Unique Vehicle Controller Manual v1.0.0

Introduction

For an indie developer, creating a video game isn't always something straight forward. Many developers struggle with where to begin when they want to start their project. Game development involves lots of learning, and that means hours upon hours of research to ensure that the project is feasible.

But there are also plenty of tools that can help speed things up as well.

Unique Vehicle Controller is a lightweight vehicle controller tool that helps create semi-realistic car mechanics with ease, Based on a custom wheel collider, Suitable for Simulation / Parking Games.

The Unique Vehicle Controller not only controls the behavior of the vehicle. It includes all the main features of the vehicle like (Lights , Visual Effects, Sound Effects, Camera Behavior...)

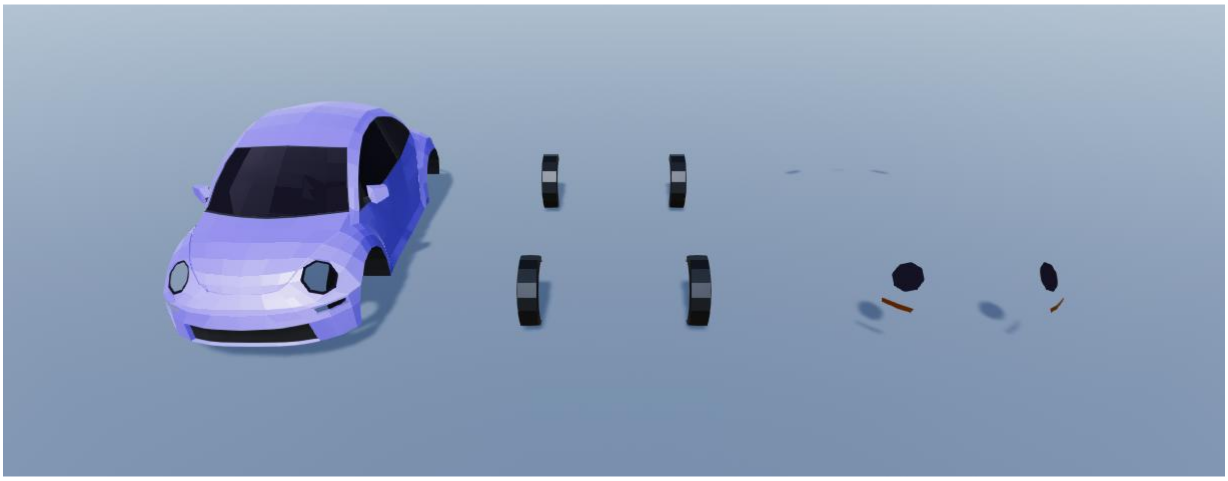
Unique Vehicle Controller also provides:

- Desktop and Mobile support
- Three drive types: All Wheel Drive, Front Wheel Drive and Rear Wheel Drive
- Anti-roll bar prevents the car from rolling over
- Ackerman steering
- Adjustable center of mass
- Anti-lock braking system (ABS)
- Power Steering Reducing Steering Efforts Needed To Turn The Wheel
- Full Camera System
- Full Vehicle Light System
- Fuel System
- Parking Sensors
- Distance Remaining & Travelled Indicators
- The most common units of speed: KM/H & MPH

Basics

Prepare Your Vehicle

Vehicles (Body, Wheels, and Head & Brake & Fog & Turn Lights) Meshes must be separated.



For a better understanding, please check out our free add-on

UVC - Unique Vehicle Controller > Add-ons > Favoured Cars Pack - Low Poly

Attach the **UVCUniqueVehicleController.cs** to your vehicle's GameObject, you will notice that all the scripts you need have been added automatically.

Learn more about these scripts bellow

Attach **UVCWheelCollider.cs** to your vehicle's wheels, Wheel Radius can be changed on **UVCUniqueVehicleController**.cs

Now drag all **Helpers** prefabs (Audio, Cameras, Effects, and Lights) that you will find in:

UVC - Unique Vehicle Controller > Prefabs > _Helpers to your vehicle's GameObject.

In order to operate the car, we need buttons to click and commands to call.

So, you need drag **Canvas** prefab into your scene. You will get an adjustable Mobile interface that enables you to play.

Don't Forget Tags

Unique Vehicle Controller

Vehicle Setup

Start Engine On Awake	Enable the checkbox to automatically start Vehicle’s engine
ABS System	Enable the checkbox to enable anti-lock braking system
Car ID	Used to store vehicle Remaining & Travelled distance, use different ID for each vehicle
Engine start duration	Time the engine takes to turn on
Center Of Mass	Center Of Mass relative to the transform's . origin. If you don't set the center of mass from a script it will be calculated automatically from all colliders attached to the rigidbody. Setting the center of mass is useful to make vehicles more stable

Engine Specifications

Max Engine Torque	Maximum torque that engine can reach
Max Engine Speed	Maximum speed that engine can reach
Brake Torque	Maximum brake torque value
Gears	Number of forward gears

Steering Setup

Steering Wheel	Interior's **steering wheel transform**[1]
Min Steering Angle	Maximum steer angle, also used for skid factor
Max Steering Angle	Minimum steer angle
Arrow Steering Speed	Determines how fast the steering wheels react on arrows steering input
Handling	Determines the way the vehicle responds and reacts to the steering inputs
Anti-Roll	Intended to force each side of the vehicle to lower, or rise, to similar heights, to reduce the sideways tilting (roll) of the vehicle on curves, sharp corners, or large bumps
Power Steering	Reduces steering effort needed to turn the wheel (Steering wheel control only)

Wheels Setup

Drive Type	The three different types of Drivetrains: All-wheel drive, front-wheel drive, and rear-wheel drive
Wheel Calipers	List of wheels calipers transform
Wheel Colliders	List of wheels. Every wheel in the list must has **UVCWheelCollider.cs** attached

GearBox Setup

Vehicle Type	Ready presets for your gearbox system, choosing custom let you specifies your own ratios
Gear Ratios	The list that specifies how many gears there are and their ratios. The ratio of the reverse gear and the first gear is the same

Input System

Visuals

Speed Text	Text that displays vehicles speed
Speed Needle	Speedometer's speed indicator
Fuel Needle	Fuel indicator
Distance Travelled Text	Text that displays vehicles travelled distance
Distance Remaining Text	Text that displays vehicles Remaining distance
Gear D | N | R | P	Gear stick lamp UI/Image

Buttons

Engine Start Button	Button that starts the engine. **StartEngine()**
Engine Start Fill	Image that displays engine starting progress **HoldStart()**
Engine Stop Button	Button that stops the engine **StopEngine()**
Light OFF Button	Button that turn off lights **TurnOffLight()**
Fog Light Button	Button that turn on fog lights **EnableFogLights()**
Low Beam Light Button	Button that turn on low beam lights **EnableLowBeamLights()**
High Beam Light Button	Button that turn on High beam lights **EnableHighBeamLights()**
Hazard Light Button	Button that turn on Hazard lights **EnableHazardLights()**

Right Blinker Button	Button that turn on right blinker
	EnableRightBlinker()
	DisableRightBlinker()

Settings Button	Button that opens Settings Menu

Speed Unit Drop	Speed unit selector **ChangeSpeedUnit()** **- 0 ➔ KM/H** **- 1 ➔ MPH**

Left Blinker Button	Button that turn on left blinker
	EnableLeftBlinker()
	DisableLeftBlinker()

Components

Gear Slider	Gear stick slider **SwitchGear()** **- D ➔ 0** **- N ➔ 1** **- R ➔ 2** **- P ➔ 3**

Steering Arrows	Arrows Steering Buttons
	CarIsGoingRight()
	CarIsGoingLeft()

Steering Wheel	UI Steering wheel gameobject

Sound Effects

Button Click	**ButtonClickSound()**
Light Button Click	**LightButtonClickSound()**
Gear Shift	**GearShiftSound()**

Parameters

Mobile	Enable when working on mobile platform
Max Speed Meter	Maximum speed the speedometer can read
Min Speed Needle Angle	**Minimum Angle**[2] the speed needle can reach
Max Speed Needle Angle	**Maximum Angle**[3] the speed needle can reach
Min Fuel Needle Angle	**Minimum Angle**[2] the fuel needle can reach
Max Fuel Needle Angle	**Maximum Angle**[3] the fuel needle can reach
Accelerator Release Speed	The time it takes for the accelerator pedal to return to its original position
Brakes Release Speed	The time it takes for the brake pedal to return to its original position

Orbit Camera

Distance

Distance	Distance between the camera and the vehicle
Min Distance	Minimum estimated distance between the camera and the vehicle when braking
Max Distance	Maximum estimated distance between the camera and the vehicle when accelerating

Speed

Speed	Camera speed when reset to original position
Speed X	Camera speed when dragging on X axis
Speed Y	Camera speed when dragging on Y axis

Transform

Y Min Limit	Minimum vertical angle to the vehicle
Y Max Limit	Maximum vertical angle to the vehicle
Off Set	The offset of the camera position to the vehicle

Camera Toggler

Attach Your Cameras

Just drag the cameras to the list in the order you want and let the script do its work.

Methods

Switch Between Cameras	public void ToggleCameras() The method switch between cameras

Fuel System

Parameters

Car ID	Used to store vehicles Current Fuel, use different ID for each vehicle
Max Fuel	Vehicle fuel tank maximum capacity
Consumption	Fuel Consumption Every x Seconds, Higher is better

Parking Sensors

Three (3) stages **(Detection, Close Detection, Too Close Detection)** with different beeping frequency and Three (3) colors

Stages

Detection	Default distance to nearby objects = (1.2m)
	State = Safety
	Color = Green

Close Detection	Default distance to nearby objects = (0.85m)
	State = Caution
	Color = Yellow

Too Close Detection	Default distance to nearby objects = (0.5m)
	State = Hazard
	Color = Red

Position

Before editing on the position, enter the edit mode by clicking the “Edit Mode” button, the **sensors will be drawn**[4] in the editor.

Vehicle Lights

Six (6) Different Types of Car Lights, Every vehicle is equipped with several different car light types, each with their own unique and important purposes. And thanks to:

1. Favoured Cars Pack - Low Poly with ***20 different car***
2. Special Cars Pack - Low Poly with ***7 unique cars***
3. Electric Cars Pack - Low Poly with ***5 different car***
4. Favoured Cars Pack - Low Poly (FREE)

Vehicles (Body, Wheels, and Head & Brake & Fog & Turn Lights) Meshes are separated so it's easier to work with.

Lights Setup

GameObject / Mesh **Fog | Brakes | Reverse | Beam | Blinkers Lights**	Drag and drop every light mesh in its proper place
Spot / Point Light **Fog | Brakes | Reverse | Beam | Blinkers Lights**	A Spot light is a light that's located at a point in the Scene and emits light in a cone shape. Used for Beam Lights. A Point light is located at a point in space and sends light out in all directions equally. Used for Brakes / Hazard / Blinkers Lights.
Material **Fog | Brakes | Reverse | Beam | Blinkers Lights**	Two states: - **Enabled ➔** The material that is attached to mesh when its light are on - **Disabled ➔** The material that is attached to mesh when its light are off
Blinking Time	The time between a blink and another

Sound System

Audio Sources

Vehicle Basics - Sound Pack includes a wide variety of typical and everyday automobile noises, making it the perfect choice for supplying the essential sounds for drive simulations and other applications that focus on cars.

This asset was initially created for Unique Vehicle Controller, other engine sound controller assets may not provide the same sound performance or quality.

The Asset comes free with Unique Vehicle Controller.

Audio Parameters

Engine Pitch Boost	Your pitch boost boosts your max pitch, by whatever your max pitch is plus the boost number. And it also the minimum value that your pitch can reach
Engine Pitch Range	Your max pitch value is determined by the pitch range, If you set it to one (1), then your max range will be between zero (if Engine Pitch Boost = 0) and one (1)
Skid Pitch Boost	Same as Engine Pitch Boost, skid sound pitch controlled by engine speed
Skid Pitch Range	Same as Engine Pitch Range, skid sound pitch controlled by engine speed

Wheel Effects

Effects Setup

Wheel Skid Prefab	Skid mark prefab for instantiating trail renderer

UI Components

Steering Wheel

Min Steering Angle	Minimum steering angle used to calculate the force applied to the steering wheel when the Power Steering is disabled
Max Steering Angle	Maximum steering angle is defined as the maximum angle the steered wheels can reach
Release Speed	How fast the steering wheel returns to its original position

Parking Sensors Config

Drag and drop every parking sensor UI element in its proper place, if it's confusing to you, just use the ready **Canvas** prefab located at:

UVC - Unique Vehicle Controller > Prefabs > _Helpers

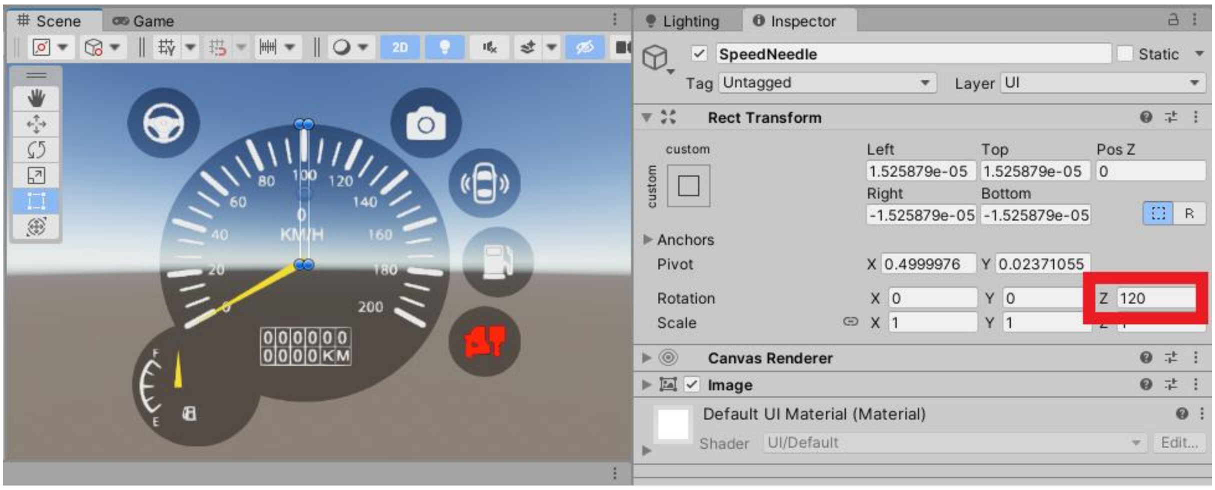
EndNotes

Steering Wheel Transform[1]

If you are having issues with rotating the steering wheel please check your wheel transform pivot

For a better understanding, please check out **Steering Wheels Collection**

Minimum Angle[2]



☑ SpeedNeedle □ Static

Tag Untagged Layer UI

Rect Transform

custom

Left	Top	Pos Z
1.525879e-05	1.525879e-05	0
Right	Bottom	
-1.525879e-05	-1.525879e-05	

Anchors

Pivot X 0.4999976 Y 0.02371055

Rotation X 0 Y 0 Z 120

Scale X 1 Y 1

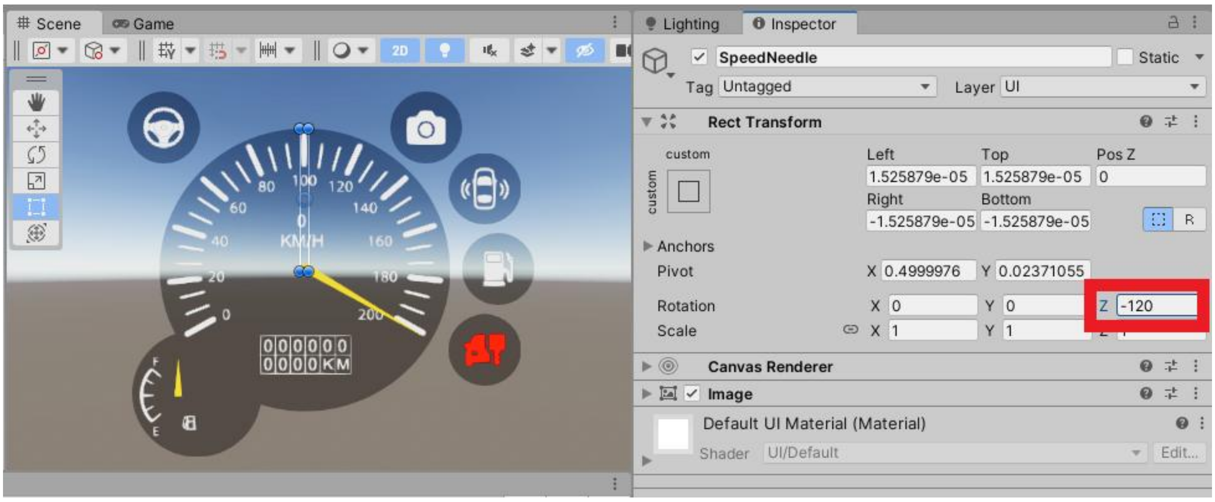
Canvas Renderer

☑ Image

Default UI Material (Material)

Shader UI/Default Edit...

Maximum Angle[3]



SpeedNeedle ☑ □ Static

Tag Untagged Layer UI

Rect Transform

custom

	Left	Top	Pos Z
	1.525879e-05	1.525879e-05	0
	Right	Bottom	
	-1.525879e-05	-1.525879e-05	

Anchors

Pivot X 0.4999976 Y 0.02371055

Rotation X 0 Y 0 Z -120

Scale X 1 Y 1

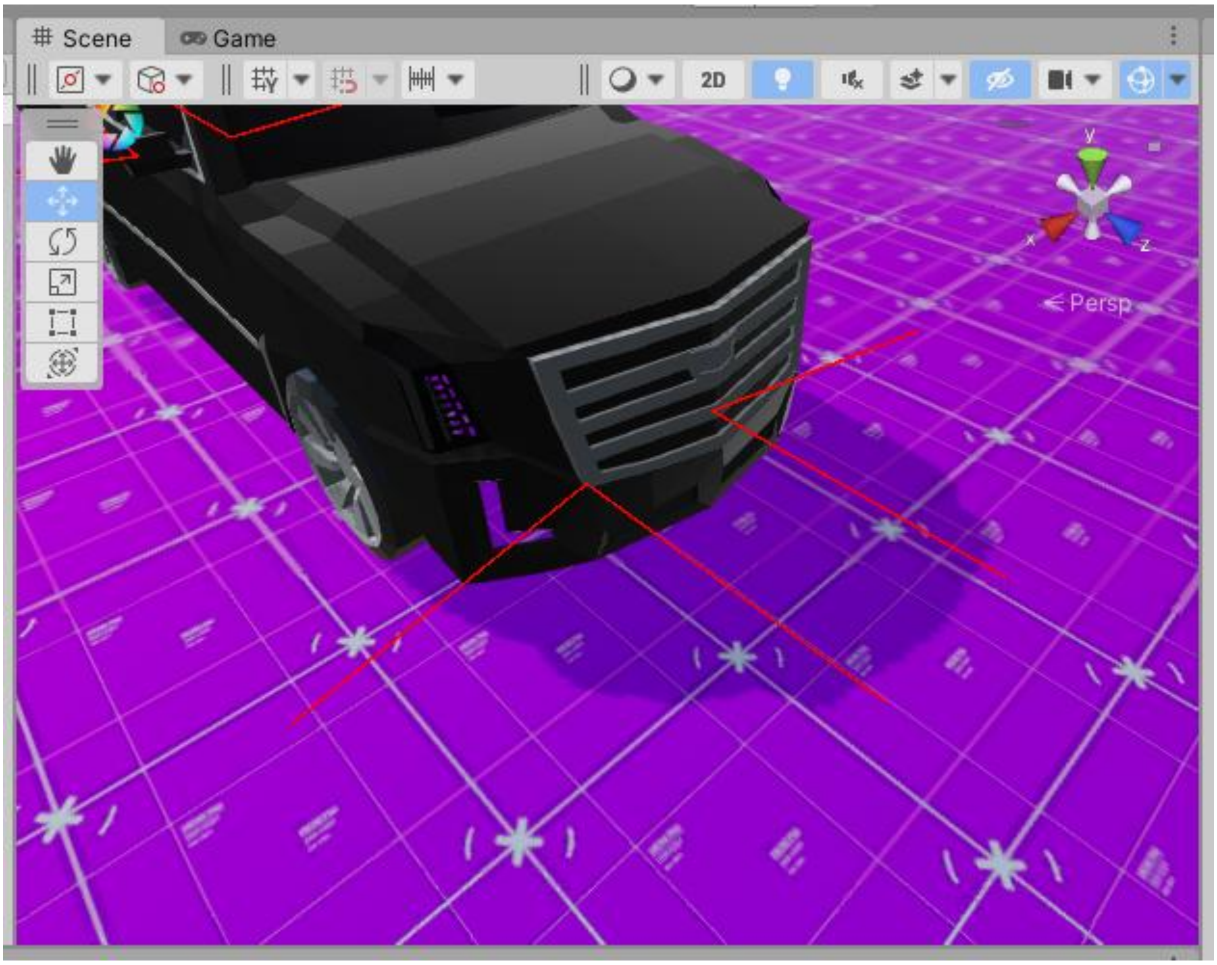
Canvas Renderer

☑ Image

Default UI Material (Material)

Shader UI/Default Edit...

Sensors Will Be Drawn[4]



Facts

- Arrows are very good for instant movement turn from one direction to the next quickly
- Steering wheel is one of the realistic controls because it has a feedback effect when the steering wheel rolls back while driving
- The sound of skidding is controlled by the speed of the car, the faster the car, the louder the sound
- Parking sensors do not operate when the speed is more than 25 KM/H or 15 MPH
- Brakes are weaker when the engine is turned off, and the brake pedal force increases and releases faster
- You can ignore the fog lights and the vehicle will work normally
- You need to turn off the engine when refueling

Support

Please send any queries or bug reports here: **imolegstudio@gmail.com**.