

WheelController3D

4.0

Generated by Doxygen 1.8.17

1 Namespace Index	2
1.1 Packages	2
2 Hierarchical Index	2
2.1 Class Hierarchy	2
3 Class Index	2
3.1 Class List	2
4 Namespace Documentation	3
4.1 NWH Namespace Reference	3
4.2 NWH.WheelController3D Namespace Reference	3
5 Class Documentation	4
5.1 NWH.WheelController3D.APITest Class Reference	4
5.1.1 Detailed Description	4
5.2 NWH.WheelController3D.Damper Class Reference	4
5.2.1 Detailed Description	5
5.2.2 Member Data Documentation	5
5.3 NWH.WheelController3D.Friction Class Reference	5
5.3.1 Detailed Description	6
5.3.2 Member Data Documentation	6
5.4 NWH.WheelController3D.FrictionPreset Class Reference	7
5.4.1 Member Function Documentation	7
5.4.2 Member Data Documentation	8
5.5 NWH.WheelController3D.FrictionPresetCollection Class Reference	8
5.6 NWH.WheelController3D.FrictionPresetEditor Class Reference	8
5.7 NWH.WheelController3D.Spring Class Reference	8
5.7.1 Detailed Description	9
5.7.2 Member Data Documentation	9
5.8 NWH.WheelController3D.Wheel Class Reference	11
5.8.1 Detailed Description	12
5.8.2 Member Function Documentation	12
5.8.3 Member Data Documentation	12
5.9 NWH.WheelController3D.WheelController Class Reference	16
5.9.1 Member Enumeration Documentation	19
5.9.2 Member Function Documentation	19
5.9.3 Member Data Documentation	21
5.9.4 Property Documentation	24
5.10 NWH.WheelController3D.WheelControllerEditor Class Reference	29
5.11 NWH.WheelController3D.WheelHit Class Reference	29
5.11.1 Member Data Documentation	30
5.11.2 Property Documentation	30

Index	31
-------	----

1 Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

NWH	3
NWH.WheelController3D	3

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

NWH.WheelController3D.Damper	4
NWH.WheelController3D.Friction MonoBehaviour	5
NWH.WheelController3D.APITest	4
NWH.WheelController3D.WheelController NUIEditor	16
NWH.WheelController3D.FrictionPresetEditor	8
NWH.WheelController3D.WheelControllerEditor ScriptableObject	29
NWH.WheelController3D.FrictionPreset	7
NWH.WheelController3D.FrictionPresetCollection	8
NWH.WheelController3D.Spring	8
NWH.WheelController3D.Wheel	11
NWH.WheelController3D.WheelHit	29

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

NWH.WheelController3D.APITest	
Class for testing API compatibility with Unity's wheel collider	4
NWH.WheelController3D.Damper	
Suspension part	4
NWH.WheelController3D.Friction	
All info related to longitudinal force calculation	5
NWH.WheelController3D.FrictionPreset	7
NWH.WheelController3D.FrictionPresetCollection	8
NWH.WheelController3D.FrictionPresetEditor	8
NWH.WheelController3D.Spring	
Suspension part	8
NWH.WheelController3D.Wheel	
Contains everything wheel related, including rim and tire	11
NWH.WheelController3D.WheelController	16
NWH.WheelController3D.WheelControllerEditor	29
NWH.WheelController3D.WheelHit	29

4 Namespace Documentation

4.1 NWH Namespace Reference

4.2 NWH.WheelController3D Namespace Reference

Classes

- class **APITest**
Class for testing API compatibility with Unity's wheel collider.
- class **Damper**
Suspension part.
- class **Friction**
All info related to longitudinal force calculation.
- class **FrictionPreset**
- class **FrictionPresetCollection**
- class **FrictionPresetEditor**
- class **Spring**
Suspension part.
- class **Wheel**
Contains everything wheel related, including rim and tire.
- class **WheelController**
- class **WheelControllerEditor**
- class **WheelHit**

5 Class Documentation

5.1 NWH.WheelController3D.APITest Class Reference

Class for testing API compatibility with Unity's wheel collider.

Inherits MonoBehaviour.

Public Attributes

- float **brakeTorque**
- Vector3 **center**
- **Friction** forwardFriction
- **WheelHit** hit
- bool **isGrounded**
- float **mass**
- float **motorTorque**
- Vector3 **position**
- float **radius**
- Quaternion **rotation**
- float **rpm**
- **Friction** sidewaysFriction
- float **steerAngle**
- float **suspensionDistance**
- **WheelController** wheel

5.1.1 Detailed Description

Class for testing API compatibility with Unity's wheel collider.

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/Test/APITest.cs

5.2 NWH.WheelController3D.Damper Class Reference

Suspension part.

Public Attributes

- float **bumpForce** = 3000.0f
Bump force of the damper.
- AnimationCurve **curve**
Curve where X axis represents speed of travel of the suspension and Y axis represents resultant force. Both values are normalized to [0,1].
- float **force**
Current damper force.
- float **reboundForce** = 3500.0f
Rebound force of the damper.

Static Public Attributes

- const float **maxVelocity** = 100f

5.2.1 Detailed Description

Suspension part.

5.2.2 Member Data Documentation

5.2.2.1 bumpForce float NWH.WheelController3D.Damper.bumpForce = 3000.0f

Bump force of the damper.

5.2.2.2 curve AnimationCurve NWH.WheelController3D.Damper.curve

Curve where X axis represents speed of travel of the suspension and Y axis represents resultant force. Both values are normalized to [0,1].

5.2.2.3 force float NWH.WheelController3D.Damper.force

Current damper force.

5.2.2.4 reboundForce float NWH.WheelController3D.Damper.reboundForce = 3500.0f

Rebound force of the damper.

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/Damper.cs

5.3 NWH.WheelController3D.Friction Class Reference

All info related to longitudinal force calculation.

Public Attributes

- float **derivative**
- float **error**
- float **force**
Current force in friction direction.
- float **forceCoefficient** = 1f
Modifies force value.
- float **integral**
- float **Kd** = 0.05f
- float **Ki** = 0.4f
- float **Kp** = 0.1f
- float **prevError**
- float **slip**
Current slip in friction direction.
- float **slipCoefficient** = 1
Modifies slip value.
- float **speed**
Speed at the point of contact with the surface.

5.3.1 Detailed Description

All info related to longitudinal force calculation.

5.3.2 Member Data Documentation

5.3.2.1 **force** float NWH.WheelController3D.Friction.force

Current force in friction direction.

5.3.2.2 **forceCoefficient** float NWH.WheelController3D.Friction.forceCoefficient = 1f

Modifies force value.

5.3.2.3 **slip** float NWH.WheelController3D.Friction.slip

Current slip in friction direction.

5.3.2.4 slipCoefficient `float NWH.WheelController3D.Friction.slipCoefficient = 1`

Modifies slip value.

5.3.2.5 speed `float NWH.WheelController3D.Friction.speed`

Speed at the point of contact with the surface.

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/Friction.cs

5.4 NWH.WheelController3D.FrictionPreset Class Reference

Inherits ScriptableObject.

Public Member Functions

- void **GenerateLUT** (ref float[] LUT, int resolution=LUT_RESOLUTION)
- void **UpdateFrictionCurve** ()
Generate Curve from B,C,D and E parameters of Pacejka's simplified magic formula

Public Attributes

- Vector4 **BCDE**
B, C, D and E parameters of short version of Pacejka's magic formula.
- float[] **LUT**

Static Public Attributes

- const int **LUT_RESOLUTION** = 50

Properties

- AnimationCurve **Curve** [get]

5.4.1 Member Function Documentation

5.4.1.1 UpdateFrictionCurve() `void NWH.WheelController3D.FrictionPreset.UpdateFrictionCurve ()`

Generate Curve from B,C,D and E parameters of Pacejka's simplified magic formula

5.4.2 Member Data Documentation

5.4.2.1 BCDE `Vector4 NWH.WheelController3D.FrictionPreset.BCDE`

B, C, D and E parameters of short version of Pacejka's magic formula.

The documentation for this class was generated from the following file:

- `C:/Unity/WheelController2/Assets/WheelController/Scripts/FrictionPreset.cs`

5.5 NWH.WheelController3D.FrictionPresetCollection Class Reference

Inherits `ScriptableObject`.

Public Attributes

- `List< FrictionPreset > frictionPresets = new List< FrictionPreset>()`

The documentation for this class was generated from the following file:

- `C:/Unity/WheelController2/Assets/WheelController/Scripts/FrictionPresetCollection.cs`

5.6 NWH.WheelController3D.FrictionPresetEditor Class Reference

Inherits `NUIEditor`.

Public Member Functions

- override `bool OnInspectorNUI ()`

The documentation for this class was generated from the following file:

- `C:/Unity/WheelController2/Assets/WheelController/Scripts/Editor/FrictionPresetEditor.cs`

5.7 NWH.WheelController3D.Spring Class Reference

Suspension part.

Public Attributes

- bool **bottomedOut**
Is the suspension currently bottomed out? True when `spring.length <= 0`.
- float **bottomOutForceCoefficient** = 1f
Coefficient modifying the force of suspension hitting bump stop (fully compressing). Too low values will result with wheel passing through ground as the reaction force will be too low, and too high values will result in vehicle overreacting and bouncing up after bottoming out. Bottoming out usually happens due to:
- float **compressionPercent**
How much is spring currently compressed. 0 means fully relaxed and 1 fully compressed.
- float **force**
Current force the spring is exerting in [N].
- AnimationCurve **forceCurve**
Force curve where X axis represents spring travel [0,1] and Y axis represents force coefficient [0, 1]. Force coefficient is multiplied by `maxForce` to get the final spring force.
- float **length**
Current length of the spring.
- float **maxForce** = 18000.0f
Maximum force spring can exert.
- float **maxLength** = 0.35f
Length of fully relaxed spring.
- bool **overExtended**
Is the spring over extended. Opposite of `bottomedOut`.
- float **prevLength**
- Vector3 **targetPoint**
- float **velocity**
Rate of change of the length of the spring in [m/s].

5.7.1 Detailed Description

Suspension part.

5.7.2 Member Data Documentation

5.7.2.1 **bottomedOut** bool NWH.WheelController3D.Spring.bottomedOut

Is the suspension currently bottomed out? True when `spring.length <= 0`.

5.7.2.2 **bottomOutForceCoefficient** float NWH.WheelController3D.Spring.bottomOutForceCoefficient = 1f

Coefficient modifying the force of suspension hitting bump stop (fully compressing). Too low values will result with wheel passing through ground as the reaction force will be too low, and too high values will result in vehicle overreacting and bouncing up after bottoming out. Bottoming out usually happens due to:

- Too weak springs
- Falling from large height
- Too large `Time.fixedDeltaTime` combined with short suspension travel

5.7.2.3 compressionPercent `float NWH.WheelController3D.Spring.compressionPercent`

How much is spring currently compressed. 0 means fully relaxed and 1 fully compressed.

5.7.2.4 force `float NWH.WheelController3D.Spring.force`

Current force the spring is exerting in [N].

5.7.2.5 forceCurve `AnimationCurve NWH.WheelController3D.Spring.forceCurve`

Force curve where X axis represents spring travel [0,1] and Y axis represents force coefficient [0, 1]. Force coefficient is multiplied by maxForce to get the final spring force.

5.7.2.6 length `float NWH.WheelController3D.Spring.length`

Current length of the spring.

5.7.2.7 maxForce `float NWH.WheelController3D.Spring.maxForce = 18000.0f`

Maximum force spring can exert.

5.7.2.8 maxLength `float NWH.WheelController3D.Spring.maxLength = 0.35f`

Length of fully relaxed spring.

5.7.2.9 overExtended `bool NWH.WheelController3D.Spring.overExtended`

Is the spring over extended. Opposite of bottomed out.

5.7.2.10 velocity `float NWH.WheelController3D.Spring.velocity`

Rate of change of the length of the spring in [m/s].

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/Spring.cs

5.8 NWH.WheelController3D.Wheel Class Reference

Contains everything wheel related, including rim and tire.

Public Member Functions

- void **Initialize** (**WheelController** wc)
Calculation of static parameters and creation of rim collider.

Public Attributes

- float **angularVelocity**
Current angular velocity of the wheel in rad/s.
- float **brakeTorque**
Brake torque applied to the wheel in Nm.
- float **camberAngle**
Current camber angle.
- float **camberAtBottom**
Camber angle at the bottom of suspension travel (fully extended).
- float **camberAtTop**
Camber angle at the top of suspension travel (fully compressed).
- Vector3 **forward**
Forward vector of the wheel in world coordinates.
- float **inertia**
Inertia of the wheel.
- Vector3 **inside**
Vector in world coordinates pointing towards the inside of the wheel.
- float **load**
Tire load in Nm.
- float **mass** = 15.0f
Mass of the wheel. Inertia is calculated from this.
- float **motorTorque**
*Motor torque applied to the wheel. Since **NWH** (p. 3) Vehicle Physics 2 the value is readonly and setting it will have no effect since torque calculation is done inside powertrain solver.*
- Vector3 **nonRotatingPositionOffset**
Position offset of the non-rotating part.
- GameObject **nonRotatingVisual**
Object representing non-rotating part of the wheel. This could be things such as brake calipers, external fenders, etc.
- float **radius** = 0.35f
Total radius of the tire in [m].
- Vector3 **right**
Vector in world coordinates pointing to the right of the wheel.
- GameObject **rimColliderGO**
GameObject containing the rim MeshCollider. This is used to prevent objects from penetrating into the wheel from sides or top, where the ground detection does not work.
- float **rimOffset**
Offset of the rim from the center of steering rotation.
- float **rotationAngle**
Current rotation angle of the wheel visual in regards to it's X axis vector.
- float **RPM**

- Current wheel RPM.*
- float **steerAngle**
 - Current steer angle of the wheel.*
- Vector3 **up**
 - Wheel** (p. 11)'s up vector in world coordinates.*
- GameObject **visual**
 - GameObject representing the visual aspect of the wheel / wheel mesh. Should not have any physics colliders attached to it.*
- Vector3 **visualPositionOffset** = Vector3.zero
 - In cases where wheel visual's model might have wrong pivot point this field can be used to center the wheel or move it in/out. It is always preferable to fix the model in modelling software or by parenting it to another, empty, transform and resetting the pivot that way. <https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html>*
- Vector3 **visualRotationOffset** = Vector3.zero
 - Use if wheel visual's model has wrong rotation or if you want to make the wheel appear to wobble (adjust Z axis to get the wobble). It is always preferable to fix the model in modelling software or by parenting it to another, empty, transform and resetting the pivot that way. <https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html>*
- float **width** = 0.25f
 - Width of the tyre.*
- Vector3 **worldPosition**
 - Position of the wheel in world coordinates.*
- Quaternion **worldRotation**
 - Rotation of the wheel in world coordinates.*

5.8.1 Detailed Description

Contains everything wheel related, including rim and tire.

5.8.2 Member Function Documentation

5.8.2.1 Initialize() void NWH.WheelController3D.Wheel.Initialize (
 WheelController wc)

Calculation of static parameters and creation of rim collider.

5.8.3 Member Data Documentation

5.8.3.1 angularVelocity float NWH.WheelController3D.Wheel.angularVelocity

Current angular velocity of the wheel in rad/s.

5.8.3.2 brakeTorque `float NWH.WheelController3D.Wheel.brakeTorque`

Brake torque applied to the wheel in Nm.

5.8.3.3 camberAngle `float NWH.WheelController3D.Wheel.camberAngle`

Current camber angle.

5.8.3.4 camberAtBottom `float NWH.WheelController3D.Wheel.camberAtBottom`

Camber angle at the bottom of suspension travel (fully extended).

5.8.3.5 camberAtTop `float NWH.WheelController3D.Wheel.camberAtTop`

Camber angle at the top of suspension travel (fully compressed).

5.8.3.6 forward `Vector3 NWH.WheelController3D.Wheel.forward`

Forward vector of the wheel in world coordinates.

5.8.3.7 inertia `float NWH.WheelController3D.Wheel.inertia`

Inertia of the wheel.

5.8.3.8 inside `Vector3 NWH.WheelController3D.Wheel.inside`

Vector in world coordinates pointing towards the inside of the wheel.

5.8.3.9 load `float NWH.WheelController3D.Wheel.load`

Tire load in Nm.

5.8.3.10 mass `float NWH.WheelController3D.Wheel.mass = 15.0f`

Mass of the wheel. Inertia is calculated from this.

5.8.3.11 motorTorque `float NWH.WheelController3D.Wheel.motorTorque`

Motor torque applied to the wheel. Since **NWH** (p. 3) Vehicle Physics 2 the value is readonly and setting it will have no effect since torque calculation is done inside powertrain solver.

5.8.3.12 nonRotatingPositionOffset `Vector3 NWH.WheelController3D.Wheel.nonRotatingPositionOffset`

Position offset of the non-rotating part.

5.8.3.13 nonRotatingVisual `GameObject NWH.WheelController3D.Wheel.nonRotatingVisual`

Object representing non-rotating part of the wheel. This could be things such as brake calipers, external fenders, etc.

5.8.3.14 radius `float NWH.WheelController3D.Wheel.radius = 0.35f`

Total radius of the tire in [m].

5.8.3.15 right `Vector3 NWH.WheelController3D.Wheel.right`

Vector in world coordinates pointing to the right of the wheel.

5.8.3.16 rimColliderGO `GameObject NWH.WheelController3D.Wheel.rimColliderGO`

GameObject containing the rim MeshCollider. This is used to prevent objects from penetrating into the wheel from sides or top, where the ground detection does not work.

5.8.3.17 rimOffset `float NWH.WheelController3D.Wheel.rimOffset`

Offset of the rim from the center of steering rotation.

5.8.3.18 rotationAngle `float NWH.WheelController3D.Wheel.rotationAngle`

Current rotation angle of the wheel visual in regards to it's X axis vector.

5.8.3.19 RPM `float NWH.WheelController3D.Wheel.RPM`

Current wheel RPM.

5.8.3.20 steerAngle `float NWH.WheelController3D.Wheel.steerAngle`

Current steer angle of the wheel.

5.8.3.21 up `Vector3 NWH.WheelController3D.Wheel.up`

Wheel (p. 11)'s up vector in world coordinates.

5.8.3.22 visual `GameObject NWH.WheelController3D.Wheel.visual`

GameObject representing the visual aspect of the wheel / wheel mesh. Should not have any physics colliders attached to it.

5.8.3.23 visualPositionOffset `Vector3 NWH.WheelController3D.Wheel.visualPositionOffset = Vector3.zero`

In cases where wheel visual's model might have wrong pivot point this field can be used to center the wheel or move it in/out. It is always preferable to fix the model in modelling software or by parenting it to another, empty, transform and resetting the pivot that way. <https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html>

5.8.3.24 visualRotationOffset `Vector3 NWH.WheelController3D.Wheel.visualRotationOffset = Vector3.zero`

Use if wheel visual's model has wrong rotation or if you want to make the wheel appear to wobble (adjust Z axis to get the wobble). It is always preferable to fix the model in modelling software or by parenting it to another, empty, transform and resetting the pivot that way. <https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html>

5.8.3.25 width `float NWH.WheelController3D.Wheel.width = 0.25f`

Width of the tyre.

5.8.3.26 worldPosition `Vector3 NWH.WheelController3D.Wheel.worldPosition`

Position of the wheel in world coordinates.

5.8.3.27 worldRotation `Quaternion NWH.WheelController3D.Wheel.worldRotation`

Rotation of the wheel in world coordinates.

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/Wheel.cs

5.9 NWH.WheelController3D.WheelController Class Reference

Inherits MonoBehaviour.

Public Types

- enum **Side** { **Left** = -1, **Right** = 1, **Center** = 0, **Auto** = 2 }
- Side of the vehicle.*

Public Member Functions

- bool **GetGroundHit** (out **WheelHit** hit)
Returns Raycast info of the wheel's hit. Always check if the function returns true before using hit info as data will only be updated when wheel is hitting the ground (isGrounded == True).
- void **GetWorldPose** (out Vector3 pos, out Quaternion quat)
Returns the position and rotation of the wheel.
- void **SetCamber** (float camberAtTop, float camberAtBottom)
Sets linear camber between the two values.
- void **SetCamber** (float **camber**)
Sets fixed camber.
- void **Initialize** ()
- void **InitializeScanParams** ()
Sets up coordinates, arrays and other fields for ground detection. Needs to be called each time a dimension of the wheel or ground detection resolution changes.
- void **SetDefaults** (bool reset=false, bool findWheelVisuals=true)
Sets default values if they have not already been set. Gets called each time Reset() is called in editor - such as adding the script to a GameObject.
- void **Step** ()
Runs one physics update of the wheel.
- **Side DetermineSide** (Vector3 pointPosition, Transform referenceTransform)
Determines on what side of the vehicle a point is.

Static Public Member Functions

- static bool **IsInLayerMask** (int layer, **LayerMask** layermask)

Determines if layer is in layermask.

Public Attributes

- **Wheel wheel**

Instance of the wheel.

- **Spring spring**

Instance of the spring.

- **Damper damper**

Instance of the damper.

- **LayerMask layerMask** = Physics.IgnoreRaycastLayer

- bool **debug**

If set to true draws detailed debug info.

- bool **useExternalUpdate**

*When true **Step()** (p. 21) will not be called each **FixedUpdate()**. Used when execution order is important and/or the other script is waiting on the result of **Step()** (p. 21).*

- GameObject **parent**

Root object of the vehicle.

- Rigidbody **parentRigidbody**

Rigidbody to which the forces will be applied.

- bool **useRimCollider** = true

If enabled mesh collider mimicking the shape of rim and wheel will be positioned so that wheel can not pass through objects in case raycast does not detect the surface in time.

- bool **autoSetupLayerMask** = true

- **Side vehicleSide** = Side.Auto

Side the wheel is on.

- **FrictionPreset activeFrictionPreset**

Current active friction preset.

- **WheelHit wheelHit** = new **WheelHit**()

*Contains point in which wheel touches ground. Not valid if **IsGrounded**.*

- bool **singleRay**

When enabled only a single raycast is used to detect ground. Very fast and should be used when performance is critical.

- AnimationCurve **loadGripCurve** = new AnimationCurve()

*Curve where X axis represents tire load as a percentage [0,1] of **maximumTireLoad** and Y axis represents tire grip force as a percentage [0,1] of **maximumTireGripForce**. Drastically influences handling.*

- float **maximumTireLoad** = 8000f

Tire load at which the grip force reaches it's maximum.

- float **maximumTireGripForce** = 8000f

Maximum total force a tire can exert on surface, no matter the load.

- bool **applyForceToOthers**

Should forces be applied to other rigidbodies when wheel is in contact with them?

- Transform **cachedTransform**

Cached value of this.transform.

- Transform **cachedVisualTransform**

Cached value of visual's transform.

Properties

- float **angularVelocity** [get, set]
Returns angular velocity of the wheel in radians. Multiply by wheel radius to get linear speed.
- float **brakeTorque** [get, set]
Brake torque on the wheel axle. [Nm] Must be positive (zero included).
- float **camber** [get]
Camber angle of the wheel. [deg] Negative angle means that the top of the wheel is closer to the vehicle than the bottom.
- Vector3 **center** [get]
The center of the wheel, measured in the world space.
- float **damperBumpForce** [get, set]
Bump force at 1 m/s spring velocity
- AnimationCurve **DamperCurve** [get, set]
***Damper** (p. 4) force curve in relation to spring velocity.*
- float **damperForce** [get]
Current damper force.
- float **damperReboundForce** [get, set]
Rebounding force at 1 m/s spring velocity
- **Friction forwardFriction** [get, set]
Returns `_Friction` object with longitudinal values.
- int **ForwardScanResolution** [get, set]
Ground scan resolution in forward direction.
- bool **isGrounded** [get]
Is the tractive surface touching the ground? Returns false if vehicle tipped over / tire sidewall is in contact.
- LayerMask **LayerMask** [get, set]
Only layers with value of 1 (ticked) will get detected by the wheel.
- float **mass** [get, set]
Mass of the wheel. [kg] Typical values would be in range [20, 200]
- float **motorTorque** [get, set]
Motor torque on the wheel axle. [Nm] Can be positive or negative based on direction.
- GameObject **NonRotatingVisual** [get, set]
Object that follows the wheel position in everything but rotation around the axle. Can be used for brake calipers, external fenders, etc.
- GameObject **Parent** [get, set]
Returns wheel's parent object.
- Vector3 **pointVelocity** [get]
Returns velocity at the wheel's center position in [m/s].
- float **radius** [get, set]
Radius (height) of the tire. [meters]
- float **rimOffset** [get, set]
Side offset of the rim. Positive value will result if wheel further from the vehicle. [meters]
- float **rpm** [get]
Rotations per minute of the wheel around the axle. [RPM]
- float **suspensionDistance** [get, set]
Maximum extension distance of wheel suspension, measured in local space. Same as `spring.maxLength`
- **Friction sideFriction** [get, set]
Returns `_Friction` object with lateral values.
- int **SideToSideScanResolution** [get, set]
Number of scan planes parallel to the wheel.
- float **speed** [get]

- Returns vehicle speed in meters per second [m/s], multiply by 3.6 for [kph] or by 2.24 for [mph].*
- bool **springBottomedOut** [get]
 - True when spring is fully compressed, i.e. there is no more spring travel.*
- float **springCompression** [get]
 - Returns value in range [0,1] where 1 means spring is fully compressed.*
- AnimationCurve **springCurve** [get, set]
 - Spring** (p. 8) force curve in relation to spring length.*
- float **springLength** [get, set]
 - Length of the spring when fully extended.*
- float **springMaximumForce** [get, set]
 - Maximum spring force. [N]*
- bool **springOverExtended** [get]
 - True when spring is fully extended.*
- float **springTravel** [get]
 - Current length (travel) of spring.*
- Vector3 **springTravelPoint** [get]
 - Point in which spring and swingarm are in contact.*
- float **springVelocity** [get]
 - Spring** (p. 8) velocity in relation to local vertical axis. [m/s] Positive on rebound (extension), negative on bump (compression).*
- float **steerAngle** [get, set]
 - Steer angle around the wheel's up axis (with add-ons ignored). [deg]*
- float **suspensionForce** [get, set]
 - Current spring force. [N] Can be written to for use in Anti-roll Bar script or similar.*
- **Side VehicleSide** [get, set]
 - Returns Enum [Side] with the corresponding side of the vehicle a wheel is at [Left, Right]*
- GameObject **Visual** [get, set]
 - Returns object that represents wheel's visual representation. Can be null in case the object is not assigned (not mandatory).*
- float **width** [get, set]
 - Width of the wheel. [meters]*
- Vector3 **worldCenter** [get]
 - The center of the wheel, measured in the world space.*

5.9.1 Member Enumeration Documentation

5.9.1.1 Side enum **NWH.WheelController3D.WheelController.Side** [strong]

Side of the vehicle.

5.9.2 Member Function Documentation

5.9.2.1 DetermineSide() **Side** NWH.WheelController3D.WheelController.DetermineSide (
 Vector3 *pointPosition*,
 Transform *referenceTransform*)

Determines on what side of the vehicle a point is.

Parameters

<i>pointPosition</i>	Position of the point in question.
<i>referenceTransform</i>	Position of the reference transform.

Returns

Enum Side [Left,Right] (int)[-1,1]

5.9.2.2 GetGroundHit() `bool NWH.WheelController3D.WheelController.GetGroundHit (`
`out WheelHit hit)`

Returns Raycast info of the wheel's hit. Always check if the function returns true before using hit info as data will only be updated when wheel is hitting the ground (isGrounded == True).

Parameters

<i>h</i>	Standard Unity RaycastHit
----------	---------------------------

Returns

5.9.2.3 GetWorldPose() `void NWH.WheelController3D.WheelController.GetWorldPose (`
`out Vector3 pos,`
`out Quaternion quat)`

Returns the position and rotation of the wheel.

5.9.2.4 InitializeScanParams() `void NWH.WheelController3D.WheelController.InitializeScanParams (`
`)`

Sets up coordinates, arrays and other fields for ground detection. Needs to be called each time a dimension of the wheel or ground detection resolution changes.

5.9.2.5 IsInLayerMask() `static bool NWH.WheelController3D.WheelController.IsInLayerMask (`
`int layer,`
`LayerMask layermask) [static]`

Determines if layer is in layermask.

5.9.2.6 SetCamber() [1/2] `void NWH.WheelController3D.WheelController.SetCamber (`
`float camber)`

Sets fixed camber.

Parameters

<i>camber</i>	
---------------	--

5.9.2.7 SetCamber() [2/2] `void NWH.WheelController3D.WheelController.SetCamber (`
`float camberAtTop,`
`float camberAtBottom)`

Sets linear camber between the two values.

Parameters

<i>camberAtTop</i>	
<i>camberAtBottom</i>	

5.9.2.8 SetDefaults() `void NWH.WheelController3D.WheelController.SetDefaults (`
`bool reset = false,`
`bool findWheelVisuals = true)`

Sets default values if they have not already been set. Gets called each time Reset() is called in editor - such as adding the script to a GameObject.

Parameters

<i>reset</i>	Sets default values even if they have already been set.
<i>findWheelVisuals</i>	Should script attempt to find wheel visuals automatically by name and position?

5.9.2.9 Step() `void NWH.WheelController3D.WheelController.Step ()`

Runs one physics update of the wheel.

5.9.3 Member Data Documentation

5.9.3.1 activeFrictionPreset **FrictionPreset** `NWH.WheelController3D.WheelController.active↔`
`FrictionPreset`

Current active friction preset.

5.9.3.2 applyForceToOthers `bool NWH.WheelController3D.WheelController.applyForceToOthers`

Should forces be applied to other rigidbodies when wheel is in contact with them?

5.9.3.3 cachedTransform `Transform NWH.WheelController3D.WheelController.cachedTransform`

Cached value of this.transform.

5.9.3.4 cachedVisualTransform `Transform NWH.WheelController3D.WheelController.cachedVisual↔
Transform`

Cached value of visual's transform.

5.9.3.5 damper `Damper NWH.WheelController3D.WheelController.damper`

Instance of the damper.

5.9.3.6 debug `bool NWH.WheelController3D.WheelController.debug`

If set to true draws detailed debug info.

5.9.3.7 loadGripCurve `AnimationCurve NWH.WheelController3D.WheelController.loadGripCurve = new
AnimationCurve()`

Curve where X axis represents tire load as a percentage [0,1] of maximumTireLoad and Y axis represents tire grip force as a percentage [0,1] of maximumTireGripForce. Drastically influences handling.

5.9.3.8 maximumTireGripForce `float NWH.WheelController3D.WheelController.maximumTireGripForce
= 8000f`

Maximum total force a tire can exert on surface, no matter the load.

5.9.3.9 maximumTireLoad `float NWH.WheelController3D.WheelController.maximumTireLoad = 8000f`

Tire load at which the grip force reaches it's maximum.

5.9.3.10 parent `GameObject NWH.WheelController3D.WheelController.parent`

Root object of the vehicle.

5.9.3.11 parentRigidbody `Rigidbody NWH.WheelController3D.WheelController.parentRigidbody`

Rigidbody to which the forces will be applied.

5.9.3.12 singleRay `bool NWH.WheelController3D.WheelController.singleRay`

When enabled only a single raycast is used to detect ground. Very fast and should be used when performance is critical.

5.9.3.13 spring `Spring NWH.WheelController3D.WheelController.spring`

Instance of the spring.

5.9.3.14 useExternalUpdate `bool NWH.WheelController3D.WheelController.useExternalUpdate`

When true **Step()** (p. 21) will not be called each **FixedUpdate()**. Used when execution order is important and/or the other script is waiting on the result of **Step()** (p. 21).

5.9.3.15 useRimCollider `bool NWH.WheelController3D.WheelController.useRimCollider = true`

If enabled mesh collider mimicking the shape of rim and wheel will be positioned so that wheel can not pass through objects in case raycast does not detect the surface in time.

5.9.3.16 vehicleSide `Side NWH.WheelController3D.WheelController.vehicleSide = Side.Auto`

Side the wheel is on.

5.9.3.17 wheel `Wheel NWH.WheelController3D.WheelController.wheel`

Instance of the wheel.

5.9.3.18 wheelHit **WheelHit** NWH.WheelController3D.WheelController.wheelHit = new **WheelHit**()

Contains point in which wheel touches ground. Not valid if !isGrounded.

5.9.4 Property Documentation

5.9.4.1 angularVelocity float NWH.WheelController3D.WheelController.angularVelocity [get], [set]

Returns angular velocity of the wheel in radians. Multiply by wheel radius to get linear speed.

5.9.4.2 brakeTorque float NWH.WheelController3D.WheelController.brakeTorque [get], [set]

Brake torque on the wheel axle. [Nm] Must be positive (zero included).

5.9.4.3 camber float NWH.WheelController3D.WheelController.camber [get]

Camber angle of the wheel. [deg] Negative angle means that the top of the wheel is closer to the vehicle than the bottom.

5.9.4.4 center Vector3 NWH.WheelController3D.WheelController.center [get]

The center of the wheel, measured in the world space.

5.9.4.5 damperBumpForce float NWH.WheelController3D.WheelController.damperBumpForce [get], [set]

Bump force at 1 m/s spring velocity

5.9.4.6 DamperCurve AnimationCurve NWH.WheelController3D.WheelController.DamperCurve [get], [set]

Damper (p. 4) force curve in relation to spring velocity.

5.9.4.7 damperForce `float NWH.WheelController3D.WheelController.damperForce [get]`

Current damper force.

5.9.4.8 damperReboundForce `float NWH.WheelController3D.WheelController.damperReboundForce [get], [set]`

Rebounding force at 1 m/s spring velocity

5.9.4.9 forwardFriction `Friction NWH.WheelController3D.WheelController.forwardFriction [get], [set]`

Returns `_Friction` object with longitudinal values.

5.9.4.10 ForwardScanResolution `int NWH.WheelController3D.WheelController.ForwardScanResolution [get], [set]`

Ground scan resolution in forward direction.

5.9.4.11 isGrounded `bool NWH.WheelController3D.WheelController.isGrounded [get]`

Is the tractive surface touching the ground? Returns false if vehicle tipped over / tire sidewall is in contact.

5.9.4.12 LayerMask `LayerMask NWH.WheelController3D.WheelController.LayerMask [get], [set]`

Only layers with value of 1 (ticked) will get detected by the wheel.

5.9.4.13 mass `float NWH.WheelController3D.WheelController.mass [get], [set]`

Mass of the wheel. [kg] Typical values would be in range [20, 200]

5.9.4.14 motorTorque `float NWH.WheelController3D.WheelController.motorTorque [get], [set]`

Motor torque on the wheel axle. [Nm] Can be positive or negative based on direction.

5.9.4.15 NonRotatingVisual `GameObject NWH.WheelController3D.WheelController.NonRotatingVisual [get], [set]`

Object that follows the wheel position in everything but rotation around the axle. Can be used for brake calipers, external fenders, etc.

5.9.4.16 Parent `GameObject NWH.WheelController3D.WheelController.Parent [get], [set]`

Returns wheel's parent object.

5.9.4.17 pointVelocity `Vector3 NWH.WheelController3D.WheelController.pointVelocity [get]`

Returns velocity at the wheel's center position in [m/s].

5.9.4.18 radius `float NWH.WheelController3D.WheelController.radius [get], [set]`

Radius (height) of the tire. [meters]

5.9.4.19 rimOffset `float NWH.WheelController3D.WheelController.rimOffset [get], [set]`

Side offset of the rim. Positive value will result if wheel further from the vehicle. [meters]

5.9.4.20 rpm `float NWH.WheelController3D.WheelController.rpm [get]`

Rotations per minute of the wheel around the axle. [RPM]

5.9.4.21 sideFriction `Friction NWH.WheelController3D.WheelController.sideFriction [get], [set]`

Returns `_Friction` object with lateral values.

5.9.4.22 SideToSideScanResolution `int NWH.WheelController3D.WheelController.SideToSideScanResolution [get], [set]`

Number of scan planes parallel to the wheel.

5.9.4.23 speed float NWH.WheelController3D.WheelController.speed [get]

Returns vehicle speed in meters per second [m/s], multiply by 3.6 for [kph] or by 2.24 for [mph].

5.9.4.24 springBottomedOut bool NWH.WheelController3D.WheelController.springBottomedOut [get]

True when spring is fully compressed, i.e. there is no more spring travel.

5.9.4.25 springCompression float NWH.WheelController3D.WheelController.springCompression [get]

Returns value in range [0,1] where 1 means spring is fully compressed.

5.9.4.26 springCurve AnimationCurve NWH.WheelController3D.WheelController.springCurve [get], [set]

Spring (p. 8) force curve in relation to spring length.

5.9.4.27 springLength float NWH.WheelController3D.WheelController.springLength [get], [set]

Length of the spring when fully extended.

5.9.4.28 springMaximumForce float NWH.WheelController3D.WheelController.springMaximumForce [get], [set]

Maximum spring force. [N]

5.9.4.29 springOverExtended bool NWH.WheelController3D.WheelController.springOverExtended [get]

True when spring is fully extended.

5.9.4.30 springTravel float NWH.WheelController3D.WheelController.springTravel [get]

Current length (travel) of spring.

5.9.4.31 springTravelPoint `Vector3 NWH.WheelController3D.WheelController.springTravelPoint [get]`

Point in which spring and swingarm are in contact.

5.9.4.32 springVelocity `float NWH.WheelController3D.WheelController.springVelocity [get]`

Spring (p.8) velocity in relation to local vertical axis. [m/s] Positive on rebound (extension), negative on bump (compression).

5.9.4.33 steerAngle `float NWH.WheelController3D.WheelController.steerAngle [get], [set]`

Steer angle around the wheel's up axis (with add-ons ignored). [deg]

5.9.4.34 suspensionDistance `float NWH.WheelController3D.WheelController.suspensionDistance [get], [set]`

Maximum extension distance of wheel suspension, measured in local space. Same as `spring.maxLength`

5.9.4.35 suspensionForce `float NWH.WheelController3D.WheelController.suspensionForce [get], [set]`

Current spring force. [N] Can be written to for use in Anti-roll Bar script or similar.

5.9.4.36 VehicleSide `side NWH.WheelController3D.WheelController.VehicleSide [get], [set]`

Returns Enum [Side] with the corresponding side of the vehicle a wheel is at [Left, Right]

5.9.4.37 Visual `GameObject NWH.WheelController3D.WheelController.Visual [get], [set]`

Returns object that represents wheel's visual representation. Can be null in case the object is not assigned (not mandatory).

5.9.4.38 width `float NWH.WheelController3D.WheelController.width [get], [set]`

Width of the wheel. [meters]

5.9.4.39 worldCenter `Vector3` NWH.WheelController3D.WheelController.worldCenter [get]

The center of the wheel, measured in the world space.

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/WheelController.cs

5.10 NWH.WheelController3D.WheelControllerEditor Class Reference

Inherits NUIEditor.

Public Member Functions

- void **OnEnable** ()
- override bool **OnInspectorNUI** ()
- override bool **UseDefaultMargins** ()

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/Editor/WheelControllerEditor.cs

5.11 NWH.WheelController3D.WheelHit Class Reference

Public Attributes

- float **angleForward**
- float **curvatureOffset**
- float **distanceFromTire**
- float **force**
The magnitude of the force being applied for the contact. [N]
- Vector3 **forwardDir**
The direction the wheel is pointing in.
- float **forwardSlip**
Tire slip in the rolling direction.
- Vector3 **groundPoint**
- Vector2 **offset**
- RaycastHit **raycastHit**
- Vector3 **sidewaysDir**
The sideways direction of the wheel.
- float **sidewaysSlip**
The slip in the sideways direction.
- bool **valid** = false
- float **weight**

Properties

- Collider **collider** [get]
- Vector3 **normal** [get]
The normal at the point of contact
- Vector3 **point** [get]
The point of contact between the wheel and the ground.

5.11.1 Member Data Documentation

5.11.1.1 force float NWH.WheelController3D.WheelHit.force

The magnitude of the force being applied for the contact. [N]

5.11.1.2 forwardDir Vector3 NWH.WheelController3D.WheelHit.forwardDir

The direction the wheel is pointing in.

5.11.1.3 forwardSlip float NWH.WheelController3D.WheelHit.forwardSlip

Tire slip in the rolling direction.

5.11.1.4 sidewaysDir Vector3 NWH.WheelController3D.WheelHit.sidewaysDir

The sideways direction of the wheel.

5.11.1.5 sidewaysSlip float NWH.WheelController3D.WheelHit.sidewaysSlip

The slip in the sideways direction.

5.11.2 Property Documentation

5.11.2.1 normal Vector3 NWH.WheelController3D.WheelHit.normal [get]

The normal at the point of contact

5.11.2.2 point Vector3 NWH.WheelController3D.WheelHit.point [get]

The point of contact between the wheel and the ground.

The documentation for this class was generated from the following file:

- C:/Unity/WheelController2/Assets/WheelController/Scripts/WheelHit.cs

Index

- activeFrictionPreset
 - NWH.WheelController3D.WheelController, 21
- angularVelocity
 - NWH.WheelController3D.Wheel, 12
 - NWH.WheelController3D.WheelController, 24
- applyForceToOthers
 - NWH.WheelController3D.WheelController, 21
- BCDE
 - NWH.WheelController3D.FrictionPreset, 8
- bottomedOut
 - NWH.WheelController3D.Spring, 9
- bottomOutForceCoefficient
 - NWH.WheelController3D.Spring, 9
- brakeTorque
 - NWH.WheelController3D.Wheel, 12
 - NWH.WheelController3D.WheelController, 24
- bumpForce
 - NWH.WheelController3D.Damper, 5
- cachedTransform
 - NWH.WheelController3D.WheelController, 22
- cachedVisualTransform
 - NWH.WheelController3D.WheelController, 22
- camber
 - NWH.WheelController3D.WheelController, 24
- camberAngle
 - NWH.WheelController3D.Wheel, 13
- camberAtBottom
 - NWH.WheelController3D.Wheel, 13
- camberAtTop
 - NWH.WheelController3D.Wheel, 13
- center
 - NWH.WheelController3D.WheelController, 24
- compressionPercent
 - NWH.WheelController3D.Spring, 9
- curve
 - NWH.WheelController3D.Damper, 5
- damper
 - NWH.WheelController3D.WheelController, 22
- damperBumpForce
 - NWH.WheelController3D.WheelController, 24
- DamperCurve
 - NWH.WheelController3D.WheelController, 24
- damperForce
 - NWH.WheelController3D.WheelController, 24
- damperReboundForce
 - NWH.WheelController3D.WheelController, 25
- debug
 - NWH.WheelController3D.WheelController, 22
- DetermineSide
 - NWH.WheelController3D.WheelController, 19
- force
 - NWH.WheelController3D.Damper, 5
 - NWH.WheelController3D.Friction, 6
 - NWH.WheelController3D.Spring, 10
 - NWH.WheelController3D.WheelHit, 30
- forceCoefficient
 - NWH.WheelController3D.Friction, 6
- forceCurve
 - NWH.WheelController3D.Spring, 10
- forward
 - NWH.WheelController3D.Wheel, 13
- forwardDir
 - NWH.WheelController3D.WheelHit, 30
- forwardFriction
 - NWH.WheelController3D.WheelController, 25
- ForwardScanResolution
 - NWH.WheelController3D.WheelController, 25
- forwardSlip
 - NWH.WheelController3D.WheelHit, 30
- GetGroundHit
 - NWH.WheelController3D.WheelController, 20
- GetWorldPose
 - NWH.WheelController3D.WheelController, 20
- inertia
 - NWH.WheelController3D.Wheel, 13
- Initialize
 - NWH.WheelController3D.Wheel, 12
- InitializeScanParams
 - NWH.WheelController3D.WheelController, 20
- inside
 - NWH.WheelController3D.Wheel, 13
- isGrounded
 - NWH.WheelController3D.WheelController, 25
- IsInLayerMask
 - NWH.WheelController3D.WheelController, 20
- LayerMask
 - NWH.WheelController3D.WheelController, 25
- length
 - NWH.WheelController3D.Spring, 10
- load
 - NWH.WheelController3D.Wheel, 13
- loadGripCurve
 - NWH.WheelController3D.WheelController, 22
- mass
 - NWH.WheelController3D.Wheel, 13
 - NWH.WheelController3D.WheelController, 25
- maxForce
 - NWH.WheelController3D.Spring, 10
- maximumTireGripForce
 - NWH.WheelController3D.WheelController, 22
- maximumTireLoad
 - NWH.WheelController3D.WheelController, 22
- maxLength
 - NWH.WheelController3D.Spring, 10

- motorTorque
 - NWH.WheelController3D.Wheel, 14
 - NWH.WheelController3D.WheelController, 25
- nonRotatingPositionOffset
 - NWH.WheelController3D.Wheel, 14
- NonRotatingVisual
 - NWH.WheelController3D.WheelController, 25
- nonRotatingVisual
 - NWH.WheelController3D.Wheel, 14
- normal
 - NWH.WheelController3D.WheelHit, 30
- NWH, 3
- NWH.WheelController3D, 3
- NWH.WheelController3D.APITest, 4
- NWH.WheelController3D.Damper, 4
 - bumpForce, 5
 - curve, 5
 - force, 5
 - reboundForce, 5
- NWH.WheelController3D.Friction, 5
 - force, 6
 - forceCoefficient, 6
 - slip, 6
 - slipCoefficient, 6
 - speed, 7
- NWH.WheelController3D.FrictionPreset, 7
 - BCDE, 8
 - UpdateFrictionCurve, 7
- NWH.WheelController3D.FrictionPresetCollection, 8
- NWH.WheelController3D.FrictionPresetEditor, 8
- NWH.WheelController3D.Spring, 8
 - bottomedOut, 9
 - bottomOutForceCoefficient, 9
 - compressionPercent, 9
 - force, 10
 - forceCurve, 10
 - length, 10
 - maxForce, 10
 - maxLength, 10
 - overExtended, 10
 - velocity, 10
- NWH.WheelController3D.Wheel, 11
 - angularVelocity, 12
 - brakeTorque, 12
 - camberAngle, 13
 - camberAtBottom, 13
 - camberAtTop, 13
 - forward, 13
 - inertia, 13
 - Initialize, 12
 - inside, 13
 - load, 13
 - mass, 13
 - motorTorque, 14
 - nonRotatingPositionOffset, 14
 - nonRotatingVisual, 14
 - radius, 14
 - right, 14
 - rimColliderGO, 14
 - rimOffset, 14
 - rotationAngle, 14
 - RPM, 15
 - steerAngle, 15
 - up, 15
 - visual, 15
 - visualPositionOffset, 15
 - visualRotationOffset, 15
 - width, 15
 - worldPosition, 16
 - worldRotation, 16
- NWH.WheelController3D.WheelController, 16
 - activeFrictionPreset, 21
 - angularVelocity, 24
 - applyForceToOthers, 21
 - brakeTorque, 24
 - cachedTransform, 22
 - cachedVisualTransform, 22
 - camber, 24
 - center, 24
 - damper, 22
 - damperBumpForce, 24
 - DamperCurve, 24
 - damperForce, 24
 - damperReboundForce, 25
 - debug, 22
 - DetermineSide, 19
 - forwardFriction, 25
 - ForwardScanResolution, 25
 - GetGroundHit, 20
 - GetWorldPose, 20
 - InitializeScanParams, 20
 - isGrounded, 25
 - IsInLayerMask, 20
 - LayerMask, 25
 - loadGripCurve, 22
 - mass, 25
 - maximumTireGripForce, 22
 - maximumTireLoad, 22
 - motorTorque, 25
 - NonRotatingVisual, 25
 - Parent, 26
 - parent, 22
 - parentRigidbody, 23
 - pointVelocity, 26
 - radius, 26
 - rimOffset, 26
 - rpm, 26
 - SetCamber, 20, 21
 - SetDefaults, 21
 - Side, 19
 - sideFriction, 26
 - SideToSideScanResolution, 26
 - singleRay, 23
 - speed, 26
 - spring, 23
 - springBottomedOut, 27

- springCompression, 27
- springCurve, 27
- springLength, 27
- springMaximumForce, 27
- springOverExtended, 27
- springTravel, 27
- springTravelPoint, 27
- springVelocity, 28
- steerAngle, 28
- Step, 21
- suspensionDistance, 28
- suspensionForce, 28
- useExternalUpdate, 23
- useRimCollider, 23
- VehicleSide, 28
- vehicleSide, 23
- Visual, 28
- wheel, 23
- wheelHit, 23
- width, 28
- worldCenter, 28
- NWH.WheelController3D.WheelControllerEditor, 29
- NWH.WheelController3D.WheelHit, 29
 - force, 30
 - forwardDir, 30
 - forwardSlip, 30
 - normal, 30
 - point, 30
 - sidewaysDir, 30
 - sidewaysSlip, 30
- overExtended
 - NWH.WheelController3D.Spring, 10
- Parent
 - NWH.WheelController3D.WheelController, 26
- parent
 - NWH.WheelController3D.WheelController, 22
- parentRigidbody
 - NWH.WheelController3D.WheelController, 23
- point
 - NWH.WheelController3D.WheelHit, 30
- pointVelocity
 - NWH.WheelController3D.WheelController, 26
- radius
 - NWH.WheelController3D.Wheel, 14
 - NWH.WheelController3D.WheelController, 26
- reboundForce
 - NWH.WheelController3D.Damper, 5
- right
 - NWH.WheelController3D.Wheel, 14
- rimColliderGO
 - NWH.WheelController3D.Wheel, 14
- rimOffset
 - NWH.WheelController3D.Wheel, 14
 - NWH.WheelController3D.WheelController, 26
- rotationAngle
 - NWH.WheelController3D.Wheel, 14
- RPM
 - NWH.WheelController3D.Wheel, 15
- rpm
 - NWH.WheelController3D.WheelController, 26
- SetCamber
 - NWH.WheelController3D.WheelController, 20, 21
- SetDefaults
 - NWH.WheelController3D.WheelController, 21
- Side
 - NWH.WheelController3D.WheelController, 19
- sideFriction
 - NWH.WheelController3D.WheelController, 26
- SideToSideScanResolution
 - NWH.WheelController3D.WheelController, 26
- sidewaysDir
 - NWH.WheelController3D.WheelHit, 30
- sidewaysSlip
 - NWH.WheelController3D.WheelHit, 30
- singleRay
 - NWH.WheelController3D.WheelController, 23
- slip
 - NWH.WheelController3D.Friction, 6
- slipCoefficient
 - NWH.WheelController3D.Friction, 6
- speed
 - NWH.WheelController3D.Friction, 7
 - NWH.WheelController3D.WheelController, 26
- spring
 - NWH.WheelController3D.WheelController, 23
- springBottomedOut
 - NWH.WheelController3D.WheelController, 27
- springCompression
 - NWH.WheelController3D.WheelController, 27
- springCurve
 - NWH.WheelController3D.WheelController, 27
- springLength
 - NWH.WheelController3D.WheelController, 27
- springMaximumForce
 - NWH.WheelController3D.WheelController, 27
- springOverExtended
 - NWH.WheelController3D.WheelController, 27
- springTravel
 - NWH.WheelController3D.WheelController, 27
- springTravelPoint
 - NWH.WheelController3D.WheelController, 27
- springVelocity
 - NWH.WheelController3D.WheelController, 28
- steerAngle
 - NWH.WheelController3D.Wheel, 15
 - NWH.WheelController3D.WheelController, 28
- Step
 - NWH.WheelController3D.WheelController, 21
- suspensionDistance
 - NWH.WheelController3D.WheelController, 28
- suspensionForce
 - NWH.WheelController3D.WheelController, 28
- up

- NWH.WheelController3D.Wheel, 15
- UpdateFrictionCurve
 - NWH.WheelController3D.FrictionPreset, 7
- useExternalUpdate
 - NWH.WheelController3D.WheelController, 23
- useRimCollider
 - NWH.WheelController3D.WheelController, 23
- VehicleSide
 - NWH.WheelController3D.WheelController, 28
- vehicleSide
 - NWH.WheelController3D.WheelController, 23
- velocity
 - NWH.WheelController3D.Spring, 10
- Visual
 - NWH.WheelController3D.WheelController, 28
- visual
 - NWH.WheelController3D.Wheel, 15
- visualPositionOffset
 - NWH.WheelController3D.Wheel, 15
- visualRotationOffset
 - NWH.WheelController3D.Wheel, 15
- wheel
 - NWH.WheelController3D.WheelController, 23
- wheelHit
 - NWH.WheelController3D.WheelController, 23
- width
 - NWH.WheelController3D.Wheel, 15
 - NWH.WheelController3D.WheelController, 28
- worldCenter
 - NWH.WheelController3D.WheelController, 28
- worldPosition
 - NWH.WheelController3D.Wheel, 16
- worldRotation
 - NWH.WheelController3D.Wheel, 16