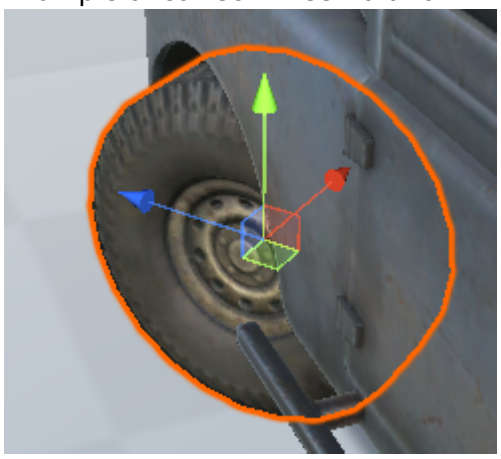
WheelController

Setup Instructions

Before you start 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 follow the following guide: https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html

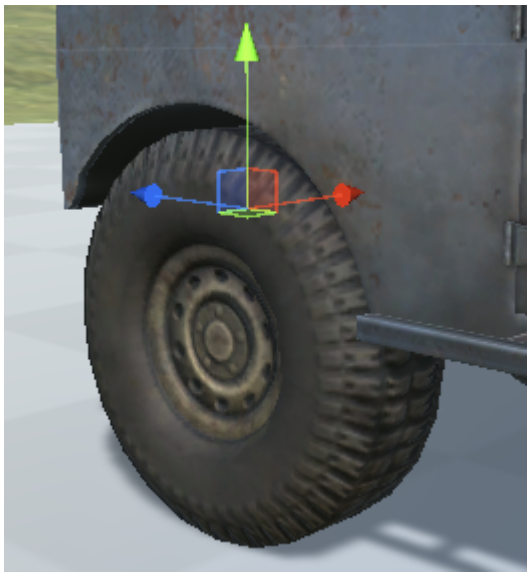
- Drag a vehicle model into the scene.
- Attach a **Rigidbody** to the model.
- Add a **collider** to the model. Make sure that the **layer** of the collider is set to Physics.IgnoreRaycast or a custom layer. For this example “WheelControllerIgnore” will be used. This step is important later on to prevent 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.
- Make sure that the **rotation** of the wheel is correct; Z-axis (blue) forward, Y-axis (green) up and X-axis (red) to the right. If not, check the link at the top of this page or use your 3D modelling software to fix the rotation. When checking switch to Local handle rotation in editor:

Example of correct wheel rotation:



- 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 above the wheel itself. 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.

Example:



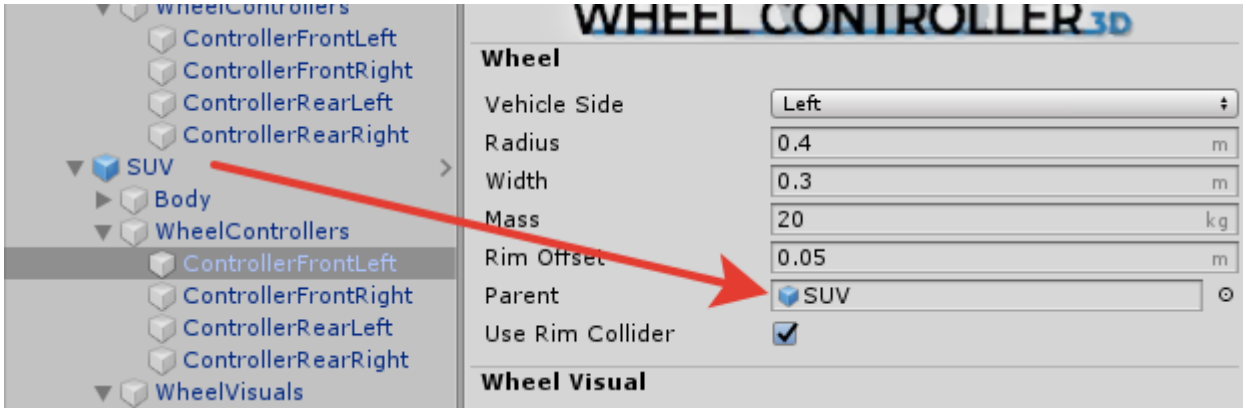
Side view:

- Attach WheelController script to object created in previous step. If everything has been set up correctly so far 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.

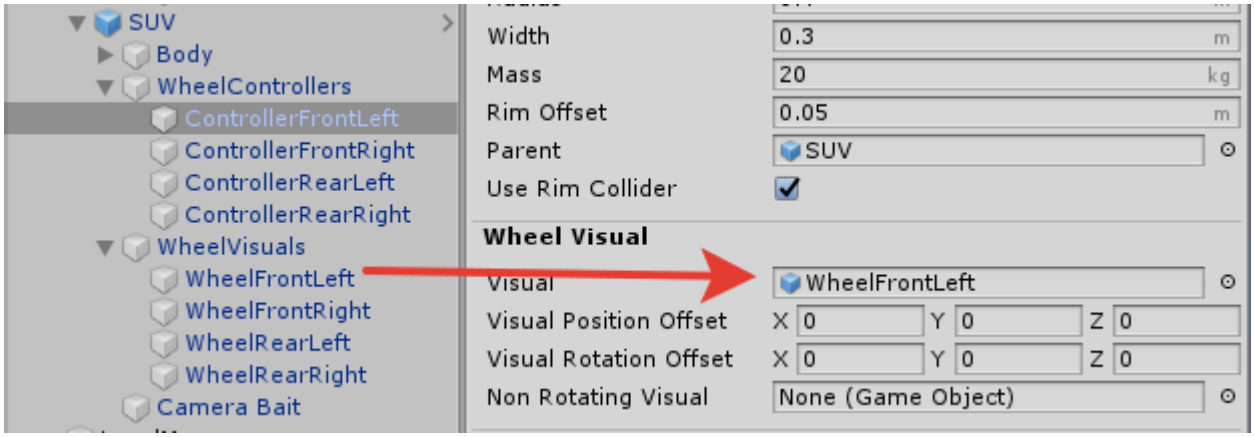
Setting Up Required Values

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:

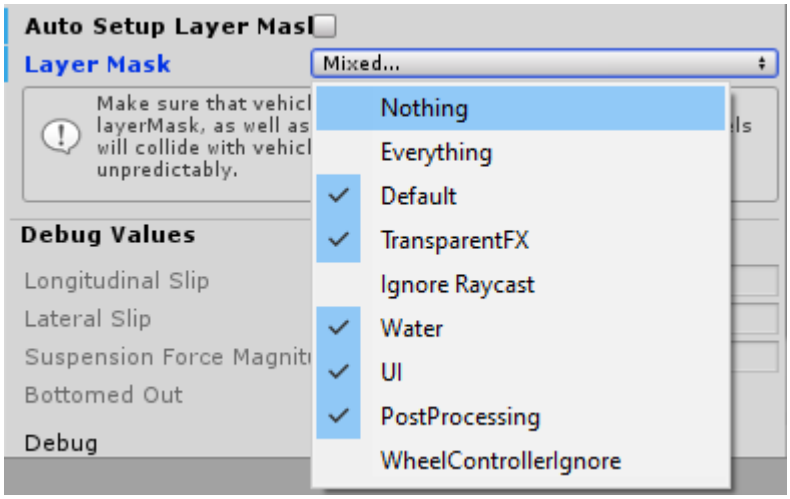
- **Parent** - parent is the GameObject containing the Rigidbody to which the forces will be applied. Usually the root of the vehicle. If it has not been assigned automatically you are most likely missing a Rigidbody on that object.



- **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.



- **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.



- **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.

A few tips for setting up vehicle suspension using WheelController:

- **Spring length** should not be shorter than ~0.2 (or 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.

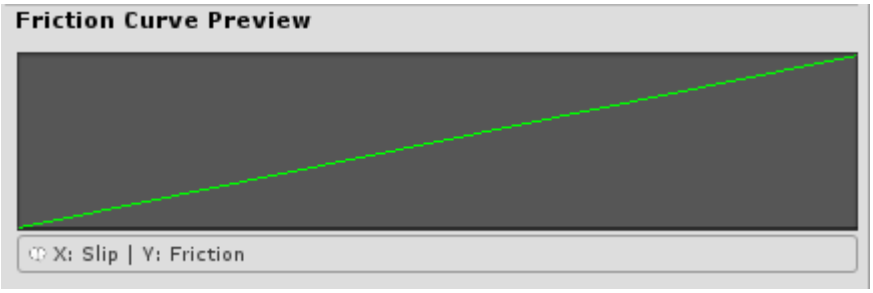
To make a vehicle drift more easily:

- Reduce *Maximum Tire Grip Force*.
- Adjust CenterOfMass to be more forward or backwards from the center of the vehicle, depending on the behavior you want.
- Set up front and rear wheels with different *Force Coefficients* or *Maximum Tire Grip Forces*. Setting any of the two values a bit lower on the rear axle will make the rear end step out more easily.

Arcade-like Behavior

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

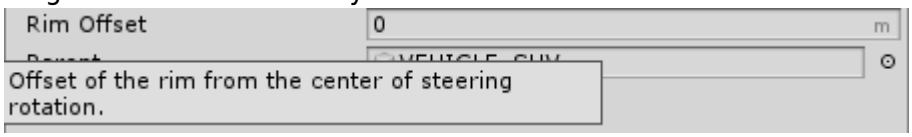
- Reduce *Slip Coefficient* and increase *Force Coefficient*. This will cause the vehicle to be less prone to slipping and spinning out. Good starting values are 0.3 for slip coefficient and 6 for force coefficient.
- Adjust the friction curve to be more linear. You might need to increase the *Force 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

To get more info about any WheelController field hover the mouse over the label:



Troubleshooting

1. My vehicle flies into the air when I press play.
    - WheelController is most likely colliding with it's parent vehicle's colliders.
    - “Queries Hit Triggers” option is enabled under Unity's physics settings and your vehicle is inside or touching a trigger-only collider. Disable “Queries Hit Triggers” or remove (untick) the layer of the trigger-only collider in WheelController's LayerMask.
2. Vehicle does not slow down.
    - By default a WheelController does not have any resistance and if Rigidbody drag is set to 0 the vehicle will continue rolling forever. To remedy this set brakeTorque to a small value from your script to represent tire friction.
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 sping strength value.
        - Too stiff friction curve / too large force coefficient / too large slip coefficient.