Realistic Car Controller V3.3 by BoneCracker Games

RCC.cs

Explained in seperate document.

RCC_AIBrakeZone.cs

Brake Zones are meant to be used for slowing AI vehicles. If you have a sharp turn on your scene, you can simply use one of these Brake Zones. It has a target speed. AI will adapt it’s speed to this target speed while in this Brake Zone. It's simple.

RCC_AIBrakeZonesContainer.cs

Used for holding a list for brake zones, and drawing gizmos for all of them on Editor.

RCC_AICarController.cs

AI Controller of RCC. It's not professional, but it does the job. Follows all waypoints, or chases the player. Must be attached to root of the vehicle. RCC_CarControllerV3.cs will not receive any inputs from player when this script attached to the vehicle.

RCC_AIWaypointsContainer.cs

Used for holding a list for waypoints, and drawing gizmos for all of them.

RCC_APIExample.cs

An example script to shows how the RCC API works. Uses the RCC.cs.

RCC_Caliper.cs

Rotates the caliper.

RCC_Camera.cs

Main RCC Camera controller. Includes 7 different camera modes with many customizable settings. It doesn't use different cameras on your scene like *other* assets. Simply it parents the camera to their positions that's all. Also supports collision detection.

RCC_CarControllerV3.cs

Main vehicle controller that includes Wheels, Steering, Suspensions, Mechanic Configuration, Stability, Lights, Sounds, and Damage. All In One script.

RCC_ChangableWheels.cs

Changes wheels (visual only) at runtime. It holds changable wheels as prefab in an array.

RCC_CharacterController.cs

Animates Driver Sofie (Credits to 3DMaesen). Simply feeds floats and bools of Sofie's animator component.

RCC_Chassis.cs

Simulates chassis movement based on vehicle rigidbody velocity.

RCC_CinematicCamera.cs

Tracks the car and keeps orientation nicely for cinematic angles. It has a pivot gameobject named "Animation Pivot". This gameobject has 3 animations itself currently.

RCC_CreateAudioSource.cs

Creates new audiosources at runtime with specified settings.

RCC_Customization.cs

Main Customization Class For RCC.

RCC_CustomizerExample.cs

A simple customizer example script used for receiving methods from UI elements and send them to RCC_Customization script. Also updates all UI elements for new spawned vehicles too.

RCC_DashboardColors.cs

Changes HUD image colors by UI Sliders.

RCC_DashboardInputs.cs

Receiving inputs from active vehicle on your scene, and feeds dashboard needles, texts, images.

RCC_DashboardObjects.cs

Receiving inputs from active vehicle on your scene, and feeds visual dashboard needles.

RCC_Demo.cs

A simple manager script for all demo scenes. It has an array of spawnable player vehicles, public methods, setting new behavior modes, restart, and quit application.

RCC_Exhaust.cs

Exhaust based on Particle System. Based on vehicle engine RPM.

RCC_FixedCamera.cs

Fixed camera system for RCC Camera. It simply parents the RCC Camera, and calculates target position, rotation, FOV, etc...

RCC_FOVForCinematicCamera.cs

Animation attached to "Animation Pivot" of the Cinematic Camera is feeding FOV float value.

RCC_GetBounds.cs

Gets total bound size of a gameobject.

RCC_GroundMaterials.cs

Configurable Ground Materials are collected in an array of class.

RCC_HoodCamera.cs

RCC Camera will be parented to this gameobject when current camera mode is Hood Camera.

RCC_Light.cs

General lighting system for vehicles. It has all kind of lights such as Headlight, Brake Light, Indicator Light, Reverse Light.

RCC_LightEmission.cs

Feeding material's emission channel for self illumin effect.

RCC_Mirror.cs

It must be attached to external camera. This external camera will be used as mirror.

RCC_MobileButtons.cs

Receiving inputs from UI buttons, and feeds active vehicles on your scene.

RCC_PoliceSiren.cs

Flashes red and blue lights with proper timing. If vehicle is an AI vehicle and chaser vehicle, toggles lights on / off automatically.

RCC_Recorder.cs

Record / Replay system. Saves player's input on record, and replays it when on playback.

RCC_SceneManager.cs

Scene manager that contains current player vehicle, current player camera, current player UI, current playe r character, recording/playing mechanim, and other vehicles as well.

RCC_Settings.cs

Stored all general shared RCC settings here.

RCC_ShadowRotConst.cs

Locks rotation of the shadow projector to avoid stretching.

RCC_Skidmarks.cs

Skidmarks Manager for RCC.

RCC_SuspensionArm.cs

Rotates and moves suspension arms based on wheelcollider suspension distance.

RCC_TruckTrailer.cs

Truck trailer has additional wheelcolliders. This script handles center of mass of the trailer, wheelcolliders, and antiroll.

RCC_UIController.cs

UI input (float) receiver from UI Button.

RCC_UIDashboardButton.cs

UI buttons used in options panel. It has an enum for all kind of buttons.

RCC_UIDashboardDisplay.cs

Handles dashboard elements.

RCC_UIDrag.cs

Mobile UI Drag used for orbiting RCC Camera.

RCC_UISliderTextReader.cs

Receives float from UI Slider, and displays the value as a text.

RCC_UISteeringWheelController.cs

UI Steering Wheel controller.

RCC_UnetNetwork.cs

Streaming player input, or receiving data from server. And then feeds the RCC.

RCC_WheelCamera.cs

RCC Camera will be parented to this gameobject when current camera mode is Wheel Camera.

RCC_WheelCollider.cs

Based on Unity's WheelCollider. Modifies few curves, settings in order to get stable and realistic physics depends on selected behavior in RCC Settings.

Photon Scripts

RCC_PhotonDemo.cs

A simple manager script for photon demo scene. It has an array of networked spawnable player vehicles, p ublic methods, restart, and quit application.

RCC_PhotonManager.cs

Connects to Photon Server, registers the player, and activates player UI panel when connected.

RCC_PhotonNetwork.cs

Syncs the player. Streams player input, or receiving data from server. And then feeds the RCC.