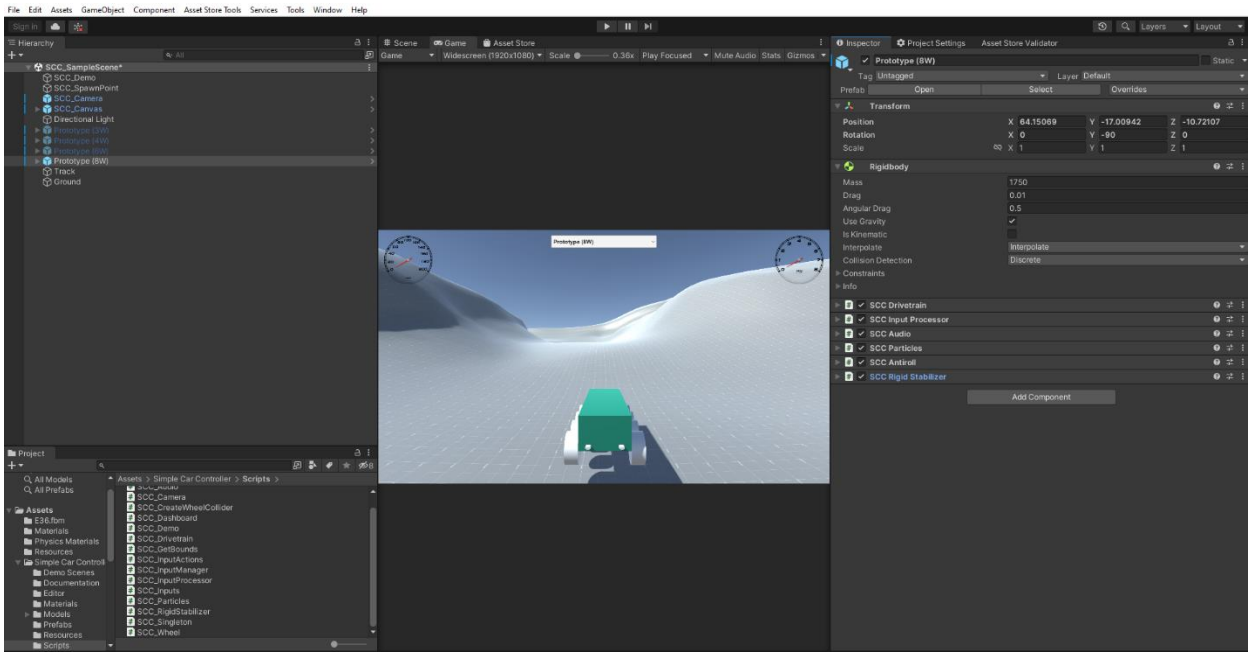
Simple Car Controller by BoneCracker Games

Thank you for purchasing and using Simple Car Controller (SCC). This documentation will guide you on definition of the following contents below.

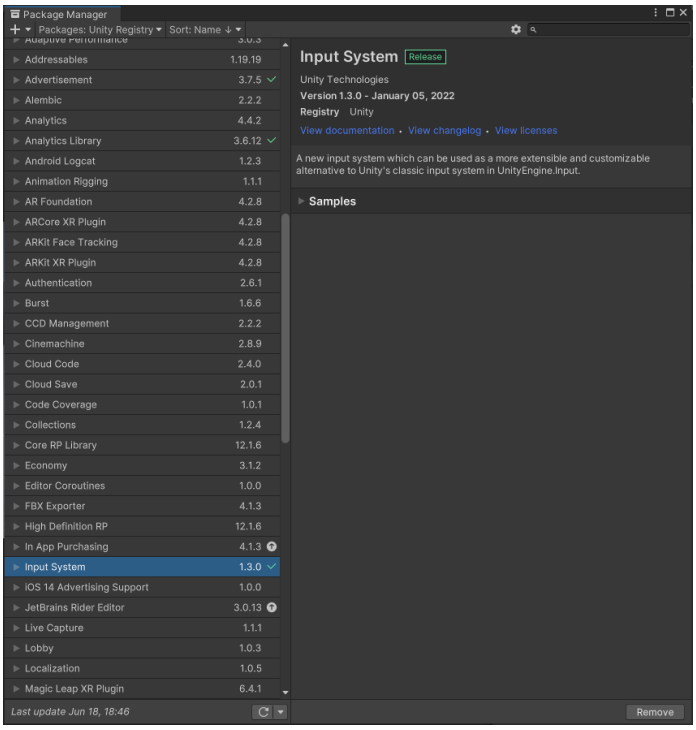
Overview

You can open the demo scene and take a look at how the system works. Demo scene has **SCC_Camera**, **SCC_Canvas**, and **SCC_Demo** scripts. Vehicles have multiple scripts such as **SCC_Drivetrain**, **SCC_InputProcessor**, **SCC_Particles**, **SCC_Audio**, etc… Detailed explanation of them can be found in this documentation.

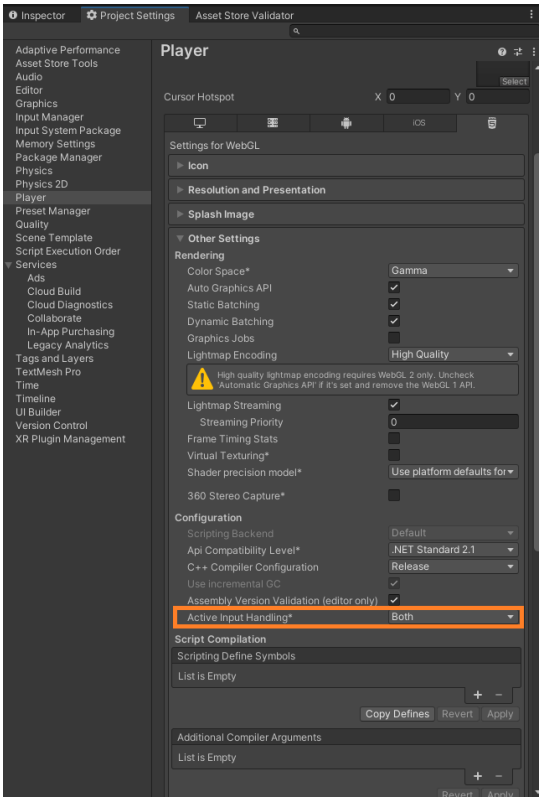


Import The Package & New Input System

SCC is using the new input system. Therefore, new input system must be installed in your project. Importing the project will install new input system automatically if you choose to install dependencies. But if you pass it, you can install new input system from the **Package Manager**. Open up the **Package Manager** through the **Window → Package Manager** and search for "**Input System**". Install it to your project. If your project is missing the new input system package, you'll get compiler errors, because inputs of the SCC is dependent on the new input system classes.



Be sure “**Active Input Handling**” in your “**Player Settings**” is set to “**New Input System**” or “**Both**”. You can access to the player settings from the “**Build Settings**”. However, these steps are necessary if you choose to install dependencies while importing the package. Only do these steps if you didn’t install dependencies.



Compatible Unity Versions

SCC is compatible with all versions of Unity after **2021.3.2f1**. SCC will still work with older versions, however I would recommend you to use **LTS** versions of Unity. If you encounter any issues with specific versions of Unity, please let me know.

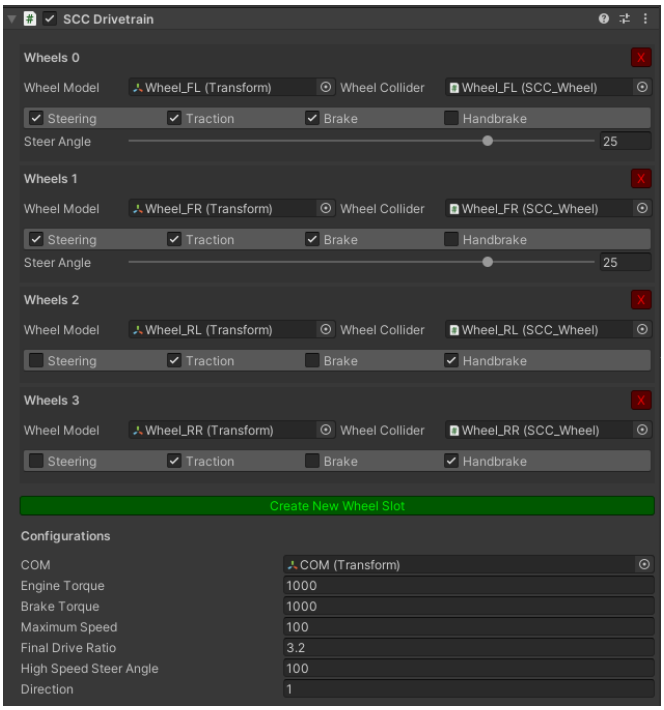
How The Demo Scene Works

Demo scene includes a camera, a canvas, vehicles, and a demo manager script. This manager script instantiates vehicles and assigns them for camera and canvas. UI dropdown menu in the canvas is using spawn method in the demo manager script to instantiate and register new vehicles.

Vehicle Components

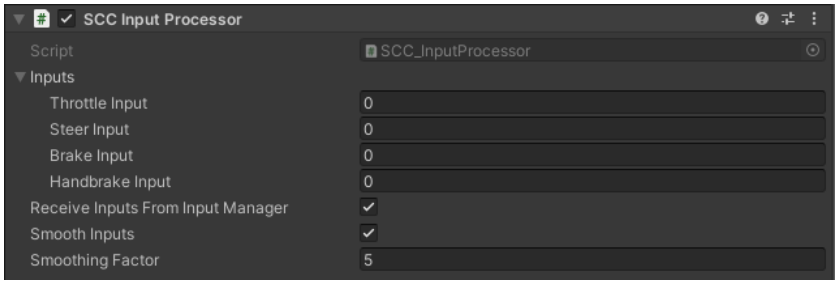
SCC_Drivetrain

Main controller of the vehicle. Stores wheels and their wheelcolliders with wheelmodels. Engine torque, speed, and other configurations can be made on this component. This component is necessary for all vehicles. When you add this component to your vehicle, all other essential components will be added as well.



SCC_InputProcessor

Input processor of the vehicle. Receives the inputs from the **SCC_InputManager**, and let other components use these inputs. It's necessary to control the vehicle with inputs. However, removing the component won't give you any errors.

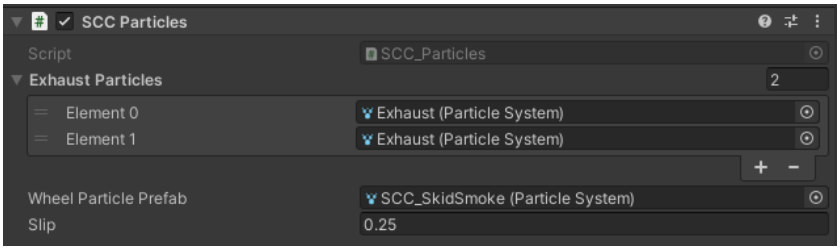


SCC_Audio

Audio component of the vehicle. It's an essential component, and not necessary. Engine on and off sounds can be selected.

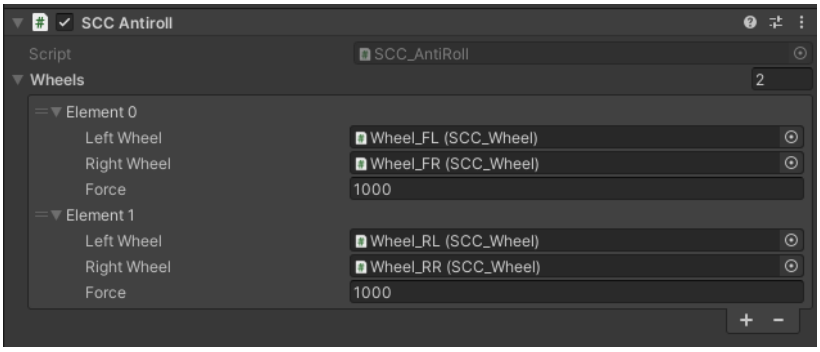
SCC_Particles

Wheel particles. It's an essential component, and not necessary. Wheel particle prefab can be selected.



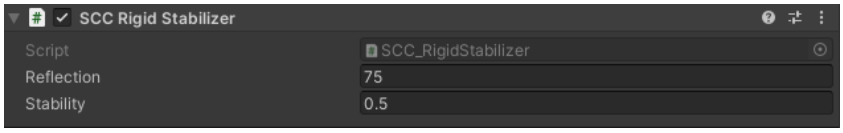
SCC_AntiRoll

Antiroll system. It's an essential component, but required to have stable vehicle physics. Keeps vehicle under control on hard turns and avoids rolling to sides. Left and right wheel can be selected with specific antiroll force amount.



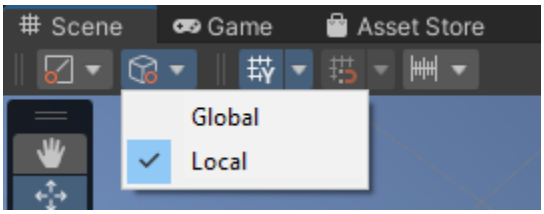
SCC_RigidStabilizer

When vehicle is not grounded, it helps to keep the vehicle straight. It's an essential component, and not necessary. Force and reflection intensity can be adjusted.

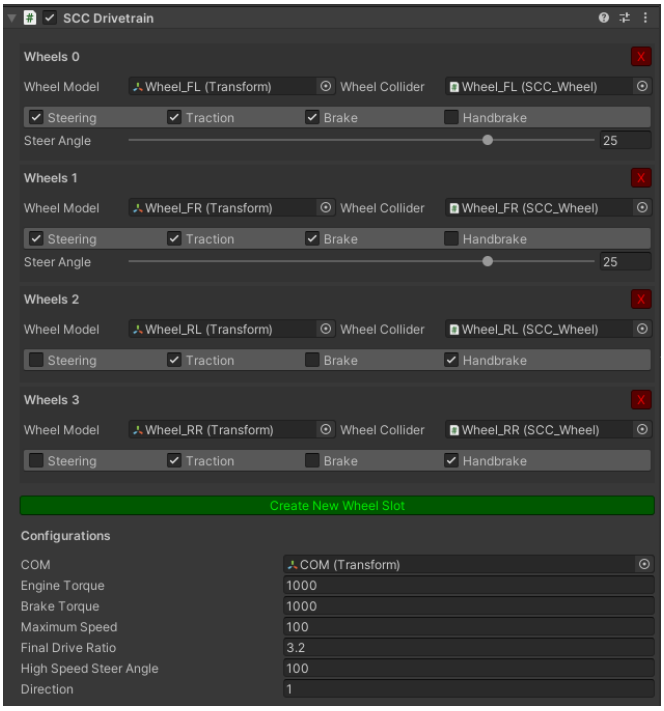


How to Create New Vehicles

Before creating new vehicles, you must be sure your vehicle models are eligible for vehicle physics. X, Y, Z directions of the vehicle must be correct and pivot positions of the wheels must be correct. X shoud be facing to right, Y is up, and Z is forward. Be sure you are in local mode while checking your model's axes.



If your model has proper axes and pivots, you are free to add the main controller to the vehicle. Simply select the vehicle model on your scene and add controller from the **Tools → BCG → SCC → Add Main Controller To Vehicle**. This will add all systems to the vehicle. After this step, you need to select wheel models and create their wheelcolliders. Simply drag and drop your wheel models to the wheel model field and click "**Create WheelCollider**" to create wheelcollider of the selected model. Do this for all wheels. Your vehicle can have many wheels as long as their wheelcolliders are set.



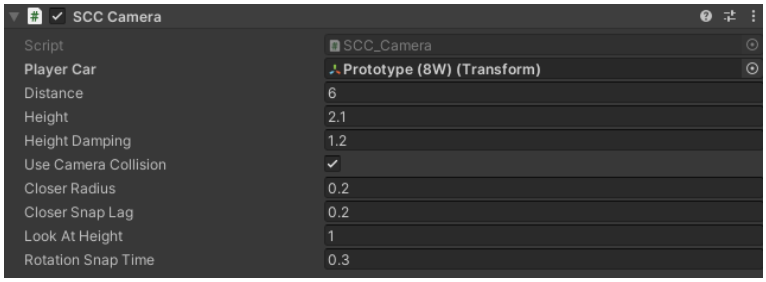
You can set steer, traction, brake and handbrake states of the wheels independently. **COM** (Center of mass) position will be created automatically, but you need to set it manually. You can check the demo vehicles. If you please **COM** too high, vehicle will roll to the sides easily.

All physics object must include colliders. Be sure your vehicle has proper collider or colliders. Simply select the main body part of your vehicle and add mesh or box collider to it. If your vehicle doesn't have the any collider, it will do weird movements when it's touch to the ground.

At this point, your vehicle is fully controllable. You can customize configurations of the vehicle from your inspector panel. If you don't need other essential components, you can remove them from the vehicle.

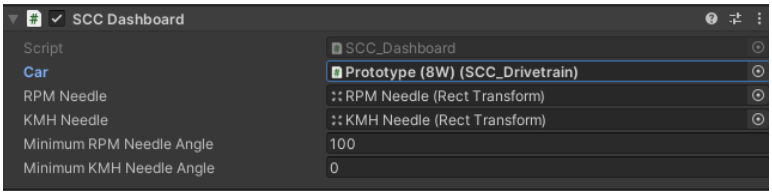
SCC_Camera

Follows the target vehicle with a few settings. In order to create it on your scene, **Tools → BCG → SCC → Add SCC Camera To Scene**. This will create a new **SCC_Camera** on your scene. Also it can be found in the prefabs folder as well, you can simply drag and drop the **SCC_Camera** prefab to your scene.



SCC_Canvas

UI Canvas for displaying the gauges, speed, and RPM. Receives them from the **SCC_Drivetrain** attached to the target vehicle. In order to create it on your scene, **Tools → BCG → SCC → Add SCC Canvas To Scene**. This will create a new SCC_Canvas on your scene. Also it can be found in the prefabs folder as well, you can simply drag and drop the **SCC_Canvas** prefab to your scene.



SCC_Camera and SCC_Canvas are not necessary for your scene, you don't have to use them. You can use your own camera controller and UI canvas as well. You are not restricted.

Support

You can contact me by sending me an email to bonecrackergames@gmail.com including your invoice number.