

Creating New Vehicles

I wrote some editor scripts to simplify creating new vehicles. Creating and configuring the vehicle with %100 progress would take only five minutes if vehicle model has correct axes and pivot positions.

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. Unity has updated EULA about it at 2018, and any models with improper axes and directions will be rejected from Asset Store.

Important

Be sure you are in **PIVOT** and **LOCAL** mode while checking axes of your model.



X should be **Right**,

Y should be **Up**,

Z should be **Forward**.

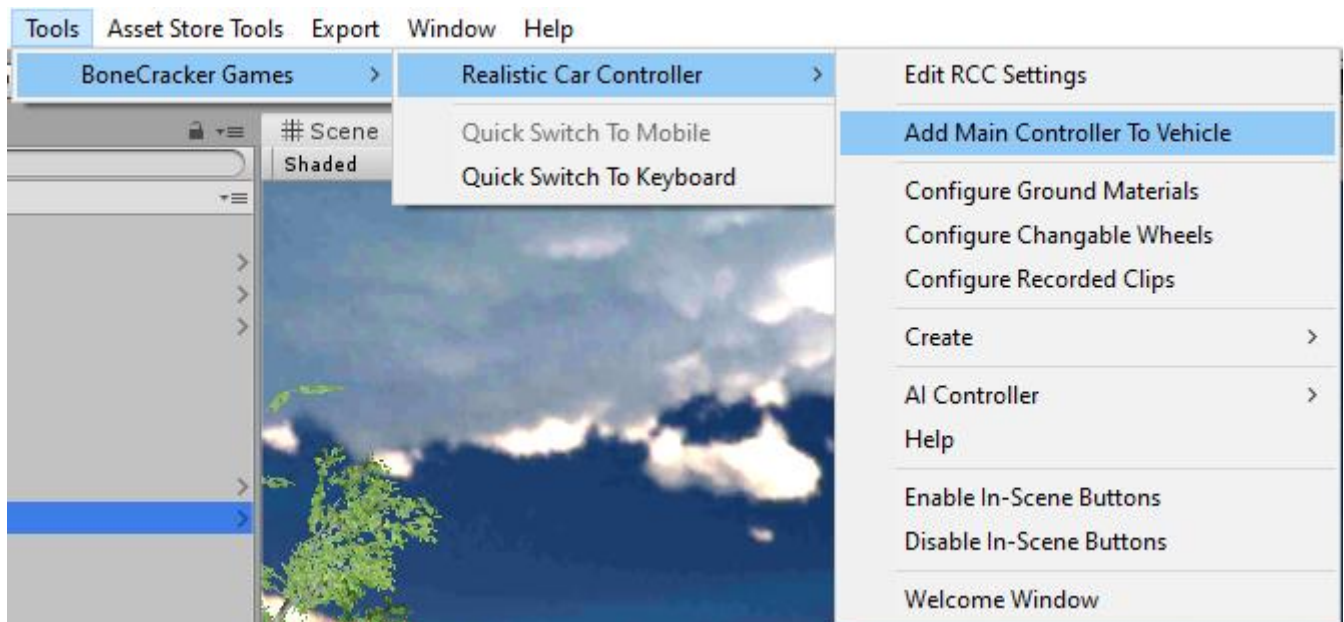


Some designers are making models with wrong directions or wrong pivot positions or both of them. This is really painful if you don't know how to fix the directions of the models. But I wrote an editor script to fix this. While adding main controller to the vehicle, it will ask you to fix pivot position of the vehicle if it's wrong. Choose “**Yes**” if your vehicle has wrong pivot position. However, script can't know vehicle model direction. **X, Y, Z** directions must be correct at least by your side.

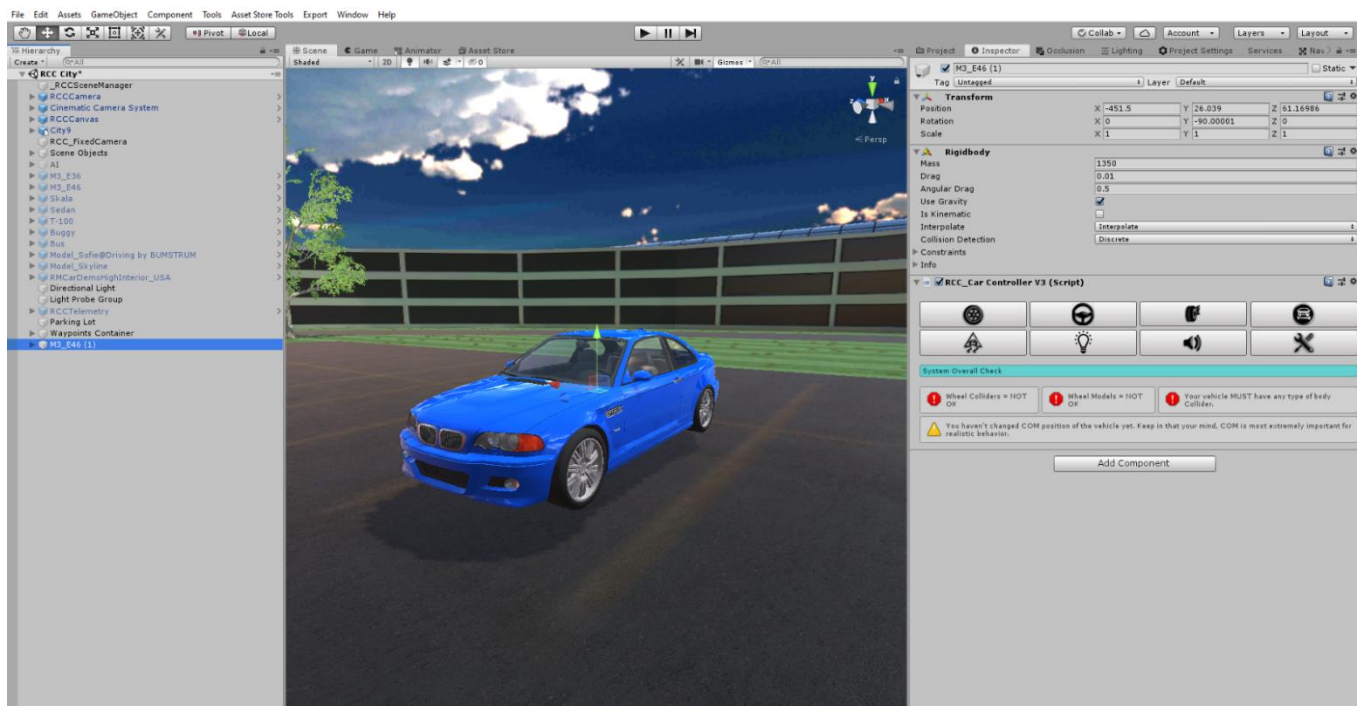
Also scale of your vehicle must be not oversized or miniature. Even if you are developing a toy car game, vehicle size should be nearly same with demo vehicles. Physics engine is calculating shape and size of the collider too. So, you have checked inputs, your vehicle and wheel models pivot positions,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

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

You have to add **Main Controller (RCC_CarControllerV3)** 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**. (**Shift + R**)

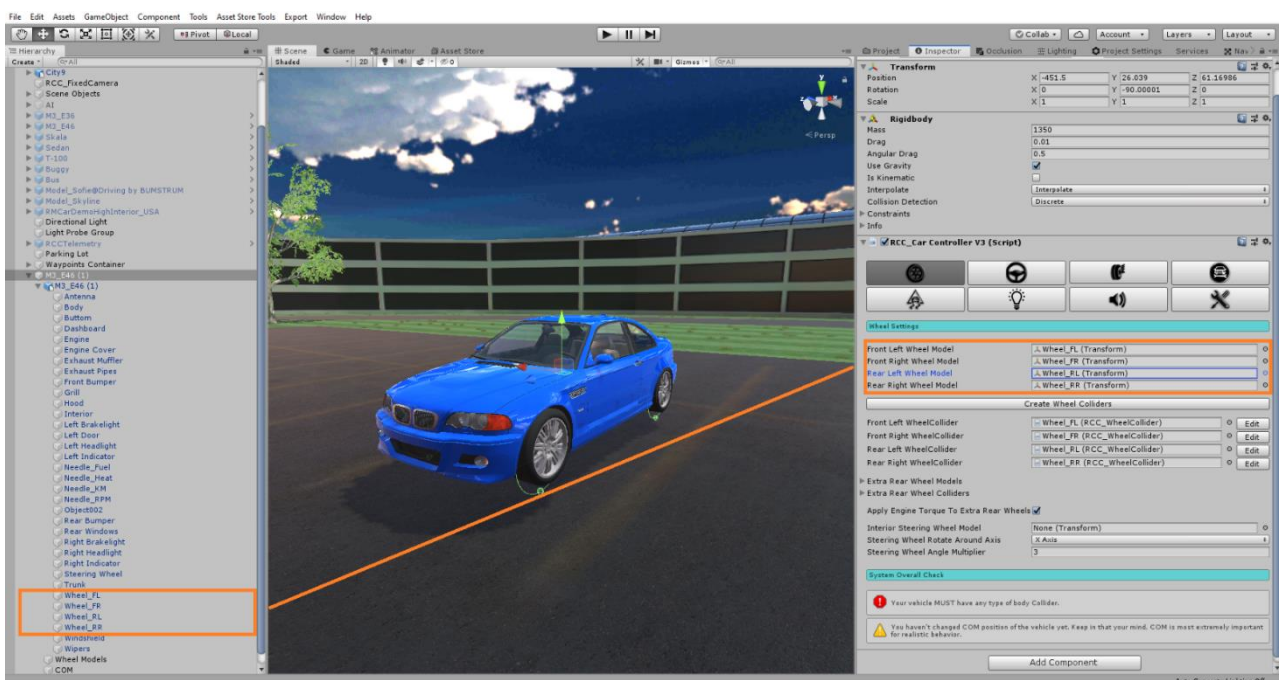


It will ask you to fix pivot position of the vehicle if it's wrong. Choose “**Yes**” if your vehicle has wrong pivot position. **RCC_Bounds.cs** will calculate the total bounds, and find center position of the model.



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 **0.25 – 0.50** for medium angular velocity. These rigidbody settings are applied when you create your vehicle automatically.

First thing we wanna do is creating the wheel colliders. Click “**Wheel**” tab in the editor script. Select all of your wheels. After selecting your wheel models, click the “**Create Wheel Colliders**” for creating wheel colliders with proper radius, suspension, damper, and friction curves automatically. Check radius and position of the wheel colliders after creating them.



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 colliders mass, spring, damp forces.

Your vehicle **MUST** have any kind of [Colliders](#) (Such as [Box Collider](#), or [Mesh Collider](#) etc...) for the body. Otherwise, physics won't work.

Editor script of the [RCC_CarControllerV3](#) will find any issues and inform you if persists. In some cases, some models have sphere colliders attached to the wheels, some models have trigger enabled mesh colliders, some models have rigidbodies on some child gameobjects, etc... I was having many emails about “**my car is bouncing, falling through the ground, won't move, etc...**”. Informer of the RCC will assist you in that case.

At this point, your vehicle model is ready to go. Wheel colliders, collider, and rigidbody are the essential components. All others components are optional and not required.

Creating Lights, Exhausts, Cameras, Sounds, Skidmarks, Smoke Effects, And Other Features

These effects are optional. And can be created with **In-Scene GUI** buttons with the fastest way. In order to enable buttons, simply go to [Tools → BCG → RCC → Enable In-Scene GUI Buttons](#). (**Shift + E**) It will enable UI panel at the left side of your editor scene window. In order to use them, select your vehicle in your scene and you will be able create lights, exhausts, cameras, mirrors, UI canvas, and vehicle camera by one click.

However, you may not use the panel. In that case, you can create them with the tabs explained below;

You can create point lights for braking and reverse gear, spot lights for headlights from [Tools → BoneCracker Games → Realistic Car Controller → Create → Lights](#). Or with In-Scene GUI Button. Place them correctly on your vehicle model.

You can create hood and wheel cameras from [Tools → BoneCracker Games → Realistic Car Controller → Create → Cameras → Hood Camera / Wheel Camera](#). Or with In-Scene GUI Buttons. Place them correctly on your vehicle model.

You can create exhaust effects from [Tools → BoneCracker Games → Realistic Car Controller → Create → Misc](#). Or with In-Scene GUI Buttons. Place them correctly on your model.

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

Smoke Effects

You will find “[RCCWheelSlipAsphalt](#)”, “[RCCWheelSlipGrass](#)” and “[RCCWheelSlipSand](#)” prefabs under Prefabs folder. You can use your own smoke prefab as you wish. These wheel particles have been selected in [Tools → BCG → RCC → Edit Ground Materials](#). Corresponding particle will be used on each ground material. If slippage of the wheel passes the target slip value in the ground materials panel, emissions will be enabled.

Engine Sound Effects

Engine sounds can be selected or changed at sound tab of the [RCC_CarControllerV3](#). RCC has three engine sound types. Modern games are using six audio clips per engine at least. Low, middle, and high RPM while pushing throttle, and same without pushing the throttle. If you have these audioclips, you can select all of them for realistic engine sound effects. But if you have only one audio clip, you can still use by selecting one source type.

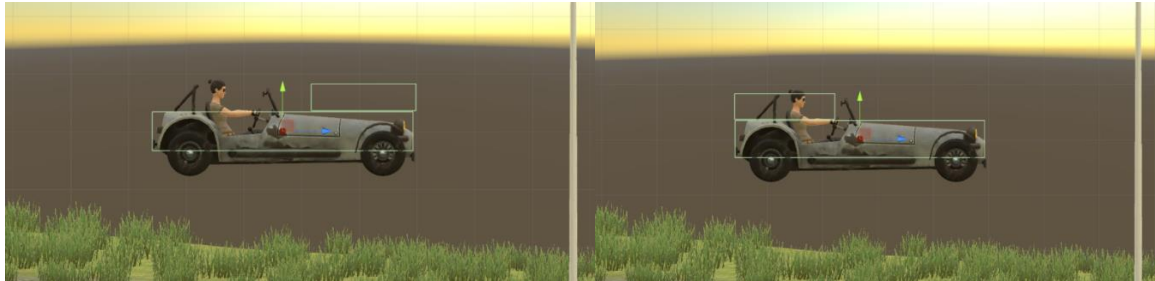
Skidmarks

Skidmarks are managed by [RCC_SkidmarksManager.cs](#). Draws meshes on the collider if slippage of the wheel collider is high enough with proper intensity. Different skidmarks can be drawn on each surface. To select, or edit them, go to [Tools → BCG → RCC → Configure Ground Materials](#). You will see each material has unique skidmarks. Double click to edit them, or just duplicate one of them in the project folder and customize.

Each wheel can have different skidmark width. Simply select your wheelcolliders and set their skidmark width. And also you can disable skidmark of any wheel too.

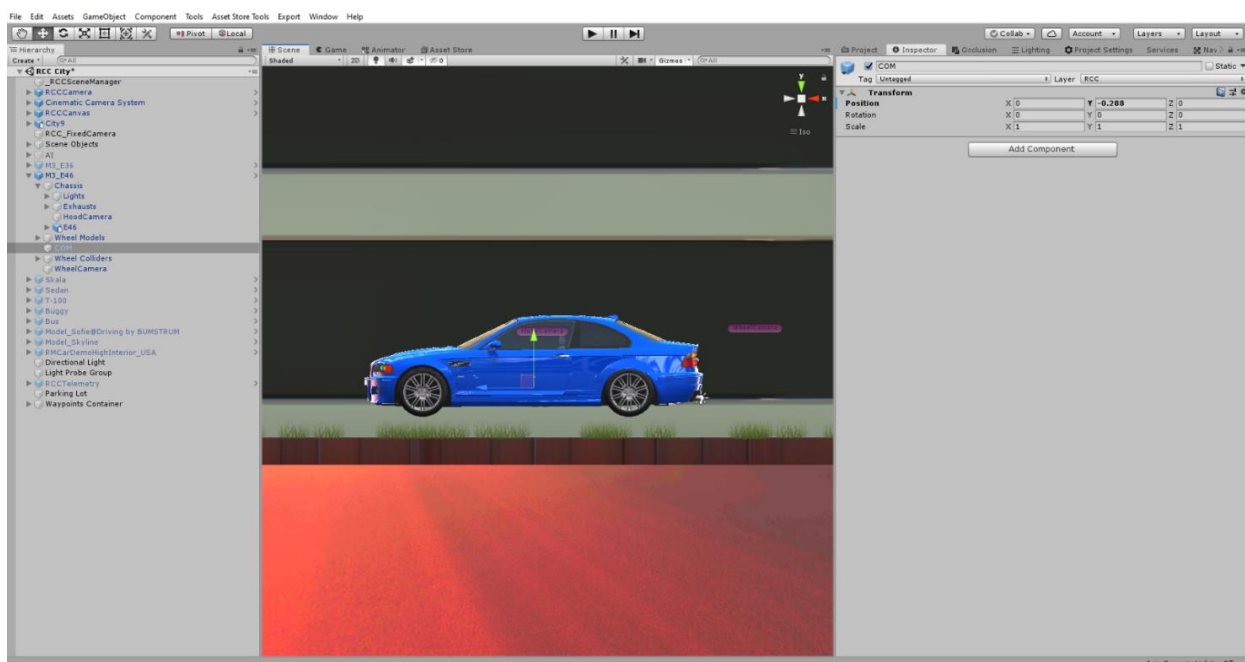
Collider Shapes Of The Vehicle

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;

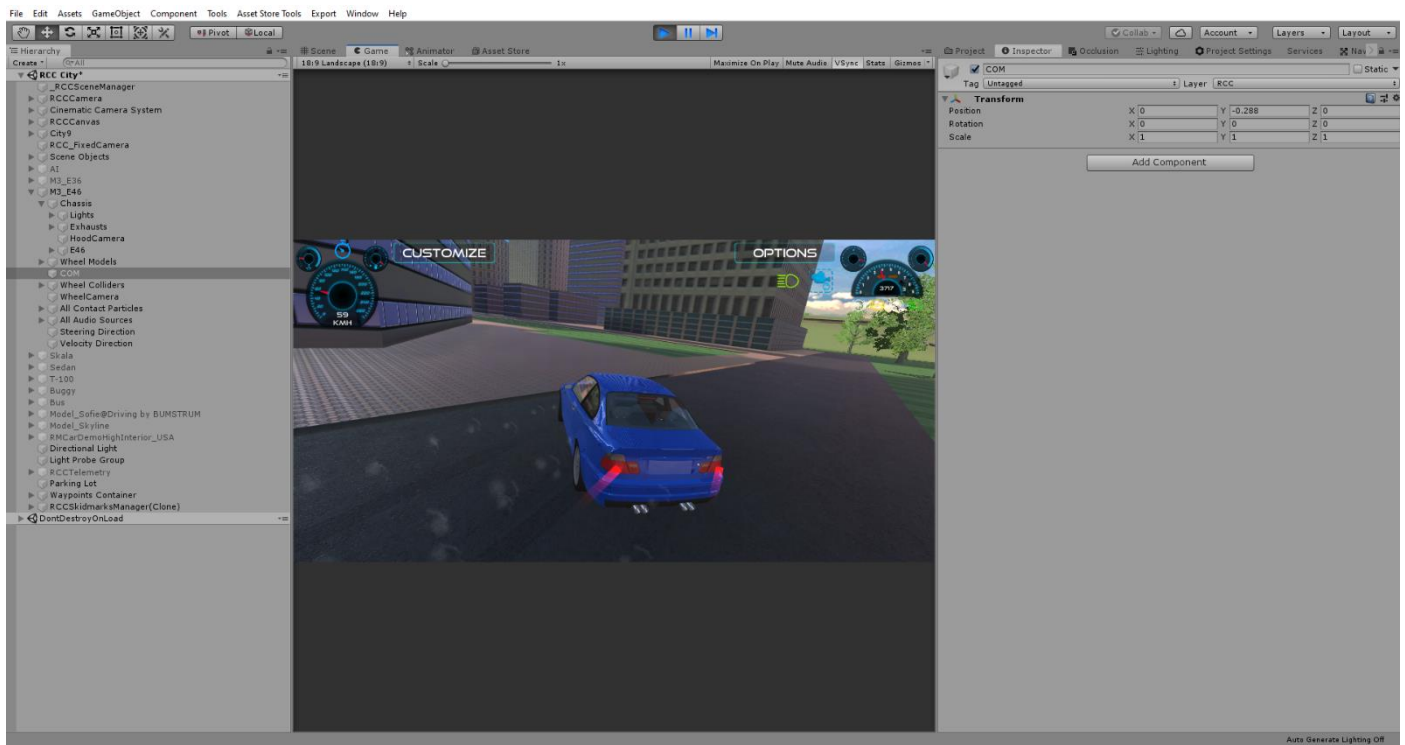


This vehicle has two box colliders. Second box collider is placed at front of the vehicle here on the first screenshot. This will bring you more controllable and stable vehicle. At the second image, second box collider is placed at rear of the vehicle. This will bring you more unstable and slippery vehicle. Differences between first colliders and second colliders are noticable. Just remember that collider shapes are effecting vehicle behavior directly. If you want to make your cars stable, keep in that your mind. And be sure to use proper mesh colliders, and box colliders on your vehicles.

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 gearbox. 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;



Runs perfectly after just few clicks.



Configure 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 slippage is equals or higher than this threshold value, corresponding assistance will be engaged immediately.

ABS = The **anti-lock braking system** (ABS). Its purpose is to keep the vehicle safe by preventing the wheels from locking when you use the brakes. You are able to keep control over the steering and your vehicle won't skid as a result of the wheels locking up.

ESP = The **electronic stability program** supports the vehicle in nearly all critical driving situations. It comprises the functions of the antilock braking system (ABS) and the traction control system, but can do considerably more. It detects vehicle skidding movements, and actively counteracts them by applying individual brake forces to each wheel.

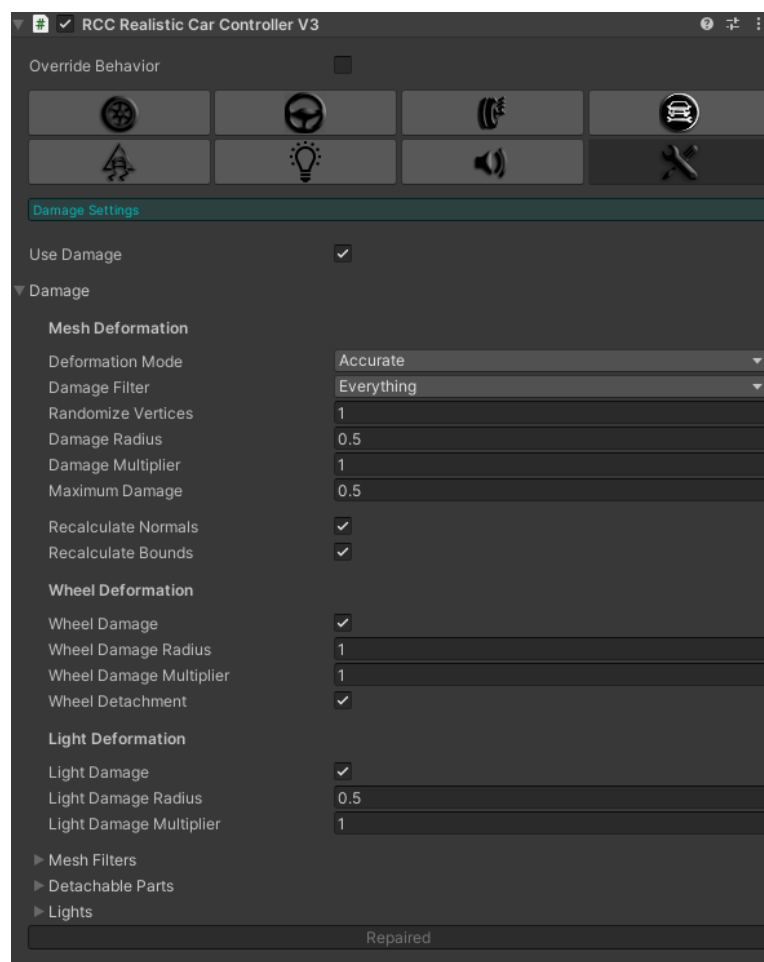
TCS = The **traction control system** detects if a loss of traction occurs among the car's wheels. Upon identifying a wheel that is losing its grip on the road, the system automatically applies the brakes to that individual one or cut down the car's engine power to the slipping wheel.

Steering Helper = The **Steering helper** 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 can be used if behavior type is Racing, Arcade, or Semi-Arcade.)

Traction Helper = The **Traction helper** will reduce stiffness of the front right or front left wheels if vehicle is skidding sideways. It will take steering angle as reference while doing this, and keep the vehicle in control.

Damage

Your vehicle body mesh wireframe topology must be reliable for realistic mesh deformation. If your vehicle body mesh has broken (unwelded) vertices or bad wireframe topology, mesh will deform buggy and unrealistically. RCC takes all meshes of the vehicle if you haven't select them specifically. If you want to select specific meshes, RCC won't take all. Damage radius, multiplier, maximum damage and more stuff can be customized at damage tab of the RCC_CarControllerV3.



Deformation mode = Fast for instant mesh deformation, and accurate for smooth time based mesh deformation. Accurate mode is more heavier.

Damage Filter = Vehicle will take damage from the colliders with these layers. You can unchoose specific layer from here.