Realistic Car Controller V3.3 by BoneCracker Games

First of all, thank you for purchasing and using Realistic Car Controller! This documentation will guide you to;

Contents

You can find more updated details on

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

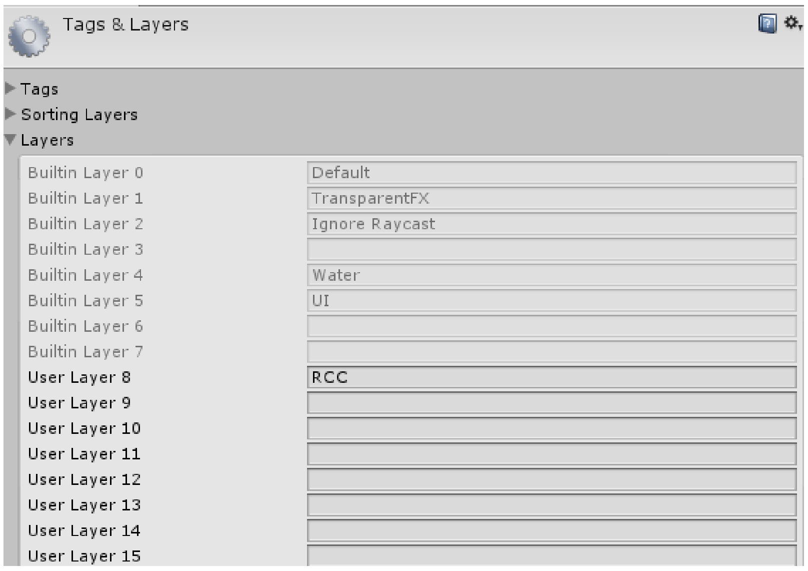
https://www.youtube.com/playlist?list=PLRXTqAVrLDpoW58lKf8XA1AWD6kDkoKb1

(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 from RealisticCarControllerV3 folder. Delete the entire folder, and import updated version.

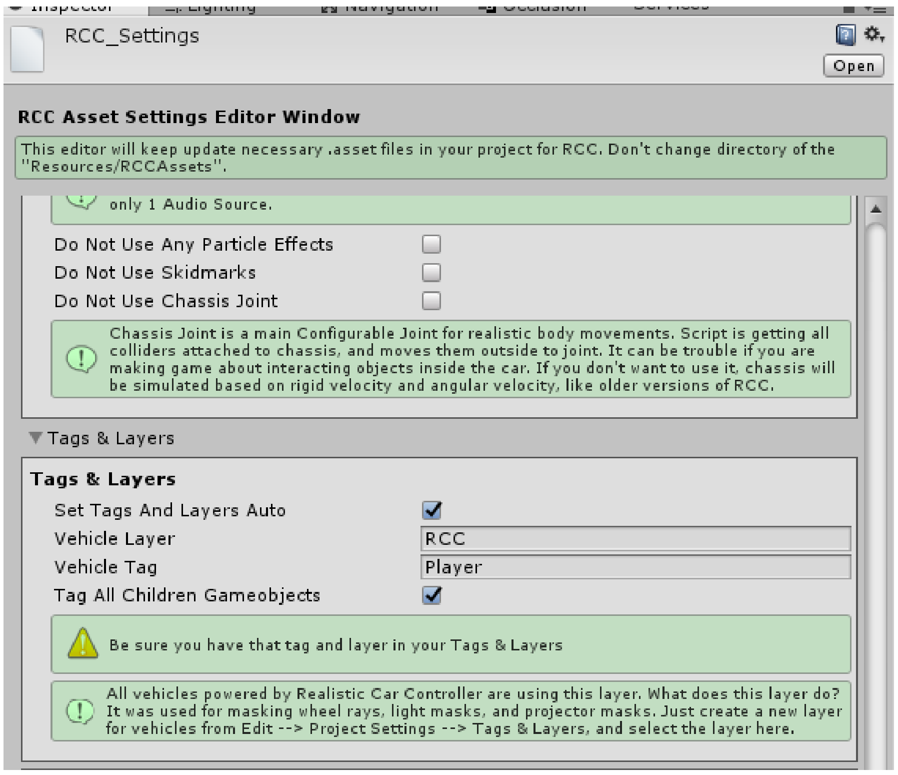
This version of the RCC is using **LayerMask** for avoid unwanted raycast hits and ignore unnecessary projector layers. Just create one layer for vehicles, and select it in **RCC Settings**.



Tags & Layers

▶ Tags
▶ Sorting Layers
▼ Layers

Builtin Layer 0	Default
Builtin Layer 1	TransparentFX
Builtin Layer 2	Ignore Raycast
Builtin Layer 3	
Builtin Layer 4	Water
Builtin Layer 5	UI
Builtin Layer 6	
Builtin Layer 7	
User Layer 8	RCC
User Layer 9	
User Layer 10	
User Layer 11	
User Layer 12	
User Layer 13	
User Layer 14	
User Layer 15	


RCC_Settings

Open

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCCAssets".

only 1 Audio Source.

Do Not Use Any Particle Effects	☐
Do Not Use Skidmarks	☐
Do Not Use Chassis Joint	☐

Chassis Joint is a main Configurable Joint for realistic body movements. Script is getting all colliders attached to chassis, and moves them outside to joint. It can be trouble if you are making game about interacting objects inside the car. If you don't want to use it, chassis will be simulated based on rigid velocity and angular velocity, like older versions of RCC.

Tags & Layers

Tags & Layers

Set Tags And Layers Auto	✓
Vehicle Layer	RCC
Vehicle Tag	Player
Tag All Children Gameobjects	✓

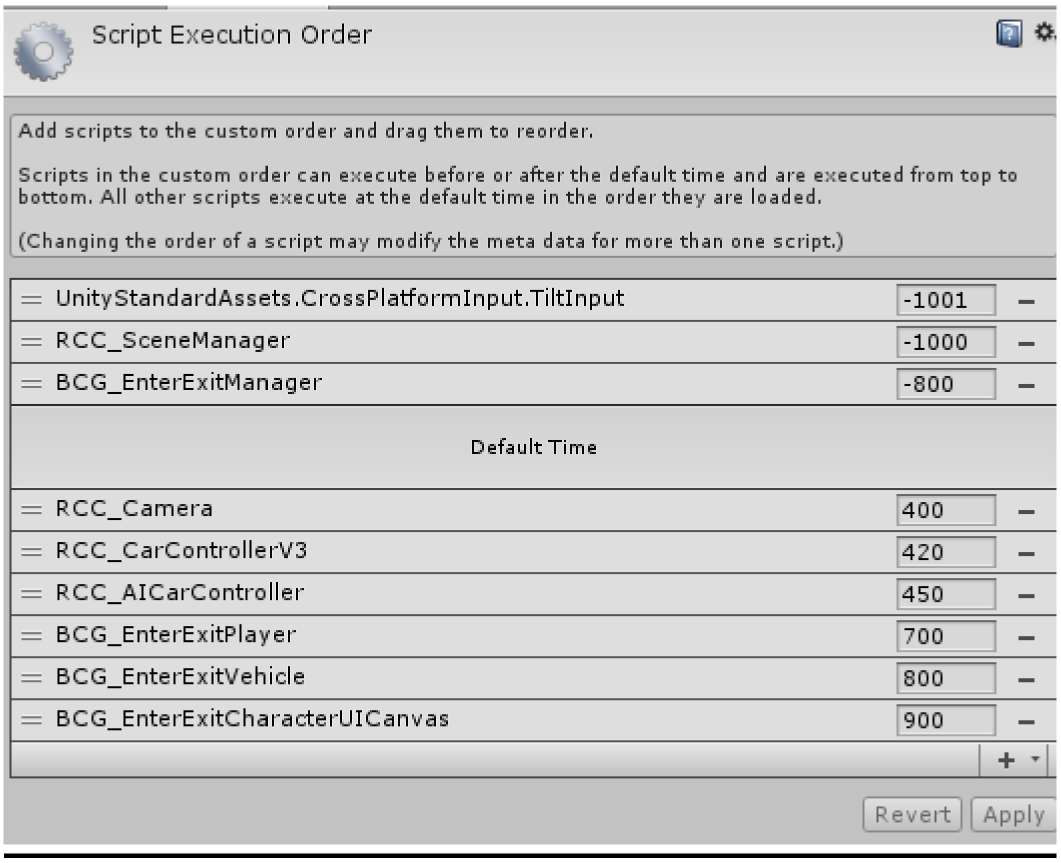
Be sure you have that tag and layer in your Tags & Layers

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.

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

Script Execution Order

This version of the 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**.


Script Execution Order

Add scripts to the custom order and drag them to reorder.

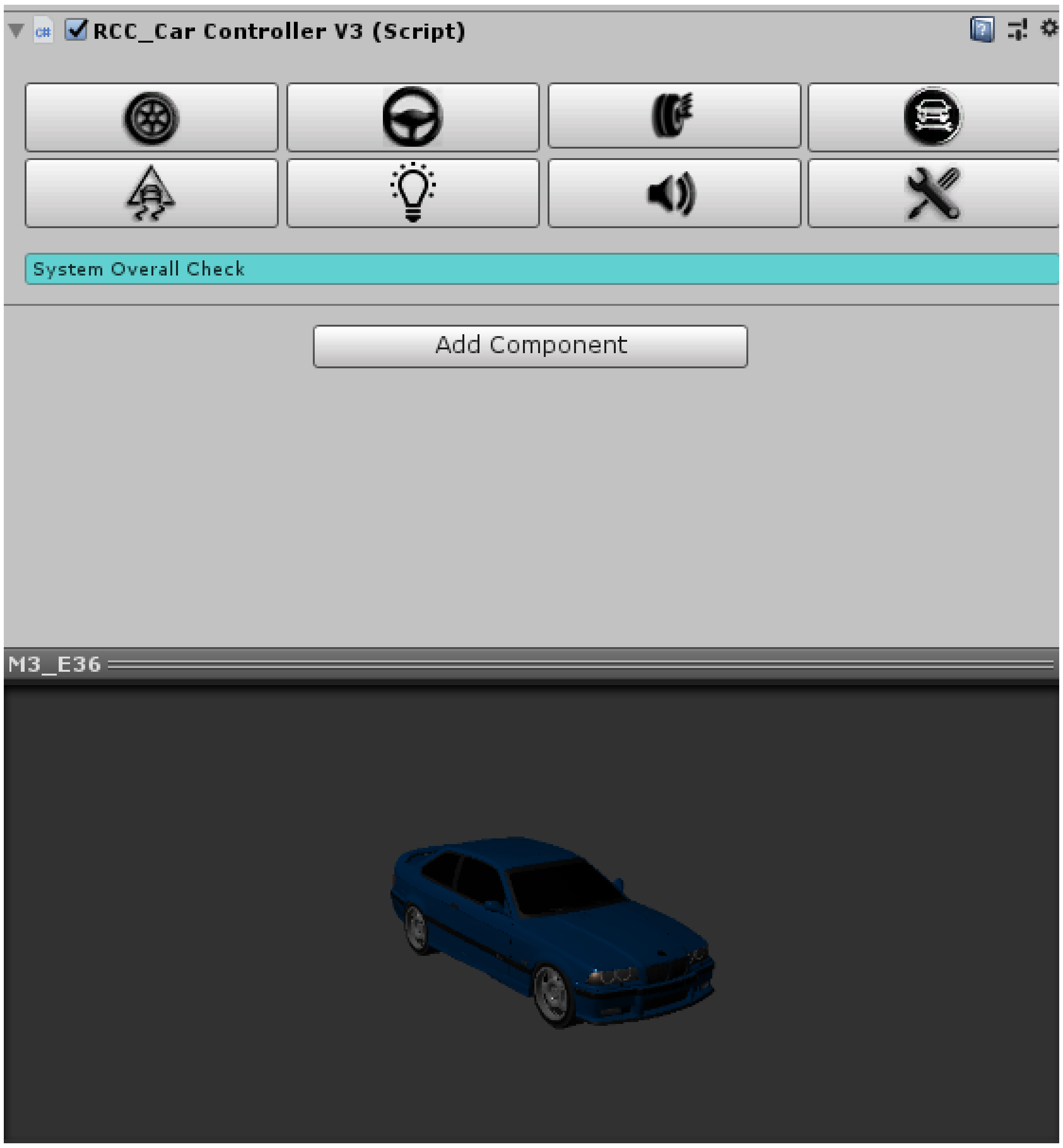
Scripts in the custom order can execute before or after the default time and are executed from top to bottom. All other scripts execute at the default time in the order they are loaded.

(Changing the order of a script may modify the meta data for more than one script.)

UnityStandardAssets.CrossPlatformInput.TiltInput	-1001
RCC_SceneManager	-1000
BCG_EnterExitManager	-800
Default Time	
RCC_Camera	400
RCC_CarControllerV3	420
RCC_AICarController	450
BCG_EnterExitPlayer	700
BCG_EnterExitVehicle	800
BCG_EnterExitCharacterUICanvas	900

General Information about RCC

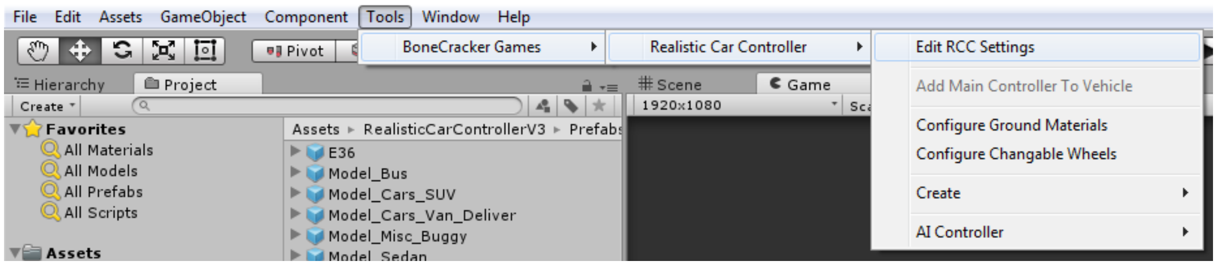
Each vehicle has it's own RCC_CarControllerV3.cs script. Each vehicle is responsible for own RCC_CarControllerV3.cs.



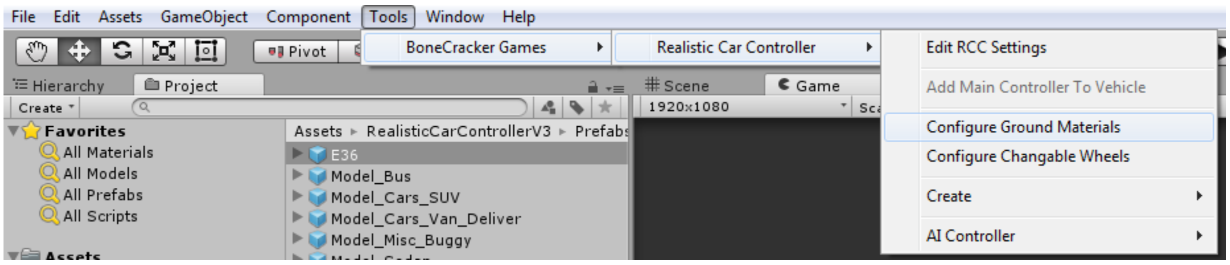
8 Main Categories for easily and understandable creating-configurating vehicles.

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

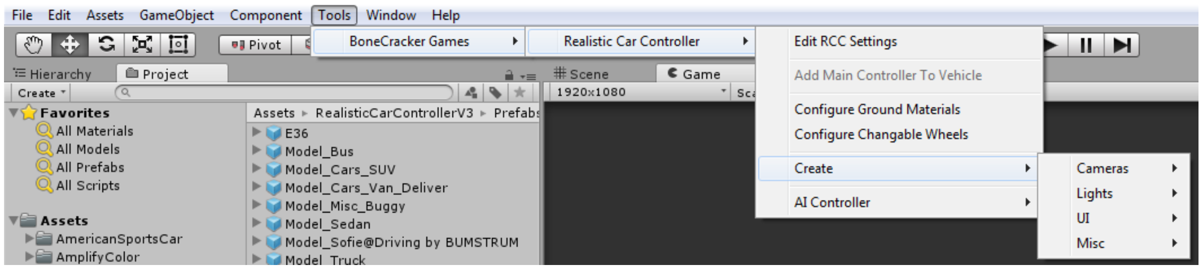
All vehicles are sharing global settings, sounds, configurations via RCC Settings.


Tools
BoneCracker Games
Realistic Car Controller
Edit RCC Settings
Add Main Controller To Vehicle
Configure Ground Materials
Configure Changable Wheels
Create
AI Controller

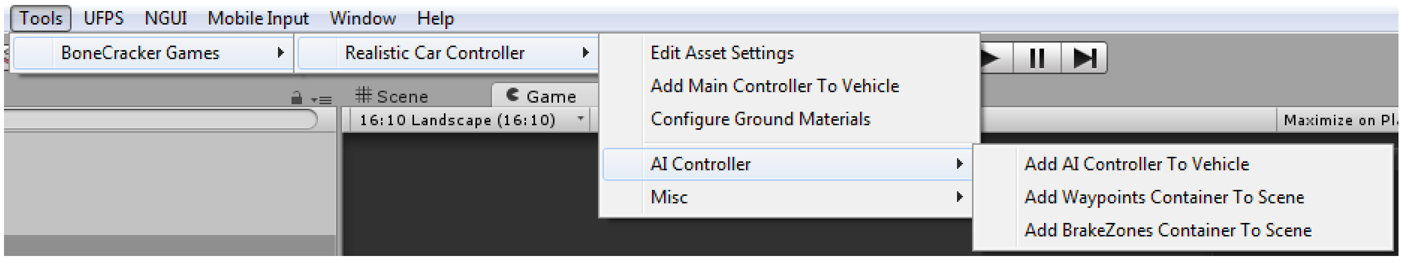
Changing ground materials physics, particles, damps, sounds, etc in **Tools → BoneCracker Games → Realistic Car Controller → Configure Ground Materials**.


Tools
BoneCracker Games
Realistic Car Controller
Edit RCC Settings
Add Main Controller To Vehicle
Configure Ground Materials
Configure Changable Wheels
Create
AI Controller

Creating lights, exhausts, mirrors, cameras, etc in **Tools → BoneCracker Games → Realistic Car Controller → Create**.

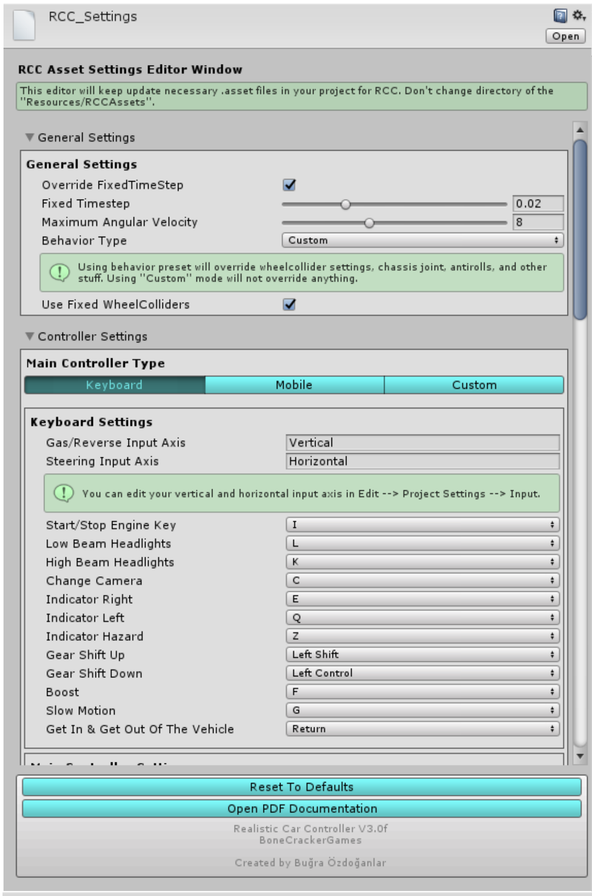


Making vehicles controlled by AI in **Tools → BoneCracker Games → Realistic Car Controller → AI Controller**.


Tools
BoneCracker Games
Realistic Car Controller
Edit Asset Settings
Add Main Controller To Vehicle
Configure Ground Materials
AI Controller
Misc
Add AI Controller To Vehicle
Add Waypoints Container To Scene
Add BrakeZones Container To Scene

RCC Settings

Main RCC Settings. It’s shared by all vehicles powered by RCC. **Tools → BoneCracker Games → Realistic Car Controller → RCC Settings**.


RCC_Settings

Open

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCCAssets".

▼ General Settings

General Settings

Override FixedTimeStep	✓
Fixed Timestep	0.02
Maximum Angular Velocity	8
Behavior Type	Custom

Using behavior preset will override wheelcollider settings, chassis joint, antirolls, and other stuff. Using "Custom" mode will not override anything.

Use Fixed WheelColliders	✓

▼ Controller Settings

Main Controller Type

Keyboard | Mobile | Custom

Keyboard Settings

Gas/Reverse Input Axis	Vertical
Steering Input Axis	Horizontal

You can edit your vertical and horizontal input axis in Edit --> Project Settings --> Input.

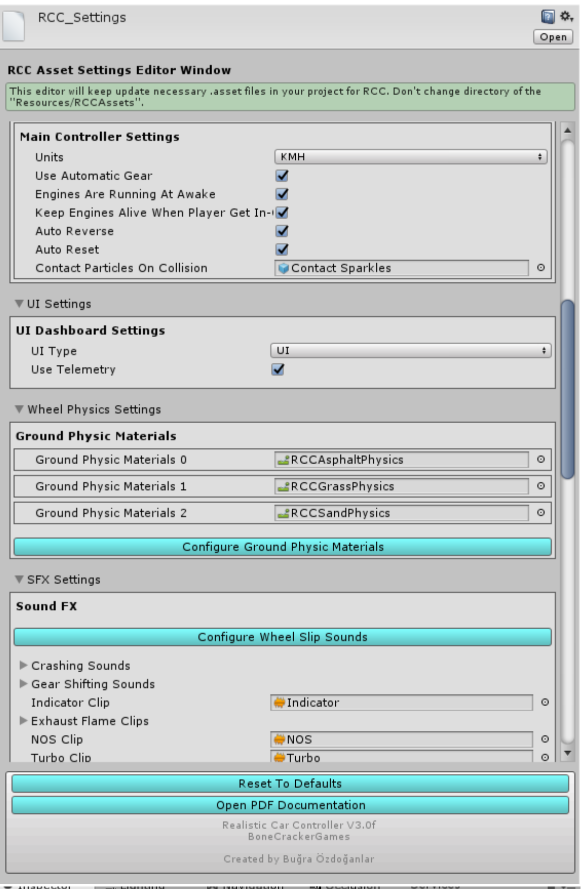
Start/Stop Engine Key	I
Low Beam Headlights	L
High Beam Headlights	K
Change Camera	C
Indicator Right	E
Indicator Left	Q
Indicator Hazard	Z
Gear Shift Up	Left Shift
Gear Shift Down	Left Control
Boost	F
Slow Motion	G
Get In & Get Out Of The Vehicle	Return

Reset To Defaults

Open PDF Documentation

Realistic Car Controller V3.0f
BoneCrackerGames

Created by Buğra Özdoğanlar


RCC_Settings

Open

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCCAssets".

Main Controller Settings

Units	KMH
Use Automatic Gear	✓
Engines Are Running At Awake	✓
Keep Engines Alive When Player Get In-	✓
Auto Reverse	✓
Auto Reset	✓
Contact Particles On Collision	Contact Sparkles

▼ UI Settings

UI Dashboard Settings

UI Type	UI
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

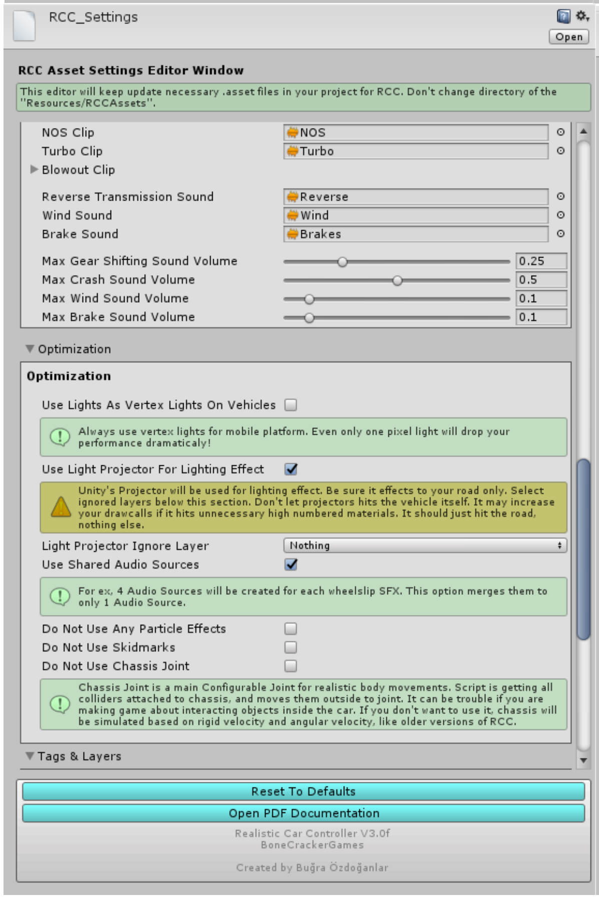
▶ Crashing Sounds	
▶ Gear Shifting Sounds	
Indicator Clip	Indicator
▶ Exhaust Flame Clips	
NOS Clip	NOS
Turbo Clip	Turbo

Reset To Defaults

Open PDF Documentation

Realistic Car Controller V3.0f
BoneCrackerGames

Created by Buğra Özdoğanlar


RCC_Settings

Open

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCCAssets".

NOS Clip	NOS
Turbo Clip	Turbo
▶ Blowout Clip	
Reverse Transmission Sound	Reverse
Wind Sound	Wind
Brake Sound	Brakes
Max Gear Shifting Sound Volume	0.25
Max Crash Sound Volume	0.5
Max Wind Sound Volume	0.1
Max Brake Sound Volume	0.1

▼ Optimization

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
Use Shared Audio Sources	✓

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	☐
Do Not Use Chassis Joint	☐

Chassis Joint is a main Configurable Joint for realistic body movements. Script is getting all colliders attached to chassis, and moves them outside to joint. It can be trouble if you are making game about interacting objects inside the car. If you don't want to use it, chassis will be simulated based on rigid velocity and angular velocity, like older versions of RCC.

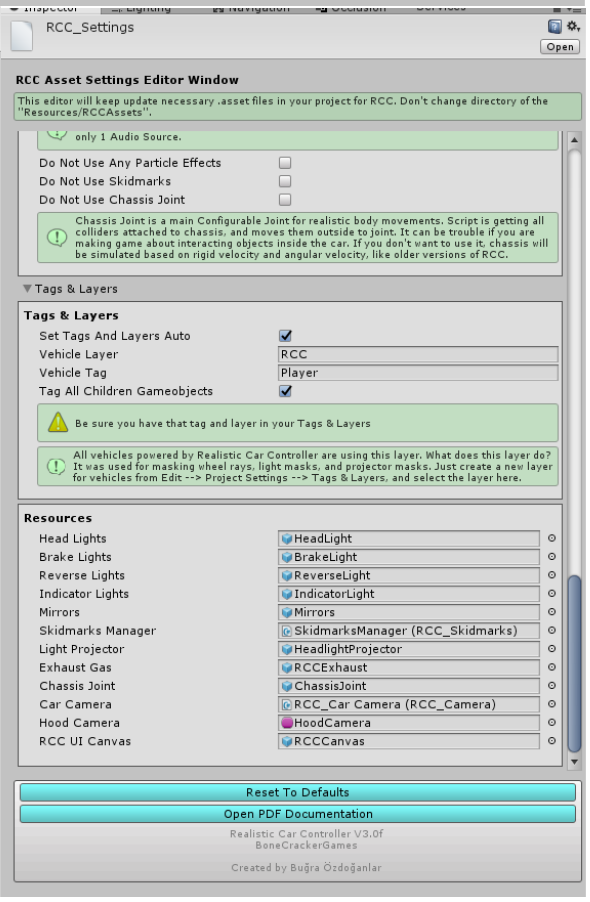
▼ Tags & Layers

Reset To Defaults

Open PDF Documentation

Realistic Car Controller V3.0f
BoneCrackerGames

Created by Buğra Özdoğanlar


RCC_Settings

Open

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCCAssets".

only 1 Audio Source.

Do Not Use Any Particle Effects	☐
Do Not Use Skidmarks	☐
Do Not Use Chassis Joint	☐

Chassis Joint is a main Configurable Joint for realistic body movements. Script is getting all colliders attached to chassis, and moves them outside to joint. It can be trouble if you are making game about interacting objects inside the car. If you don't want to use it, chassis will be simulated based on rigid velocity and angular velocity, like older versions of RCC.

▼ 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
Mirrors	Mirrors
Skidmarks Manager	SkidmarksManager (RCC_Skidmarks)
Light Projector	HeadlightProjector
Exhaust Gas	RCCExhaust
Chassis Joint	ChassisJoint
Car Camera	RCC_Car Camera (RCC_Camera)
Hood Camera	HoodCamera
RCC UI Canvas	RCCCanvas

Reset To Defaults

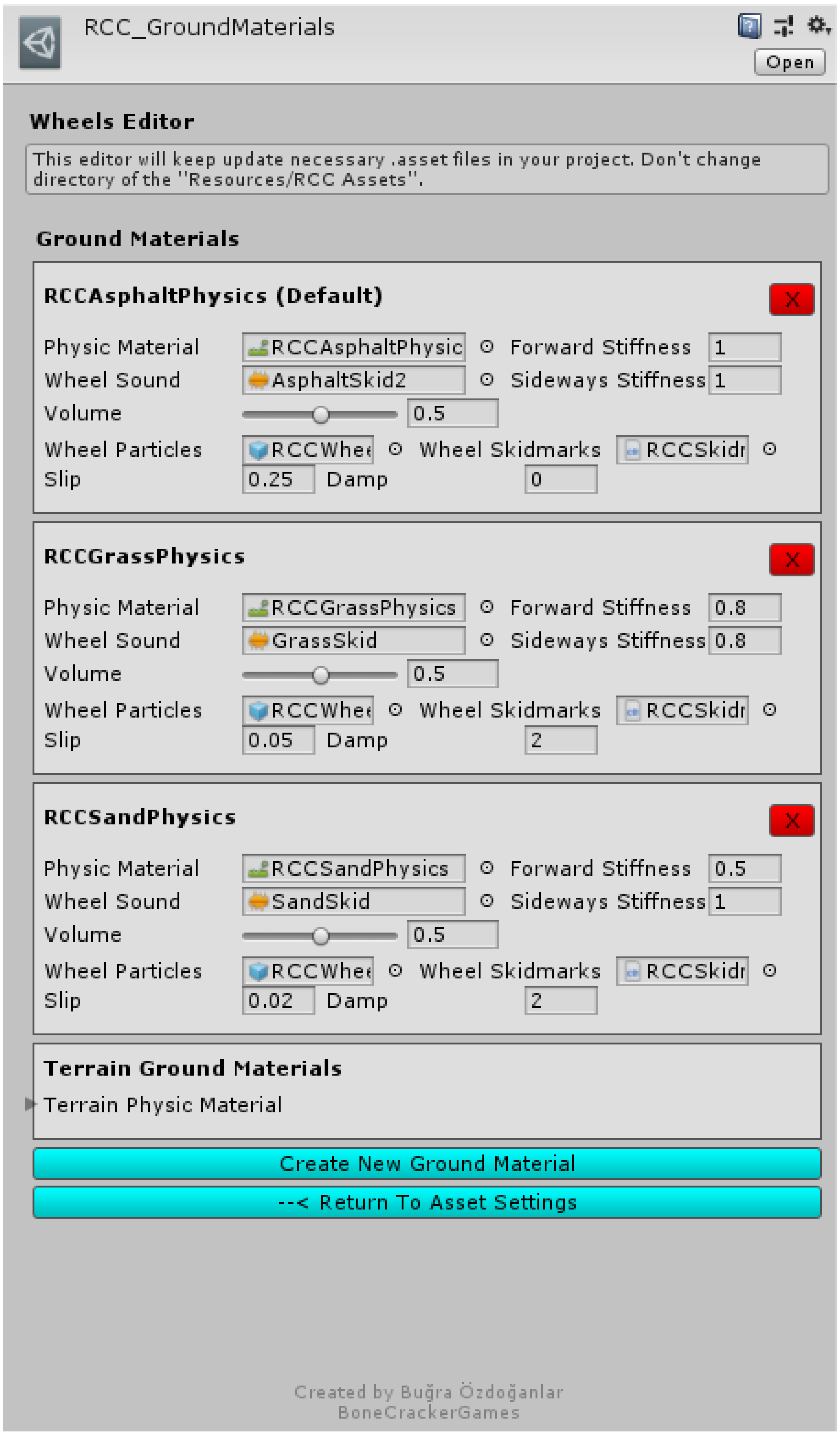
Open PDF Documentation

Realistic Car Controller V3.0f
BoneCrackerGames

Created by Buğra Özdoğanlar

Configurable Ground Materials

Changing or adding new ground materials physics, particles, damps, sounds, etc in **Tools → BoneCracker Games → Realistic Car Controller → Configure Ground Materials**.



RCC_GroundMaterials

Open

Wheels Editor

This editor will keep update necessary .asset files in your project. Don't change directory of the "Resources/RCC Assets".

Ground Materials

RCCAsphaltPhysics (Default)

Physic Material	RCCAsphaltPhysic	Forward Stiffness	1
Wheel Sound	AsphaltSkid2	Sideways Stiffness	1
Volume	0.5		
Wheel Particles	RCCWhee	Wheel Skidmarks	RCCSkidr
Slip	0.25	Damp	0

RCCGrassPhysics

Physic Material	RCCGrassPhysics	Forward Stiffness	0.8
Wheel Sound	GrassSkid	Sideways Stiffness	0.8
Volume	0.5		
Wheel Particles	RCCWhee	Wheel Skidmarks	RCCSkidr
Slip	0.05	Damp	2

RCCSandPhysics

Physic Material	RCCSandPhysics	Forward Stiffness	0.5
Wheel Sound	SandSkid	Sideways Stiffness	1
Volume	0.5		
Wheel Particles	RCCWhee	Wheel Skidmarks	RCCSkidr
Slip	0.02	Damp	2

Terrain Ground Materials

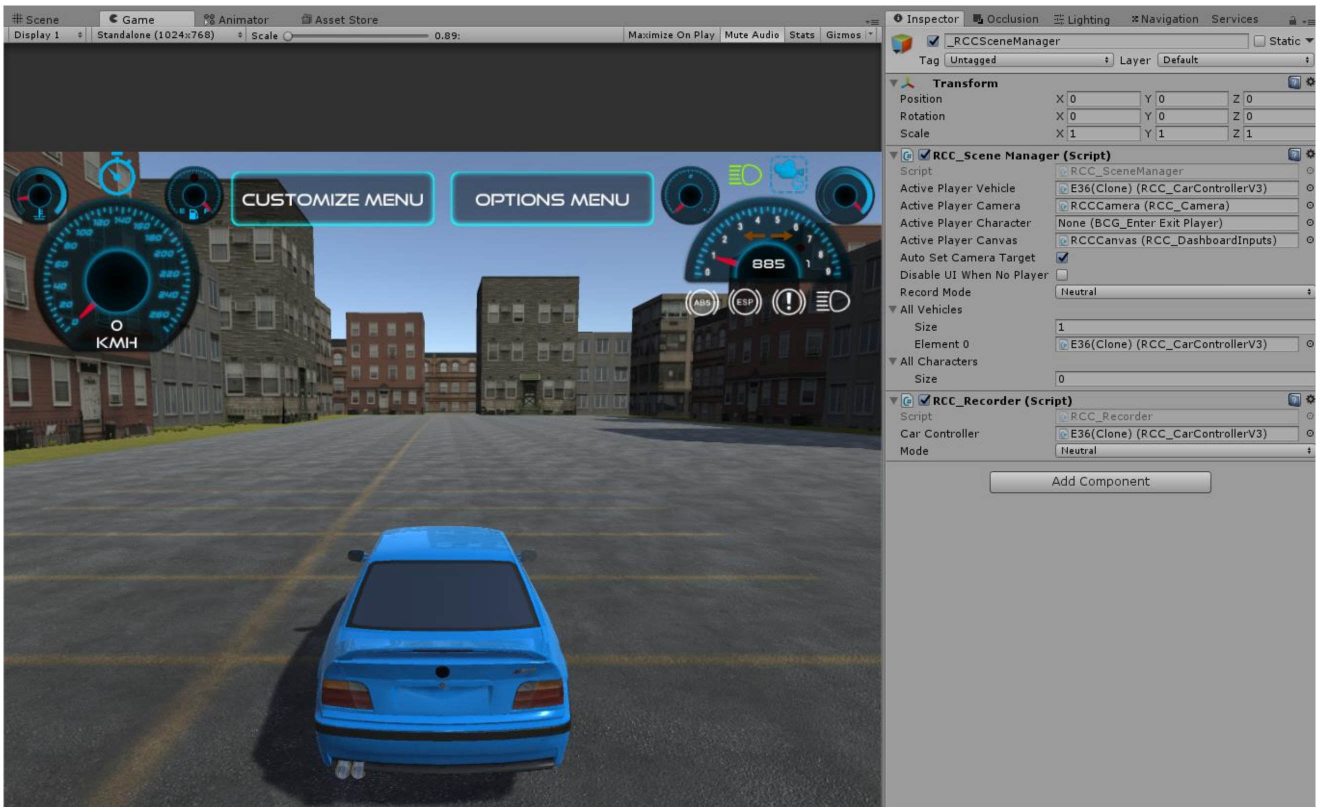
Terrain Physic Material

X
Create New Ground Material
--< Return To Asset Settings
Created by Buğra Özdoğanlar
BoneCrackerGames

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

RCC Scene Manager

Every scene will have this manager. **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.



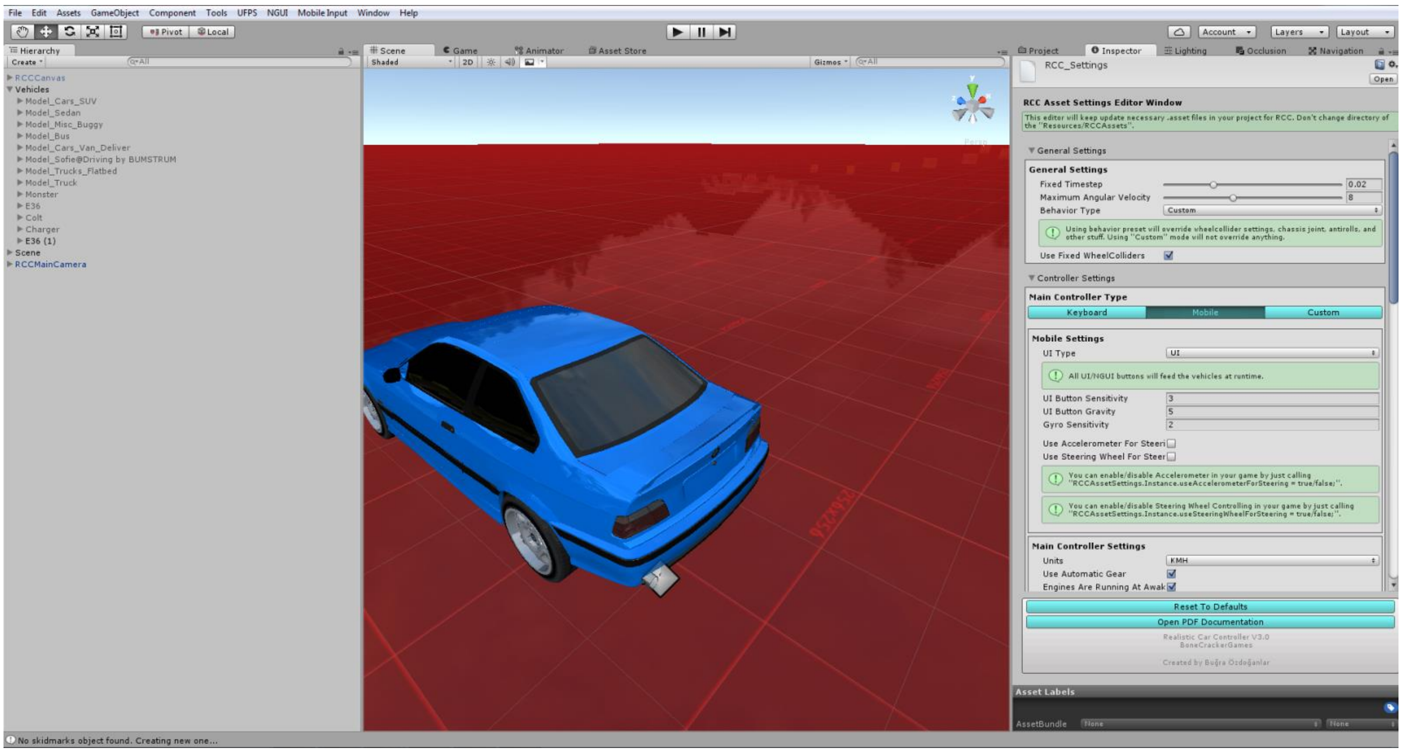
_RCCSceneManager	☐ Static
Tag	Untagged
Layer	Default
Transform	
Position	X 0 Y 0 Z 0
Rotation	X 0 Y 0 Z 0
Scale	X 1 Y 1 Z 1
RCC_Scene Manager (Script)	
Script	RCC_SceneManager
Active Player Vehicle	E36(Clone) (RCC_CarControllerV3)
Active Player Camera	RCCCamera (RCC_Camera)
Active Player Character	None (BCG_Enter Exit Player)
Active Player Canvas	RCCCanvas (RCC_DashboardInputs)
Auto Set Camera Target	✓
Disable UI When No Player	☐
Record Mode	Neutral
All Vehicles	
Size	1
Element 0	E36(Clone) (RCC_CarControllerV3)
All Characters	
Size	0
RCC_Recorder (Script)	
Script	RCC_Recorder
Car Controller	E36(Clone) (RCC_CarControllerV3)
Mode	Neutral

Main Controllers

RCC supports 3 controller types ready to use. Keyboard, Mobile, and Xbox. Each controller has unique settings in RCC Settings. Also you can use your own inputs in Custom mode.

Mobile Controller

Package contains **Mobile Controller** ready to use with UI. Now let's make our vehicles can be controller by Mobile Controller. Just open up **RCC Settings** from **Tools → BoneCracker Games → Realistic Car Controller → RCC Settings**. You can change your inputs for **Keyboard**, **Mobile Controllers**, **Xbox**, and **Custom Controllers** here.



RCC_Settings

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCCAssets".

General Settings

General Settings

Fixed Timestep 0.02
Maximum Angular Velocity 8
Behavior Type Custom

Using behavior preset will override wheelcollider settings, chassis joint, antirolls, and other stuff. Using "Custom" mode will not override anything.

Use Fixed WheelColliders ✓

Controller Settings

Main Controller Type

Keyboard [✓ Mobile] Custom

Mobile Settings

UI Type UI

All UI/NGUI buttons will feed the vehicles at runtime.

UI Button Sensitivity 3
UI Button Gravity 5
Gyro Sensitivity 2

Use Accelerometer For Steeri ☐
Use Steering Wheel For Steer ☐

You can enable/disable Accelerometer in your game by just calling "RCCAssetSettings.Instance.useAccelerometerForSteering = true/false;".

You can enable/disable Steering Wheel Controlling in your game by just calling "RCCAssetSettings.Instance.useSteeringWheelForSteering = true/false;".

Main Controller Settings

Units KMH
Use Automatic Gear ✓
Engines Are Running At Awak ✓

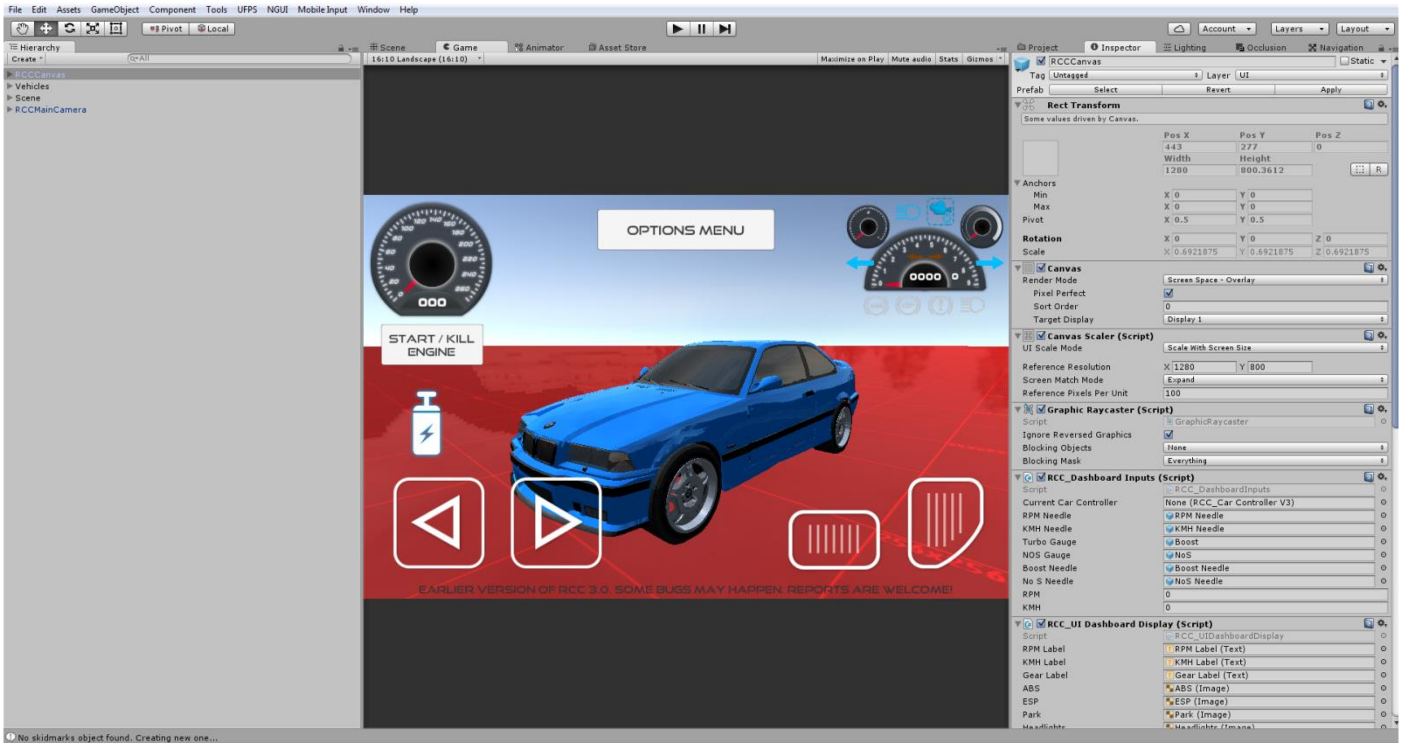
Realistic Car Controller V3.0
BoneCrackerGames

Created by Buğra Özdoğanlar

AssetBundle None None

No skidmarks object found. Creating new one...

Default Mobile Controller type is "**UI Controller**". You will find "**RCCCanvas**" prefab under **RealisticCarControllerV3/Prefabs**. Drag and drop to your Hierarchy.



RCCCanvas			Static
Tag	Untagged	Layer	UI
Prefab	Select	Revert	Apply

Rect Transform

Some values driven by Canvas.

	Pos X	Pos Y	Pos Z
	443	277	0
	Width	Height	
	1280	800.3612	
Anchors			
Min	X 0	Y 0	
Max	X 0	Y 0	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 0.6921875	Y 0.6921875	Z 0.6921875

Canvas

Render Mode	Screen Space - Overlay
Pixel Perfect	✓
Sort Order	0
Target Display	Display 1

Canvas Scaler (Script)

UI Scale Mode	Scale With Screen Size
Reference Resolution	X 1280 Y 800
Screen Match Mode	Expand
Reference Pixels Per Unit	100

Graphic Raycaster (Script)

Script	GraphicRaycaster
Ignore Reversed Graphics	✓
Blocking Objects	None
Blocking Mask	Everything

RCC_Dashboard Inputs (Script)

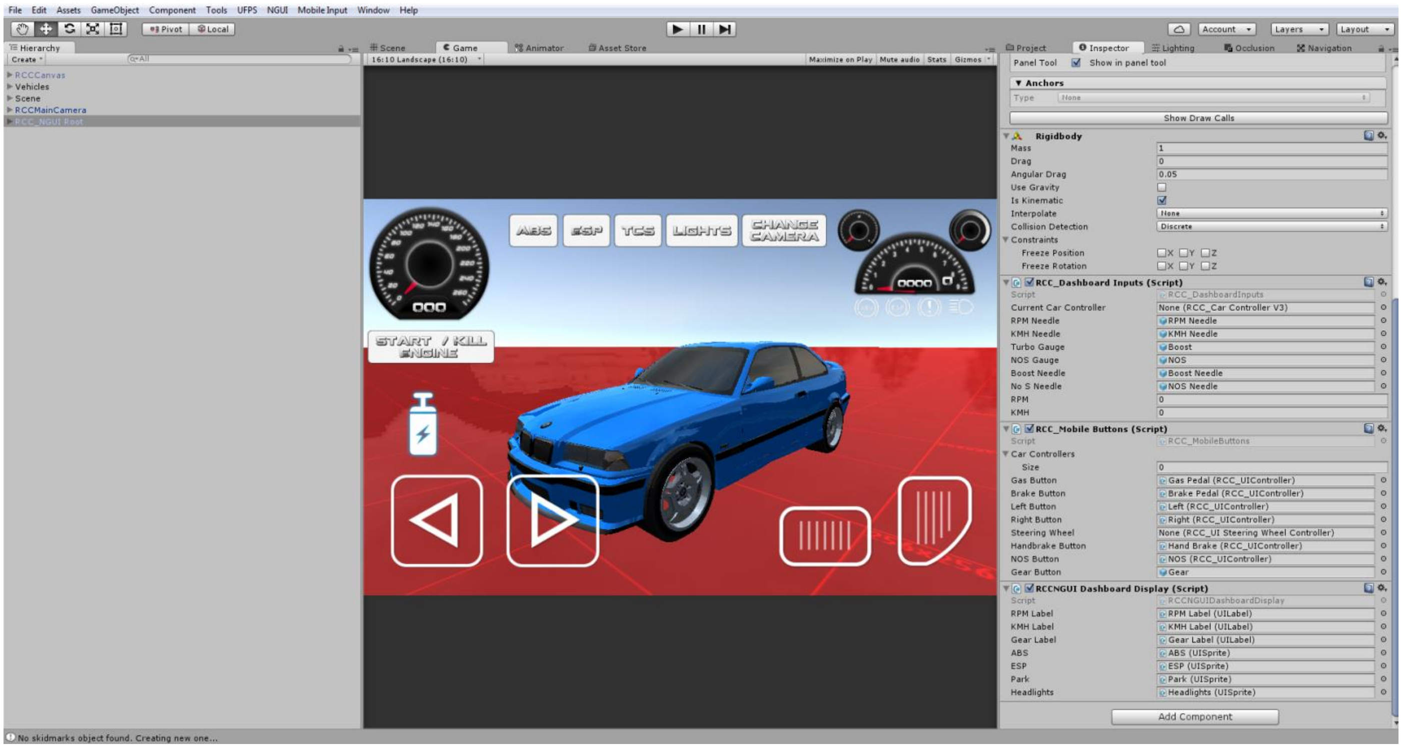
Script	RCC_DashboardInputs
Current Car Controller	None (RCC_Car Controller V3)
RPM Needle	RPM Needle
KMH Needle	KMH Needle
Turbo Gauge	Boost
NOS Gauge	NoS
Boost Needle	Boost Needle
No S Needle	NoS Needle
RPM	0
KMH	0

RCC_UI Dashboard Display (Script)

Script	RCC_UIDashboardDisplay
RPM Label	RPM Label (Text)
KMH Label	KMH Label (Text)
Gear Label	Gear Label (Text)
ABS	ABS (Image)
ESP	ESP (Image)
Park	Park (Image)
Headlights	Headlights (Image)

No skidmarks object found. Creating new one...

Each UI controller gameobject has "**RCC_UIController.cs**" script for inputs. These buttons feeds car controller with normalized float values. You can adjust UI buttons sensitivity and gravity from RCC Settings



Panel Tool	✓ Show in panel tool

Anchors

Type	None

Show Draw Calls

Rigidbody

Mass	1
Drag	0
Angular Drag	0.05
Use Gravity	☐
Is Kinematic	✓
Interpolate	None
Collision Detection	Discrete
Constraints	
Freeze Position	☐X ☐Y ☐Z
Freeze Rotation	☐X ☐Y ☐Z

RCC_Dashboard Inputs (Script)

Script	RCC_DashboardInputs
Current Car Controller	None (RCC_Car Controller V3)
RPM Needle	RPM Needle
KMH Needle	KMH Needle
Turbo Gauge	Boost
NOS Gauge	NOS
Boost Needle	Boost Needle
No S Needle	NOS Needle
RPM	0
KMH	0

RCC_Mobile Buttons (Script)

Script	RCC_MobileButtons
Car Controllers	
Size	0
Gas Button	Gas Pedal (RCC_UIController)
Brake Button	Brake Pedal (RCC_UIController)
Left Button	Left (RCC_UIController)
Right Button	Right (RCC_UIController)
Steering Wheel	None (RCC_UI Steering Wheel Controller)
Handbrake Button	Hand Brake (RCC_UIController)
NOS Button	NOS (RCC_UIController)
Gear Button	Gear

RCCNGUI Dashboard Display (Script)

Script	RCCNGUIDashboardDisplay
RPM Label	RPM Label (UILabel)
KMH Label	KMH Label (UILabel)
Gear Label	Gear Label (UILabel)
ABS	ABS (UISprite)
ESP	ESP (UISprite)
Park	Park (UISprite)
Headlights	Headlights (UISprite)

Add Component

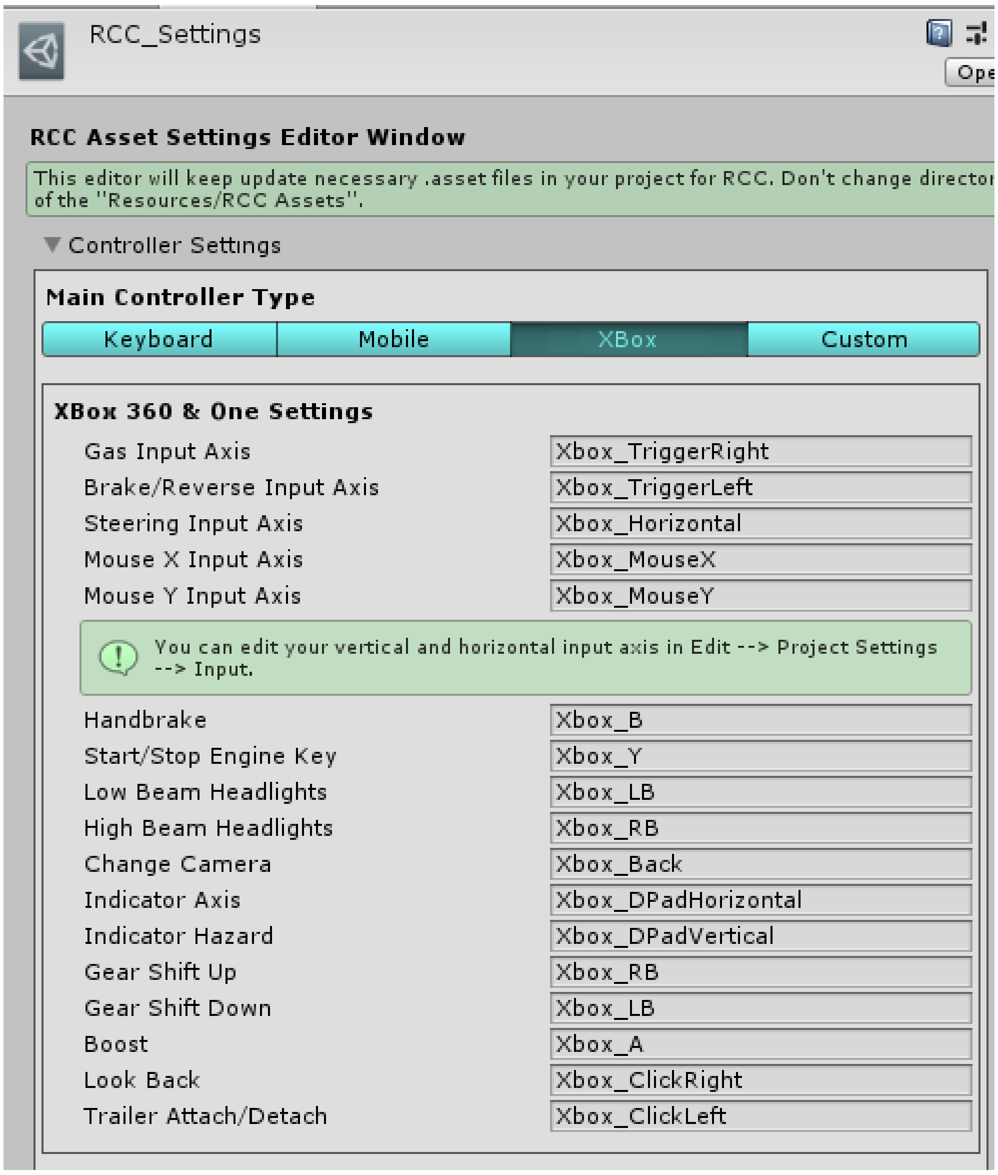
No skidmarks object found. Creating new one...

About Mobile Usement On City Scene

City scene has lot of specular maps with alpha channels. Textures with alpha channels are heavy for mobile devices. In Demo APK from my website is not using any texture with alpha channels. Also all standard shaders are replaced with mobile shaders in **RCC City Mobile** scene. If you build an APK without editing materials, you will get performance loss.

Xbox Controller

This version comes up with Xbox controller support. You can configure map from **Tools → BCG → RCC → Edit Settings**. Select "**Xbox**" controller, you will be able to edit them.



RCC_Settings

Ope

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change director of the "Resources/RCC Assets".

▼ Controller Settings

Main Controller Type

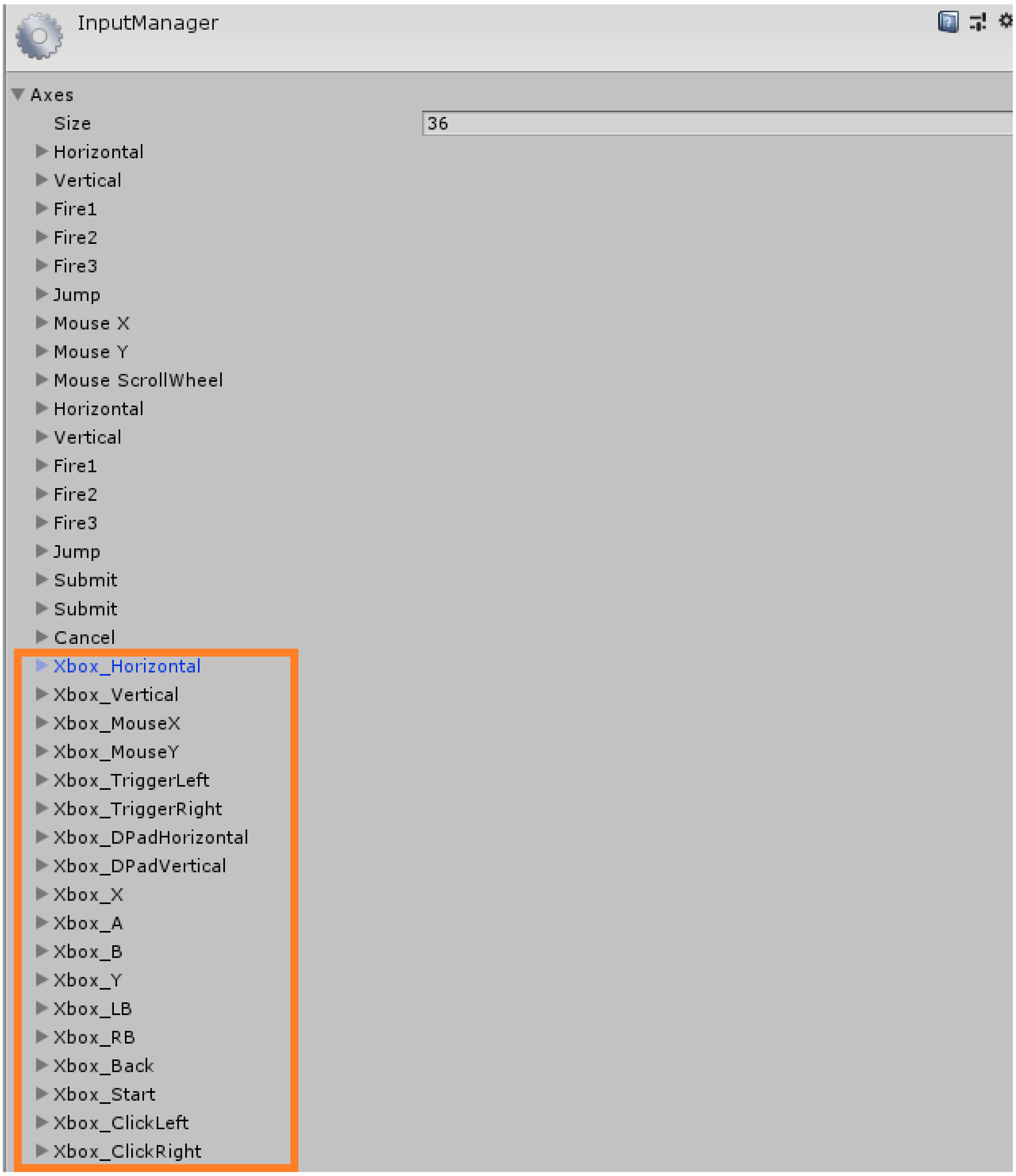
XBox 360 & One Settings

Gas Input Axis	Xbox_TriggerRight
Brake/Reverse Input Axis	Xbox_TriggerLeft
Steering Input Axis	Xbox_Horizontal
Mouse X Input Axis	Xbox_MouseX
Mouse Y Input Axis	Xbox_MouseY

You can edit your vertical and horizontal input axis in Edit --> Project Settings --> Input.

Handbrake	Xbox_B
Start/Stop Engine Key	Xbox_Y
Low Beam Headlights	Xbox_LB
High Beam Headlights	Xbox_RB
Change Camera	Xbox_Back
Indicator Axis	Xbox_DPadHorizontal
Indicator Hazard	Xbox_DPadVertical
Gear Shift Up	Xbox_RB
Gear Shift Down	Xbox_LB
Boost	Xbox_A
Look Back	Xbox_ClickRight
Trailer Attach/Detach	Xbox_ClickLeft

Xbox inputs are using Unity's Input Manager class. Therefore, all proper Xbox controller inputs must be typed in correctly. You have to do it one by one in Input Manager. But RCC comes with **InputManager.asset** file ready to use it. You will find "**InputManager.rar**" file in **RealisticCarControllerV3/For Xbox** folder. Extract it into "**Project Settings**" of your project. Overwrite when asked. If your project hasn't configured InputManager, overwrite it.



InputManager

Axes

Size 36

Horizontal
Vertical
Fire1
Fire2
Fire3
Jump
Mouse X
Mouse Y
Mouse ScrollWheel
Horizontal
Vertical
Fire1
Fire2
Fire3
Jump
Submit
Submit
Cancel
Xbox_Horizontal
Xbox_Vertical
Xbox_MouseX
Xbox_MouseY
Xbox_TriggerLeft
Xbox_TriggerRight
Xbox_DPadHorizontal
Xbox_DPadVertical
Xbox_X
Xbox_A
Xbox_B
Xbox_Y
Xbox_LB
Xbox_RB
Xbox_Back
Xbox_Start
Xbox_ClickLeft
Xbox_ClickRight

You will get this preset when you have imported it.

Creating New Vehicles

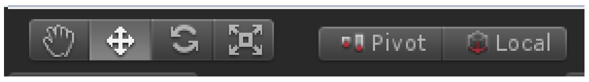
Some developers struggling with creating new vehicles. So, i have improved some editor scripts for simplify creating new vehicles.

Warning

Script and behavior depends on vehicle **X**, **Y**, **Z** directions and pivots. So, your vehicle model and wheel models transform directions ~~**should**~~ MUST be correct. Just check the demo vehicles.

Important

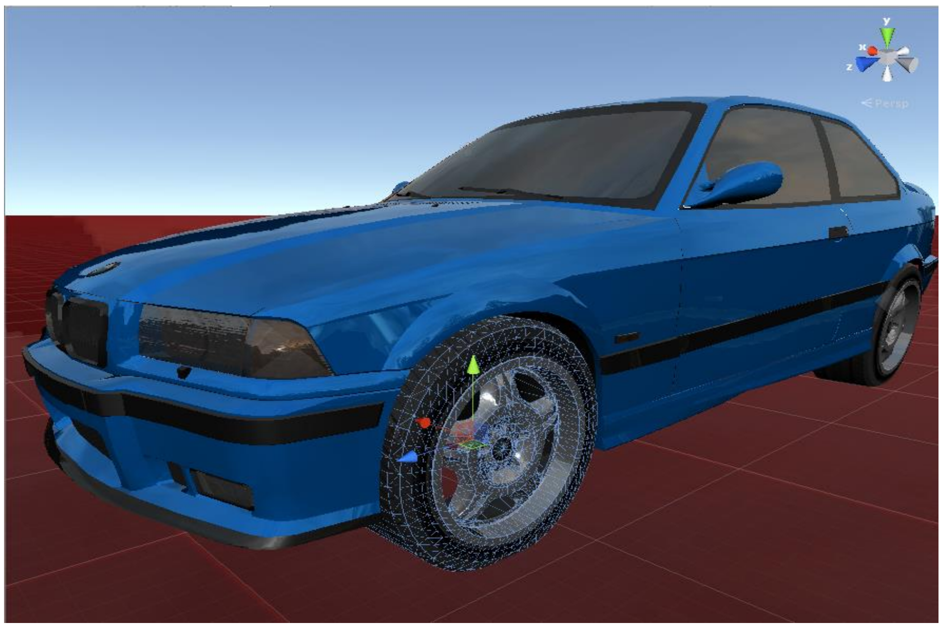
Be sure you are in **PIVOT** and **LOCAL** mode while checking directions.



X should **Right**,

Y should **Up**,

Z should **Forward**.



Many designers are making models with wrong directions or wrong pivot positions or both of them. This is really painful if you don't know how to fix models directions. 2 ways for fixing this;

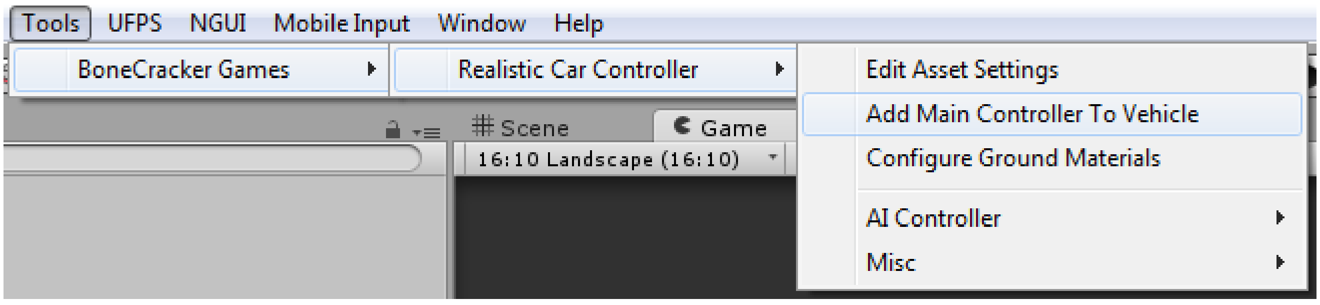
1 – Create a empty Gameobject at the center of the model, and rotate the new Gameobject to correct directions. Then parent your model in to this new Gameobject. It's simple.

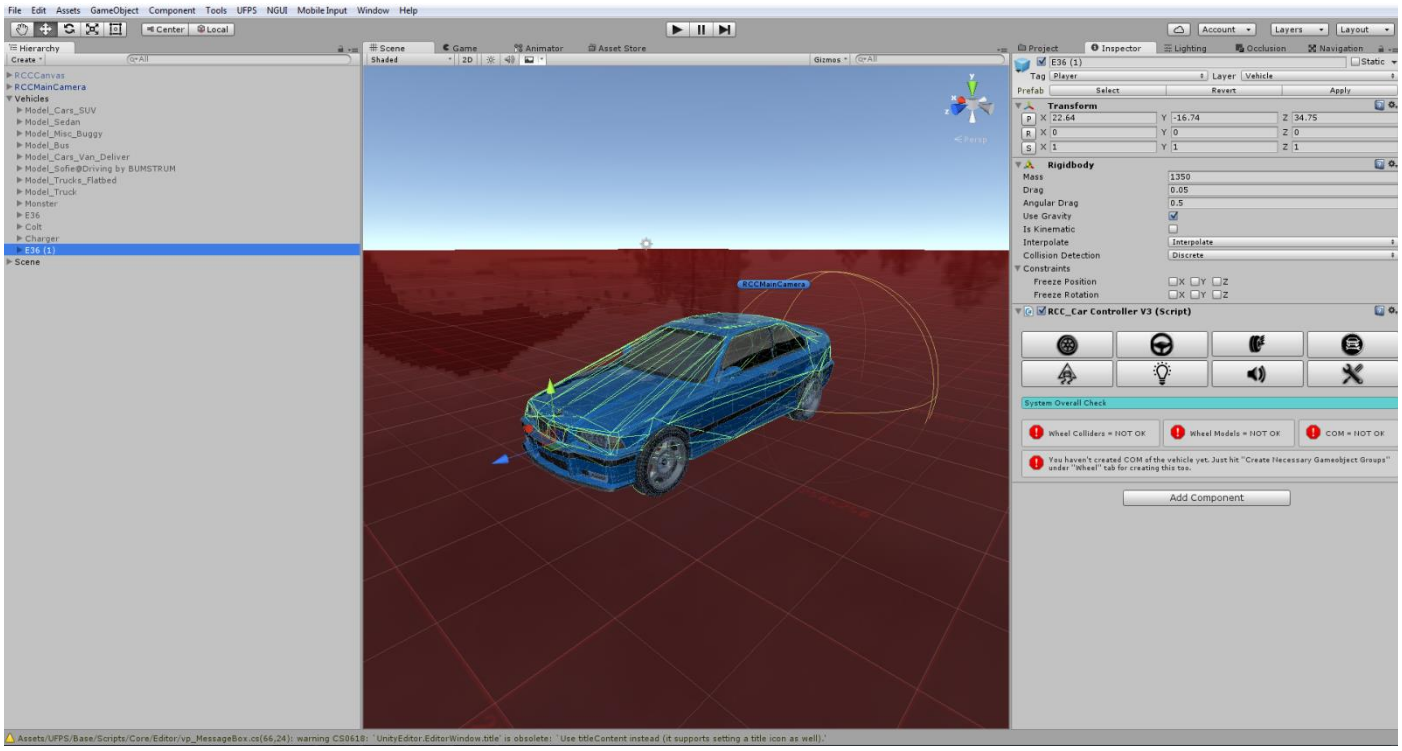
2 – Fix model's directions and pivot positions inside your Designing Software. You can check videos on Youtube for how to do this in 3ds Max.

Also scale of your vehicle must be not oversized or miniature. Even if you want to make a toy car game, car size should be nearly same with demo vehicles. PhysX is calculating size of the collider too. So, you have checked inputs, your vehicle and wheel models pivot positions, and their X, Y, Z directions. Everything is OK right? Then...

Drag and drop your vehicle model to your current scene and let's get started;

You have to add **Main Controller** to root of the vehicle. Just select your vehicle model on your scene, and click **Tools → BoneCracker Games → Realistic Car Controller → Add Main Controller To Vehicle**.





E36 (1) Static
Tag Player Layer Vehicle
Prefab Select Revert Apply

Transform
P X 22.64 Y -16.74 Z 34.75
R X 0 Y 0 Z 0
S X 1 Y 1 Z 1

Rigidbody
Mass 1350
Drag 0.05
Angular Drag 0.5
Use Gravity ✓
Is Kinematic ☐
Interpolate Interpolate
Collision Detection Discrete
Constraints
Freeze Position ☐X ☐Y ☐Z
Freeze Rotation ☐X ☐Y ☐Z

RCC_Car Controller V3 (Script)

System Overall Check

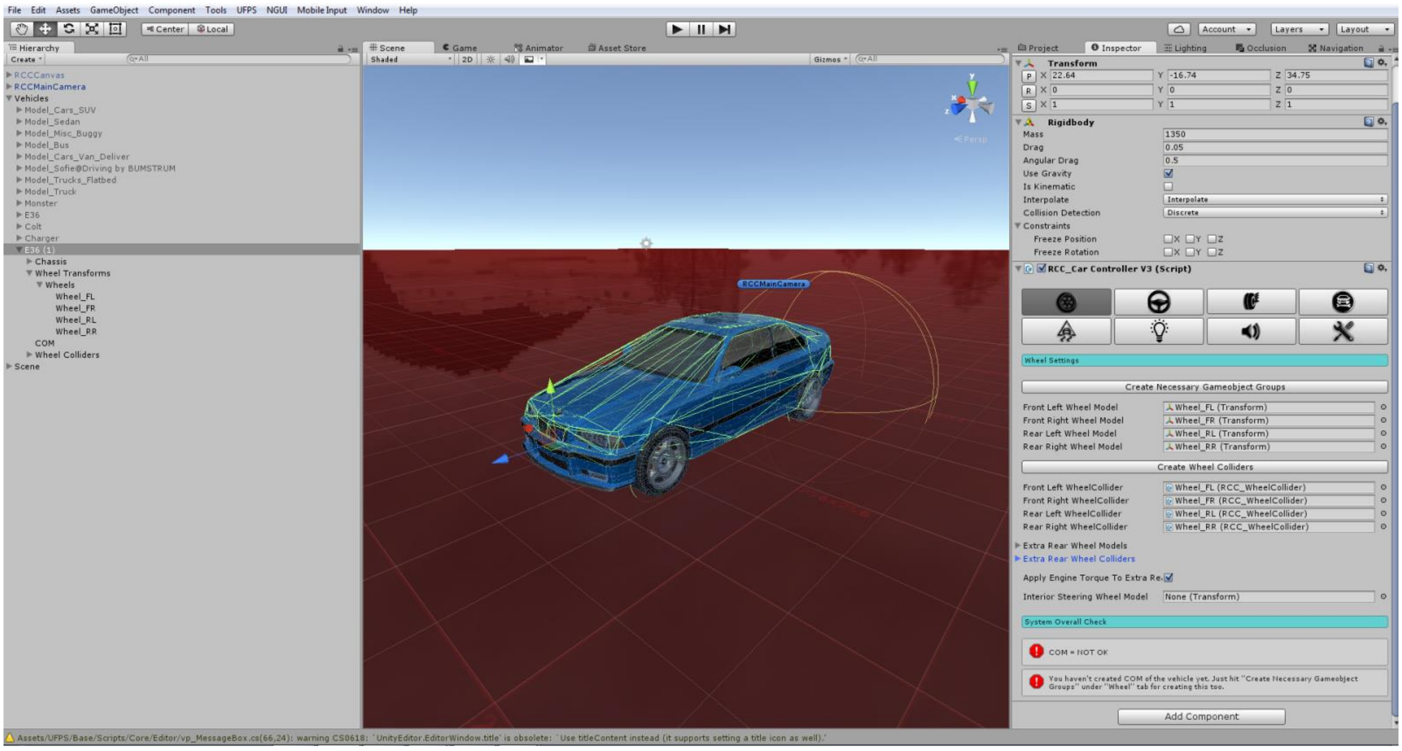
Wheel Colliders = NOT OK
Wheel Models = NOT OK
COM = NOT OK

You haven't created COM of the vehicle yet. Just hit "Create Necessary Gameobject Groups" under "Wheel" tab for creating this too.

```
Assets/UFPS/Base/Scripts/Core/Editor/vp_MessageBox.cs(66,24): warning CS0618: `UnityEditor.EditorWindow.title' is obsolete: `Use titleContent instead (it supports setting a title icon as well).'
```

As soon as when you added **RCC_CarControllerV3.cs** script to your car, Rigidbody component will be added automatically. Set your mass to around 1250-1500 for this type of the vehicles. Interpolate Mode = Interpolate, Angular drag is around 0.1 – 0.25 for medium angular velocity. These rigidbody settings are applied when you create your vehicle automatically.

After rigidbody settings, click “**Wheel**” tab in the editor script. Select all of your wheels. After selecting your wheel models, hit the “**Create Wheel Colliders**” for creating Wheel Colliders with proper radius, suspension, damper, and friction curves automatically.

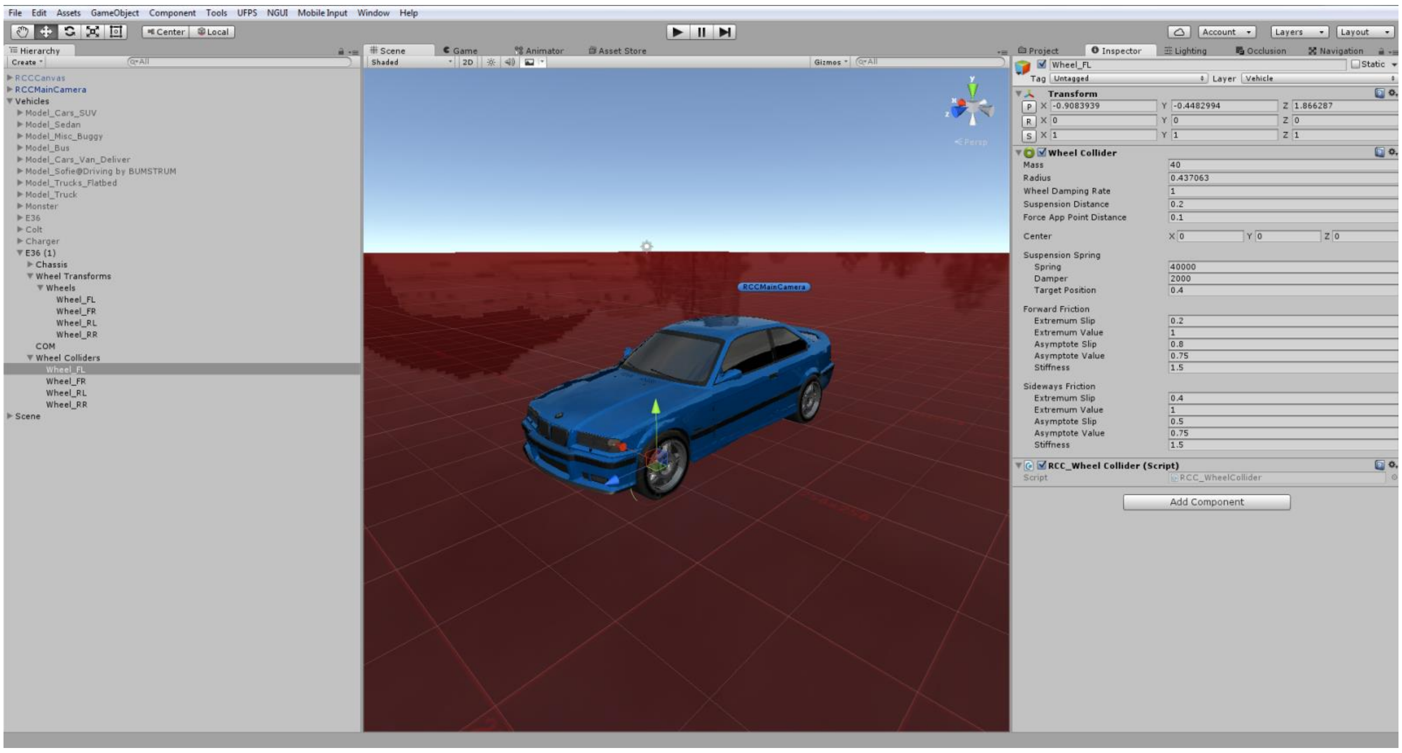


```
Transform
P  X 22.64    Y -16.74   Z 34.75
R  X 0        Y 0        Z 0
S  X 1        Y 1        Z 1

Rigidbody
Mass                 1350
Drag                 0.05
Angular Drag         0.5
Use Gravity          ☑
Is Kinematic         ☐
Interpolate          Interpolate
Collision Detection  Discrete
Constraints
  Freeze Position    ☐X ☐Y ☐Z
  Freeze Rotation    ☐X ☐Y ☐Z

RCC_Car Controller V3 (Script)
Wheel Settings
[Create Necessary Gameobject Groups]
Front Left Wheel Model        Wheel_FL (Transform)
Front Right Wheel Model       Wheel_FR (Transform)
Rear Left Wheel Model         Wheel_RL (Transform)
Rear Right Wheel Model        Wheel_RR (Transform)
[Create Wheel Colliders]
Front Left WheelCollider      Wheel_FL (RCC_WheelCollider)
Front Right WheelCollider     Wheel_FR (RCC_WheelCollider)
Rear Left WheelCollider       Wheel_RL (RCC_WheelCollider)
Rear Right WheelCollider      Wheel_RR (RCC_WheelCollider)
▶ Extra Rear Wheel Models
▶ Extra Rear Wheel Colliders
Apply Engine Torque To Extra Re ☑
Interior Steering Wheel Model None (Transform)
System Overall Check
COM = NOT OK
You haven't created COM of the vehicle yet. Just hit "Create Necessary Gameobject Groups" under "Wheel" tab for creating this too.
[Add Component]
```

```
Assets/UFPS/Base/Scripts/Core/Editor/vp_MessageBox.cs(66,24): warning CS0618: `UnityEditor.EditorWindow.title' is obsolete: `Use titleContent instead (it supports setting a title icon as well).'
```



```
Wheel_FL                      Static
Tag Untagged                  Layer Vehicle

Transform
P  X -0.9083939   Y -0.4482994   Z 1.866287
R  X 0            Y 0            Z 0
S  X 1            Y 1            Z 1

Wheel Collider
Mass                      40
Radius                    0.437063
Wheel Damping Rate        1
Suspension Distance       0.2
Force App Point Distance  0.1
Center                    X 0   Y 0   Z 0
Suspension Spring
  Spring                  40000
  Damper                  2000
  Target Position         0.4
Forward Friction
  Extremum Slip           0.2
  Extremum Value          1
  Asymptote Slip          0.8
  Asymptote Value         0.75
  Stiffness               1.5
Sideways Friction
  Extremum Slip           0.4
  Extremum Value          1
  Asymptote Slip          0.5
  Asymptote Value         0.75
  Stiffness               1.5

RCC_Wheel Collider (Script)
Script                    RCC_WheelCollider
[Add Component]
```

Generated Wheel Colliders settings are fine for 1250-1500 mass vehicles. If you have heavy vehicle such as bus or truck, you must increase Wheel Collider's mass. This value is overrided by **RCC_WheelCollider.cs** right now.

Your vehicle **MUST** have one of any **Colliders** (Such as **Box Collider**, or **Mesh Collider** etc...) Otherwise, physics won't work.

Spawning New Vehicles With Code

You don't have to use ~~GameObject.Instantiate()~~ for spawning new vehicles. You can spawn new vehicles by just one line of code. You can take a look at API documentation.

Spawning New Vehicles With Given Position, Rotation, Sets It's Controllable, And Engine State

You can spawn new vehicles by;

RCC.SpawnRCC(RCC_CarControllerV3 vehiclePrefab, Vector3 position, Quaternion rotation, bool registerAsPlayerVehicle, bool isControllable, bool isEngineRunning
);

As you can see, you can spawn your vehicle with given configuration by only one line of code.

Registering Vehicle As Player Vehicle

You can register the vehicle as player vehicle by;

RCC.RegisterPlayerVehicle(RCC_CarControllerV3 vehicle);

De-Registering Player Vehicle

You can de-register the player vehicle by;

RCC.DeRegisterPlayerVehicle ();

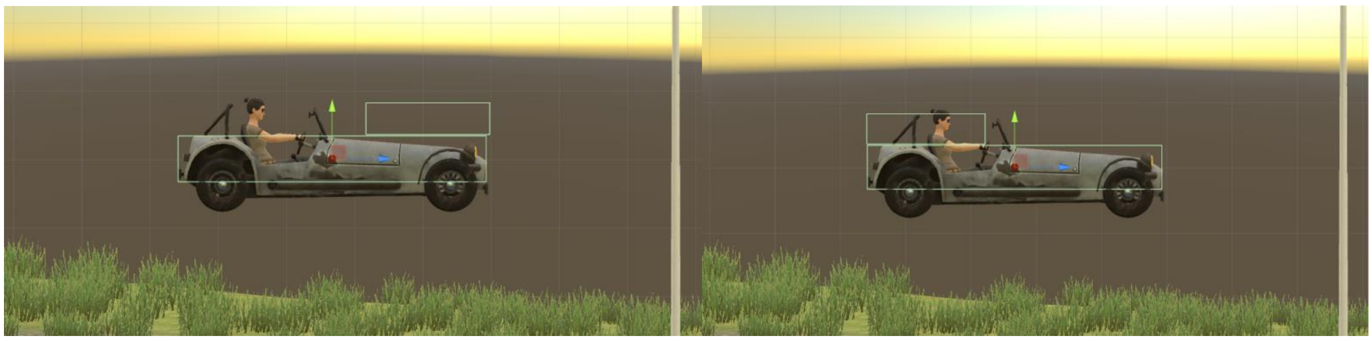
Setting Controllable State Of The Player Vehicle

You can set controllable state of the player vehicle by;

RCC.SetControl(RCC_CarControllerV3 vehicle, bool controlState)

Collider Shapes Of The Vehicle

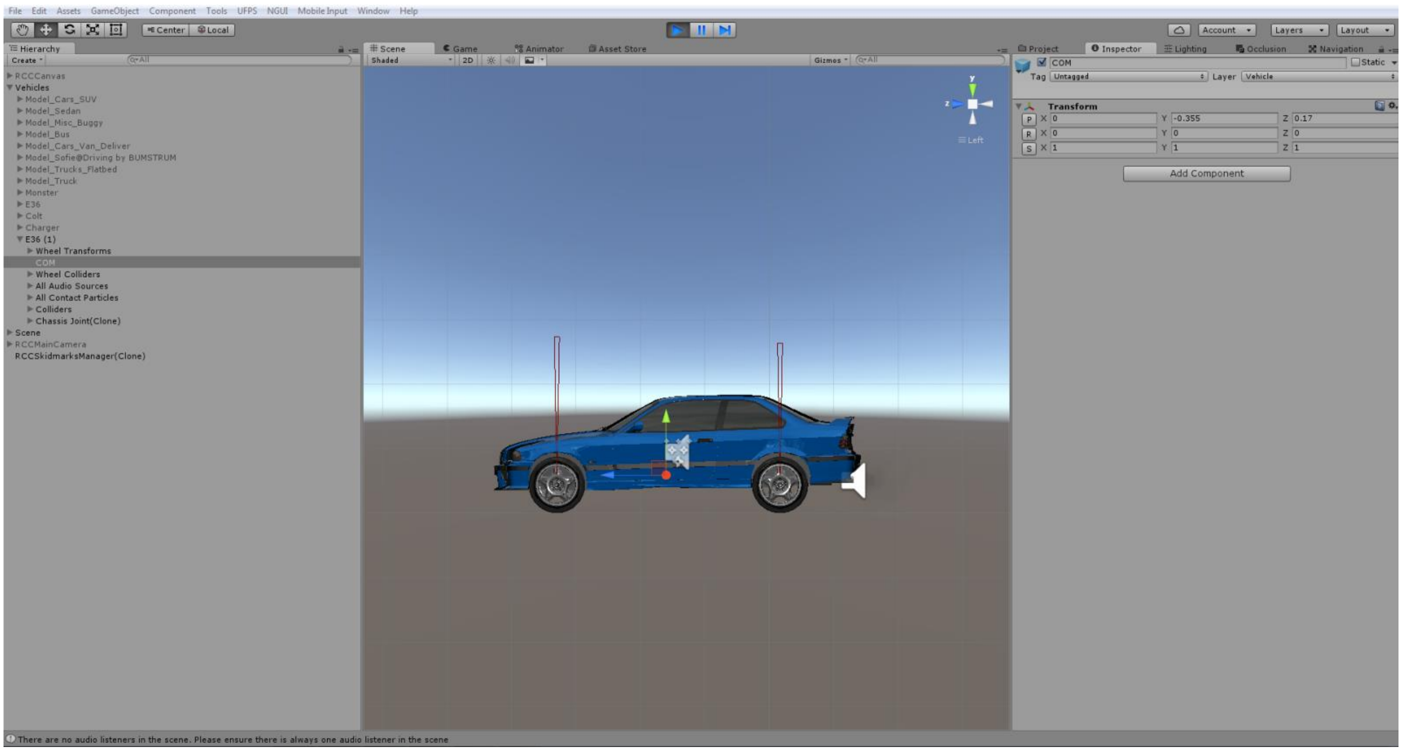
This is one of the most important thing for physics behavior. Most devs are using mesh colliders for their vehicles. Remember that, shape of your vehicle collider will affect physics behavior directly. Let me explain this;



This vehicle has 2 Box Colliders. Second Box Collider is placed at front of the vehicle here on first screenshot. This will bring you more controllable and stable vehicle. Usually, second Box Collider should be at center Z of the vehicle for this model. But this is high speed vehicle, so it must be stable. That's why I did this.

Second Box Collider is placed at rear of the vehicle on second screenshot. This will bring you more unstable and slippy vehicle. Differences between first colliders and second colliders are huge. Just remember that collider shapes are effecting vehicle behavior directly. If you want to make your cars stable, keep in that your mind.

After end up with wheel models and wheel colliders configurations, place your **COM** to correct place. This is our **Center Of Mass**. And COM's position is effecting whole behavior. Usually COM of the vehicle is at just below about front seats. Engine and transmission is at front of the vehicle, and they are heavy. This model is RWD. It has shaft at middle and differential at back of the vehicle. So, I'll just set it to just like this;

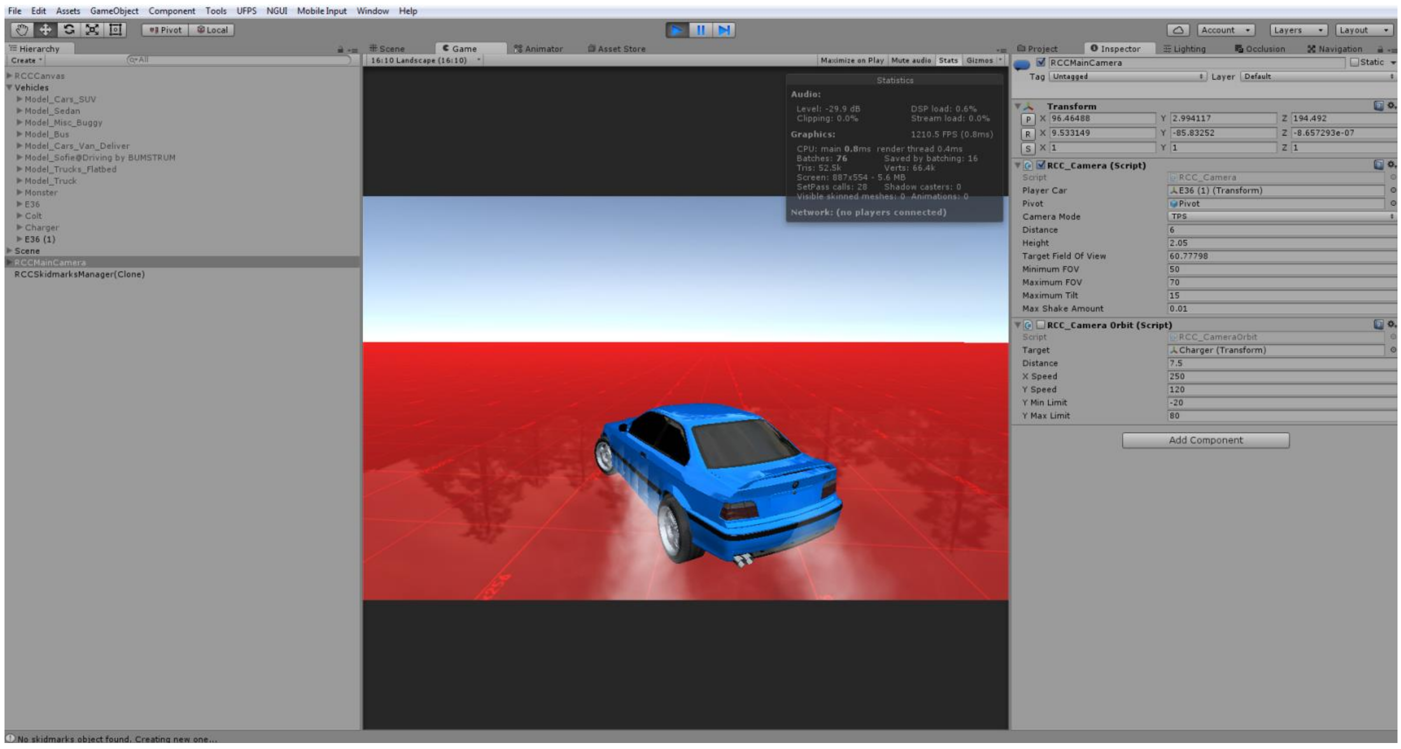


```
RCCCanvas
▼ Vehicles
    Model_Cars_SUV
    Model_Sedan
    Model_Misc_Buggy
    Model_Bus
    Model_Cars_Van_Deliver
    Model_Sofie@Driving by BUMSTRUM
    Model_Trucks_Flatbed
    Model_Truck
    Monster
    E36
    Colt
    Charger
  ▼ E36 (1)
      Wheel Transforms
      COM
      Wheel Colliders
      All Audio Sources
      All Contact Particles
      Colliders
      Chassis Joint(Clone)
Scene
RCCMainCamera
RCCSkidmarksManager(Clone)
```

```
COM                                  Static
Tag Untagged            Layer Vehicle
Transform
P  X 0        Y -0.355    Z 0.17
R  X 0        Y 0         Z 0
S  X 1        Y 1         Z 1
```

```
There are no audio listeners in the scene. Please ensure there is always one audio listener in the scene
```

Runs perfectly after just few clicks.



```
Statistics
Audio:
  Level: -29.9 dB                DSP load: 0.6%
  Clipping: 0.0%                 Stream load: 0.0%
Graphics:                        1210.5 FPS (0.8ms)
  CPU: main 0.8ms  render thread 0.4ms
  Batches: 76            Saved by batching: 16
  Tris: 52.5k            Verts: 66.4k
  Screen: 887x554 - 5.6 MB
  SetPass calls: 28      Shadow casters: 0
  Visible skinned meshes: 0  Animations: 0
Network: (no players connected)
```

```
RCCCanvas
▼ Vehicles
    Model_Cars_SUV
    Model_Sedan
    Model_Misc_Buggy
    Model_Bus
    Model_Cars_Van_Deliver
    Model_Sofie@Driving by BUMSTRUM
    Model_Trucks_Flatbed
    Model_Truck
    Monster
    E36
    Colt
    Charger
    E36 (1)
Scene
RCCMainCamera
RCCSkidmarksManager(Clone)
```

```
RCCMainCamera                        Static
Tag Untagged            Layer Default
Transform
P  X 96.46488   Y 2.994117    Z 194.492
R  X 9.533149   Y -85.83252   Z -8.657293e-07
S  X 1          Y 1           Z 1
RCC_Camera (Script)
Script                 RCC_Camera
Player Car             E36 (1) (Transform)
Pivot                  Pivot
Camera Mode            TPS
Distance               6
Height                 2.05
Target Field Of View   60.77798
Minimum FOV            50
Maximum FOV            70
Maximum Tilt           15
Max Shake Amount       0.01
RCC_Camera Orbit (Script)
Script                 RCC_CameraOrbit
Target                 Charger (Transform)
Distance               7.5
X Speed                250
Y Speed                120
Y Min Limit            -20
Y Max Limit            80
```

```
No skidmarks object found. Creating new one...
```

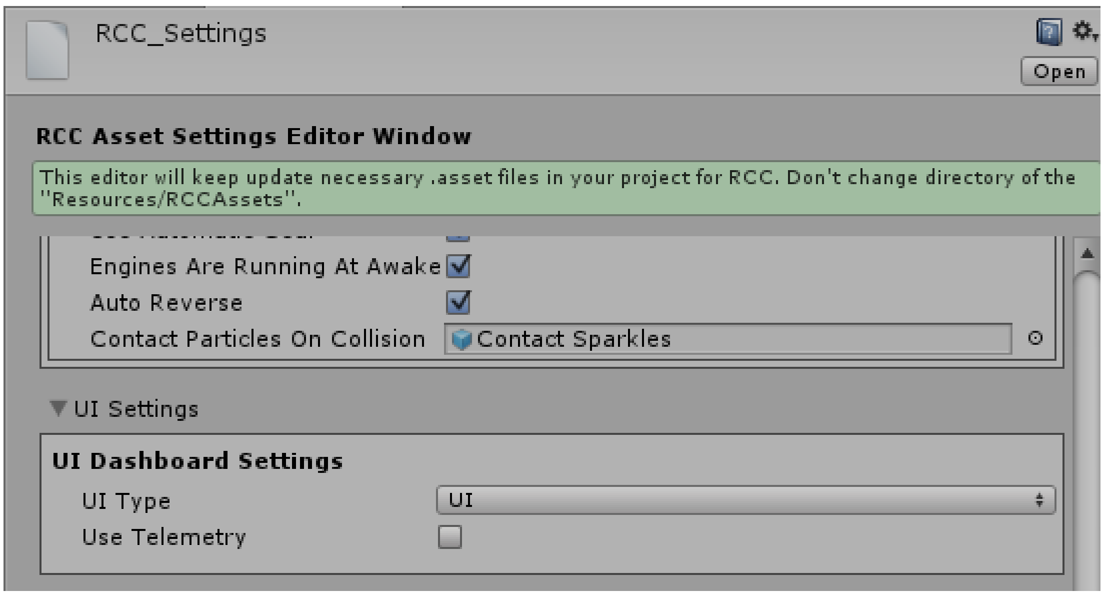
Configurate your vehicle as you wish. If you want to use manual gear, you need to set it from **RCC Settings**. Also you can select which key to shift up and shift down here.

Driving Assistances

Main Controller has **ABS**, **TCS**, **ESP, Steering Assistance, and Traction Assistance**. Threshold means, if wheel slip is equals or higher than this threshold value, corresponding assistance will be engaged immediately. Steering assistance will apply local Y axis torque to vehicle for more easily turns. But results more unrealistic turns like arcade games. 0.1f would be good for all vehicles. (0.2f will be used if behavior type is Racing or Semi-Arcade.)

Dashboard Configuration

You can access dashboard from RCC Settings.



RCC_Settings

Open

RCC Asset Settings Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCCAssets".

Engines Are Running At Awake	✓
Auto Reverse	✓
Contact Particles On Collision	Contact Sparkles

UI Settings

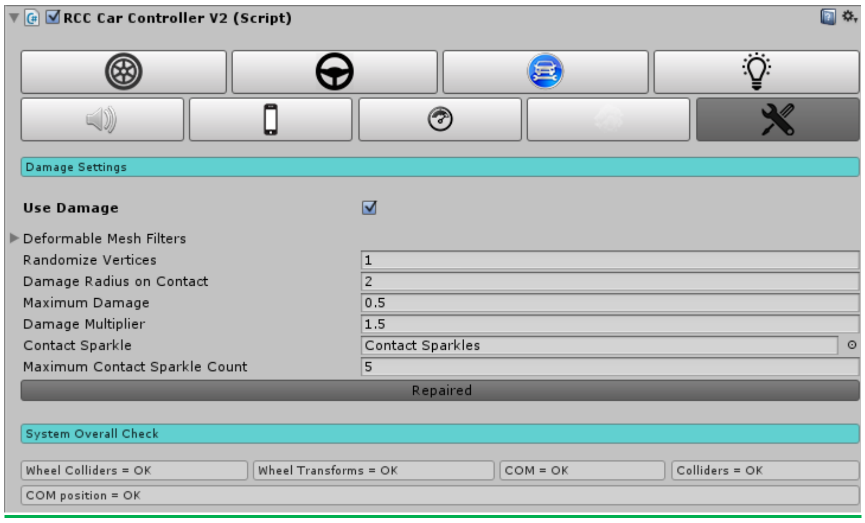
UI Dashboard Settings

UI Type	UI
Use Telemetry	☐

Default Dashboard Type is "**UI**". Script will access "**RCC_DashboardInputs.cs**" and "**RCC_DashboardDisplay.cs**" scripts for displaying dashboard. These scripts are attached on RCCCanvas root.

It's extremely easy to create and customize your own dashboard and controller HUD on Unity UI instead of writing your own legacy OnGUI() method.

Damage (Based on Mesh Deformation)


RCC Car Controller V2 (Script)

Damage Settings

Use Damage	✓
Deformable Mesh Filters	
Randomize Vertices	1
Damage Radius on Contact	2
Maximum Damage	0.5
Damage Multiplier	1.5
Contact Sparkle	Contact Sparkles
Maximum Contact Sparkle Count	5

Repaired

System Overall Check

Wheel Colliders = OK
Wheel Transforms = OK
COM = OK
Colliders = OK
COM position = OK

Your vehicle body mesh wireframe topology must be reliable for realistic vertices movement. If your vehicle body mesh has broken (unwelded) vertices or bad wireframe topology, mesh will deform buggy and unrealisticly. RCC takes all meshes of the vehicle if you haven't select them specifically. If you select specific meshes, RCC won't take all.

Creating Lights, Sounds, Skidmarks, Smoke Effects

These effects are optional.

Create point lights for braking and reverse gear, spot lights for headlights. Place them correctly on your vehicle model. You can create all kind of lights from **Tools → BoneCracker Games → Realistic Car Controller → Create → Lights**.

Script doesn't Instantiate, Destroy any smoke particles, or any kind of stuff. Just enabling/disabling particle emitters to avoid garbage memory.

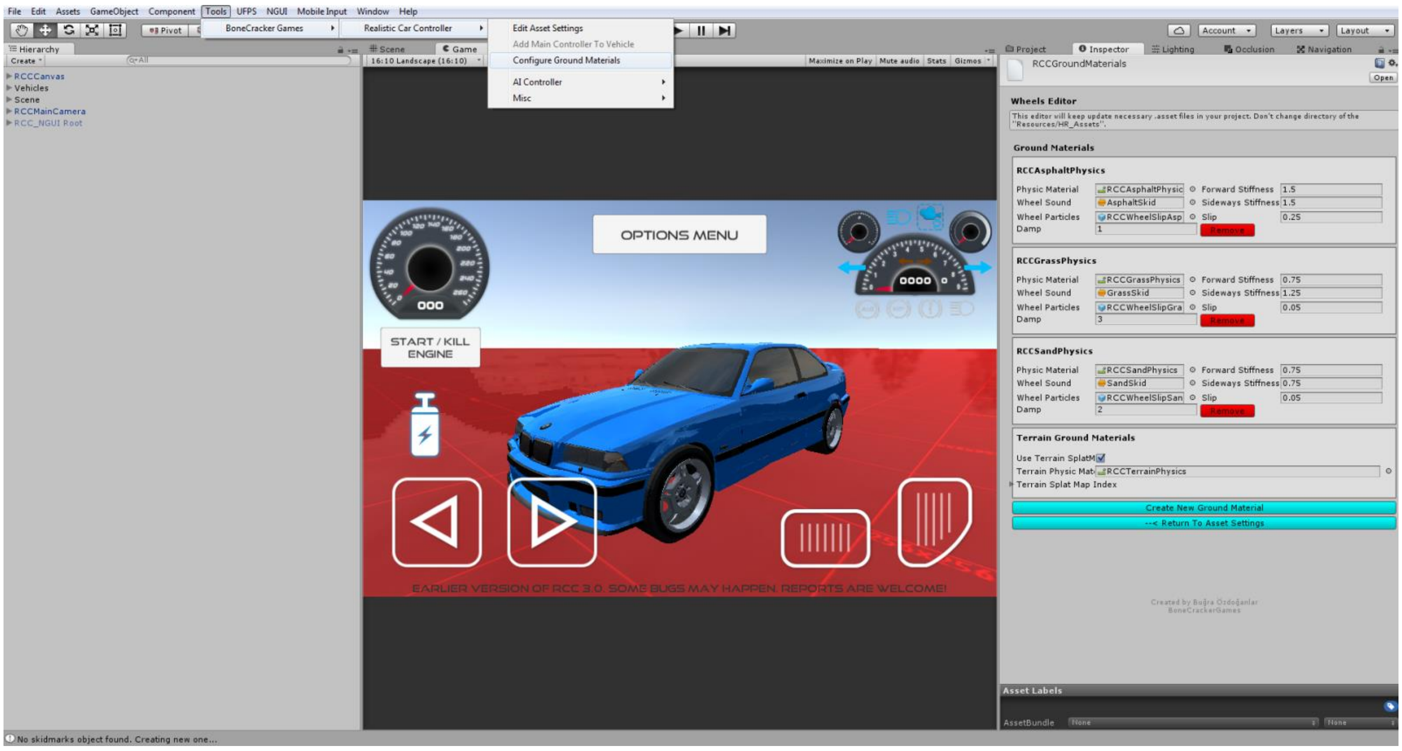
You will find "**RCCWheelSlipAsphalt**", "**RCCWheelSlipGrass**" and "**RCCWheelSlipSand**" prefabs under Prefabs folder. You can use your own smoke prefab as you wish.

If you want to use exhaust effects, you can create it from **Tools → BoneCracker Games → Realistic Car Controller → Create → Misc**. You need to place it to your model correctly. That's it.

You will find "**RCCWheelSkidmarks.cs**" under Scripts folder. Scene must have "**RCCSkidmarksManager**" to create skid meshes. You will find the "**RCCSkidmarksManager**" prefab under Prefabs folder. Drag and drop to your Hierarchy. If your scene doesn't have, RCC_WheelCollider will create itself.

Ground Materials

If you want to get variable tire grips on varied surfaces, you can use this feature. Open up **Tools → BoneCracker Games → Realistic Car Controller → Configure Ground Materials**.



RCCGroundMaterials

Wheels Editor

This editor will keep update necessary .asset files in your project. Don't change directory of the "Resources/HR_Assets".

Ground Materials

RCCAsphaltPhysics

Physic Material	RCCAsphaltPhysic	Forward Stiffness	1.5
Wheel Sound	AsphaltSkid	Sideways Stiffness	1.5
Wheel Particles	RCCWheelSlipAsp	Slip	0.25
Damp	1	Remove	

RCCGrassPhysics

Physic Material	RCCGrassPhysics	Forward Stiffness	0.75
Wheel Sound	GrassSkid	Sideways Stiffness	1.25
Wheel Particles	RCCWheelSlipGra	Slip	0.05
Damp	3	Remove	

RCCSandPhysics

Physic Material	RCCSandPhysics	Forward Stiffness	0.75
Wheel Sound	SandSkid	Sideways Stiffness	0.75
Wheel Particles	RCCWheelSlipSan	Slip	0.05
Damp	2	Remove	

Terrain Ground Materials

Use Terrain SplatM ✓
Terrain Physic Mat RCCTerrainPhysics
Terrain Splat Map Index

Create New Ground Material
--< Return To Asset Settings

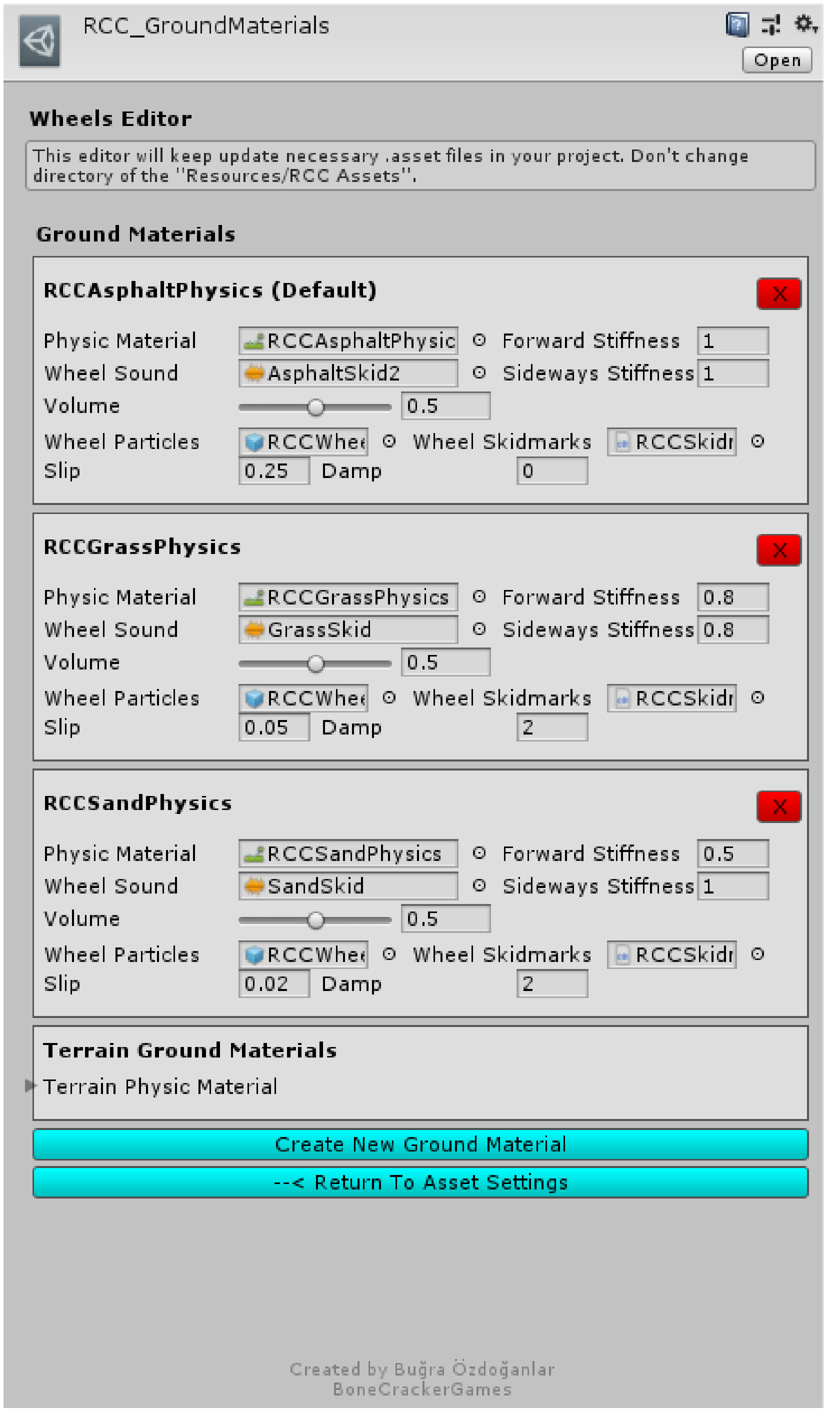
Created by Buğra Özdoğanlar
BoneCrackerGames

No skidmarks object found. Creating new one...

Currently 3 surfaces available such as Asphalt(Default), Sand, and Grass. You will find "**RCCAsphaltPhysics**", "**RCCGrassPhysics**" and "**RCCSandPhysics**" Physic Materials in "**Resources**" folder. If your scene ground is not a Unity Terrain, and made by individual gameobjects, you have to assign each ground gameobject collider's Physic Material to corresponding one. For ex. Select your grass ground gameobject, and select it's collider's Physic Material as "**RCCGrassPhysics**".

Adjusting Ground Particles, Wheel Sounds, Damp, Forward and Sideway Stiffness, Slip On Different Grounds

You can adjust each materials Ground Particles, Wheel Sounds, Damp, Forward and Sideway Stiffness, Slip here.


RCC_GroundMaterials

Open

Wheels Editor

This editor will keep update necessary .asset files in your project. Don't change directory of the "Resources/RCC Assets".

Ground Materials

RCCAsphaltPhysics (Default)

Physic Material	RCCAsphaltPhysic	Forward Stiffness	1
Wheel Sound	AsphaltSkid2	Sideways Stiffness	1
Volume	0.5		
Wheel Particles	RCCWhee	Wheel Skidmarks	RCCSkidı
Slip	0.25	Damp	0

RCCGrassPhysics

Physic Material	RCCGrassPhysics	Forward Stiffness	0.8
Wheel Sound	GrassSkid	Sideways Stiffness	0.8
Volume	0.5		
Wheel Particles	RCCWhee	Wheel Skidmarks	RCCSkidı
Slip	0.05	Damp	2

RCCSandPhysics

Physic Material	RCCSandPhysics	Forward Stiffness	0.5
Wheel Sound	SandSkid	Sideways Stiffness	1
Volume	0.5		
Wheel Particles	RCCWhee	Wheel Skidmarks	RCCSkidı
Slip	0.02	Damp	2

Terrain Ground Materials

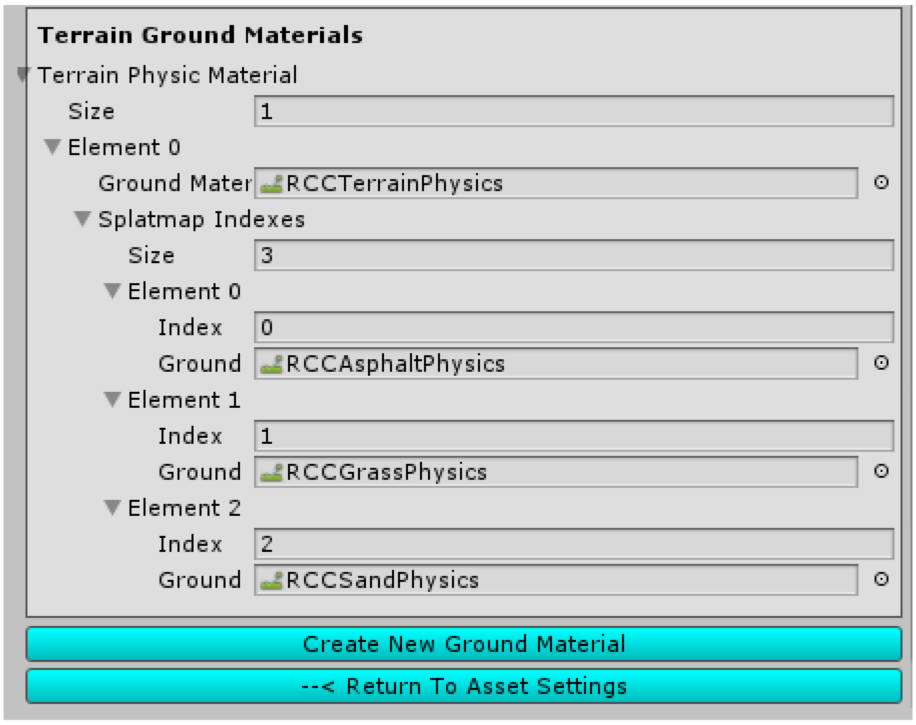
Terrain Physic Material

Create New Ground Material
--< Return To Asset Settings
Created by Buğra Özdoğanlar
BoneCrackerGames

As i said, these are optional effects. If you don’t want to use them, just leave.

Terrain Ground Materials

If your scene has a Unity Terrain as a ground, your terrain textures will decide which surface your on. Open up **Tools → BoneCracker Games → Realistic Car Controller → Configure Ground Materials**. You will be able to configure existing ground materials, remove, or add new ones.



Terrain Ground Materials

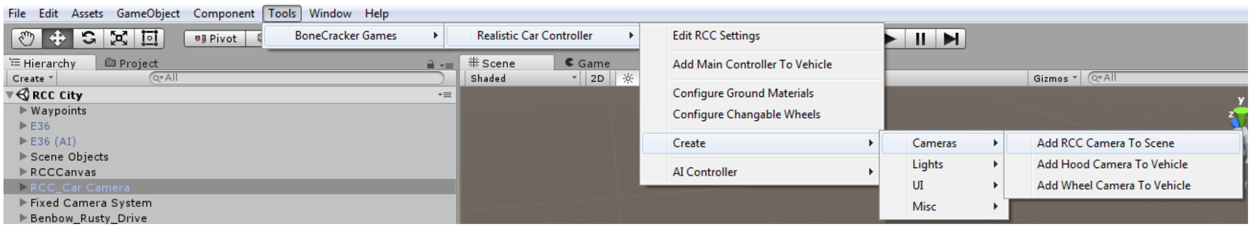
Terrain Physic Material
- Size: 1
- Element 0
  - Ground Mater: RCCTerrainPhysics
  - Splatmap Indexes
    - Size: 3
    - Element 0
      - Index: 0
      - Ground: RCCAsphaltPhysics
    - Element 1
      - Index: 1
      - Ground: RCCGrassPhysics
    - Element 2
      - Index: 2
      - Ground: RCCSandPhysics

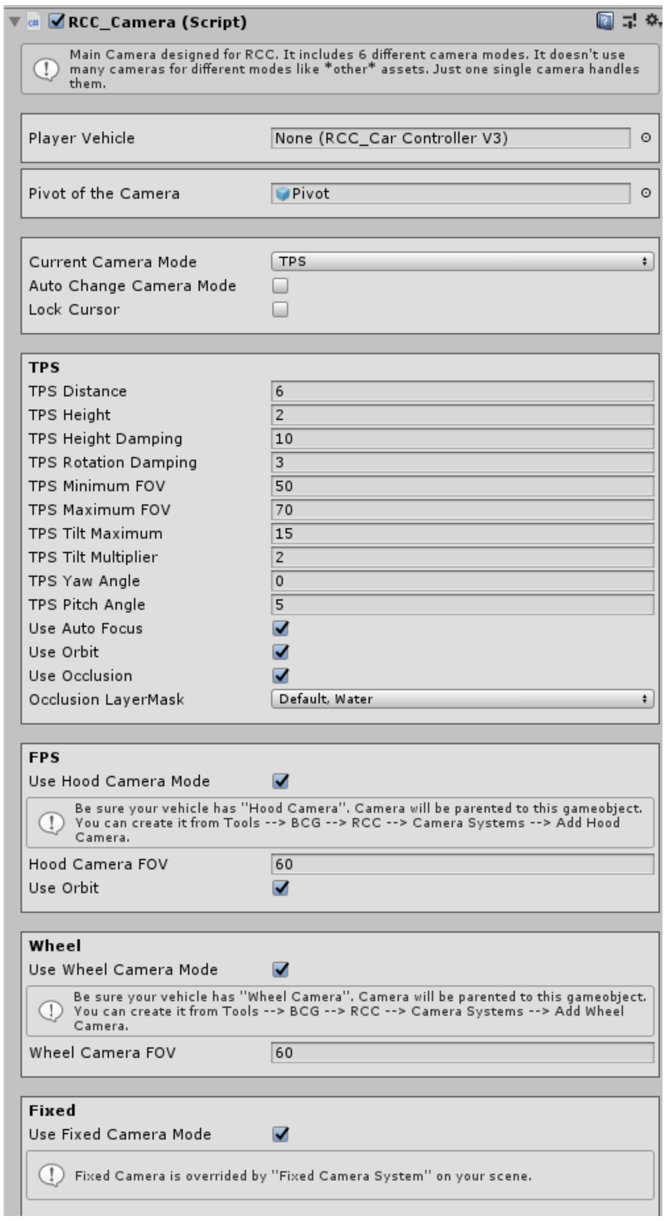
Select each index of your terrain texture slot for corresponding physics material. And yes, supports multiple terrains as well. Just be sure each terrain has unique physic material.

RCC Camera System

Main Camera system desiged for using with RCC. Includes 6 different camera modes with many customizable settings. It doesn't use different individual cameras on your scene like *other assets. Simply it parents the camera to their positions that's all. No need to be Einstein.

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**





RCC_Camera (Script)

Main Camera designed for RCC. It includes 6 different camera modes. It doesn't use many cameras for different modes like *other* assets. Just one single camera handles them.

Player Vehicle	None (RCC_Car Controller V3)
Pivot of the Camera	Pivot
Current Camera Mode	TPS
Auto Change Camera Mode	☐
Lock Cursor	☐

TPS

TPS Distance	6
TPS Height	2
TPS Height Damping	10
TPS Rotation Damping	3
TPS Minimum FOV	50
TPS Maximum FOV	70
TPS Tilt Maximum	15
TPS Tilt Multiplier	2
TPS Yaw Angle	0
TPS Pitch Angle	5
Use Auto Focus	✓
Use Orbit	✓
Use Occlusion	✓
Occlusion LayerMask	Default, Water

FPS

Use Hood Camera Mode ✓

Be sure your vehicle has "Hood Camera". Camera will be parented to this gameobject. You can create it from Tools --> BCG --> RCC --> Camera Systems --> Add Hood Camera.

Hood Camera FOV	60
Use Orbit	✓

Wheel

Use Wheel Camera Mode ✓

Be sure your vehicle has "Wheel Camera". Camera will be parented to this gameobject. You can create it from Tools --> BCG --> RCC --> Camera Systems --> Add Wheel Camera.

Wheel Camera FOV 60

Fixed

Use Fixed Camera Mode ✓

Fixed Camera is overrided by "Fixed Camera System" on your scene.

Record / Replay

Complete physics based record / replay system. Only player vehicle can record / replay. All you have to do is press "**R**" for start recording, and "**P**" for start replay. These keys can be changed in **RCC Settings**. And of course, there is a UI button for mobile.

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

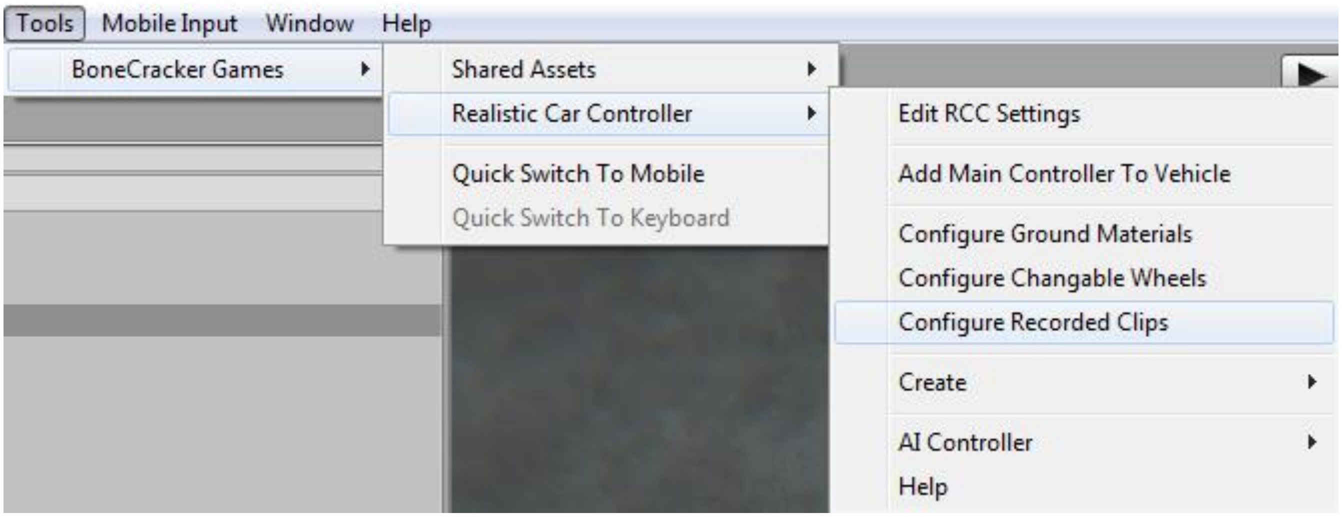
RCC. StartStopReplay ();

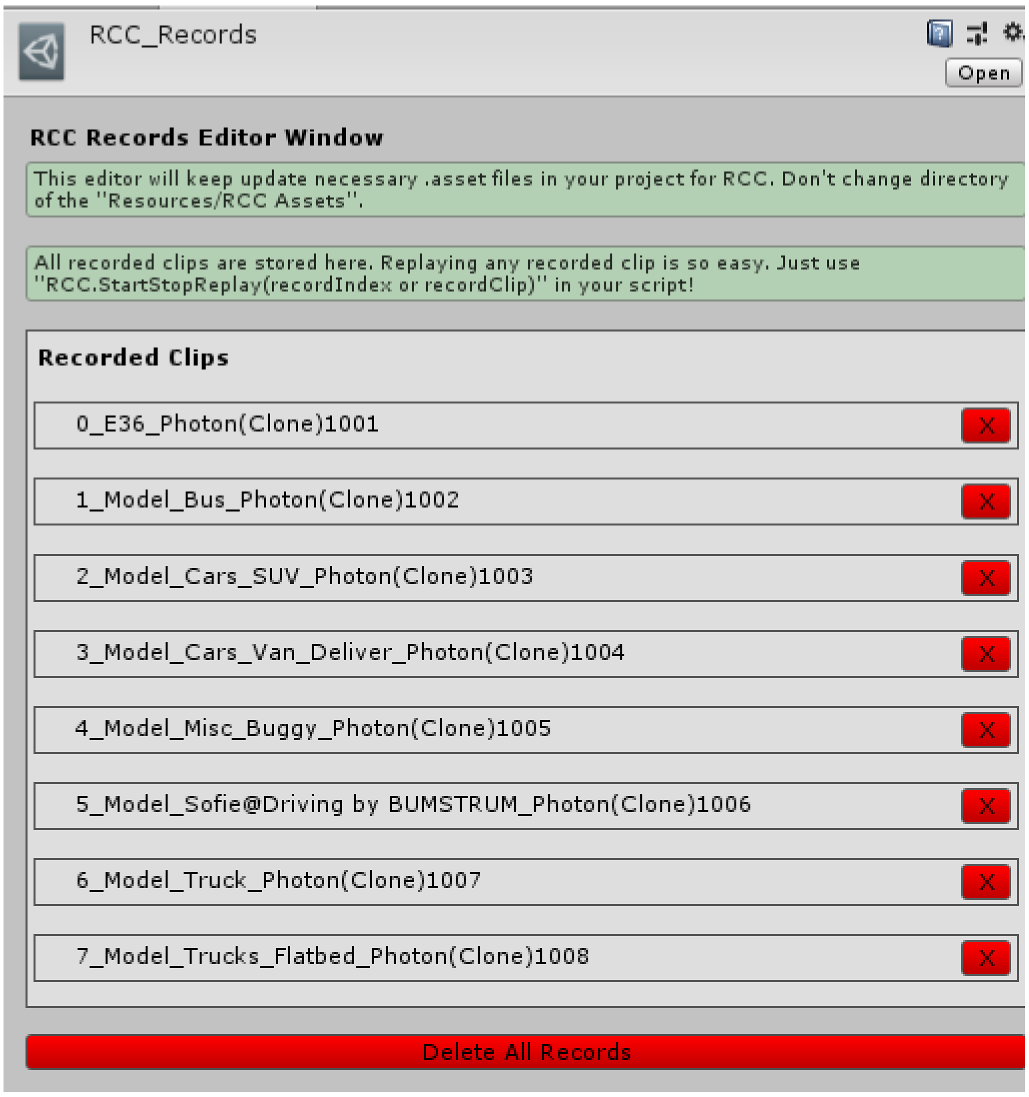
RCC. StartStopReplay (RCC_Recorder.Recorded recordedClip);

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**.




RCC_Records

Open

RCC Records Editor Window

This editor will keep update necessary .asset files in your project for RCC. Don't change directory of the "Resources/RCC Assets".

All recorded clips are stored here. Replaying any recorded clip is so easy. Just use "RCC.StartStopReplay(recordIndex or recordClip)" in your script!

Recorded Clips

- 0_E36_Photon(Clone)1001 X
- 1_Model_Bus_Photon(Clone)1002 X
- 2_Model_Cars_SUV_Photon(Clone)1003 X
- 3_Model_Cars_Van_Deliver_Photon(Clone)1004 X
- 4_Model_Misc_Buggy_Photon(Clone)1005 X
- 5_Model_Sofie@Driving by BUMSTRUM_Photon(Clone)1006 X
- 6_Model_Truck_Photon(Clone)1007 X
- 7_Model_Trucks_Flatbed_Photon(Clone)1008 X

Delete All Records

You can replay any clip on any vehicle by. For ex, you have recorded footage with E36, but you can replay it on bus.

Customization

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

How The Customization Panel Works

I've wrote a example script called "**RCC_CustomizerExample.cs**". Script is attached to RCCCanvas. UI Buttons in Customization Panel sends methods to thix 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 car. So, we have to call it;

RCC_Customization.SetFrontSuspensionsDistances (targetRCC, targetValue);

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

RCC_Customization. RepairCar (targetRCC);

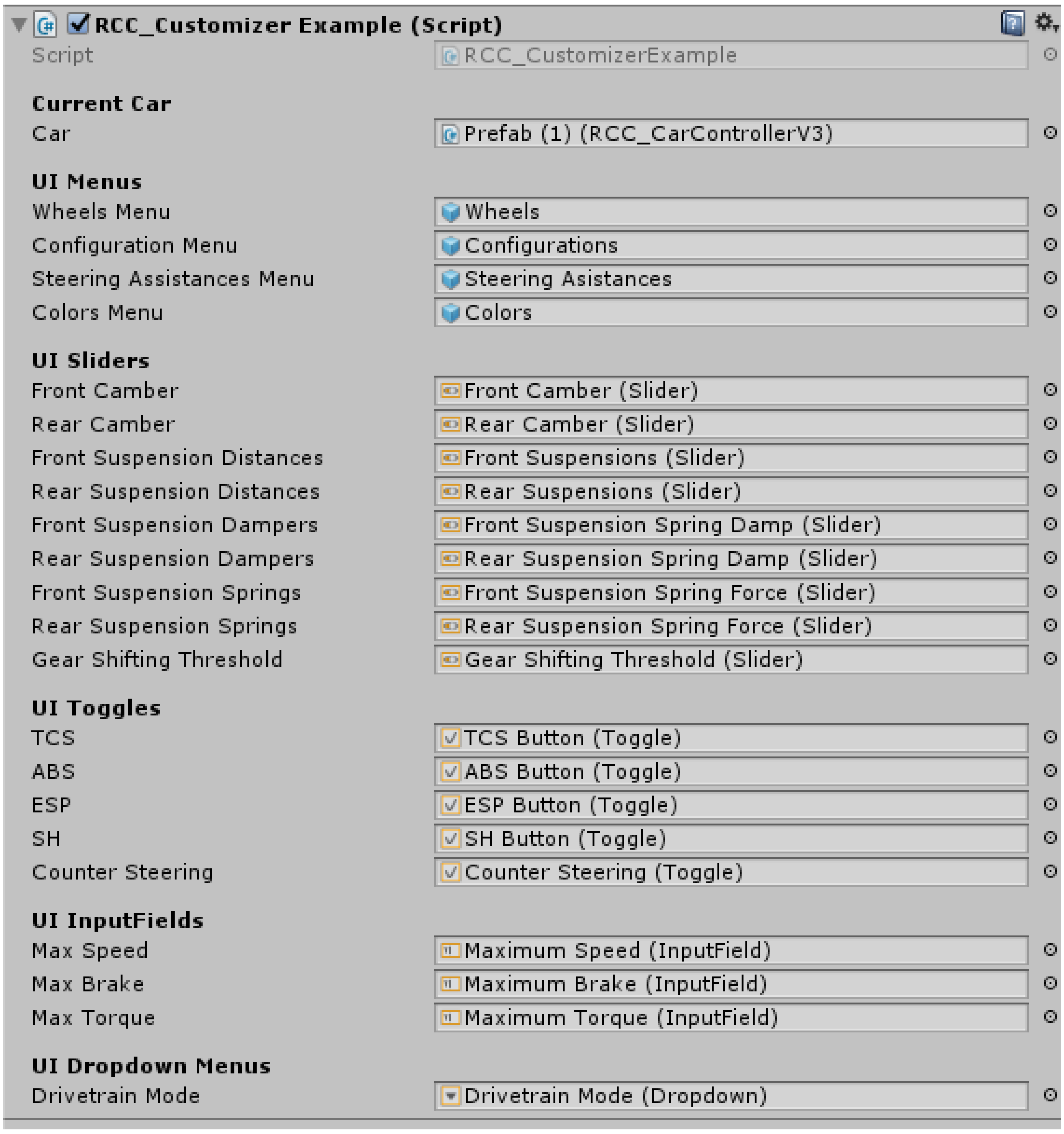
We want to change drivetrain of our car to AWD. So, we have to call it;

RCC_Customization. SetDrivetrainMode (targetRCC, RCC_CarControllerV3 WheelType.AWD

);

And goes on...

RCC_CustomizerExample.cs attached to RCC Canvas.



RCC_Customizer Example (Script)

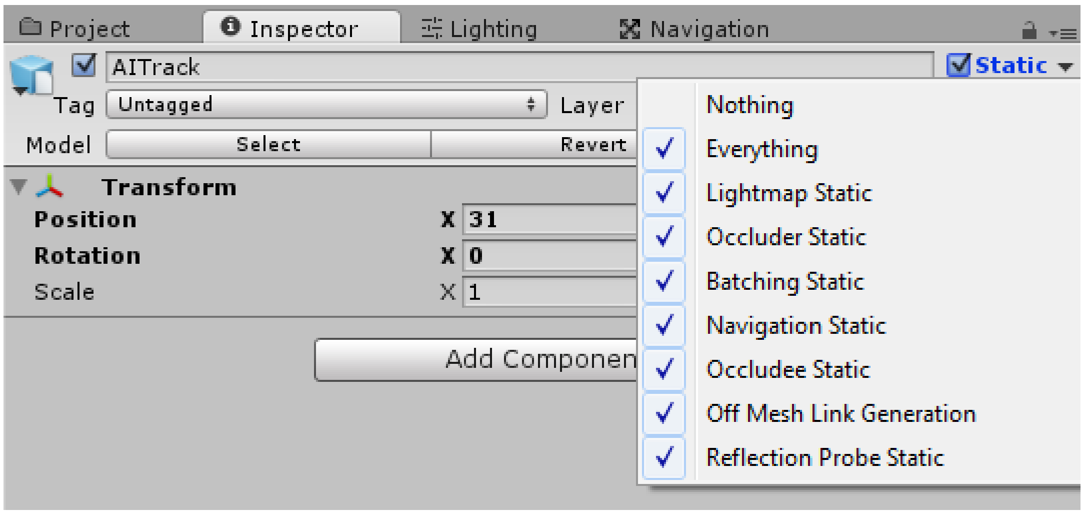
Script	RCC_CustomizerExample
Current Car	
Car	Prefab (1) (RCC_CarControllerV3)
UI Menus	
Wheels Menu	Wheels
Configuration Menu	Configurations
Steering Assistances Menu	Steering Asistances
Colors Menu	Colors
UI Sliders	
Front Camber	Front Camber (Slider)
Rear Camber	Rear Camber (Slider)
Front Suspension Distances	Front Suspensions (Slider)
Rear Suspension Distances	Rear Suspensions (Slider)
Front Suspension Dampers	Front Suspension Spring Damp (Slider)
Rear Suspension Dampers	Rear Suspension Spring Damp (Slider)
Front Suspension Springs	Front Suspension Spring Force (Slider)
Rear Suspension Springs	Rear Suspension Spring Force (Slider)
Gear Shifting Threshold	Gear Shifting Threshold (Slider)
UI Toggles	
TCS	TCS Button (Toggle)
ABS	ABS Button (Toggle)
ESP	ESP Button (Toggle)
SH	SH Button (Toggle)
Counter Steering	Counter Steering (Toggle)
UI InputFields	
Max Speed	Maximum Speed (InputField)
Max Brake	Maximum Brake (InputField)
Max Torque	Maximum Torque (InputField)
UI Dropdown Menus	
Drivetrain Mode	Drivetrain Mode (Dropdown)

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

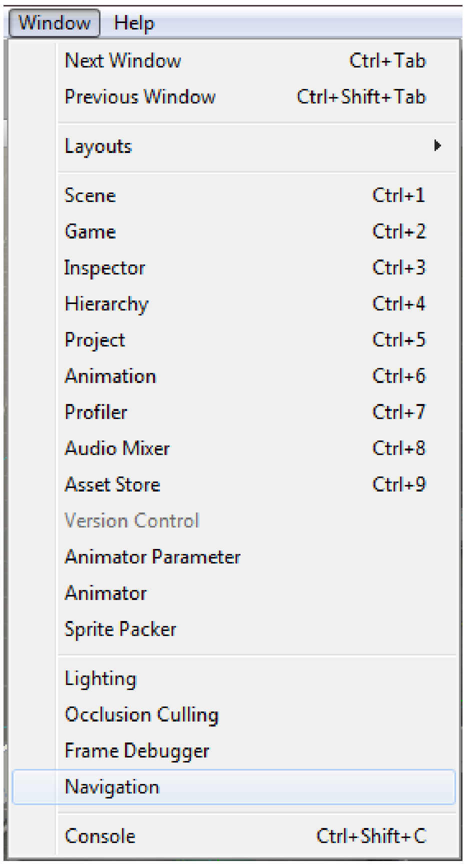
AI Configuration

Creating NavMesh For Scene

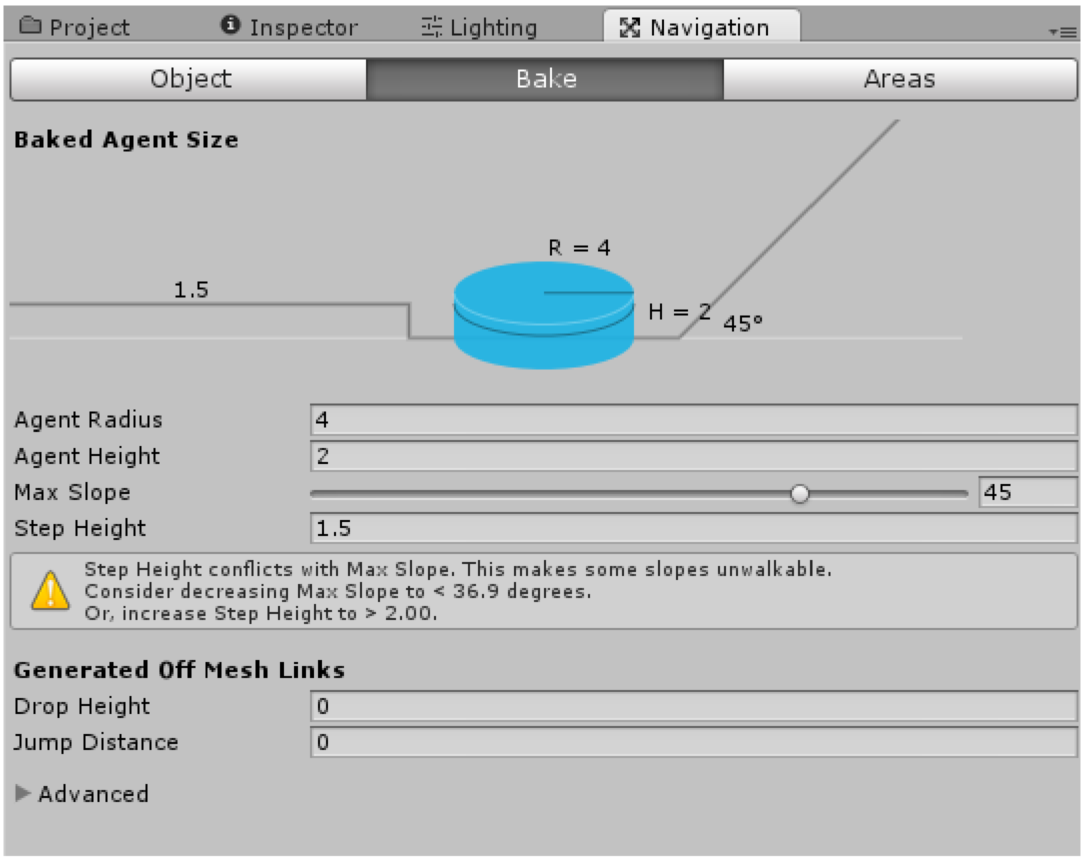
AI is based on **Unity's Nav Mesh**. Therefore, you must bake and create navigation mesh for your scene. Select your all static objects (including road too). And set them as "**Static**".



When all your static objects are marked as "**Static**", then you can bake your navigation mesh. Open "**Navigation**" window from **Window → Navigation**.



Default settings should be like this;



Baked Agent Size

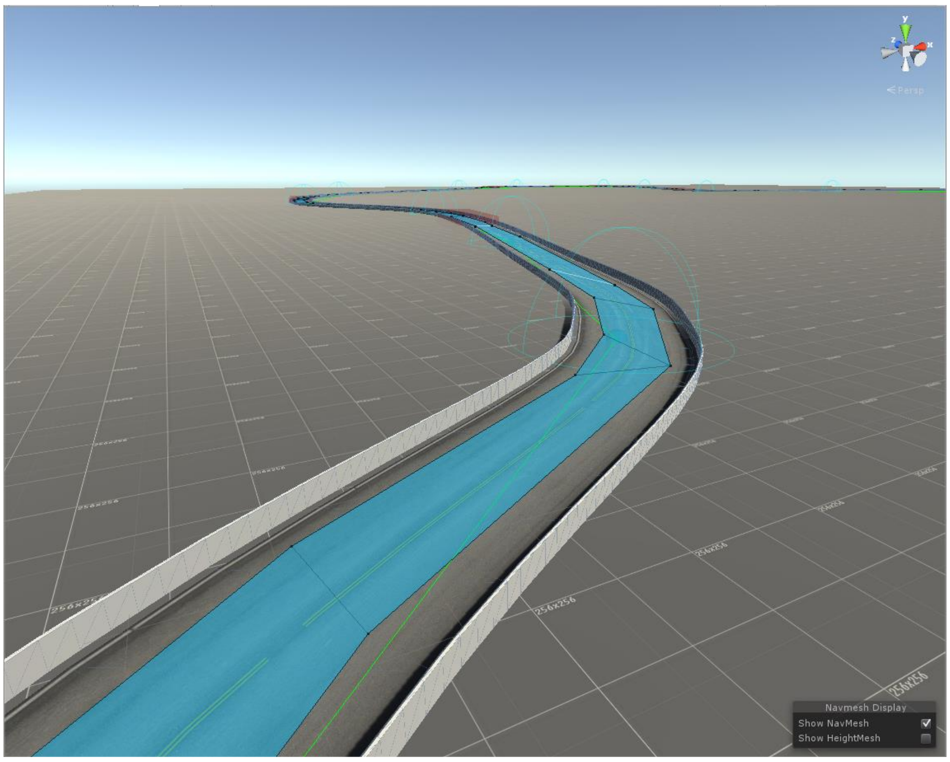
Agent Radius 4
Agent Height 2
Max Slope 45
Step Height 1.5

Step Height conflicts with Max Slope. This makes some slopes unwalkable.
Consider decreasing Max Slope to < 36.9 degrees.
Or, increase Step Height to > 2.00.

Generated Off Mesh Links
Drop Height 0
Jump Distance 0

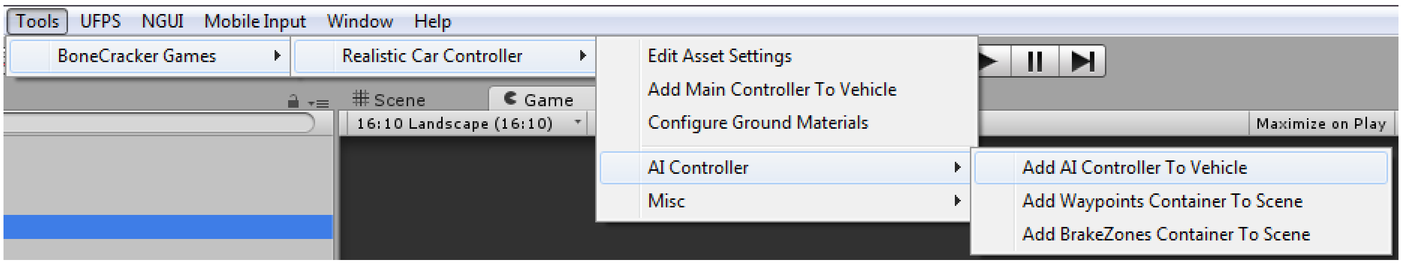
Advanced

And then, click the bake button and bake your scene. Check your blue navigation mesh. Should be like this;

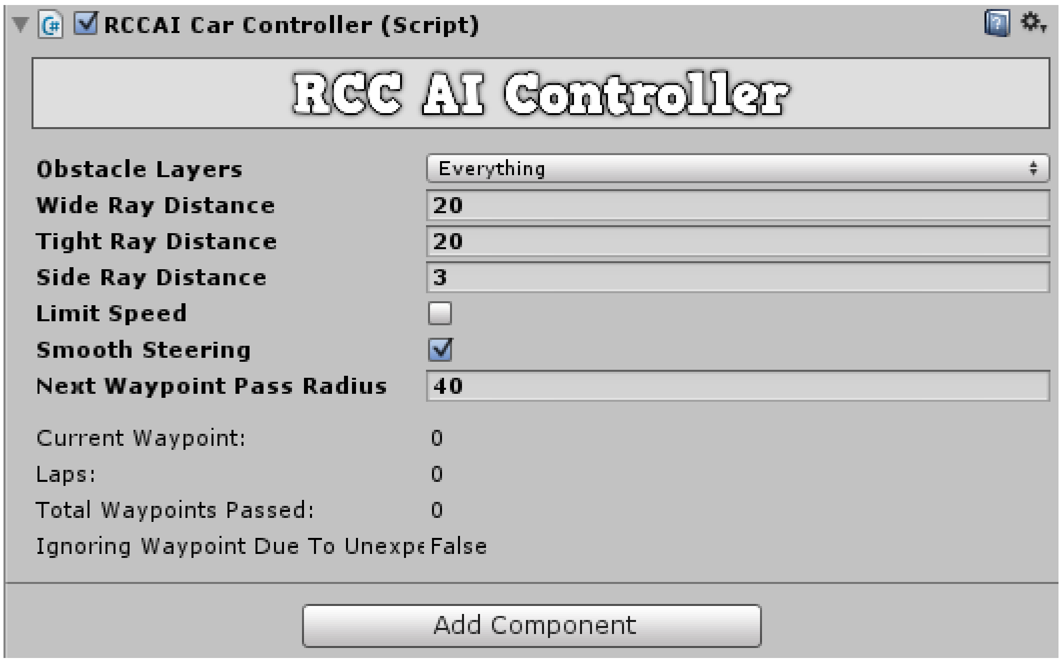


Adding AI Controller To Vehicle

First, build and configurate your vehicle. When everything works fine and results are as expected, you can add AI Controller to your vehicle by clicking "**Tools → BoneCracker Games → RCC → AI Controller → Add AI Controller To Vehicle**"



This will add "RCC_AIController" to root of your vehicle.


RCCAI Car Controller (Script)

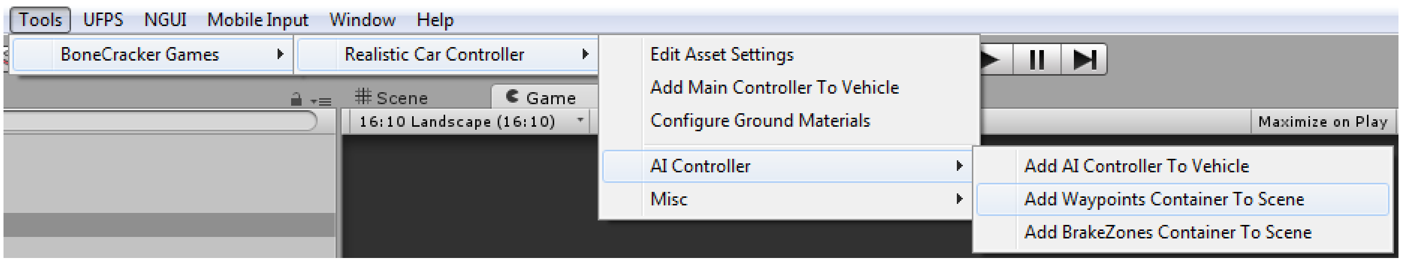
RCC AI Controller

Obstacle Layers: Everything
Wide Ray Distance: 20
Tight Ray Distance: 20
Side Ray Distance: 3
Limit Speed: ☐
Smooth Steering: ✓
Next Waypoint Pass Radius: 40

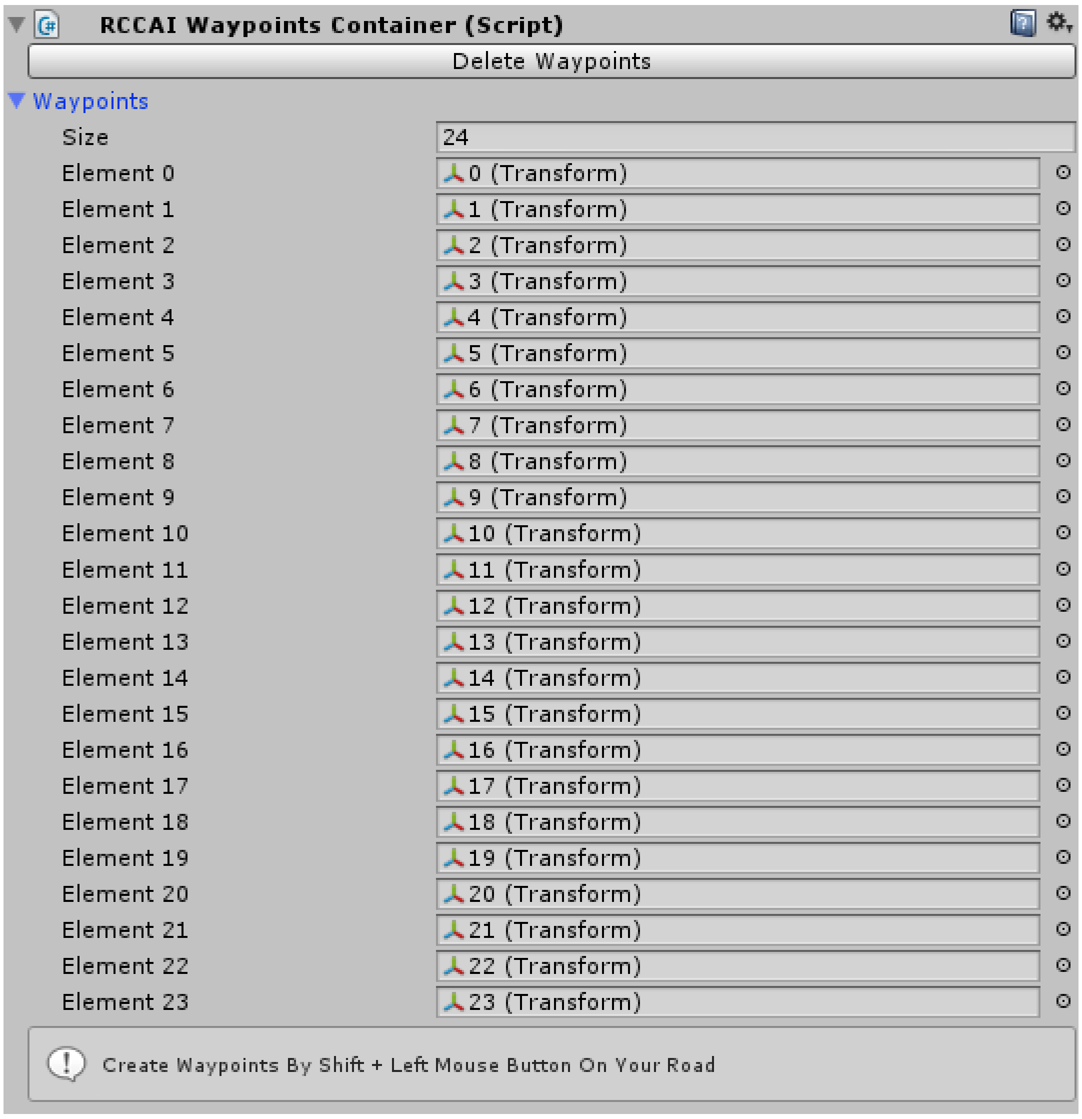
Current Waypoint: 0
Laps: 0
Total Waypoints Passed: 0
Ignoring Waypoint Due To Unexpe False

Vehicle will use "Nav Mesh Agent" for road path based on your waypoints, and will use raycasts for dynamic objects. If you have specified gameobjects for ignoring raycasts for this gameobject, you can select your gameobject layer from obstale layers.

Adding Waypoints Container To Scene



This will add "RCC AI Waypoints Container" to your scene. Simply hold Shift and left click on your road to create a new waypoint. Create your path with them.


RCCAI Waypoints Container (Script)

Delete Waypoints

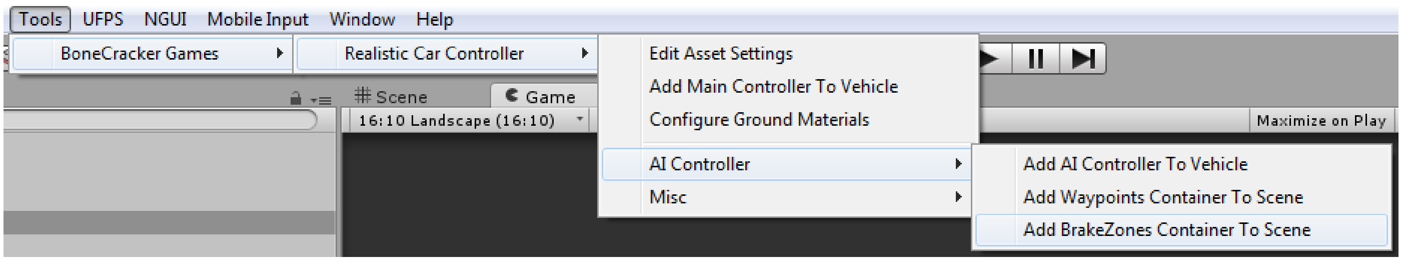
Waypoints

Size	24
Element 0	0 (Transform)
Element 1	1 (Transform)
Element 2	2 (Transform)
Element 3	3 (Transform)
Element 4	4 (Transform)
Element 5	5 (Transform)
Element 6	6 (Transform)
Element 7	7 (Transform)
Element 8	8 (Transform)
Element 9	9 (Transform)
Element 10	10 (Transform)
Element 11	11 (Transform)
Element 12	12 (Transform)
Element 13	13 (Transform)
Element 14	14 (Transform)
Element 15	15 (Transform)
Element 16	16 (Transform)
Element 17	17 (Transform)
Element 18	18 (Transform)
Element 19	19 (Transform)
Element 20	20 (Transform)
Element 21	21 (Transform)
Element 22	22 (Transform)
Element 23	23 (Transform)

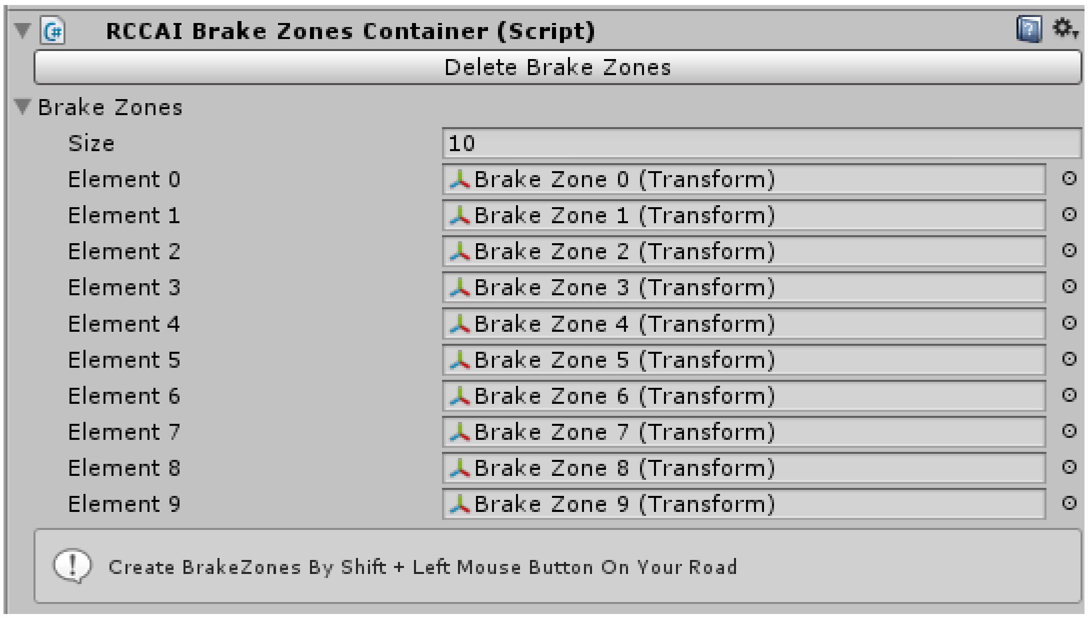
Create Waypoints By Shift + Left Mouse Button On Your Road

Info: Do not use CTRL+D for duplicate any waypoint.

Adding BrakeZones Container To Scene



This will add "RCC AI BrakeZones Container" to your scene. Simply hold Shift and left click on your road to create a new brake zone.


RCCAI Brake Zones Container (Script)

Delete Brake Zones

Brake Zones

Size	10
Element 0	Brake Zone 0 (Transform)
Element 1	Brake Zone 1 (Transform)
Element 2	Brake Zone 2 (Transform)
Element 3	Brake Zone 3 (Transform)
Element 4	Brake Zone 4 (Transform)
Element 5	Brake Zone 5 (Transform)
Element 6	Brake Zone 6 (Transform)
Element 7	Brake Zone 7 (Transform)
Element 8	Brake Zone 8 (Transform)
Element 9	Brake Zone 9 (Transform)

Create BrakeZones By Shift + Left Mouse Button On Your Road

Info: Do not use CTRL+D for duplicate any brake zone.

Each brake zone has a target speed. Vehicle will adapt its speed to this target speed when in this brakezone. You can scale your brakezones to fit to your road section.

Enter-Exit System

First, you have to import "**BCG Shared Assets**" into your project. It will add necessary scripts and resources to use your FPS or TPS player for enter-exit vehicles. Package is located in "**For BCG Shared Assets**" folder. There will be two new scenes in your "**Demo Scenes**" folder. All you have to do is, add necessary scripts to vehicle and player. Get out positions are created automatically in script if you don't select it in **RCC_EnterExitCar**.

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.

Multiplayer with Photon2

You have to import the necessary scripts to the project. First, download and import **Photon2**. Pass your AppID to Photon setup (expalined below), and then import "**RCC_PhotonNecessaryScripts**" in Scripts/Photon folder. Now you can test the Photon demo scene.

There should be a new scene named "**RCC City Photon2**" in Demo Scenes folder. Same scene with regular city scene. Only difference is, this scene has **Photon Network Manager**.

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

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

2 – If your connection to server succeeded, you will be in lobby. Now you can create your room, or join someones room, or join randomly room here.

3 – Congrats, now you are in an online multiplayer based room in realtime.

As soon as you have imported it to your project, it will ask your AppID. Pass it. And now you are ready to develop your realtime based multiplayer levels. It's free. And of course you are limited with 20 CCU.

Photon2 has many simple methods in their API. It's extremely easy to understand. Let me explain how the demo scene works;

Demo scene has a gameobject with script named "**RCC_PhotonManager**". This script handles multiplayer section of the scene. Uses these methods (You can find all methods from Photon2's docs);

Photon.Pun.PhotonNetwork.ConnectUsingSettings ();

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

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

Once connection to the server established, we have to join lobby.

Photon.Pun.PhotonNetwork.JoinLobby ();

As soon as we are connected to 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 room name. I didn't use it, because there are no any room list in demo.

I take a string that belongs to 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.cs** and **RCC_PhotonNetwork.cs**. These scripts are necessary for each vehicle. **RCC_PhotonNetwork.cs** is observed by **PhotonView.cs**.

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

These vehicles are not instantiated or destroyed with regular **GameObject.Instantiate** or Destroy. You have to 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. Therefore, there are two canvases in resources folder. One of them is using **RCC_Demo.cs**, other one is using **RCC_PhotonDemo.cs**.

F.A.Q.

Q. My vehicle is too unstable!

A. Might be wrongly placed COM. Do not use wrong shaped colliders. Do not use overlapped colliders. Use driving assistances.

Q. My vehicle is moving backwards!

A. Your vehicle model has wrong axes. Z must be looking at forward.

Q. My vehicle wheels or wheelcolliders are facing to wrong direction!

A. Be sure your wheel models x, y, z and pivot positions are correct (It's explained above)

Q. Why I can't drift like in your video?

A. Many things. First of all, COM. Your center of mass must be at correct position. Turn on Drifting is related with gizmos and check your wheel forces while car stands. Should be %55 front, %45 rear. Check out your collider shape (It is explained above). Apply high engine torque. Be sure you have selected "**Drift**" behavior from **RCC Settings**. This will override your vehicle settings. And also **RCC_WheelCollider** will use **Drift()** method to change wheelcollider friction curves at runtime.

Q. I'm getting huge performance loss and red exceptions!

A. Maybe you've found a bug. Send me your error immediately, so I can fix it.

Q. I'm getting "BCG_EnterExit" not found error.

A. If your project doesn't have BCG Shared Assets, you have to import it. Or delete "BCG_ENTEREXIT" key in Build Settings. Go to **File → Build Settings → Player Settings → Other Settings**. You will see "BCG_ENTEREXIT" in scripting define symbols. Delete it.

Credits

Extreme Vehicle Pack by Vertigo Games

Package contains sponsored vehicles by Vertigo Games. You can get this asset from this link;

http://u3d.as/4xM

Sofie With Animations by 3DMaesen

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

http://u3d.as/2vg

Sound Effects

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

So, how you holding up there? 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 mail me if you used any of my Assets on your game. I'd like to see it in action!

Made by Bugra Özdoganlar

Contact@BoneCrackerGames.com