

Wheel Controller

WHEEL CONTROLLER^{3D}

Show Advanced Settings ☒

Wheel

Vehicle Side

Left

Radius

0.3806881

m

Width

0.2145097

m

Mass

29

kg

Rim Offset

0

m

Drag Torque

40

Nm

Parent

Niva

Use Rim Collider

☒

Wheel Model

Visual

WheelLO

Visual Position Offset

X 0

Y 0

Z 0

Visual Rotation Offset

X 0

Y 0

Z 0

Non-Rotating Visual (opt.)

None (Game Object)

Spring

Max Force

22000

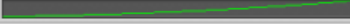
Nm

Max Length

0.42

m

Force Curve



! X: Spring compression [%], Y: Force coefficient

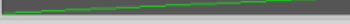
Damper

Bump Force

2300

Ns/m

Bump Curve




! X: Spring velocity (normalized) [m/s/10], Y: Force coefficient (normalized)

Rebound Force

2500

Ns/m

Rebound Curve



! X: Spring velocity (normalized) [m/s/10], Y: Force coefficient (normalized)

Geometry

Camber At Top

0

deg

Camber At Bottom

0

deg

Smooth Movement

☐

Squat

0.319

Friction

Slip Circle Shape

1.05

Curve Based Load Coefficient

☐

Active Friction Preset

DefaultTireFrictionPreset (FrictionPreset)

Friction Preset

! This is a ScriptableObject. All changes are global.

Pacejka Parameters

B

10.72

C

1.9

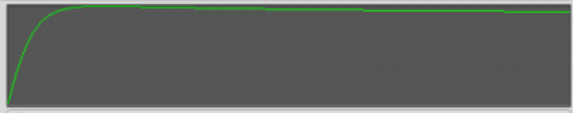
D

0.883

E

0.99

Friction Curve Preview



! X: Slip | Y: Friction

Longitudinal

Slip Coefficient

1

x100 %

Force Coefficient

0.9

00 %

Lateral

Slip Coefficient

1

x100 %

Force Coefficient

0.9

00 %

Ground Detection

Single Ray

☐

Longitudinal Scan Resolution

13

Lateral Scan Resolution

2

! Ray count: 26

Apply Force To Others

☐

Auto Setup Layer Mask

☒

! Vehicle colliders will be set to ignore Raycast layer while Auto Setup Layer Mask is active. Use manual setup if you need to be able to raycast the vehicle.

Debug Values

Longitudinal Slip

0

Lateral Slip

0

Suspension Force Magnitude

0

Bottomed Out

☐

Motor Torque

0

Brake Torque

0

Debug

☐

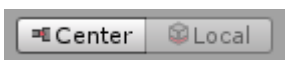
<http://nwhvehiclephysics.com/>

Printed on 2020/12/16 21:27

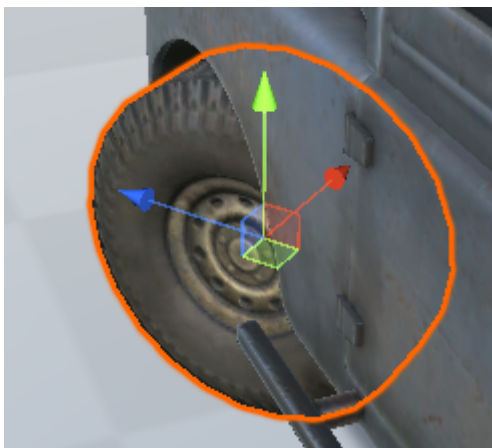
Setup Instructions

Make sure that your model has proper rotation for Unity which is Z axis forward, X axis right and Y axis up. In case your model has different rotation either fix it in modelling software or check out the following guide: <https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html>

- Drag a vehicle model into the scene.
- Attach a Rigidbody to the model.
- Add a Collider to the model. Make sure that the layer of the collider is set to Physics.IgnoreRaycast or a custom layer. For this example WheelControllerIgnore will be used. This step is important later on to prevent wheels from detecting the vehicle as ground.
- [Optional] Change name of wheel objects to contain 'wheel' in the name. E.g. Wheel_FL, wheel_front_left or similar. This will allow the script to auto-find them later.



Center/Local handle switch.



Wheel model with Unity-correct rotation and pivot.

- Since it is important that WheelController script is not attached to the wheel itself, a new GameObject needs to be created.
- This game object will hold the WheelController script and act as an anchor for suspension which means it needs to be positioned directly above the wheel axle.
- To achieve this duplicate the wheel object, remove all the components from it (MeshRenderer, MeshFilter, etc.) and drag it upwards to the point where you want the beginning of spring travel to be. Afterwards, you can move this object which will in turn move the wheel position relative to the vehicle.



Position of WheelController object in relation to wheel.

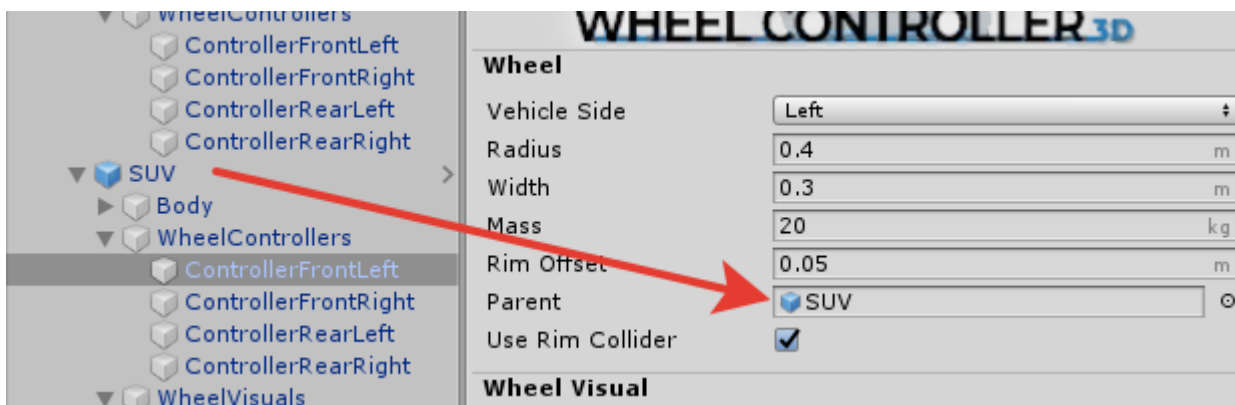


Side view.

- Attach WheelController to object created in previous step. WheelController will fill in all the necessary fields by itself. To check if the values have been (correctly) filled in check out the next section.

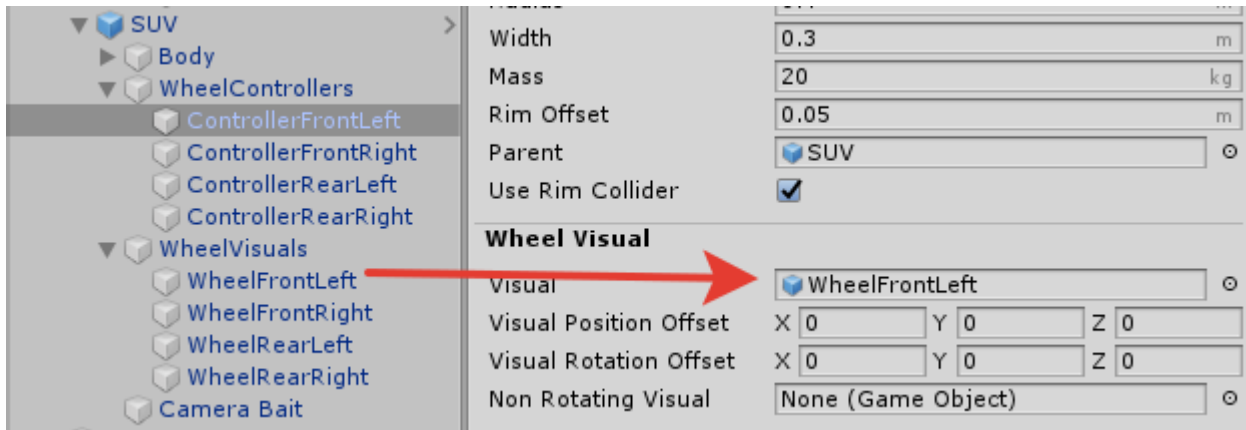
Script will try to set up all the values automatically when added to a GameObject. If for some reason this fails check the following fields:

- Parent - parent is the GameObject containing the Rigidbody to which the forces will be applied. Usually the root of the vehicle.



Parent (root) object location in vehicle hierarchy.

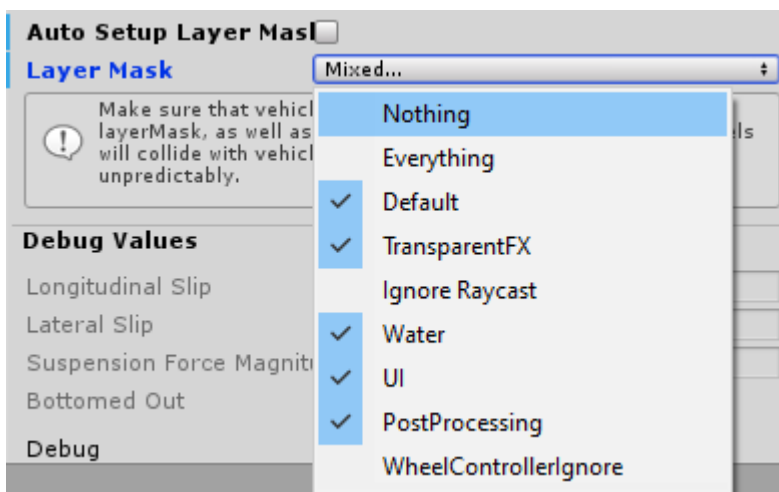
- **Visual** - this is the object that represents the wheel mesh. It will be assigned automatically if the object contains "wheel" or "whl" in its name. If not, assign it manually.



Visual object location in vehicle hierarchy.

- **Layer Mask**- only ticked layers will be detected by WheelController. Make sure that **Physics.IgnoreRaycast** and all you vehicle's collider layers are unticked.

If your vehicle's colliders have no layer assigned create a new layer, assign it to them and make sure it is unticked under **Layer Mask** field. If this step is skipped vehicle will fly into the air as the wheels will detect the vehicle as ground and suspension will become and remain fully compressed.



Example manual LayerMask configuration.

- **Width and Radius** - adjust the values so that the wheel gizmo corresponds to the size of the wheel mesh.

The rest of the fields do not require manual setup. Your wheel should now be fully functional.

Adjusting Vehicle Behavior

If you have no knowledge of how suspension works I would recommend checking out [this link](#) first. Also, check out [FrictionPreset](#) page.

Tips

- Spring Length should not be shorter than ~ 0.2 (or $\text{Time.fixedDeltaTime} * 10$). Using very short spring rates might result in suspension bottoming out when wheel travel in one frame is larger than the total length of the spring. This usually results in vehicle bouncing off the ground instead of suspension absorbing the hit. If you want the vehicle to sit lower do not reduce the spring length but rather consider moving the WheelController object up which will in turn make the wheel move up effectively reducing the ride height.
- A good starting point for Maximum Spring Force on 4 wheeled vehicles is vehicle mass times 10.

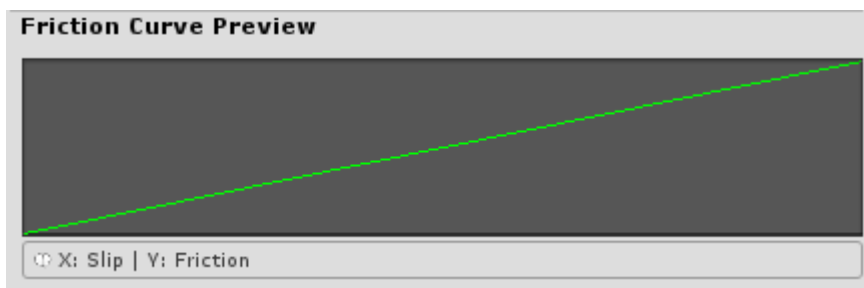
Achieving Oversteer

- Reduce Maximum Tire Grip Force on the rear wheels relative to the front wheels.
- Adjust CenterOfMass forwards. If you are using standalone WheelController3D you can use the provided script. If you are using NWH Vehicle Physics this can be found under Settings tab of VehicleController inspector.
- Increase spring rate on the rear wheels.
- Increase roll-bar force on the rear wheels.

Arcade-like Behavior

By default the asset is geared towards realism. To achieve arcade-like behavior:

- Reduce Slip Coefficient and increase Force Coefficient. This will cause the vehicle to be less prone to slipping and spinning out. Good starting values are 0.3 for slip coefficient and 6 for force coefficient.
- Adjust the friction curve to be more linear. You might need to increase the Force Coefficient to make vehicle slide less afterwards (for the curve in the image below force coefficient of 12 was used):



- Lower the center of mass to reduce the chance of vehicle flipping over and make vehicle lean less making it behave more like a go kart. Do not lower center of mass below the axle height or the vehicle might begin to lean in the opposite direction.

Field Explanations

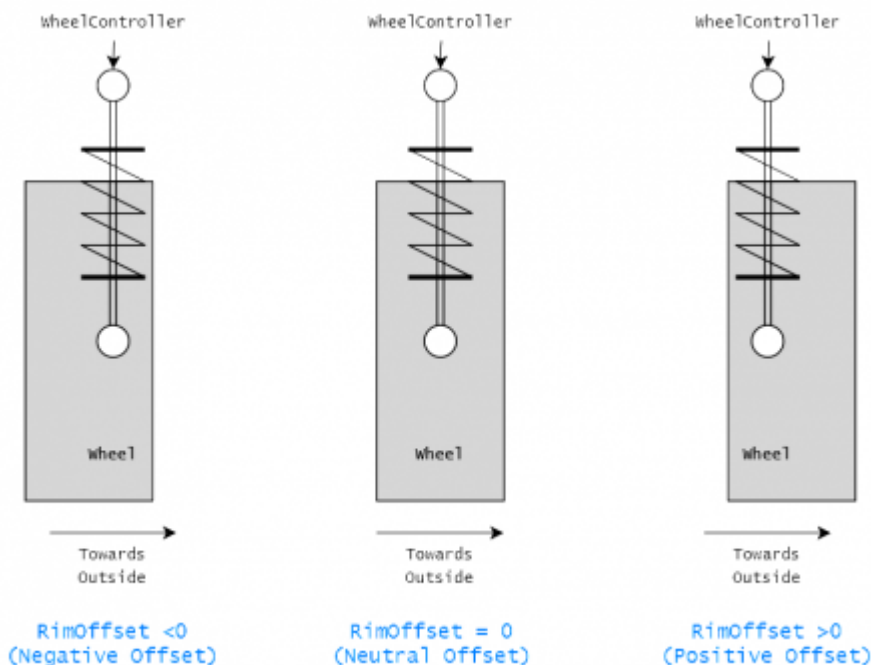
Field Explanations

Wheel

Wheel	
Vehicle Side	Left
Radius	0.34 m
Width	0.26 m
Mass	14 kg
Rim Offset	0.05 m
Drag Torque	35 Nm
Parent	Sports Car
Use Rim Collider	<input type="checkbox"/>

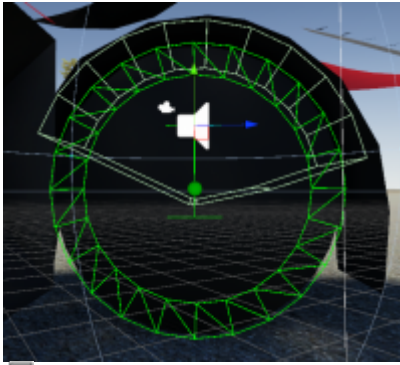
Wheel section of WheelController inspector.

- Vehicle Side is auto-detected and it determines which side of the vehicle is on. If it is a motorcycle use Center.
- Radius and Width need to be adjusted manually to fit the wheel model. Turn on gizmos to be able to see wheel gizmo.
- Rim Offset is equal to the distance between the suspension center and the rim center.



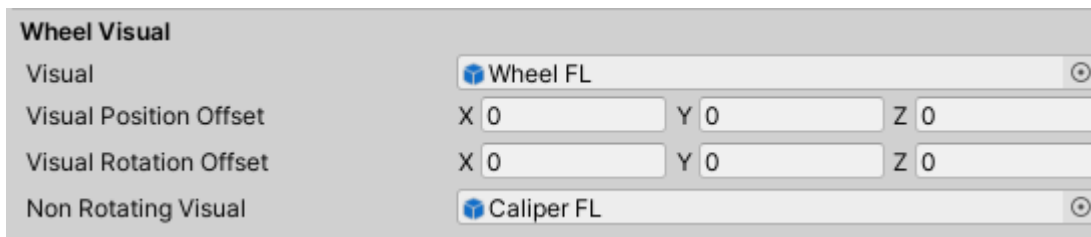
Rim Offset. Frontal wheel view.

- Drag Torque is brake torque that is constantly applied to imitate rolling resistance.
- Parent is a parent object to which the forces will be applied. Must have a Rigidbody attached.
- If Use Rim Collider is true a half-circle mesh collider is generated and placed on the top half of the wheel to prevent that section of the wheel detecting objects as ground which can cause the wheel to jump suddenly. Also prevents side of the wheel from going through objects as the ground detection only happens on the bottom half of the tire thread.



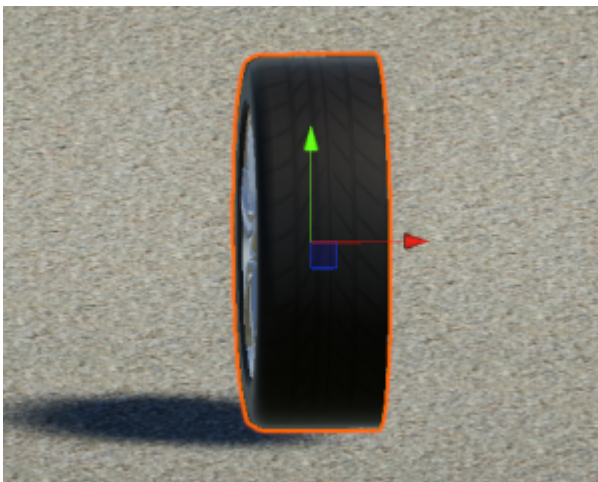
RimCollider example.

Wheel Visual



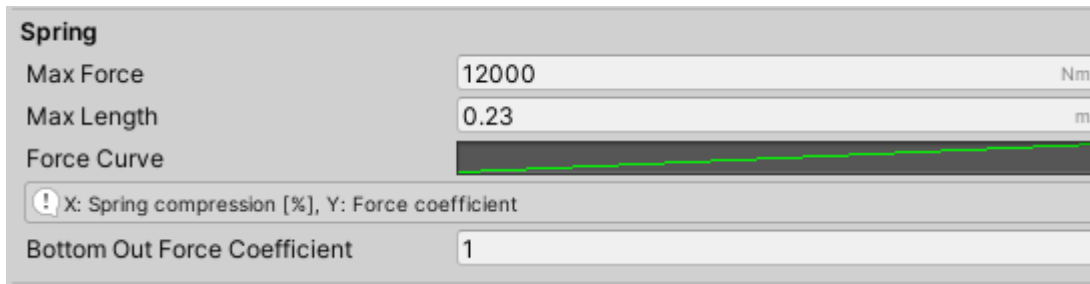
Wheel Visual section of WheelController inspector.

- Visual is a model representing the wheel. It needs to have **Unity-correct rotation and centered pivot**. If the rotation or pivot on the model is wrong it should be fixed either in 3D software (e.g. Blender) or by following the steps in this [link](#). The method from the link can be applied both for rotation and pivot.
- Since WheelController version 4.x (included with NWH Vehicle Physics 2) rotation and pivot can also be adjusted through Visual Position Offset and Visual Rotation Offset. Visual Rotation Offset's Z-axis can also be used to introduce wheel wobble if the wheel is damaged.



Wheel visual with Unity-correct rotation and centered pivot.

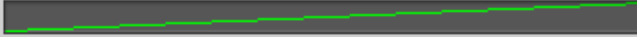
Spring



Spring

Max Force Nm

Max Length m

Force Curve 

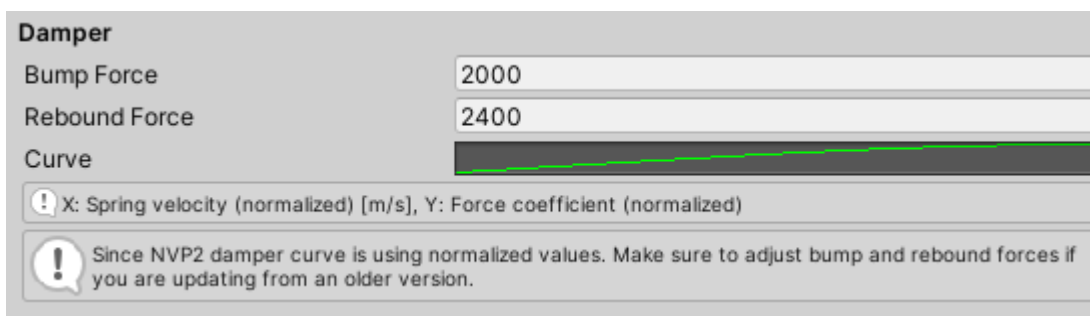
! X: Spring compression [%], Y: Force coefficient

Bottom Out Force Coefficient

Spring section of WheelController inspector.

- Max Force is the maximum force the spring will be able to exert when fully compressed.
- Max Length is the maximum extended spring length. Unlike in Unity's WheelCollider there is no target spring position setting as the spring will fully extend (as it should) when in the air. Therefore, the resting length of the spring is not known beforehand. Because of that the best way to position the vehicle to be in its resting position is to enter play mode, let the vehicle settle and then copy over the Transform values.
- Force Curve is an AnimationCurve with normalized values on both axes (both 0 to 1). X represents spring compression (0 for relaxed, 1 for compressed) and Y axis represents force. Actual X and Y maximum values are adjusted through Max Force and Max Length.

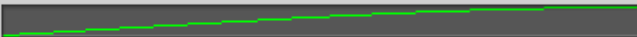
Damper



Damper

Bump Force

Rebound Force

Curve 

! X: Spring velocity (normalized) [m/s], Y: Force coefficient (normalized)

! Since NVP2 damper curve is using normalized values. Make sure to adjust bump and rebound forces if you are updating from an older version.

Damper section of WheelController inspector.

- Bump Force - maximum damper force when damper is getting compressed.
- Rebound Force - maximum damper force when damper is getting extended.
- Curve - damper curve that determines relationship between spring velocity (X-axis, normalized) and force (Y-axis, normalized).

Geometry



Geometry

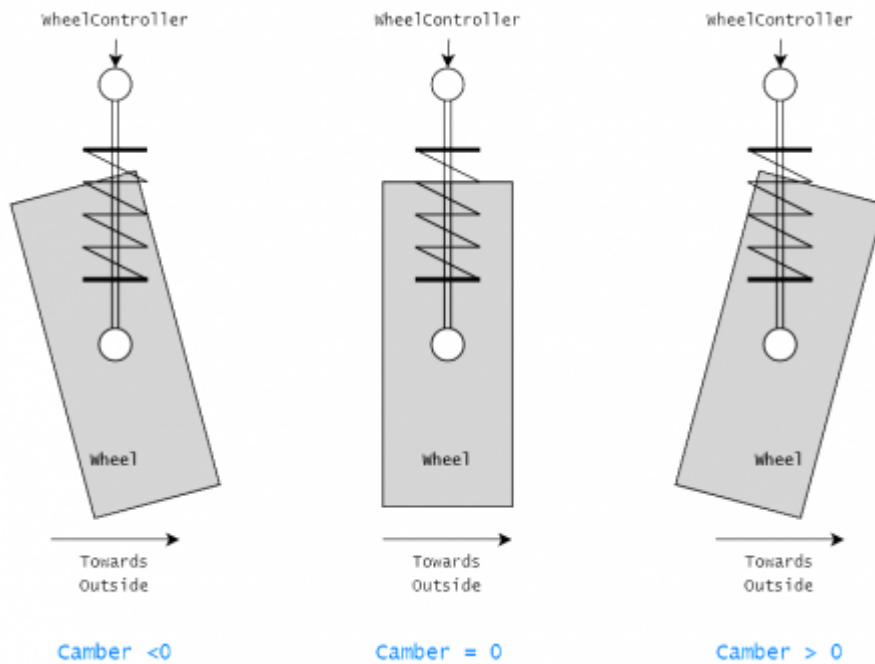
Camber At Top deg

Camber At Bottom deg

Squat  0.4

Geometry section of WheelController inspector.

- Camber At Top and Camber At Bottom indicate camber at top and bottom of suspension travel.



Negative, neutral and positive camber.

- Squat - as tire applies force to the ground through friction a torque in the opposite direction gets generated around wheel axle. Squat field determines percentage of that torque which gets transferred to the vehicle. 1 (100%) will result in rear end squatting on acceleration while -1 (-100%) will do exactly the opposite. In real world this depends on the suspension setup.

Friction

Friction

Active Friction Preset
DefaultFrictionPreset (FrictionPreset)

> Friction Preset ?

Longitudinal

Slip Coefficient
1
x100 %

Force Coefficient
1
x100 %

Lateral

Slip Coefficient
1
x100 %

Force Coefficient
1
x100 %

Load and Forces

Load Grip Curve

! X: Tire Load [N], Y: Max Tire Friction Force [N]

Maximum Tire Load
10500
Nm

Maximum Tire Grip Force
8400
Nm

Friction section of WheelController inspector.

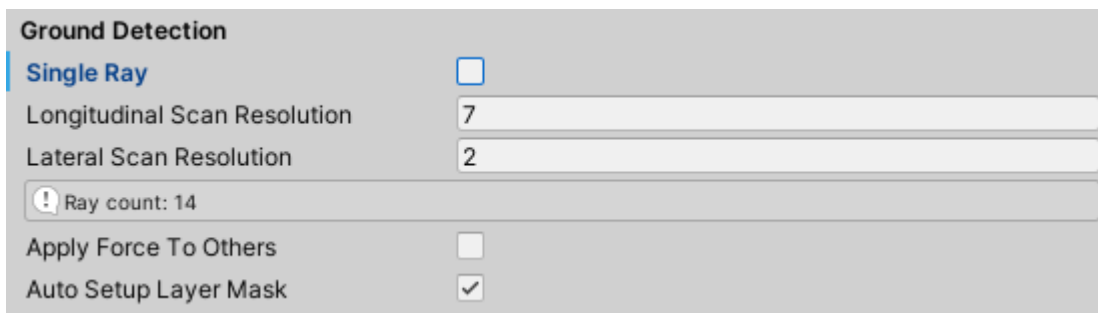
- Active Friction Preset - currently active friction preset / curve. Check [Friction Preset](#) page for more info.
- Slip Coefficient - coefficient by which the slip will be multiplied. Can be used to modify friction behavior without modifying currently active curve.
- Force Coefficient - coefficient by which the tire friction force will be multiplied. Roughly

equal to D parameter of Friction Curve's *Pacejka Parameters*.

Load and Forces

Tire's grip increases with load placed on it. How much depends on the tire but in most tires this grip/load relation is linear until certain point when it starts to level off. Load Grip Curve imitates that relation where X-axis represents percentage (0 to 1) of Maximum Tire Load and Y-axis represents percentage (0 to 1) of Maximum Tire Grip Force. Wheel behavior heavily depends on these values.

Ground Detection



Ground Detection section of WheelController inspector.

Ground detection in WheelController happens by firing one or multiple Raycasts and working out from those the shape of the terrain. This is heavily optimized and raycasts are batched. Still, number of rays does affect performance as there is some post-processing of received hit data required.

- **Single Ray** - select to use only one ray in the middle of the wheel. Equivalent to what Unity's WheelCollider is doing. Adequate for racing games and games where there are not many obstacles.
- Multiple rays can be used when better ground detection resolution is required, such as in off-road games. Untick **Single Ray** and two fields, **Longitudinal Scan Resolution** and **Lateral Scan Resolution**, will show up. These two fields determine the resolution of the grid of rays that will be cast.
- **Apply Force To Others** - if true wheel will interact with other Rigidbodies.

Ground Detection - Layer Mask

Since Raycast is used to detect ground the wheel will also detect other objects, such as the vehicle itself. To prevent this a Layer Mask is used. By default **Auto Setup Layer Mask** is ticked and all the layers assigned to the Parents (Vehicle's) colliders will be automatically removed from the Layer Mask (i.e. will be ignored).

- If setting up the Layer Mask manually it is important to also un-tick **IgnoreRaycast** layer.
- Do not use **Default** as body collider layer.

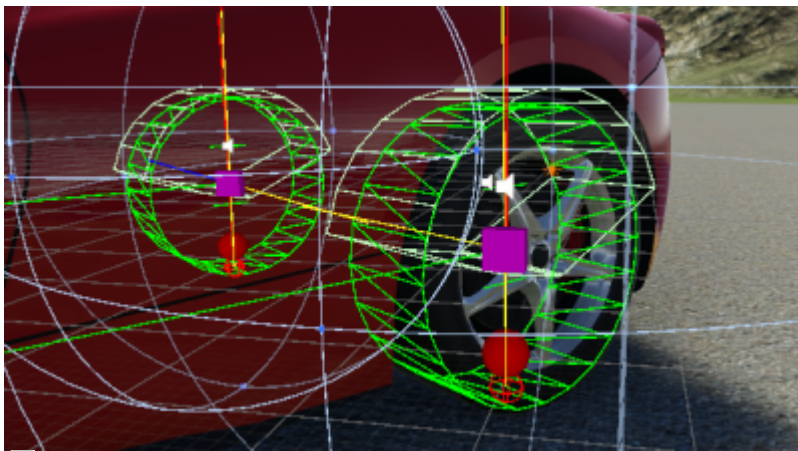
Debug Values

Debug Values	
Longitudinal Slip	<input type="text" value="0"/>
Lateral Slip	<input type="text" value="0"/>
Suspension Force Magnitude	<input type="text" value="0"/>
Bottomed Out	<input type="checkbox"/>
Motor Torque	<input type="text" value="0"/>
Brake Torque	<input type="text" value="0"/>
Debug	<input type="checkbox"/>

Debug section of WheelController inspector.

Debug sections shows the most important values of the wheel.

- Enable Debug to show more detailed debug gizmos.



Wheel gizmo with Debug enabled.

- Suspension Force Magnitude value can be used to balance the vehicle's weight distribution. Adjust Rigidbody's center of mass until all wheels have the same force magnitude on the flat surface to get 50/50 weight distribution. If using WheelController asset adjusting center of mass is done through an included script, if using NWH Vehicle Physics 2 this is done through Settings tab of VehicleController.

2020/02/07 19:41 · Aron Rescec

Troubleshooting

1. My vehicle flies into the air when I press play.
 - Vehicle does not have a collider.
 - Rigidbody mass is 1.
 - WheelController is most likely colliding with it's parent vehicle's colliders.
2. Vehicle does not slow down.
 - Increase Drag Torque
3. Vehicle seems to be vibrating when standing still.
 - This can be caused by any of the following:
 - Time.fixedDeltaTime is too high. Values higher than default 0.02 are not recommended, while 0.1667 (60Hz physics update) will greatly increase the smoothness of the physics at a small performance cost and is recommended.
 - Center of mass of the vehicle is too high. Try lowering the center of mass slightly with included CenterOfMass script. High center of mass has the effect of trying to balance a

pole on the tip of your finger and is more prone to oscillation.

- Extremely high spring strength value.
- Too stiff Friction Curve or too large Force Coefficient / Slip Coefficient.

From:

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

Permanent link:

<http://nwhvehiclephysics.com/doku.php/Setup/WheelController>

Last update: **2020/12/16 21:08**

