

# Kart-Racing Physics

[Overview](#)

[Scene Setup](#)

[A note on creating tracks](#)

[Adding a Kart to your scene](#)

[Physics Parameters](#)

[The Kart Object](#)

[Swapping in your own models](#)

[Player Controller](#)

[Triggering Effects](#)

## Overview

Kart-Racing Physics provides an easily tunable arcade-style physics simulation in the vein of classic racing games like Mario-Kart. The focus is on creating a fun racing experience, and not on a realistic simulation of a real vehicle. This means we can do things that a real car wouldn't be able to do simply because it feels more fun, or in order to provide visual feedback to the player.

It also means that instead of the vehicle having complex real-world parameters that need tuning (such as engine torque curves), we can use simple meaningful parameters like top-speed, acceleration time, and so on, which makes tuning the vehicles to get the right feel much more straightforward.

In addition to the basic kart physics Kart-Racing Physics also provides some simple 'effects' for you to use. These can be used to very easily add powerups to your game to add another layer of gameplay alongside simple racing. The effects include speed-boost, speed-penalty, 720 degree spin, a wiggle and a jump. We'll go over how to use each of these in more detail later.

## Scene Setup

There is very little scene setup that needs to be done before we get going. If you are creating a new scene from scratch, you will need to create a new tag named "OffRoad". To do this go to the top of the inspector window and select Tag->Add Tag... and add a new tag with the name "OffRoad". Then select all the ground meshes in the scene that should be considered off-road areas and assign them this tag.

## A note on creating tracks

Use the MeshCollider component to add collision to your track meshes. For the best results, try and make the track mesh as smooth as possible. Hitting even a small bump in the track at high speed can cause the kart to jump.

## Adding a Kart to your scene

The best way to get started is simply to drag the example Kart prefab into your scene. You can find the prefab in the folder KartRacingPhysics/Prefabs/

We'll cover how to swap out the models for your own ones a little later on. For now let's take a look at the various physics parameters.

## Physics Parameters

The physics parameters for the Kart Controller are designed to be simple to understand, whilst still giving a lot of freedom to design vehicles that have different capabilities and feels.

Select the Kart in the hierarchy, then if you look in the inspector you'll find it has both a rigid body component and a Kart Controller script attached. We shouldn't need to change any of the rigid body settings, so we'll leave those as they are and just concentrate on changing the Kart Controller parameters. Let's go through them one by one:

- **Top Speed (MPH)** - This is the top speed of your vehicle in miles per hour. Note that the kart can travel faster than this for short bursts during a speed-boost.
- **Acceleration Time** - This is the time in seconds that it will take the kart to accelerate from stationary to the top speed.
- **Traction** - This is a value between 0 and 1 which determines how much traction the kart has whilst turning. Low values will make the kart slide on corners as if it's driving on ice. Higher values will allow the kart to turn corners quickly without slipping.
- **Deceleration Speed** - This is a value between 0 and 1 which determines how quickly the kart will come to a stop when the throttle is released. Low values will mean the kart slows down very gradually, whilst higher values will cause the kart to come to an abrupt halt.

Below those basic parameters, there are two foldouts - Vehicle Setup, and Advanced parameters. The vehicle setup parameters will be useful when we get to swapping in our own models, or when building a kart from scratch. We'll come back to these later.

Now let's take a look at the more advanced parameters. These are things that you probably won't need to alter very often, but they can give some more customizability when you really want to change the feel of the kart.

- **Steering Speed** - This controls the speed at which the kart turns. Because of the non-physical nature of the simulation we can alter the turning speed without having to specify complex tire traction curves. Higher values will make the kart turn more quickly.
- **Off-Road Drag** - This value is used as a multiplier to control the drag force that is applied when the vehicle is off road. For example a value of 2 means that the kart will experience twice the drag force when it is off-road compared to when it is on the track - thereby slowing it down.
- **Air Drag** - Similar to the off-road drag this is a multiplier that affects the drag force the kart feels when it is in the air (i.e. when the wheels are not touching the ground). A value of

0.5 means it will feel half the amount of drag compared to when the wheels are in contact with the ground.

- Visual Oversteer - This is a value between 0 and 1 which controls how much the body of the kart should be rotated when it is turning. It has no real physical basis, but can help to make it feel like the kart is sliding its way around corners more. It can be useful to get a very arcadey feel. A value of 0 means that the body will not rotate at all, giving a more physically correct look.

## The Kart Object

It is very simple to swap out the kart models for your own ones, but before we do that let's just take a quick look at the way the Kart prefab is structured.

Expand the Kart object in the hierarchy so we can see the various elements. It should look something like this:

Kart	← contains rigid body and Kart Controller script
Body	← contains box collider component
BodyMesh	← mesh object for the kart body
Wheels	
FrontLeft	← gives position of front left wheel
WheelMesh	← should be positioned at (0,0,0) relative to parent
FrontRight	← gives position of front right wheel
WheelMesh	← should be positioned at (0,0,0) relative to parent
RearLeft	← etc
WheelMesh	
RearRight	
WheelMesh	

As we've already seen, the root 'Kart' object contains a rigid body component as well as the Kart Controller script. It also has on it an example player controller which simply sends the pad or keyboard input to the Kart Controller. Beneath that object in the hierarchy you will find a child named 'Body'. This contains the collision box for the kart, and is also the parent for all the meshes. This is so that the body of the vehicle can rotate independently of the simulation, so it can for example spin around 720 degrees whilst still travelling forwards. Underneath this body object you'll find the main body mesh and another parent object that contains the 4 wheels. Each wheel has its own parent transform which gives the position of the centre of that wheel. It is important that the rotation of this transform is set to (0,0,0) so that the wheels can be correctly rotated by the controller script as the vehicle moves and steers. The wheels don't contain any collision components as these are all created automatically by the Kart Controller script when the game starts.

## Swapping in your own models

In order to swap in your own models simply drag your kart body mesh(es) into the hierarchy and parent them to the 'Body' object. Then simply scale and position them as necessary, remembering to alter the dimensions of the collision box (on the 'Body' object) if needed.

Replace the example wheel meshes with your own ones, each under the correct parent object.

In order to make sure the wheels rotate around the correct point make sure that the origin of the mesh is at the center of the wheel and that the transforms are set up as follows.

WheelParent - positioned at the point the wheel should rotate around. rotation should be (0,0,0)

WheelMesh - child of WheelParent, positioned at (0,0,0) relative to its parent and rotated as necessary.

As for any physics simulation try to keep the size of the Kart roughly to a real-world scale. For reference the size of the example model is roughly 1 meter long, by about 0.7 meters wide.

Now it's time to adjust the vehicle setup parameters in the root Kart object, so select the Kart object and toggle open the Vehicle Setup foldout in the inspector. The various transform references shouldn't need to be altered, but for reference they should be setup as follows:

- Kart Body - this should reference the 'Body' object (not the body mesh)
- Front Left - this should reference the 'FrontLeft' wheel parent object (not the wheel mesh)
- Front Right - the 'FrontRight' wheel object
- similar for Rear Left and Rear Right

The radius values should be set to the radius of the front and rear wheels respectively (in meters).

Finally the Max Steer Angle parameter gives the angle (in degrees) that the front wheels should rotate when the steering is at full lock.

## Player Controller

In the Example Scene/Scripts folder you'll find an example player controller script. This is what takes gamepad or keyboard input and feeds it to the actual Kart Controller. There are only two parameters that the Kart Controller needs - a thrust value, and a steering value. The thrust value is effectively how far down the throttle is pressed and should be between -1 and 1 (where 0 = throttle is not pressed at all, 1 = full throttle forwards and -1 = full throttle in reverse). Likewise the steering value also varies from -1 to 1, with -1 indicating full lock left, and +1 indicating full lock right.

When implementing AI drivers, you'll need to replace the player controller script with an AI controller script that calculates the thrust and steering values itself, before feeding them to the Kart Controller.

## Triggering Effects

The Kart Controller script supports several non-physical effects that you can apply to your vehicles during gameplay. They can be triggered by calling simple functions on the Kart Controller script. For an example of how these can be used, have a look at the various pickup items in the example scene.

The built-in effect are:

- SpeedBoost - Gives the kart a temporary boost in speed. You can specify the new top speed in miles-per-hour, the new acceleration time, how long the speed boost should last before starting to fade out, and the fade out time.
- SpeedPenalty - Gives the kart a temporary reduction in speed.
- Spin - Spins the kart 720 degrees around the Y-axis. Quite often this is accompanied by a speed-penalty at the same time.
- Wiggle - Wiggles the kart back and forth around the Y-axis a few times. This could be used with a speed boost to give a wheel spin effect at the start of the race or alternatively it could be used with a speed penalty to make something like an oil-slick that slows the kart down.
- Jump - Makes the kart jump vertically into the air by the specified amount.

These effects can be combined together to implement a whole range of different gameplay features or powerups.