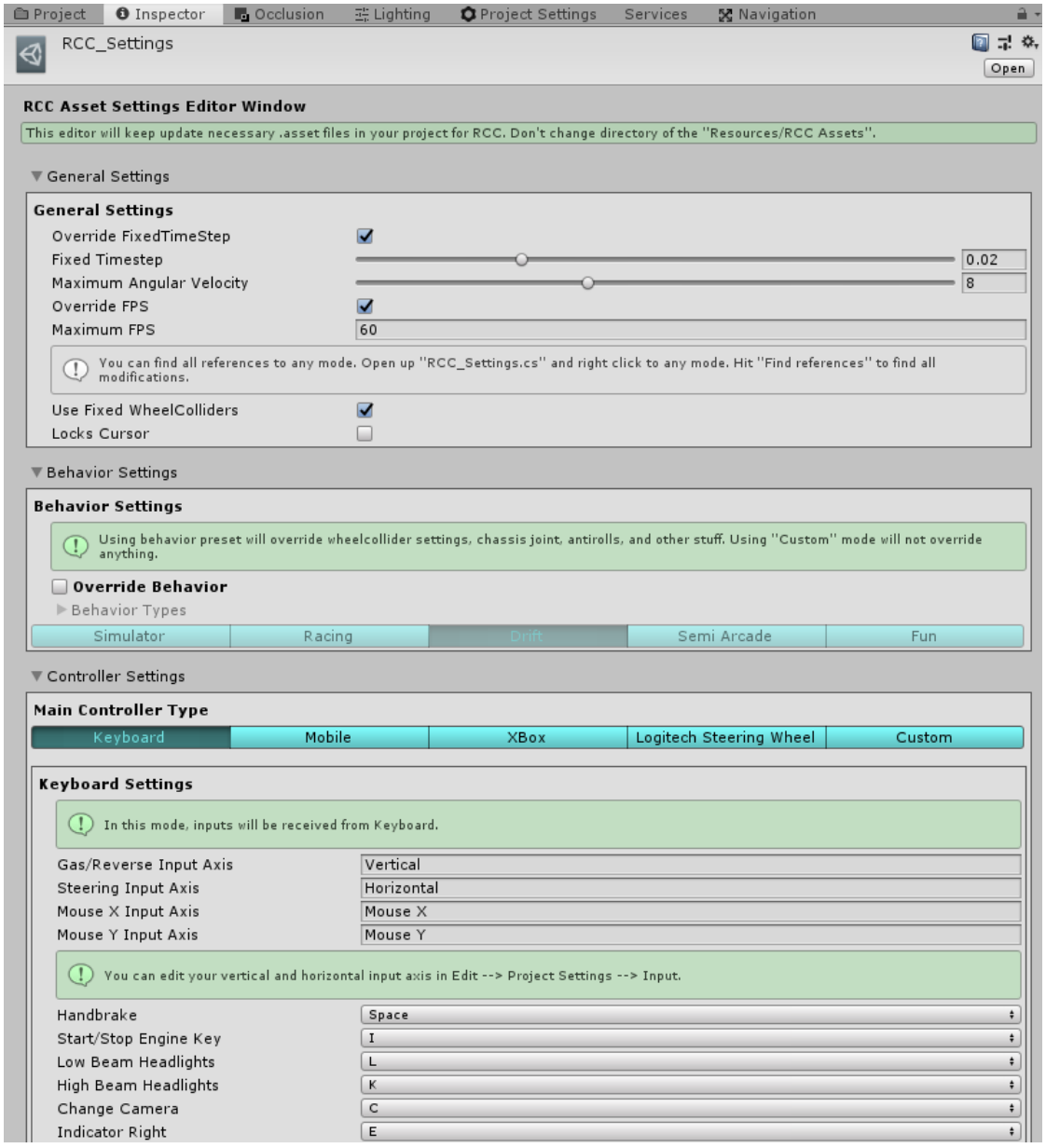


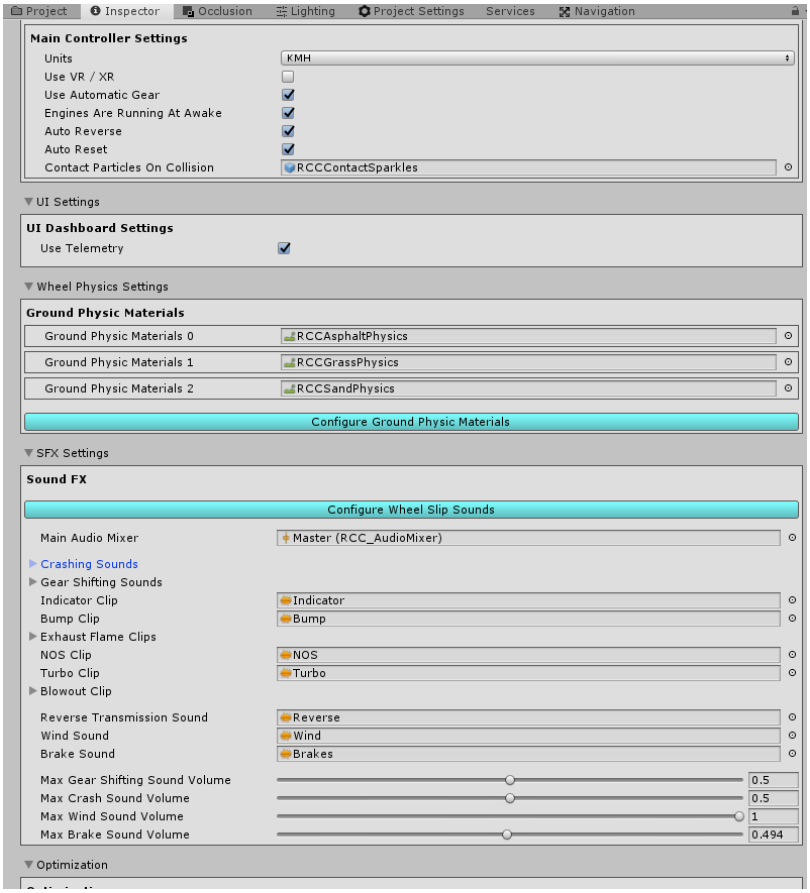
Making vehicles controlled by AI in **Tools → BoneCracker Games → Realistic Car Controller → AI Controller.** (Detailed explanation in documentation named “**Realistic Car Controller V3.53 AI**”)

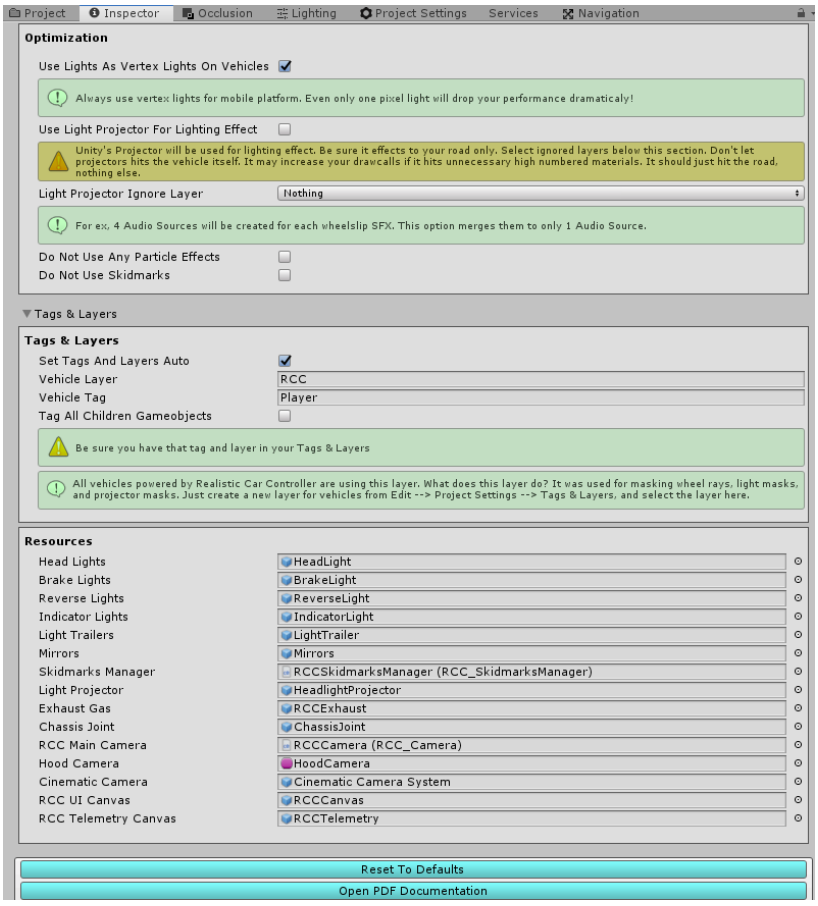


RCC Settings

Main RCC Settings. It's shared by all vehicles powered by RCC. **Tools → BoneCracker Games → Realistic Car Controller → RCC Settings**. (Detailed explanation in documentation named "**Realistic Car Controller V3.53 RCC_Settings**")

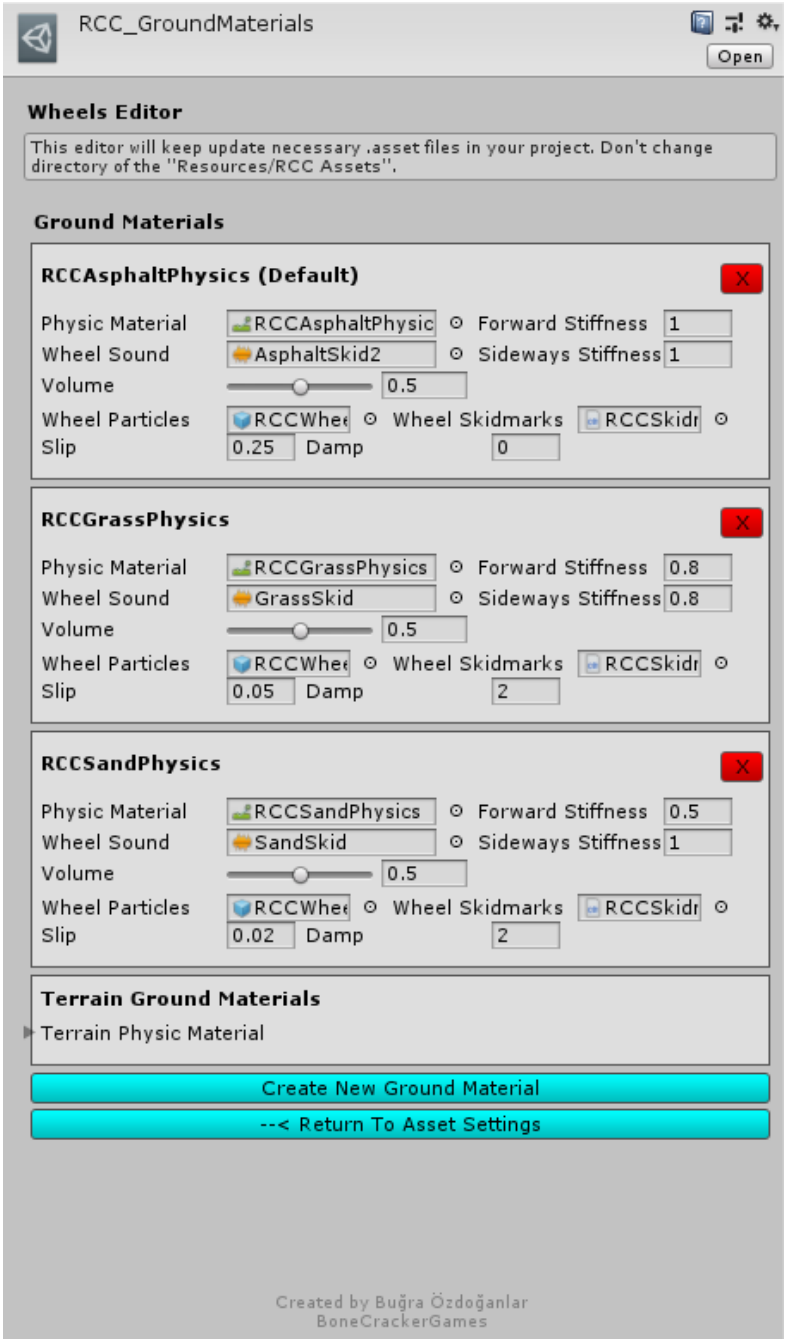



Project
Inspector
Occlusion
Lighting
Project Settings
Services
Navigation
Main Controller Settings
Units
KMH
Use VR / XR
Use Automatic Gear
Engines Are Running At Awake
Auto Reverse
Auto Reset
Contact Particles On Collision
RCCContactSparkles
UI Settings
UI Dashboard Settings
Use Telemetry
Wheel Physics Settings
Ground Physic Materials
Ground Physic Materials 0
RCCAsphaltPhysics
Ground Physic Materials 1
RCCGrassPhysics
Ground Physic Materials 2
RCCSandPhysics
Configure Ground Physic Materials
SFX Settings
Sound FX
Configure Wheel Slip Sounds
Main Audio Mixer
Master (RCC_AudioMixer)
Crashing Sounds
Gear Shifting Sounds
Indicator Clip
Indicator
Bump Clip
Bump
Exhaust Flame Clips
NOS Clip
NOS
Turbo Clip
Turbo
Blowout Clip
Reverse Transmission Sound
Reverse
Wind Sound
Wind
Brake Sound
Brakes
Max Gear Shifting Sound Volume
0.5
Max Crash Sound Volume
0.5
Max Wind Sound Volume
1
Max Brake Sound Volume
0.494
Optimization


Project
Inspector
Occlusion
Lighting
Project Settings
Services
Navigation
Optimization
Use Lights As Vertex Lights On Vehicles
Always use vertex lights for mobile platform. Even only one pixel light will drop your performance dramaticaly!
Use Light Projector For Lighting Effect
Unity's Projector will be used for lighting effect. Be sure it effects to your road only. Select ignored layers below this section. Don't let projectors hits the vehicle itself. It may increase your drawcalls if it hits unnecessary high numbered materials. It should just hit the road, nothing else.
Light Projector Ignore Layer
Nothing
For ex, 4 Audio Sources will be created for each wheelslip SFX. This option merges them to only 1 Audio Source.
Do Not Use Any Particle Effects
Do Not Use Skidmarks
Tags & Layers
Tags & Layers
Set Tags And Layers Auto
Vehicle Layer
RCC
Vehicle Tag
Player
Tag All Children Gameobjects
Be sure you have that tag and layer in your Tags & Layers
All vehicles powered by Realistic Car Controller are using this layer. What does this layer do? It was used for masking wheel rays, light masks, and projector masks. Just create a new layer for vehicles from Edit --> Project Settings --> Tags & Layers, and select the layer here.
Resources
Head Lights
HeadLight
Brake Lights
BrakeLight
Reverse Lights
ReverseLight
Indicator Lights
IndicatorLight
Light Trailers
LightTrailer
Mirrors
Mirrors
Skidmarks Manager
RCCSkidmarksManager (RCC_SkidmarksManager)
Light Projector
HeadlightProjector
Exhaust Gas
RCCExhaust
Chassis Joint
ChassisJoint
RCC Main Camera
RCCCamera (RCC_Camera)
Hood Camera
HoodCamera
Cinematic Camera
Cinematic Camera System
RCC UI Canvas
RCCCanvas
RCC Telemetry Canvas
RCCTelemetry
Reset To Defaults
Open PDF Documentation

Configurable Ground Materials

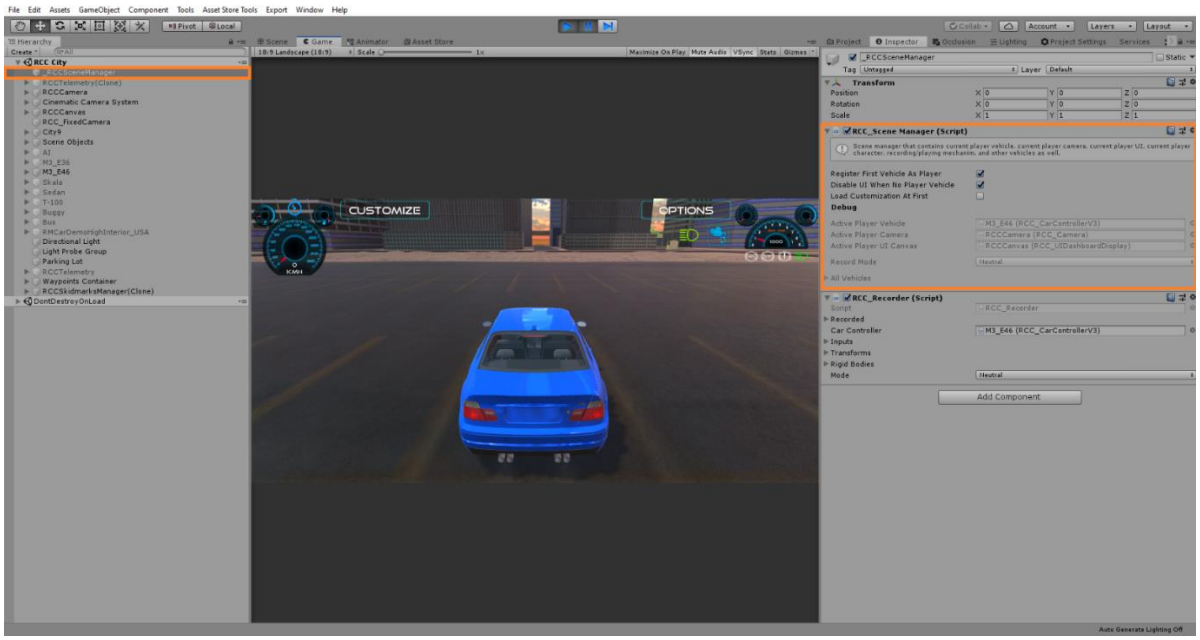
Changing or adding new ground materials physics, particles, damps, sounds, etc in **Tools → BoneCracker Games → Realistic Car Controller → Configure Ground Materials**.
(Detailed explanation in documentation named "**Realistic Car Controller V3.53 RCC_GroundMaterials**")



If WheelCollider hits a collider with one of the physic material in the list, changes will be applied to WheelCollider. You can check out demo scenes.

RCC Scene Manager

Every scene will have this manager automatically. **RCC Scene Manager** contains current player vehicle, current player camera, current player UI, current player character, recording / replay mechanim, and other vehicles as well. Instead of finding current car controller, or camera on scene, RCC Scene Manager will find it and manage it only. All other scripts depending on player vehicle will take reference of the RCC Scene Manager. For ex, finding player vehicle on scene is **RCC_SceneManager.Instance.activePlayerVehicle**. All other codes can be found at scripts documentation.



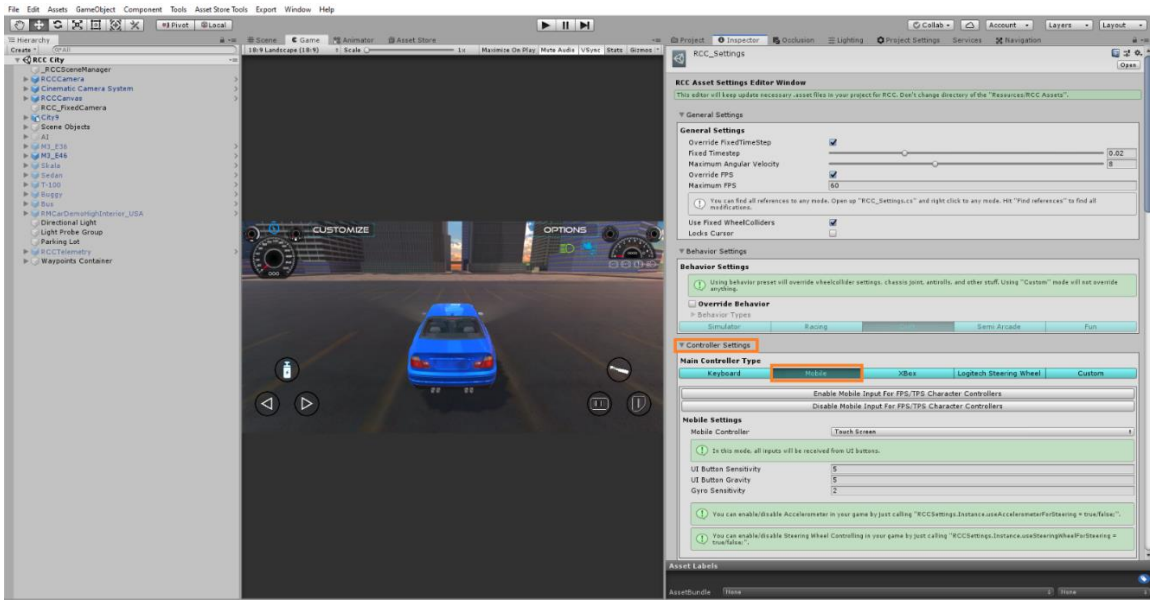
Controller Types

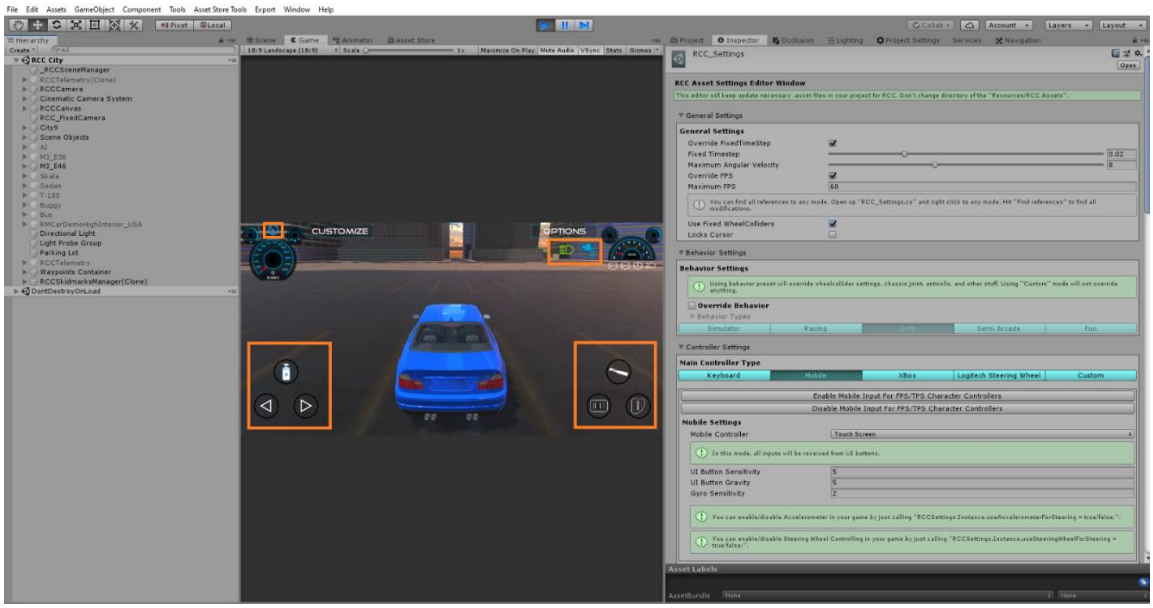
RCC supports all controller types with the new input system. Each controller can be changed directly from **RCC_InputActions** (Detailed explanation in documentation named "**Realistic Car Controller V3.53 New Input System**")

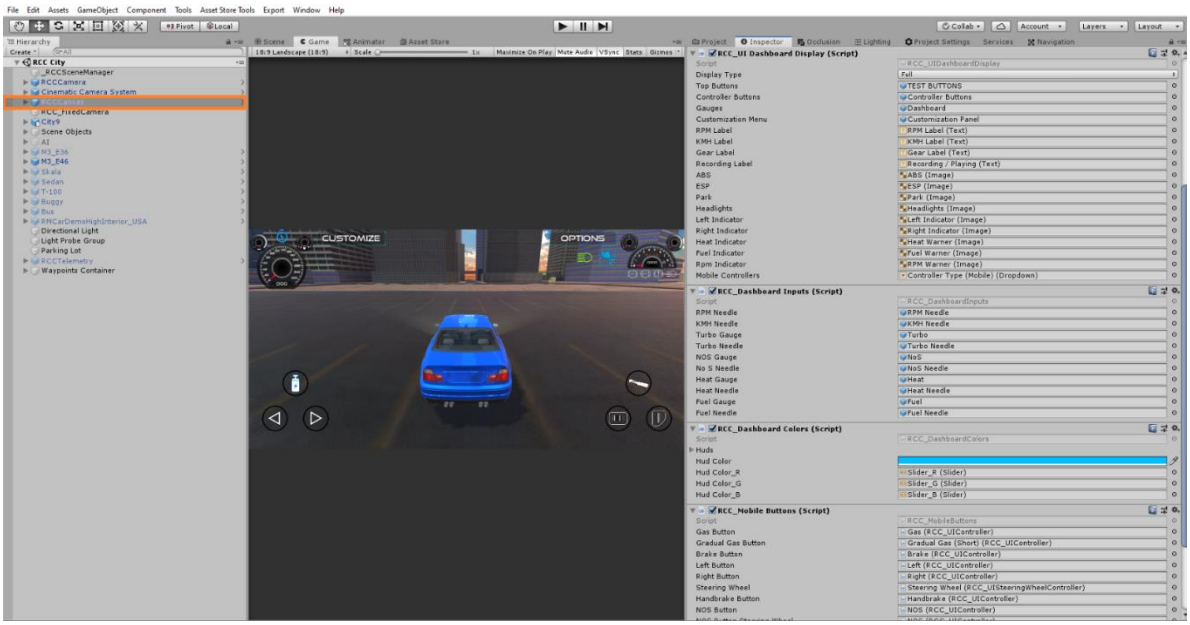
Logitech Steering requires **Logitech Gaming SDK** installed in your project.

Mobile Controller

Mobile controller is using my own input system instead of the new input manager. Each UI controller button has “**RCC_UIController.cs**” script for inputs. These buttons feeds **RCC_InputManager** with normalized float values. You can adjust UI buttons sensitivity and gravity from **RCC Settings**. Switching mobile controller to the new input manager is easy, however I don’t recommend to do this. Because UI buttons will simulate gamepad buttons in this case.







About Mobile Usement On City Scene

City scene has lot of specular maps with alpha channels. Textures with alpha channels and bump maps are heavy for mobile devices. In Demo APK in my website is not using any texture with alpha channels. Also all standard shaders are replaced with mobile shaders in **RCC City Mobile** scene at the demo. If you build an APK without editing materials, you may get performance loss on low-end devices.

Keyboard Shortcuts

Keyboard shortcuts can be used if "**Use Shortcuts**" is enabled in the **RCC Settings**. It's disabled by default.

Shift + R = Add main controller to the vehicle

Shift + S = RCC Settings

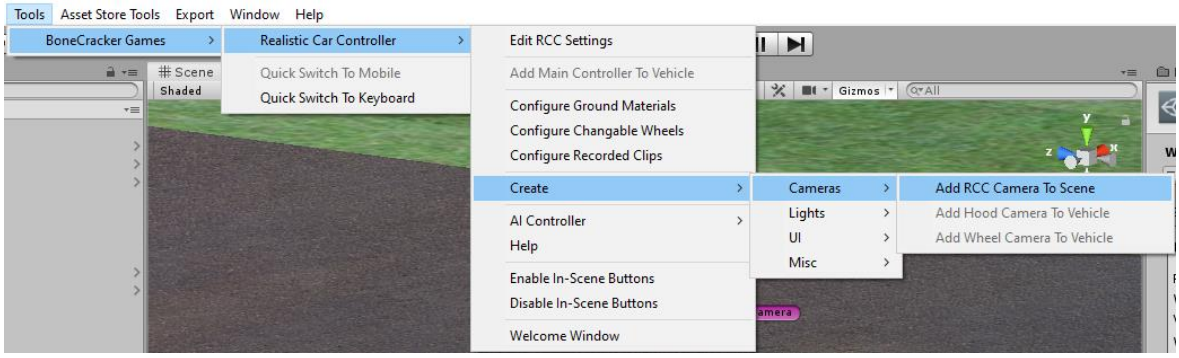
Shift + E = Enable In-Scene editor buttons

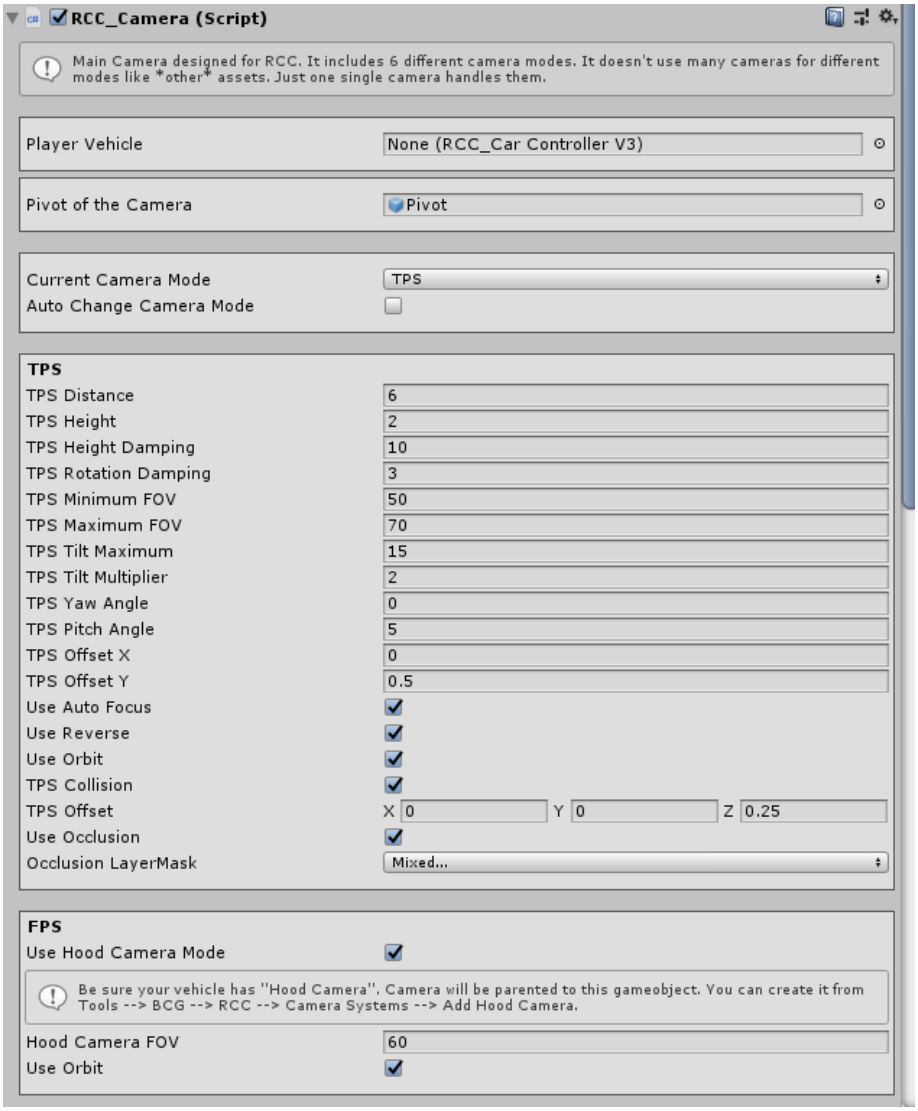
Some packages may conflict with the shortcuts. For example, removing road key in EasyRoads (Shift + R) will conflict with this. To remove or change any shortcut, disable "**Use Shortcuts**" in the **RCC Settings**.

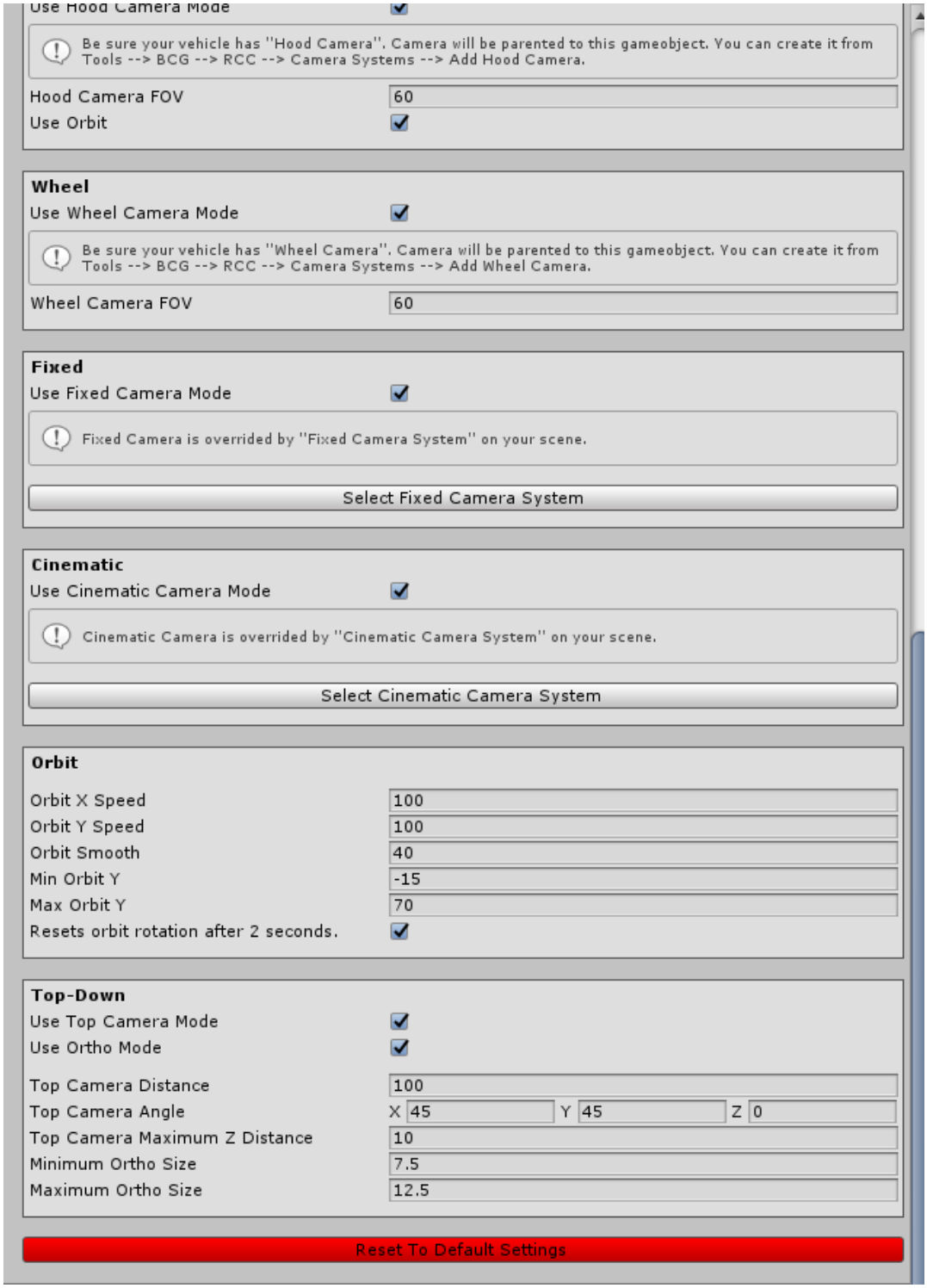
RCC Camera

Main camera system designed for using with RCC. Related with vehicle stats and includes six different camera modes with many customizable settings. It doesn't use different individual cameras on your scene. Simply it parents the camera to their positions, and that's all.

If your scene doesn't have RCC Camera, you can create it from **Tools → BoneCracker Games → Realistic Car Controller → Create → Cameras → Add RCC Camera To Scene**.







Each camera mode can be customized here. **TPS** mode is required, and all other modes are optional. If you don’t want to use hood, wheel, fixed, cinematic camera, top-down modes, you can just disable them here.

Record / Replay

Complete physics and input based record / replay system. Player vehicle and all active AI vehicles can record / replay. All you have to do is press “**R**” for start recording, and “**P**” for start replay. These buttons can be changed in **RCC_InputActions**. And of course, there is a UI button for mobile.

RCC_Recorder can be found at attached to **_RCCSceneManager** on your scene. You can enable or disable it. Script will be added at awake, or you can add it by manually if enabled. You can use RCC’s API for start record / replay at runtime. For ex;

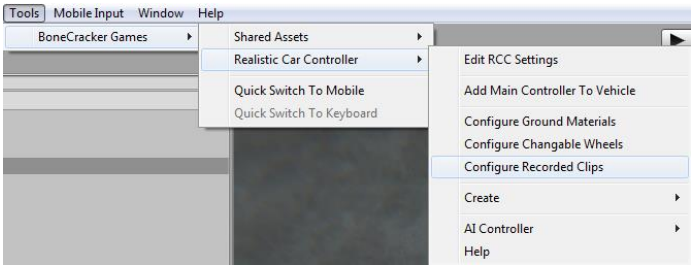
RCC. StartStopReplay ();

RCC. StartStopReplay (RCC_Recorder.Recorded recordedClip);

RCC. StartStopReplay (int index);

RCC. StartStopReplay (RCC_Recorder.Recorded recordedClip);

All records are stored in **RCC_Records**. You can access it from **Tools → BCG → RCC → Configure Recorded Clips**.



Customization

You can customize your vehicles by just calling a single method. Please take a look at "**Realistic Car Controller V3.53 Scripts**" documentation. All methods in **RCC_Customization** are explained there.

How The Customization Panel Works

I wrote a example script called "**RCC_CustomizerExample.cs**" which uses static methods in **RCC_Customization.cs**. Script is attached to **RCCCanvas**. UI buttons in customization panel sends methods to this example script. And this example script uses static methods in **RCC_Customization.cs** for making changes. Let me explain it with simple examples;

We want to change front suspension distance of our vehicle. So, we have to call;

RCC_Customization.SetFrontSuspensionsDistances (targetRCC, targetValue);

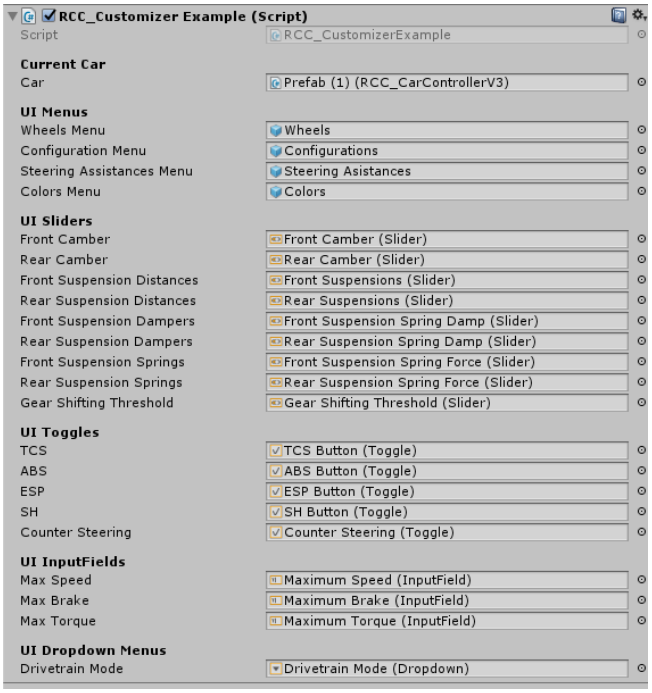
We want to repair our car. So, we have to call;

RCC_Customization. RepairCar (targetRCC);

We want to change drivetrain of our car to AWD. So, we have to call it;

```
RCC_Customization. SetDrivetrainMode (targetRCC,
RCC_CarControllerV3 WheelType.AWD);
```

And goes on... Simply take a look at all methods in **RCC_CustomizerExample.cs** script, you will see how I customized the player vehicle by using **RCC_Customization.cs** script.



This example script handles all UI menus, buttons, sliders, toggles, inputfields, and dropdown menus of the customization panel. It just receives inputs from UI, and fires necessary actions.

Credits

Driver Sofie, her animations, and her car model made by 3DMaesen. You can access 3DMaesen asset store from this link;

http://u3d.as/2vg

All sounds in package are completely royalt free. You can use them on any personal or commercial projects. You can't redistribute / resell them.

License

You can use this package for your personal / commercial projects. **But you can't resell or redistribute any asset in the package on any store (not even any single asset in package)**. I got many reports from my customers about some fake developers are reselling my package on other stores. This is strictly forbidden. You can't resell or redistribute ANY asset from Unity's Asset Store, unless if developer gave you special license for making this. If anyone violates this, he will be banned, and his revenue from package sellings will be interrupted. You can read Unity EULA from this link;

http://unity3d.com/legal/as_terms

You can ask me anything about my assets! If you want change **minor things** in the package, don't waste your time by editing scripts. Just tell me, I'll do my best with no cost. I don't take any projects right now, and I'm not available for hire. Please email me if you used any of my assets in your game, I'd like to see it in action!

Ekrem Buğra Özdoğanlar

Bonecrackergames@gmail.com