Ilumisoft

Arcade Racing Kit

Documentation

Inhalt

Welcome

Welcome to the Arcade Racing Kit Documentation. In the following articles you will find all the information you need to get started.

Online Documentation

For the best experience, we recommend using our online documentation!

Setup Project

- Open the Unity Hub and create a new **3D project** suing **Unity 2019.4**
- Download and import Arcade Vehicle Kit from the **Asset Store** or the **Package Manager**

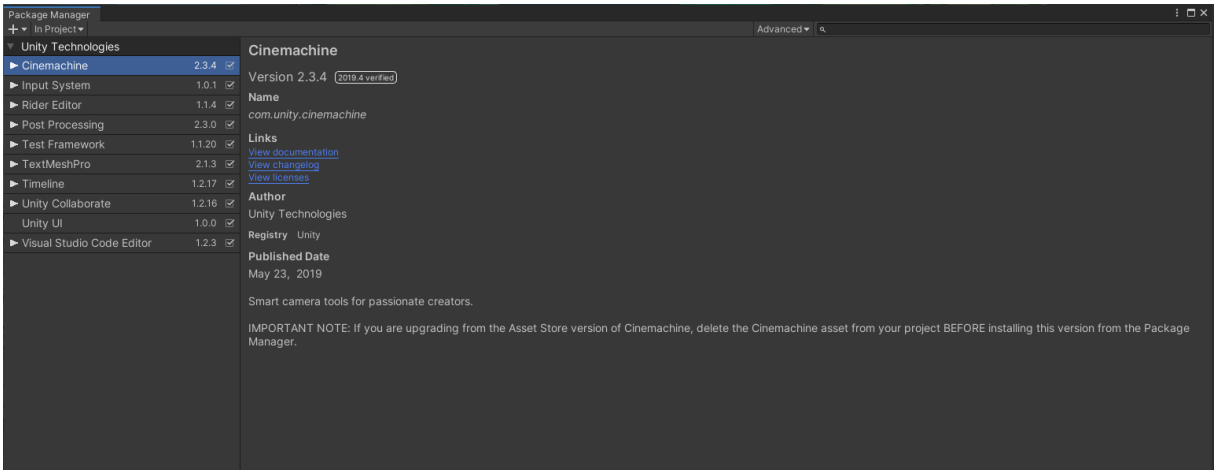
Setup Required Packages

To work properly, **Arcade Racing Kit** requires the following packages from the Package Manager

- **Cinemachine 2.3.4**
- **Post Processing 2.3.0**
- **Input System 1.0.1**

While Unity should automatically install all required packages when **Arcade Racing Kit** gets imported, it can still happen that errors occur. In that case follow these steps:

1. In the Unity Editor click **Help -> Reset Packages to defaults.** This will probably result in alot of error messages being thrown due to missing packages.
2. Open the Package Manager by clicking **Window -> Package Manager.**
3. Select and install all the packages (with the mentioned versions) from the list above



When installing the **Input System**, a popup will occur asking whether you want to disable the old input system and restart Unity. Confirm this message!

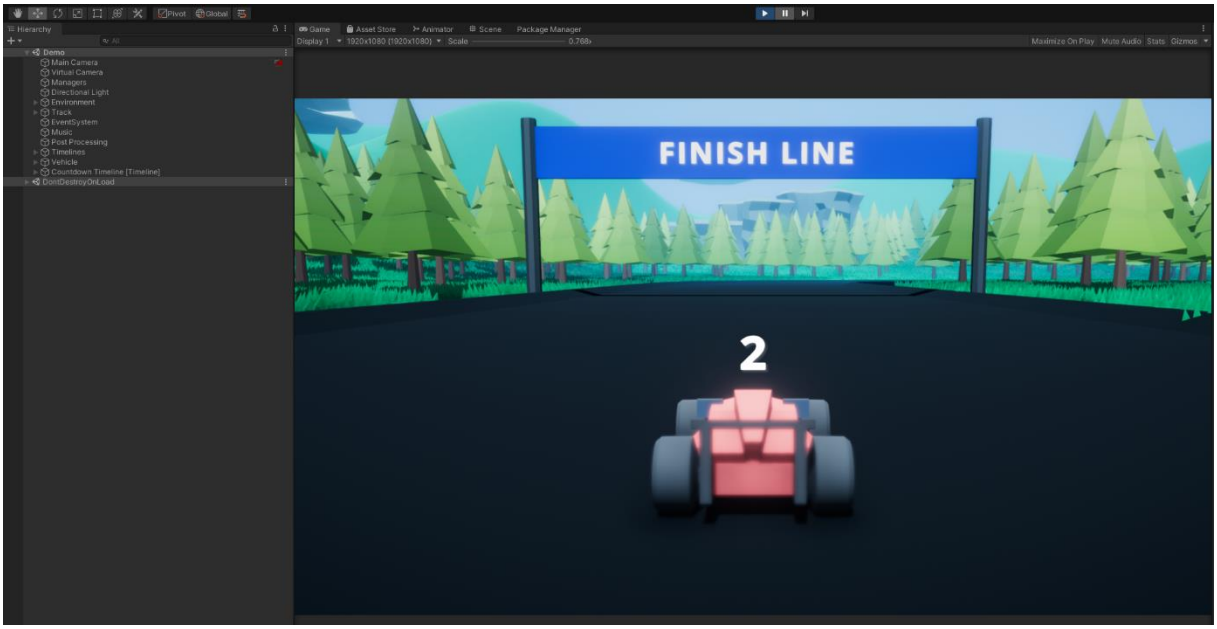
Run the Example Scene

We added an example scene to the project, which contains a clean setup of all systems and is the best way to get started with Arcade Racing Kit!

To run the example scene open

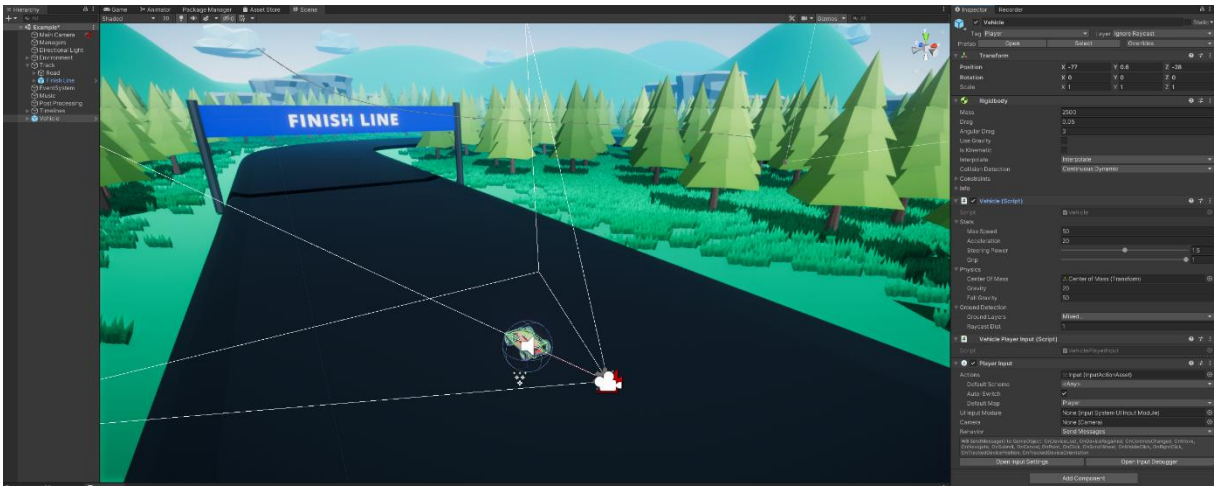
Assets/Ilumisoft/Arcade Racing Kit/Example/Scenes/Example.unity

and start the Playmode**.**



Most Important Components

- Vehicle

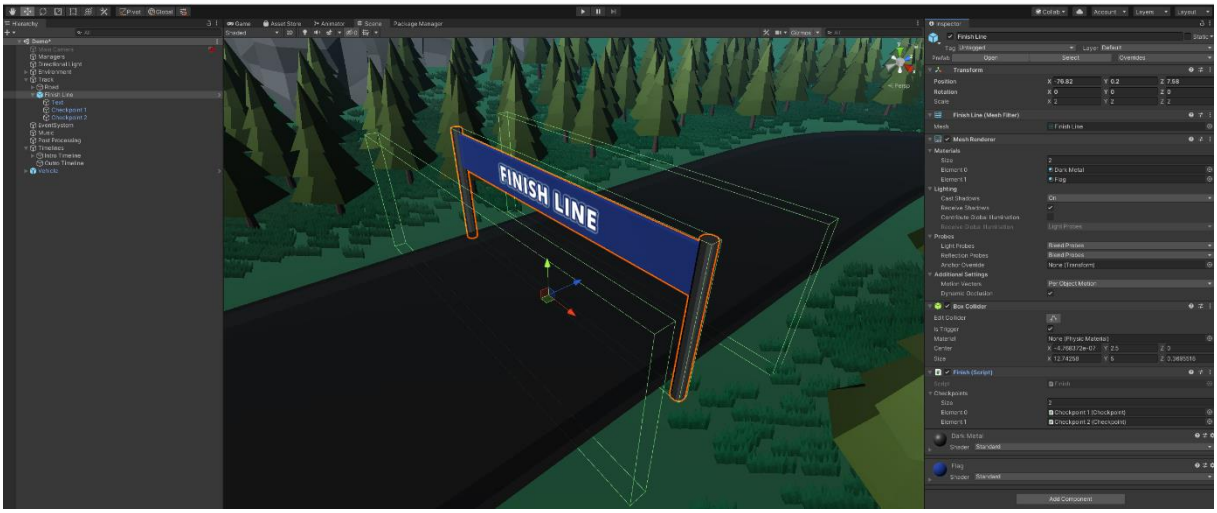


The vehicle is the main component of the kit. When you add a new vehicle component to a game object, the following behaviours are added automatically

- **Vehicle** Main component performing the driving logic
- **Rigidbody**
- **Vehicle Player Input** An input source collecting the player input via Unity's new input system.

To learn how to setup a vehicle from scratch, follow our tutorial. For an example of a fully setup vehicle, check out the Example scene.

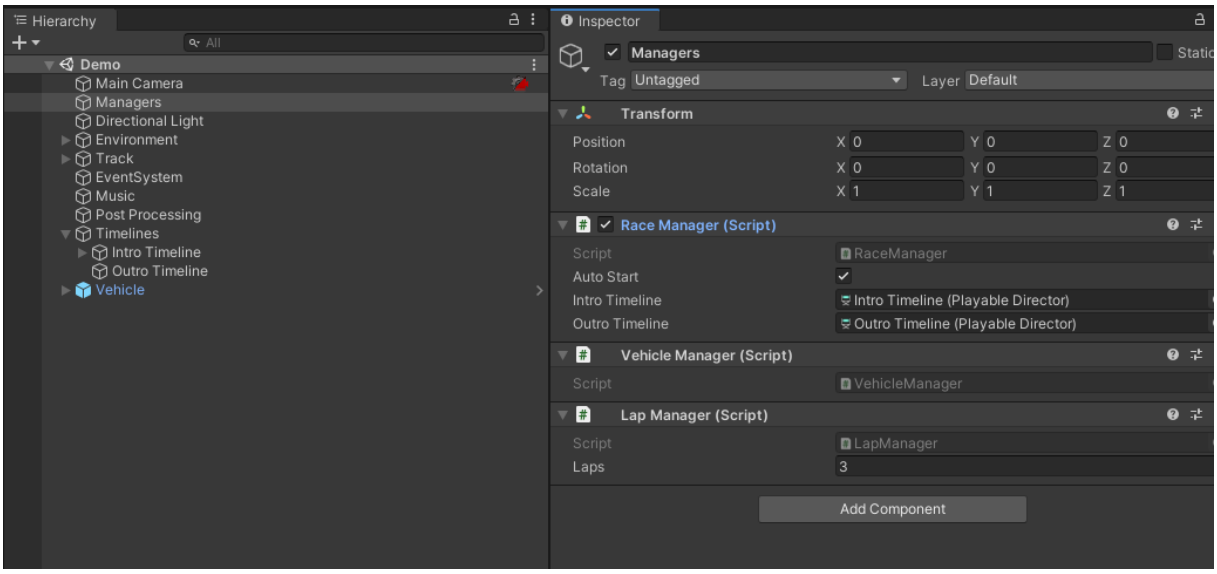
- Finish Line



The finish line marks the finish of a lap and invokes an event when a vehicle is entering it.

To work properly it should be setup with at least two checkpoints. When setup this way the Finish Line will only consider vehicles which first passed Checkpoint 1, then Checkpoint 2 and finally the Finish Line itself, in order to prevent cheating. Check out the example scene to see this in action!

- Race Manager



The race manager controls the state of the race. It will trigger the intro sequence, check whether the player has finished the race and informs all subscribers when the race has begun/ended.

Properties

- **Auto Start** If set to true, the race will be started automatically, when the scene has been loaded
- **Intro Timeline (optional)** Before the race starts, the race manager will lock the movement of all vehicles, play the intro timeline and unlock the vehicles, when the timeline is not playing anymore. That allows you to define in avery flexible way, what should happen before the race begins. E.g. in the demo scene we use the intro timleine to show off the track and play the countdown.
- **Outro Timeline (optional)** The outro timeline is played after the race has been finished. In the demo we use it to disable the movement of the player vehicle and show the race finished UI.

Events

- **OnRaceStarted** This event will be invoked right after the vehicles have been unlocked and the race has been started
- **OnRaceFinished** This event will be invoked right after the player has completed the race

- Vehicle Manager

The vehicle manager tracks all vehicles in the scene and allows all other systems to get access to them in a very performant and flexible way. When a vehicle is created, it automatically registers itself.

The vehicle manager is required by the Race Manager and will be added automatically, if you add the Race Manager to the scene.

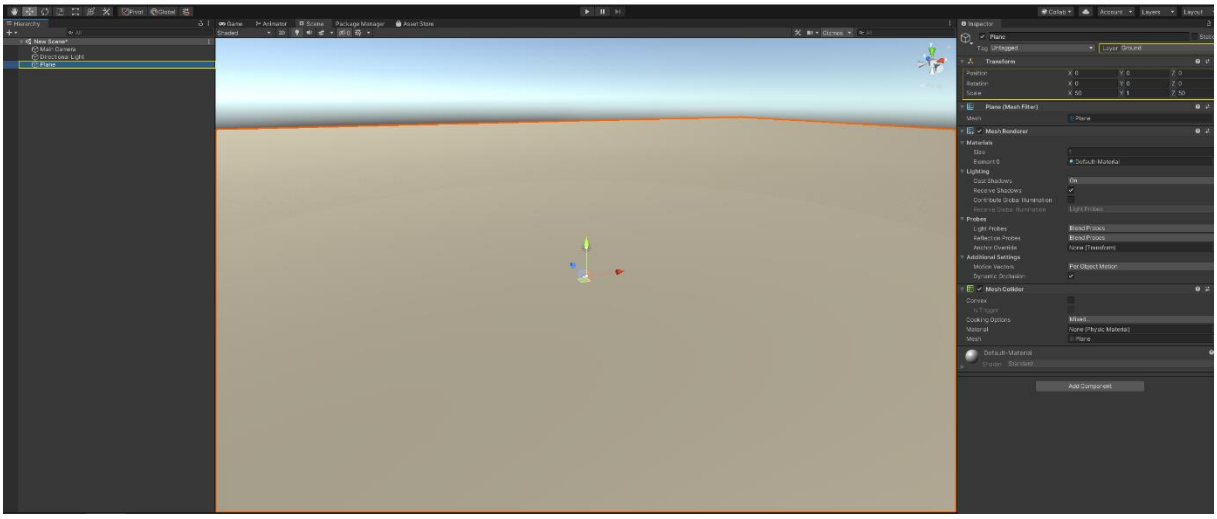
- Lap Manager

The Lap Manager keeps track of how many laps each vehicle has passed and invokes an event when a vehicle passes the Finish Line. In order to work properly, it requires a Finish Line to be in the scene.

Tutorial: Setup a Vehicle from Scratch

1. Scene Setup

1. Create a new scene and open it
2. Create a new 3D Object -> Plane and set the following values

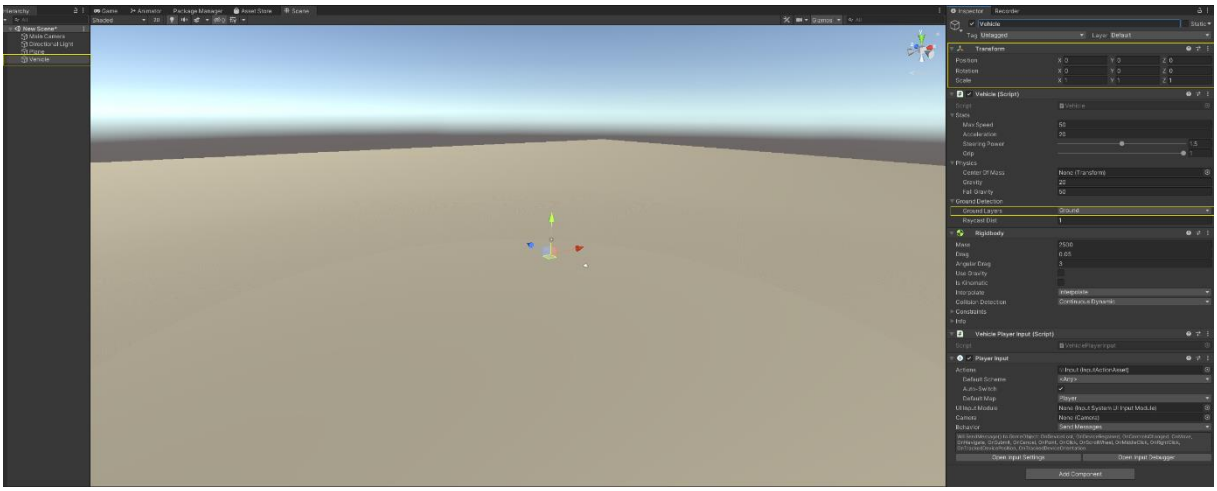


Plane values

Tip: If no Ground Layer exists in your project, create it

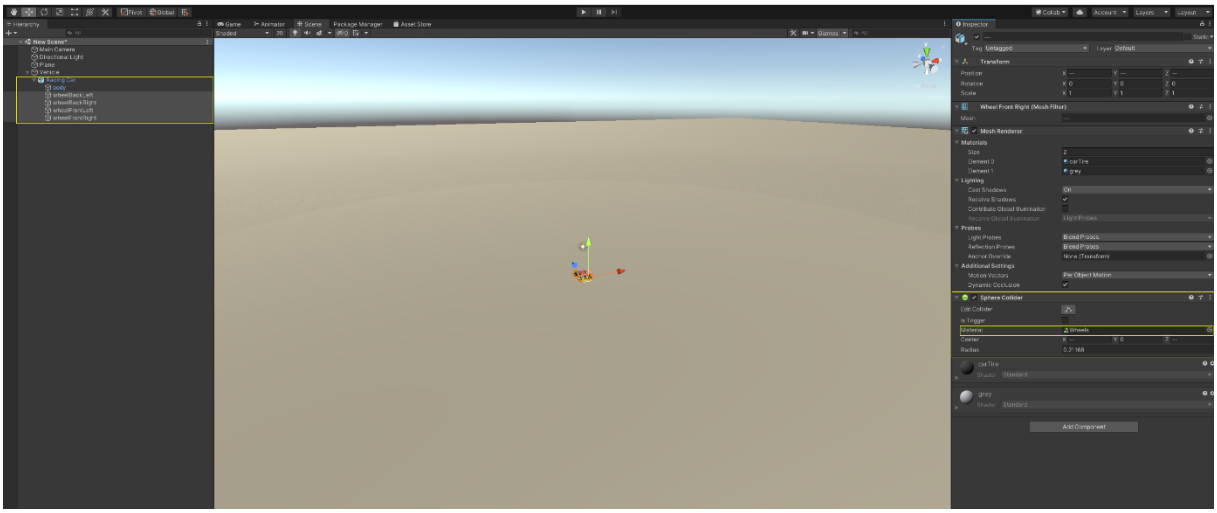
2. Vehicle Setup

Create a new **Game Object** and add a **Vehicle** component. Set the Ground Layers to **"Ground"**



4. Drag and drop the car model from **Assets/Ilumisoft/Arcade Racing Kit/Example/Models/Kenney Racing Kit/Racing Car** onto the vehicle

5. Select the four wheels, add a **sphere collider** to them and set the **Physics Material** to **Wheels**

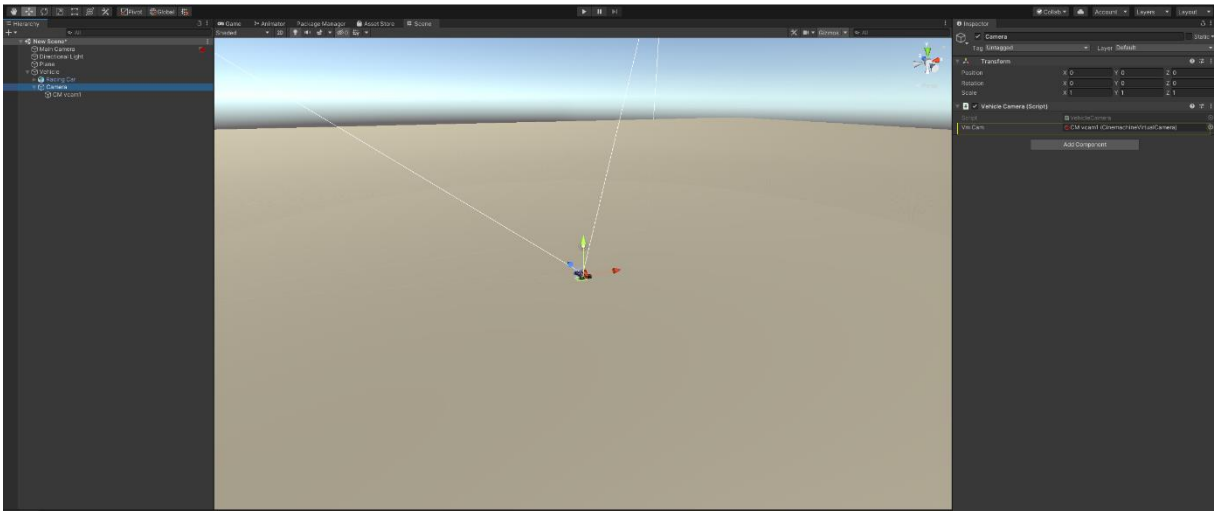


Wheel setup

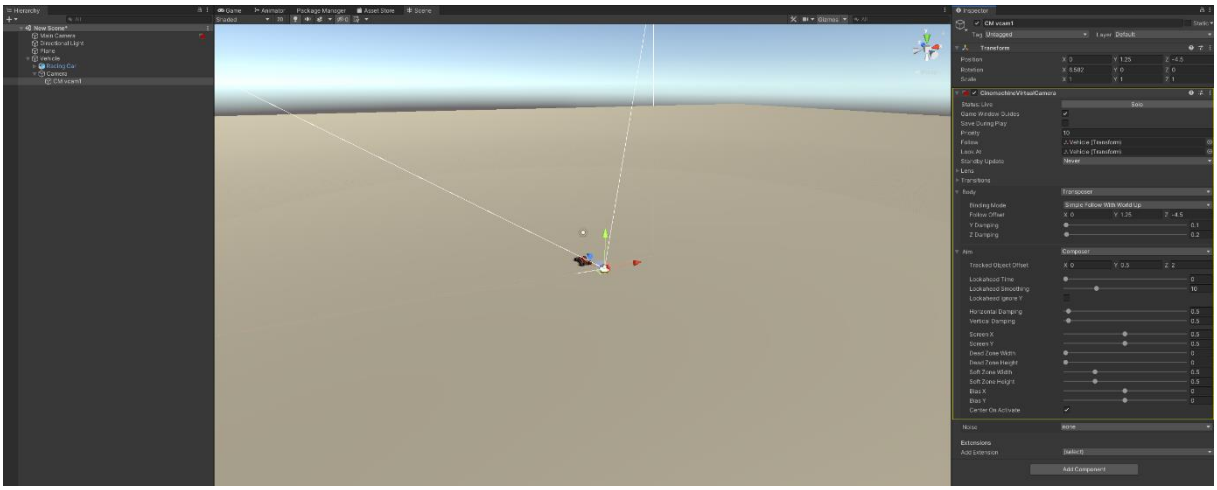
Tip: Setting the **Wheels** physics material is especially important, otherwise the vehicle will not be able to move!

3. Camera Setup

1. Create a new **Game Object**, name it **Vehicle Camera** and attach it to the vehicle
2. Add a **Vehicle Camera** component to the Game Object
3. Select **Cinemachine->Create Virtual Camera** and attach it to the **Vehicle Camera** Game Object
4. In the **Vehicle Camera** component, set the **VmCam** field to the created virtual camera
5. Set the virtual camera inspector fields to the values in **image 3.5**



3.4 Vehicle Camera



3.5 Virtual Camera Setup

Tip: If the Virtual Camera is not positioned behind the vehicle after setting up the values, disable the game object, reset the transform, and then reenable the game object

Congratulations! You have created a basic vehicle from scratch. When entering playmode, the vehicle should be able to drive properly and the camera should be following up.

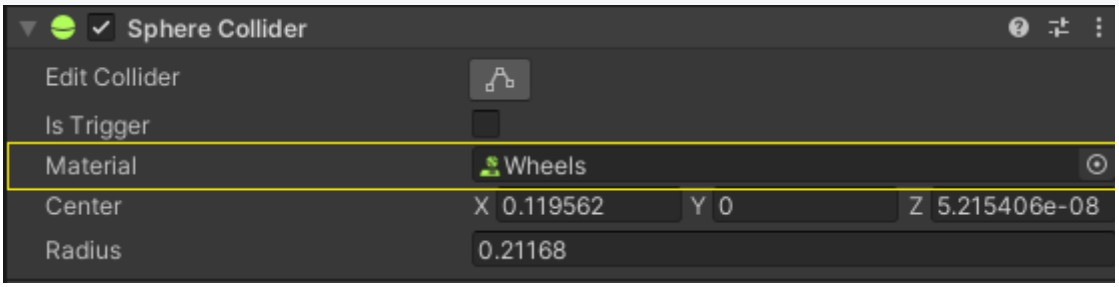
4. Extend the Vehicle

We created all vehicle scripts to work in a very modular way. With the basic setup, you can extend the vehicle further by adding a custom center of mass, effects, audio, and UI. To see a complete example, look at the example scene.

Troubleshooting

- The vehicle cannot drive

Make sure, the wheels of the vehicle have a sphere collider attached and use the Wheels Physics material



Wheel Physics material

Make sure, the **Ground** layer is setup properly in the **Vehicle Ground Detection** and on the game object of the racing track. The vehicle will only detect a collider as ground, if it's layer is set in the vehicles list of Ground Layers!

- The camera is stuttering in the Editor

In Unity 2019.4 we noticed that **Cinemachine** seems to stutter if an Game Object is selected in the inspector when in **Playmode**. **Deselecting** any selected object fixed that!

(Please not this issue is related to Cinemachine and not to the Arcade Racing Kit itself)

- The vehicle hops on composed track parts

If your track is composed of multiple small colliders, it can happen that the vehicle hops sometimes when driving over a contact point of two track parts, due to the way Unity's Physics System work. This can be fixed by setting

Project Settings -> Physics-> Default Contact Offset to **0.0001**

Nevertheless, we recommed to make your track being composed of as few as possible colliders.

Support

You have a problem with the package and need help or do you have a suggestion how the documentation or the package could be improved? Do not hesitate to get in touch with me via **support@ilumisoft.de**.