

Before going through this troubleshooting guide please check that you have the latest version of the asset.

Vehicle physics is behaving weirdly (jitter, jumping, etc.).

- Check the model rotation as per [this](#) guide.
- Check the model scale. The root object and WheelControllers should have a scale of [1,1,1].
- Make sure that the mass of the vehicle is set through the *VehicleController* ⇒ *Settings* tab, instead of the RigidBody inspector.
- Make sure that the dimensions of the vehicle are set correctly in *VehicleController* ⇒ *Settings* and update the inertia value if any changes were made.
- If the layers of the vehicle body colliders were changed from the default of *Physics.IgnoreRaycast*, untick the *WheelController* ⇒ *Auto Setup Layer Mask* and untick all the layers used for the vehicle body colliders (BoxColliders, MeshColliders, etc.). This is so the wheels do not detect the vehicle itself as ground.
- Check that the *Project Settings* ⇒ *Time* ⇒ *Fixed Delta Time* is 0.02 (default) or lower.
- Check that *Wheel Controller* suspension length is larger than 0.1 or so.
- Vehicle should not be further than roughly 10,000 units from the origin. This will degrade physics quality due to the floating-point precision. Consider using floating origin for open-world games.

There is no input.

- Check that there is a *VehicleInputProvider* and a *SceneInputProvider* script present in the scene. The full name will depend on the input method used, e.g. *RewiredVehicleInputProvider* or *InputSystemVehicleInputProvider*. Only one of each script should be present per scene.
- Check that the vehicle is *AWAKE* during play mode. This can be checked through the *ASLEEP* or *AWAKE* button at the top of the *VehicleController* inspector.
- Check that the vehicle is receiving input by clicking on *VehicleController* ⇒ *Control* ⇒ *Input*. The sliders and checkboxes there should react to user input if the vehicle is awake.

Raycasting the vehicle does not work.

WheelController (by default) sets the layer of the vehicle colliders to *Physics.IgnoreRaycast* to prevent the wheels from hitting the vehicle itself. This will however make raycasting the vehicle from an outside script impossible. The solution to this is to manually setup the layers and *WheelController* layer mask:

- Select all the colliders on the vehicle (BoxColliders, SphereColliders, MeshColliders, etc.) and assign a custom layer to them - e.g. *VehicleLayer*.
- Untick *Auto Setup Layer Mask* on the *WheelControllers* attached to the vehicle. A layer mask dropdown will appear. Make sure to untick the *VehicleLayer* on that list.

WheelControllers will now ignore the vehicle itself but the vehicle will be Raycastable.

How to make the vehicle feel more arcade?

NWH Vehicle Physics 2 is by default set up more towards realism / simcade style of vehicles and tries to be as physically accurate as possible. However, sometimes games require a more arcade approach. Here are a few tweaks to get more arcade behavior:

- Adjust Lateral Slip Coefficient of WheelController to a lower value, e.g. 0.5 or 0.7. This will reduce the tendency of the vehicle to snap oversteer and in general lose traction.
- Adjust Longitudinal Slip Coefficient of WheelController to a lower value. This will make wheel spin less likely to happen with short bursts of torque (e.g. gear changes).
- Adjust Slip Circle Shape of WheelController to 1. This will make wheel spin have less influence on the lateral grip, however, it will be harder to do powerslides.
- Set the center of mass to be a bit lower than realistic, e.g. a few centimeters above the floor of the vehicle. This will reduce leaning in the corners.
- Reduce VehicleController > Settings > Inertia to make the vehicle change direction more easily and feel more like an RC car. Overdoing it might cause instabilities and jitter.
- For more advanced users adjusting the wheel friction curve of the friction presets can also help. However, in most cases adjusting slip coefficients will be adequate.

Vehicle bounces after falling from height.

This is caused by the suspension bottoming out which triggers de-penetration code whose main job is to prevent the wheels from going into the ground. Bottoming out is caused by either too soft suspension or suspension travel that is shorter than the distance traveled in one physics frame, resulting is suspension that goes from fully extended to fully compressed (and further) before the code has a chance to update. There are a few ways this behaviour can be fixed:

- If the jump is small and the issue is not suspension travel, using stiffer suspension - especially bump damping - will solve the issue.
- If possible having a collider on the underside of the vehicle that prevents the suspension from fully compressing can be a solution.
- Spring values that can not hold the vehicle weight should be avoided. E.g. Max spring force setting of 10,000N (~1000kg) when using 4 wheels on a 10,000kg vehicle will not be enough and the vehicle will bottom out even while stationary.
- Increasing physics update rate by reducing Time.fixedDeltaTime. Default of 50Hz is quite low for vehicle physics and 100Hz will have much better results (Settings > Time > Fixed Delta Time of 0.01).
- If bottoming out can not be avoided there are 'Depenetration Spring' and 'Depenetration Damper' values under WheelController that can be tweaked to fit the vehicle.

From:

<http://nwhvehiclephysics.com/> - **NWH Vehicle Physics 2 Documentation**

Permanent link:

<http://nwhvehiclephysics.com/doku.php/Troubleshooting>

Last update: **2021/12/10 14:58**

