

Toy Car Simulator

Thank you for purchasing and using TCS!

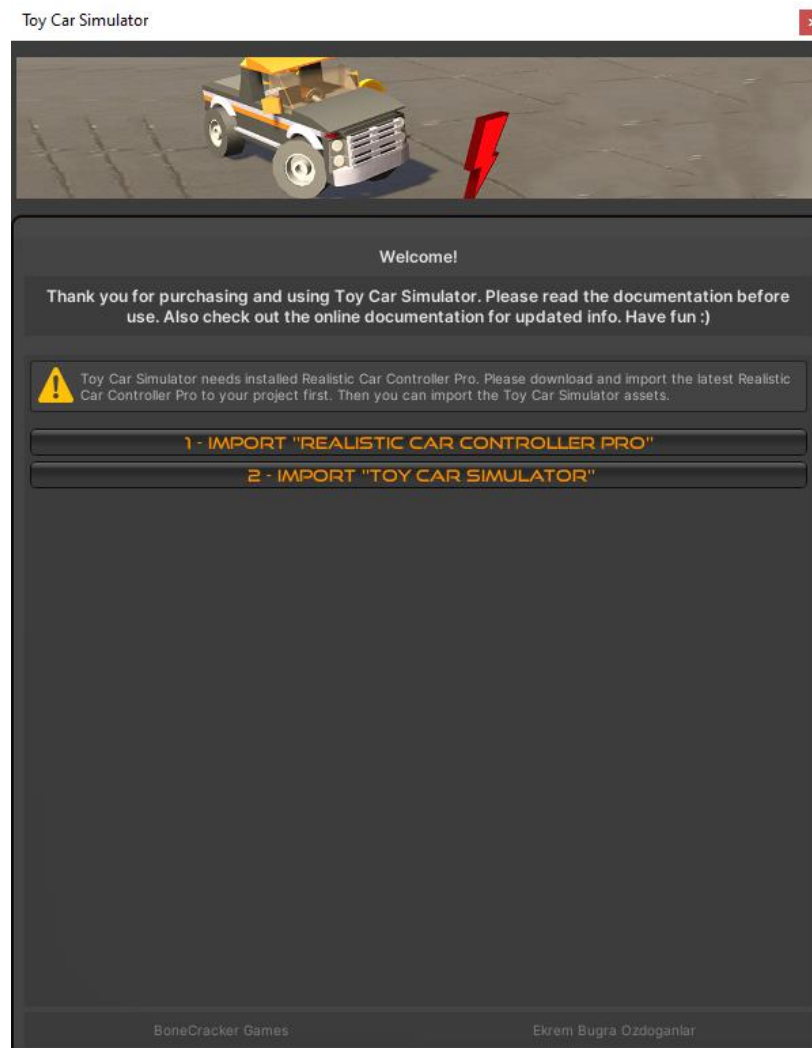
Contents

How to install the project	2
How to install Realistic Car Controller Pro to the project	4
How to customize the main settings, and soundtracks	5
How the mainmenu system works	5
How the gameplay system works.....	6
How to add / remove / edit vehicles	6
How to create a new vehicle from the scratch (with tutorial video)	7
How to add / remove / edit scenes	9
How to add / remove / edit items	10
Scripts	11
TCS_API	11
TCS_AudioSource.....	11
TCS_GameplayManager	11
TCS_Item	11
TCS_MainMenuManager.....	11
TCS_Player.....	11
TCS_PlayerCars	11
TCS_Settings	11
TCS_ShowroomCamera.....	12
TCS_Soundtrack	12
TCS_UI_CoinsAnimator / TCS_UI_EnergyAnimator / TCS_UI_OverspeedAnimator	12
TCS_UIManager	12
How to use the simple API	12
Vehicles.....	12
Deleting demo contents of the Realistic Car Controller Pro	13
URP.....	13

How to install the project

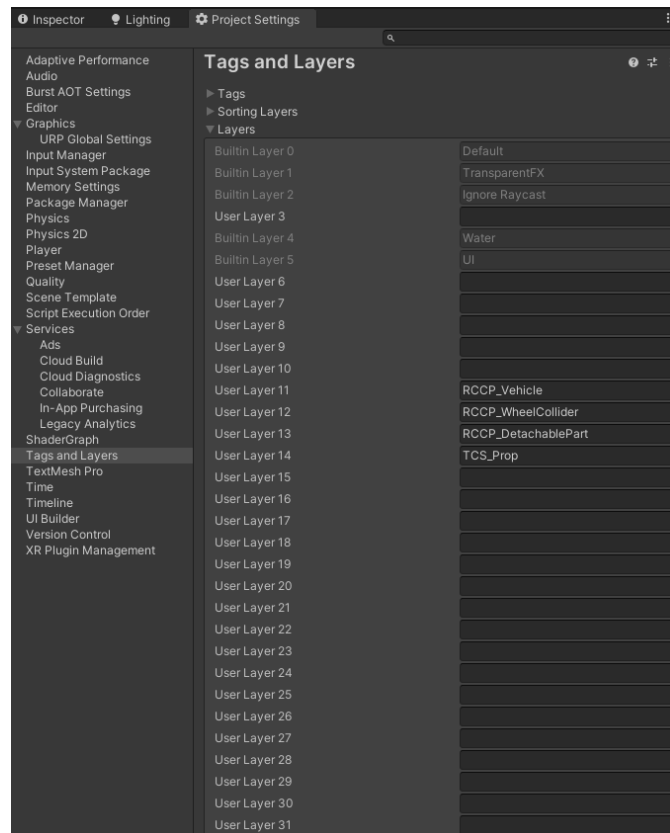
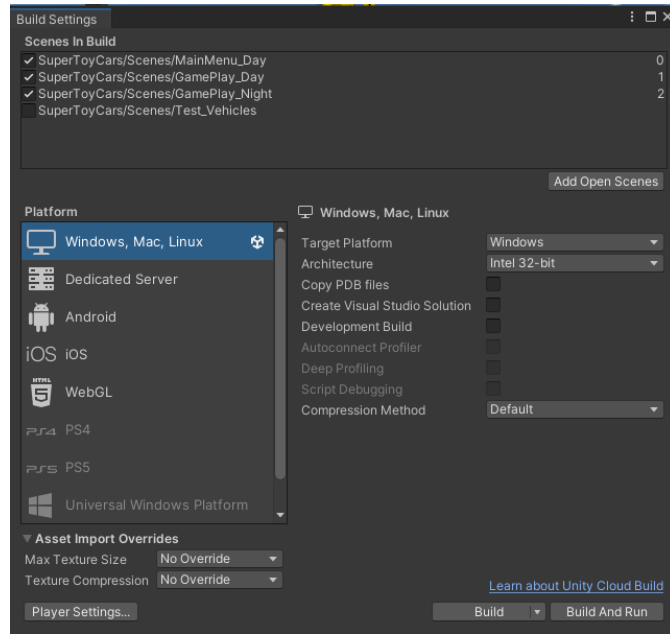
Be sure to install dependencies while importing the package. Import screen will ask you to install dependencies or not, please choose “**Yes**”. After importing the package, welcome window will appear and it will ask you to import the **Realistic Car Controller Pro** to the project first. Dependencies are input system, and textmesh pro packages.

There would be two very simple installation steps. First, you need to import **Realistic Car Controller Pro**, and then, you need to import **Toy Car Simulator**. Please wait while importing the packages, let the editor loads them up. Don't try to import both packages at the same time.



Project won't run without Realistic Car Controller Pro. First step is importing the Realistic Car Controller Pro. After installation of the Realistic Car Controller Pro, you need to import remain

Toy Car Simulator assets into the project. After those two installation process, We don't want to miss anything. So, we need to check the installation of the project. Open up the **Build Settings** (**File → Build Settings**), and check every scene in the list. After that, check the tags and layers from the **Project Settings** (**Edit → Project Settings**). They should be exactly like this;



Open up the main menu scene, pick a vehicle, and take a ride. If everything works fine without any errors, your import should be fine.

How to install Realistic Car Controller Pro to the project

As I mentioned in the “[How to install the project](#)” title, welcome window will appear after installation of the package. It will ask you to import Realistic Car Controller Pro to the project. Once you import the Realistic Car Controller Pro to the project, please wait the editor to compile scripts and import all necessary assets. Once it's done, you're good to go.

Welcome window of the Realistic Car Controller Pro will appear, but don't add any demo scenes of the Realistic Car Controller Pro, otherwise your build settings will be corrupted.

Note: Realistic Car Controller Pro will be installed with the demo content. If you are not planning to use demo content of the Realistic Car Controller Pro, you can delete and remove them from the project. Otherwise your build size will be increased. It's explained in the title named “[Deleting demo contents of the Realistic Car Controller Pro](#)”.

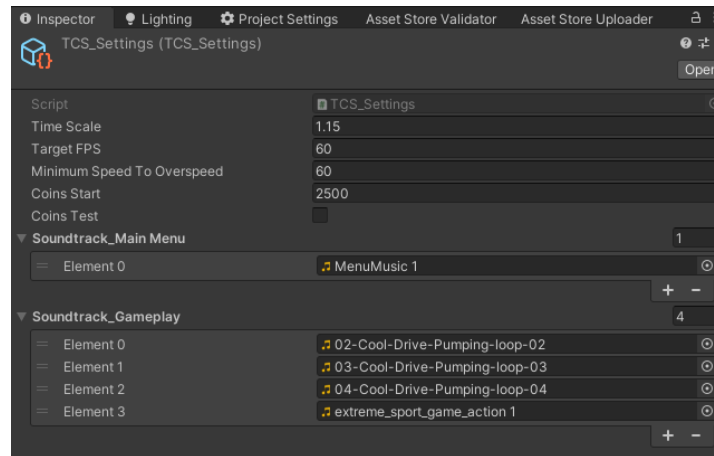
Note: Project is using URP, and all materials of the Toy Car Simulator is converted to URP materials. But not Realistic Car Controller Pro's materials. If you try to load any model of Realistic Car Controller Pro, you will see pink materials because all materials of the Realistic Car Controller Pro is using built-in materials. You'll need to convert all materials of the model to URP material. In order to do this, select all material of the models, and convert them through the **Edit → Rendering → Materials → Convert selected built-in materials to URP materials**.

TCS only override RCCP_Settings in the Realistic Car Controller Pro. There would be a new behavior option named “TCS” in the RCCP Settings. In order to access to the RCCP Settings, go to Tools → BCG → RCCP → Edit Settings.

Recommended: I would suggest you to read documentations of the Realistic Car Controller Pro to understand how the vehicles work.

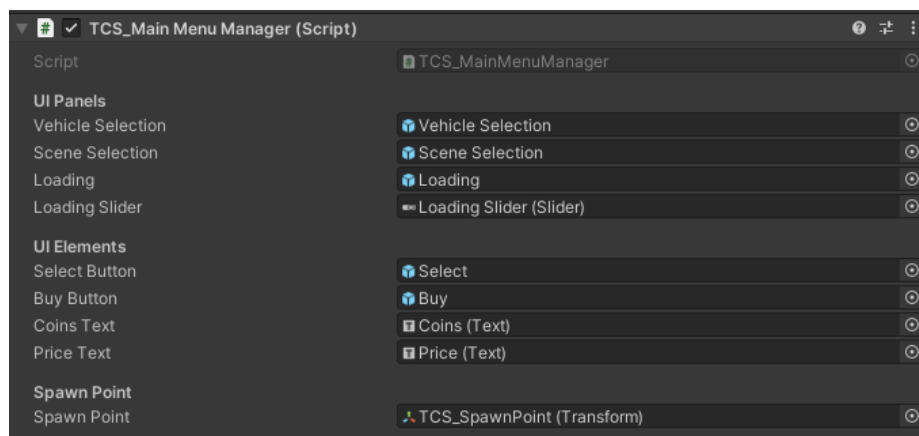
How to customize the main settings, and soundtracks

Toy Car Simulator includes general shared settings. You can customize these basic settings if you like. You can access to the settings from the **Tools → TCS → Edit Settings**.



How the mainmenu system works

Main menu includes a manager script named “**TCS_MainMenuManager**”, a spawn point, and a showroom camera (Main camera). Basically, it instantiates all selectable player vehicles at once, disables all of them, and enables only selected vehicle by the player. Saves the selected vehicle with index, this index will be used to spawn selected vehicle on the other scenes. And manages the UI. When player pushes the next / previous vehicle buttons, manager script will enable only selected vehicle in the list. All UI buttons are using methods in the manager script.



Main menu must contain these objects;

An empty gameobject with “[TCS_MainMenuManager](#)”

An empty gameobject for the spawn point (It’s just a transform, and should be selected by the [TCS_MainMenuManager](#))

Main Camera (With [TCS_ShowroomCamera](#) attached)

And UI panels / buttons / elements for vehicle selection, scene selection, loading and loading slider (Optional). UI is optional, but I think it’s required to control the scene.

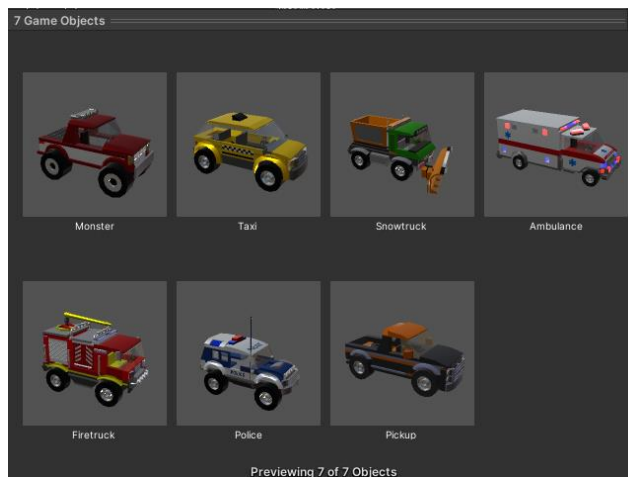
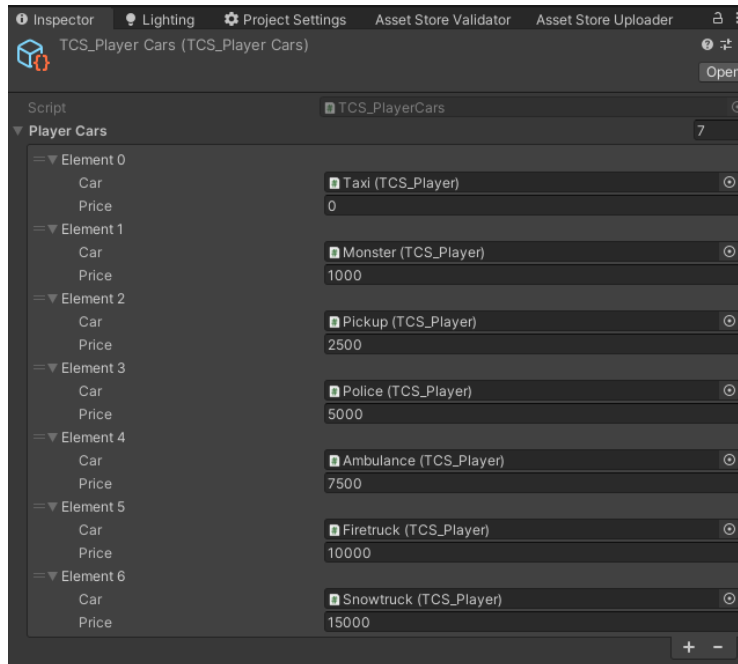
How the gameplay system works

Gameplay scenes includes a manager script named “[TCS_GameplayManager](#)”. It loads the latest selected vehicle by the player with index value. Spawns the correct player vehicle at the spawn point. Registers it as a player vehicle with RCCP’s API. Calls events when player spawns / dies. Let the other scripts access to this manager script to do actions on the player.

[TCS_GameplayManager](#) requires only a spawn point. All gameplay scenes must have this script.

How to add / remove / edit vehicles

All selectable player vehicles are stored in the [TCS_PlayerCars](#). In order to access to the list, go to **Tools** → **BCG** → **TCS** → **Player Vehicles**. Or “[TCS_PlayerCars](#)” asset in the “[Resources](#)” folder. You can set each vehicle’s price here as well. To add new vehicles, increase the list number by one and select your new vehicle prefab. How to create a new vehicle has been explained below. If you want to remove any specific vehicle, right click on it and select “[Delete Array Element](#)”. This will remove the stated vehicle from the list, but it will still exist in your project as a prefab. If you want to delete the vehicle prefab too, all vehicles are stored in the “[Prefabs/Vehicles](#)” folder. You can delete it from here as well.



How to create a new vehicle from the scratch (with tutorial video)

All vehicles are powered by Realistic Car Controller Pro. You can read documentations of Realistic Car Controller Proto how to create new vehicles, or you can just watch this tutorial to build your own vehicles below;

<https://youtu.be/hYJRIMTeWwo>

If you've used Realistic Car Controller Pro earlier, everything is the same. You create your own vehicle with Realistic Car Controller Pro's directions, and add the "TCS_Player" script to the vehicle. Create a prefab version of your vehicle, and add it to the list. And it's done.

1. Open up the scene named "Test_Vehicles".
2. Drag and drop your model to the scene.
3. Add main controller through the **Tools** → **BCG** → **RCCP** → **Add Main Controller**.
4. Create your vehicle with Realistic Car Controller Pro and take a test ride.
5. Be sure your vehicle is operating correctly.
6. Add "TCS_Player" script to the vehicle
7. Create prefab version of your vehicle by simply clicking the "Create Prefab" button.
8. Add your prefab to the list by simply clicking the "Add To List" button.
9. All other settings such as engine torque, brake, speed, battery consumption, etc up to you.

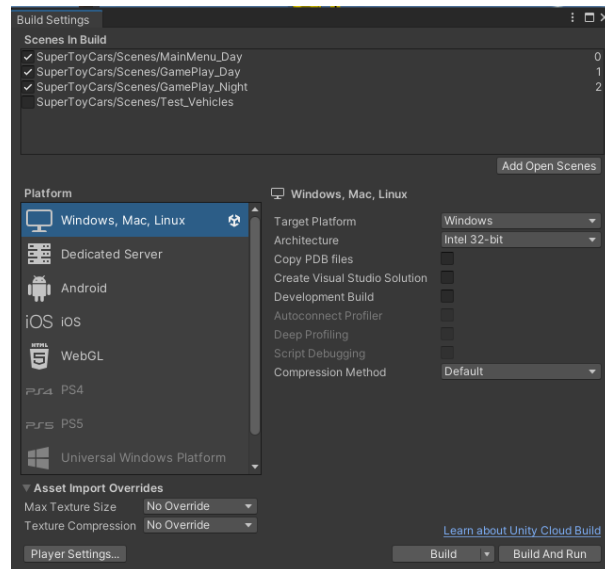


Note: If you've done changes on your prefab, be sure to click "Save Prefab" button to save the changes. Otherwise your changes will not be saved.

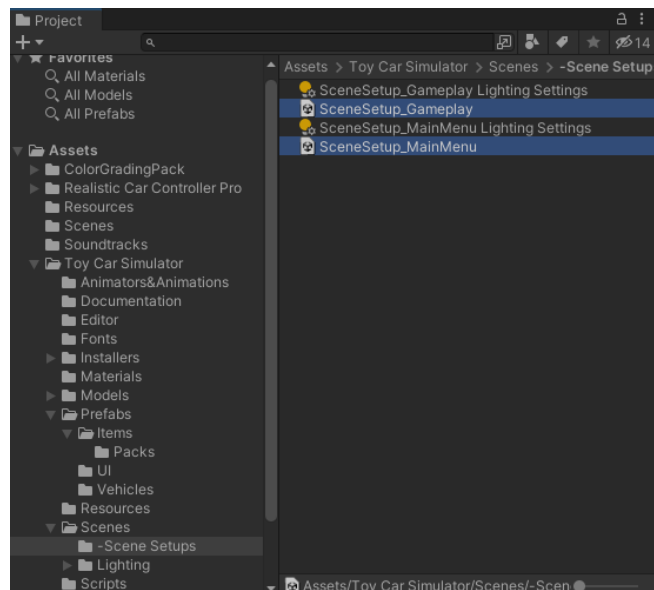
Note: All systems related to the vehicle physics is related to Realistic Car Controller Pro. If you want to edit acceleration, speed, steering, handling, or other settings, you need to tweak settings of the Realistic Car Controller Pro. TCS has nothing to do with the vehicle physics.

How to add / remove / edit scenes

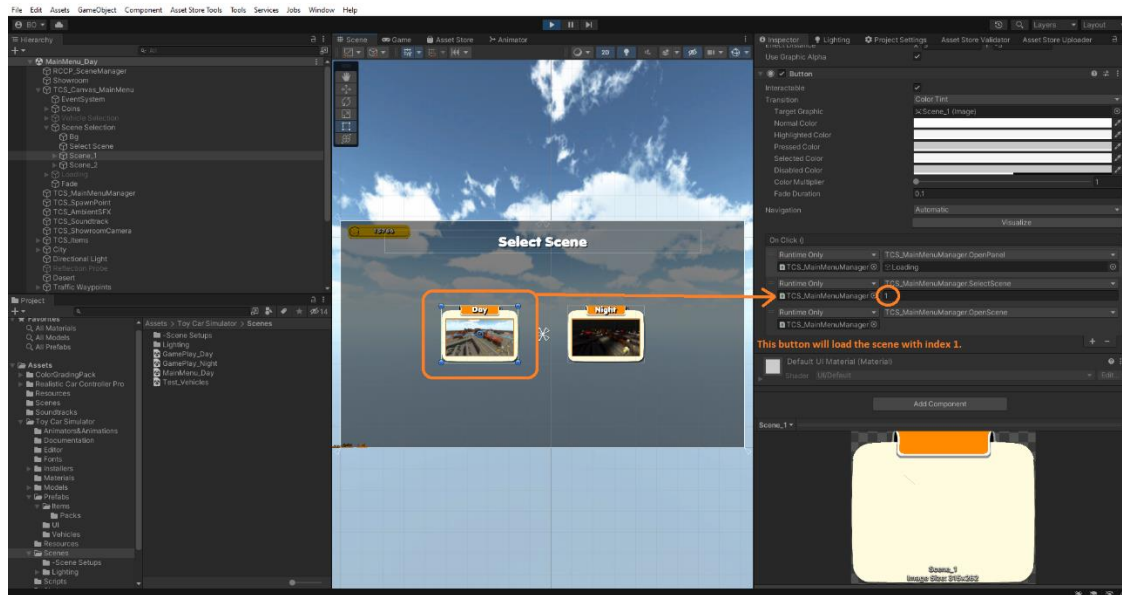
All included scenes are added to the build settings. To access build settings, go to **File → Build Settings**. If you want to remove or add any scene, this list will be used.



So, how to create a new main menu or gameplay scenes? There are two template scenes for the main menu setup and gameplay setup. These scenes don't contain any models, just necessary objects and scripts. You can use these template scenes to build your own scenes. These scenes are located in the “**Scenes/-Scene Setups**” folder. One for main menu, and the other one for gameplay.



When you created your own scene for the main menu, or gameplay, be sure to add it to your build settings. Open up the build settings and add your new level to the list. Each scene has an index value. We'll be using these index values to load the corresponding scene selected by the player at the main menu. Each scene selection button in the main menu scene uses a method in the **TCS_MainMenuManager**. Each button has a target index value. If this UI button should open the night scene, index of the night scene must be used.



How to add / remove / edit items

All items have **TCS_Item** script attached to them. You can specify the battery amount (if it's a battery item) and coins amount (if it's a coins item). Currently demo has **25, 50, 100** and **500** coins pack. But you can add your own coins. Items must have a box collider with trigger enabled. When player triggers with the collider, item will be taken by the player. Items have a few options to edit.

Scripts

TCS_API

TCS contains simple API to do such things like add / consume coins, lock / unlock vehicles, get coins. API methods have been explained below.

TCS_AudioSource

Creating and managing audiosources at runtime with one line of code.

TCS_GameplayManager

Spawns the player, observes the gameplay, and manages the scene.

TCS_Item

Attached to the pickup items such as coins and batteries. Box collider with trigger enabled must be added.

TCS_MainMenuManager

Instantiates all selectable player vehicles at once, disables all of them, and enables only selected vehicle by the player. Saves the selected vehicle with index, this index will be used to spawn selected vehicle on the other scenes. And manages the UI. When player pushes the next / previous vehicle buttons, manager script will enable only selected vehicle in the list. All UI buttons are using methods in the manager script.

TCS_Player

Attached to the player vehicle. It contains battery amount, collected coins and overspeed coins. Controls the main car controller component ([RCCP_CarController](#)) to enable / disable the control of the vehicle. When battery dies, disables the control of the vehicle and let the manager script knows it.

TCS_PlayerCars

List of all selectable player vehicles with prices. If price is 0, vehicle will be unlocked by default.

Tools → BCG → TCS → Player Vehicles.

TCS_Settings

Contains global shared settings and soundtracks. **Tools → BCG → TCS → Edit Settings.**

TCS_ShowroomCamera

Used in main menu camera for orbiting effect. Has a target point to look with variable speed.

TCS_Soundtrack

Creates soundtracks at runtime. All soundtracks are taken from the [TCS_Settings](#).

TCS_UI_CoinsAnimator / TCS_UI_EnergyAnimator / TCS_UI_OverspeedAnimator

Used on UI elements to animate specific animations when necessary.

TCS_UIManager

Manages the main UI canvas. Listening events of the player (spawn / die), enables / disables the panels, displaying texts and feeding slider values (for coins, battery, and nos).

How to use the simple API

TCS contains simple API to do such things like add / consume coins, lock / unlock vehicles, get coins. Open up the [TCS_API](#) script for the methods. Also they are explained below;

TCS_API.GetCoins = returns player coins as int

TCS_API.ChangeCoins(amount) = changes the coins with x amount. X amount can be negative value too.

TCS_API.LockVehicle(int) = locks the target vehicle with int index value.

TCS_API.UnlockVehicle(int) = unlocks the target vehicle with int index value.

Vehicles

All vehicles are powered by Realistic Car Controller Pro. In order to create / edit / customize vehicles, you need to read documentation of the [Realistic Car Controller Pro](#). You can create new vehicles same as other [Realistic Car Controller Pro](#) vehicles.

Deleting demo contents of the Realistic Car Controller Pro

Realistic Car Controller Pro includes 9 demo vehicles. These vehicles will be included in the build, because they have been selected by a scriptableObject placed in the resources folder. But TCS is not using any of them. If you are not planning to use any demo content of the Realistic Car Controller Pro, I would recommend you to delete them. In order to delete the demo content, go to **Tools → BCG → RCC Pro → Welcome Window** and click delete demo contents button. This will delete all unused demo contents from the project and reduce the total size of your build.

URP

Project is using universal render pipeline (**URP**), and it should be installed automatically if you choose install dependencies while importing the package. But you may want to update your URP to the latest version for some Unity versions. If you are having pink materials, or missing materials on your scene, you can update your URP to the latest version. In order to do that, open up the **Package Manager** through the **Window → Package Manager**. And search for **Universal RP**. Clicking update button will update your URP to the latest version and it will ask you to update all materials in the project. Choose “**Yes**” to update all materials in the project.