

Suspension 2.0

Perfect Games, January 2019

Hello! Thanks for downloading the asset, I hope you find it useful.

If you liked the asset, you can rate it on the asset store page. This helps a lot in promoting assets.

You can also take a look at my other assets:

An asset with excellent car physics:

[Universal Car Controller](#)

ACC series, well optimized for mobile projects:

[Arcade Car Controller \(Lite version\)](#)

[Arcade Car Controller \(Standard version\)](#)

[Arcade Car Controller \(Multiplayer version\)](#)

Third-party assets that I myself use and recommend to everyone:

[EasyRoads3D Pro v3](#) - An asset with which it is easy to create roads for games of any genre.

[Gaia 2 - Terrain & Scene Generator](#) - An asset with which it is easy to create and generate terrains of any complexity.

Suspension 2.0

This is an asset, a set of scripts that change the position and rotation of objects. The physical behavior of the car or the collider does not affect.

This asset contains scripts:

- **ArcMoveFromWheelCollider.cs** - This script is needed to move the wheel in an arc.
- **Axle.cs** - This script is needed to change the position and rotation the axle.
- **LookAt.cs** - This script is needed to look an object while maintaining rotation along the z axis.
- **LookAtOneAxis.cs** - This script is needed to rotate object by only one axis.
- **MoveBySteerAngle.cs** - This script is needed to move the object based on the angle of angle of the WheelCollider.
- **SetPosition.cs** - This script is needed to move the object to target point with offset.
- **SetRotationByRPM.cs** - This script is needed to rotate object from WheelCollider.rpm.
- **SetRotationBySteerAngle.cs** - This script is needed to rotate object from WheelCollider.steerAngle.
- **SpringSheet.cs** - This script is needed to change the geometry of the suspension sheets.
- **SuspensionHelper.cs** - Contains helper methods.
- **WorldPosFromWheelCollider.cs** - This script is needed to change position and rotation from WheelCollider.

Code style:

Variables that need to be changed from the editor are labeled as private and with “SerializeField” attribute.

```
[SerializeField] WheelCollider WheelCollider;
```

Private and public fields or properties belonging to the class begin with a capital letter:

```
[SerializeField] WheelCollider WheelCollider;
```

```
float StartDistance;
```

```
Vector3 TargetPoint { get { return Target.TransformPoint(Offset); } }
```

Parameters of the method and variables declared in the body of the method begin with a small letter.

```
public VerticesGroup (Mesh deformingMesh, List<Vector3> vertices) {  
    StartCenterPoint = Vector3.zero;  
  
    var verticesIndexHashSet = new HashSet<int>();  
    VerticesCenterOffset = new Dictionary<int, Vector3>();  
  
    foreach (var vert in vertices) {  
        StartCenterPoint += vert;  
    }  
    StartCenterPoint /= vertices.Count;  
}
```

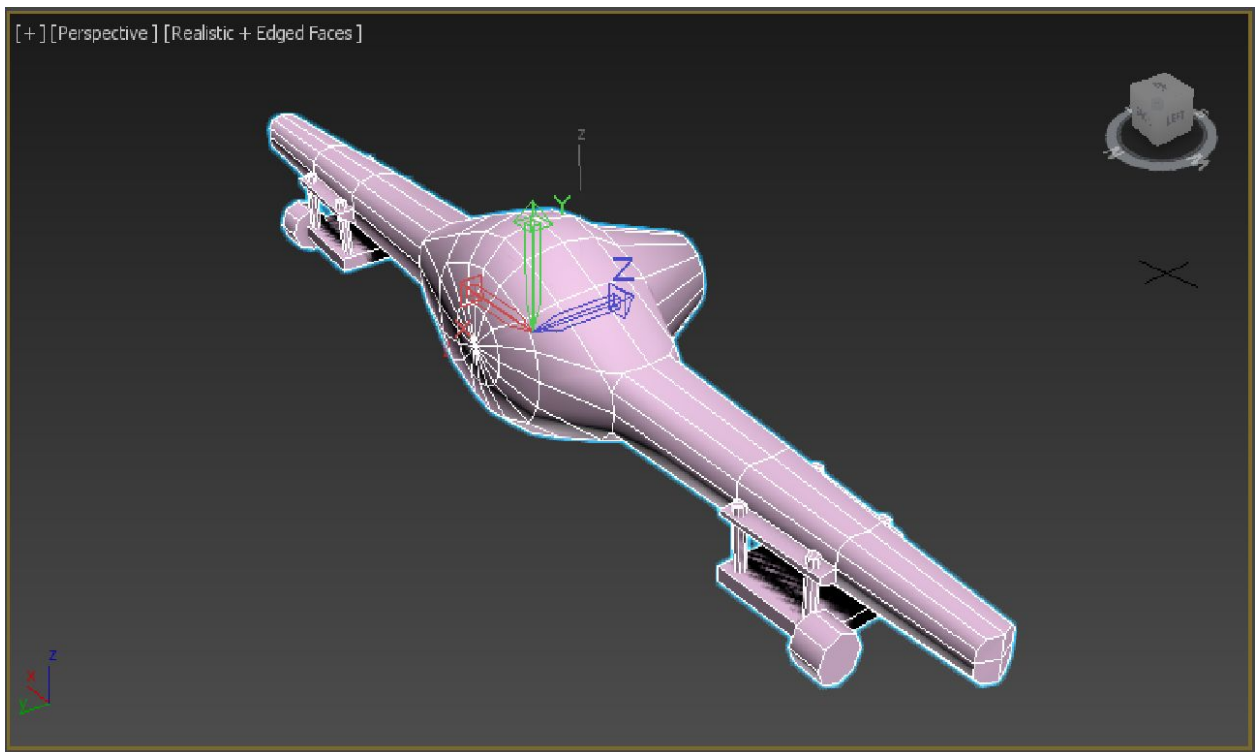
All fields, properties and method names are chosen to maximize the destination.

Model preparation:

Everything is done intuitively, in the demo scene presents an example of the work of all scripts. The rotation of the pivot points of the model is not important, the settings can set the correct direction (except SpringSheet.cs and Axle.cs).

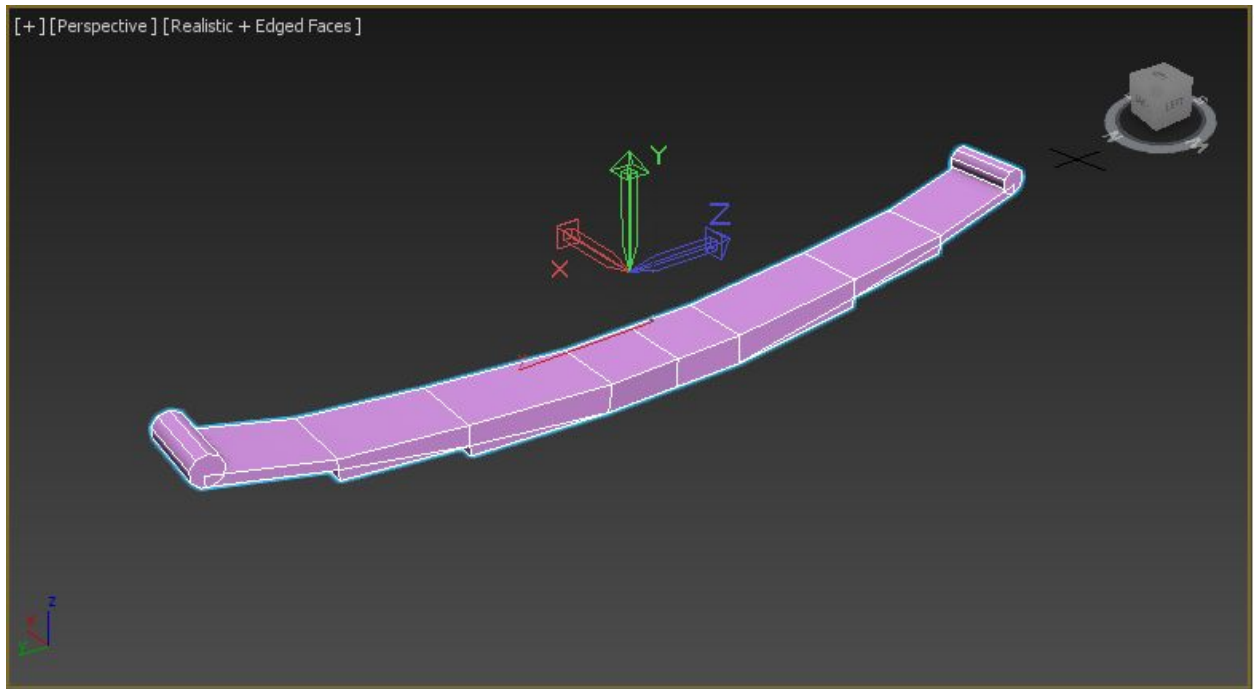
Axle

The pivot point should be rotated as follows:



SpringSheet

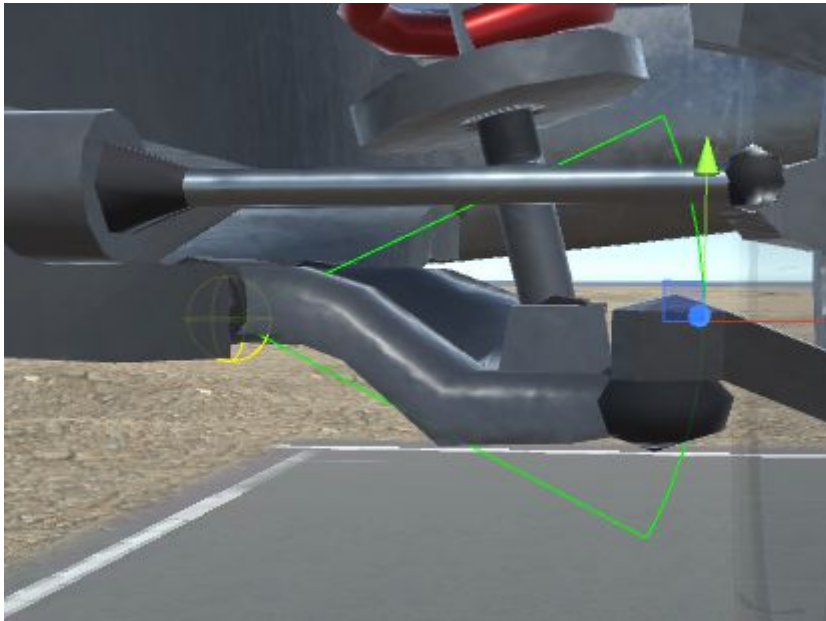
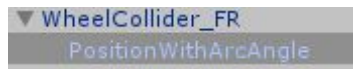
The pivot point should be rotated as follows:



How to use:

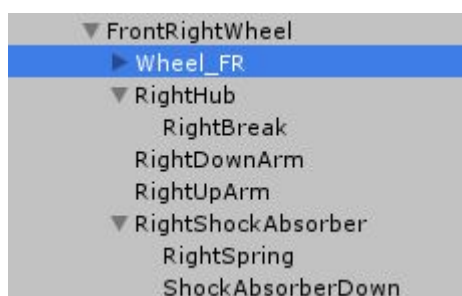
Independent suspension On the example of the right wheel:

1. Wheel position with angle of arc:



This is the main position of the wheel, depending on the position of the wheel on the axis-y will change the position of the axis-x.

Hierarchy all objects of front right wheel



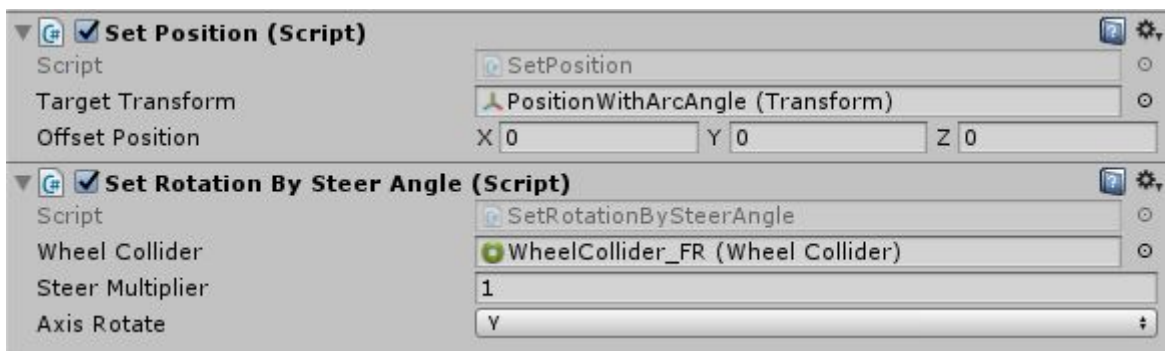
2. Wheel view position and rotatin:

The wheel view must be assigned the position of this object using a script "SetPosition.cs"



Rotation of the wheel view is attached using a script "WorldPosFromWheelCollider.cs", need only set checkbox "SetRotation".

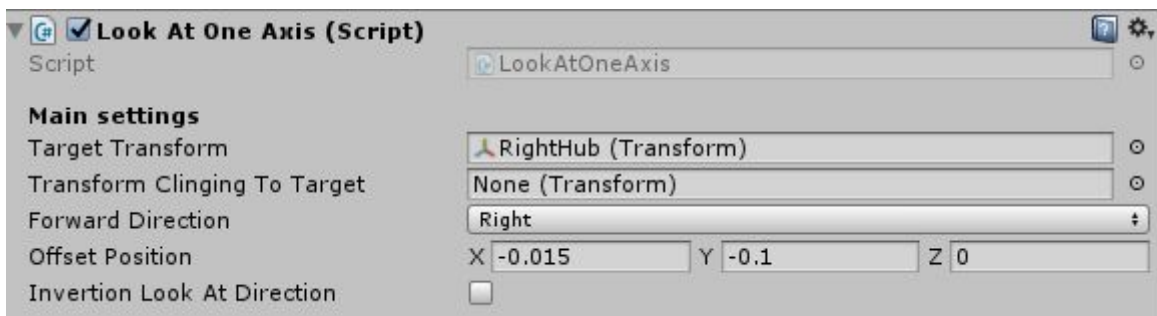
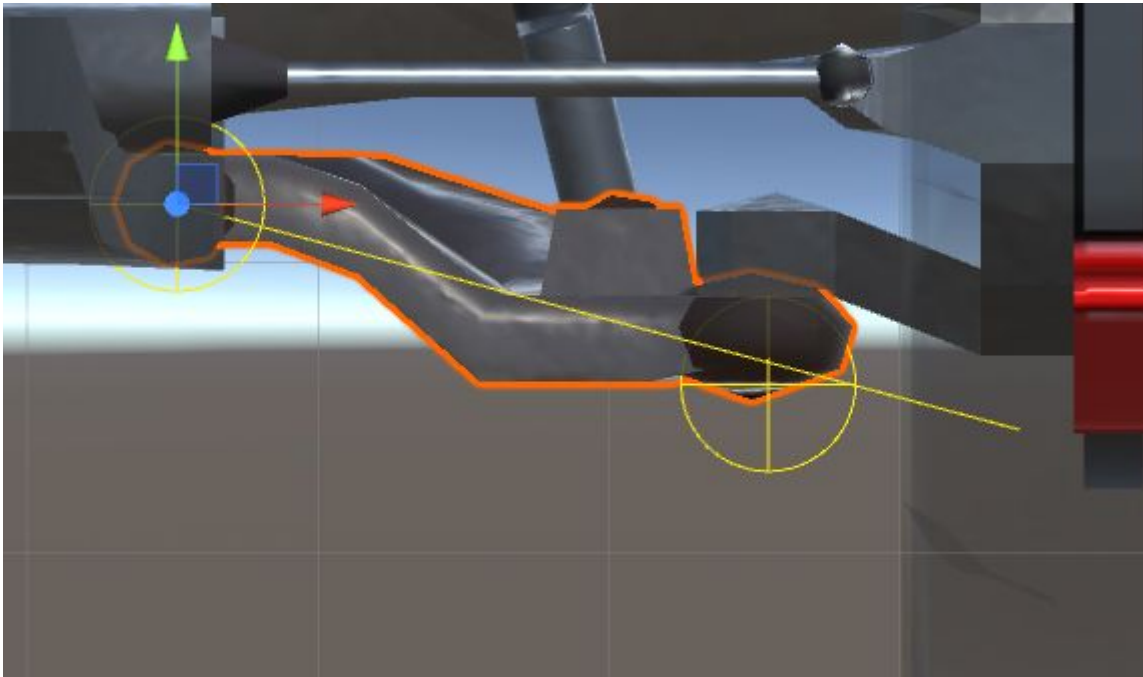
3. Hub:



The position is set in the same way as the wheel. Rotation is set using the script "SetRotationBySteerAngle.cs"

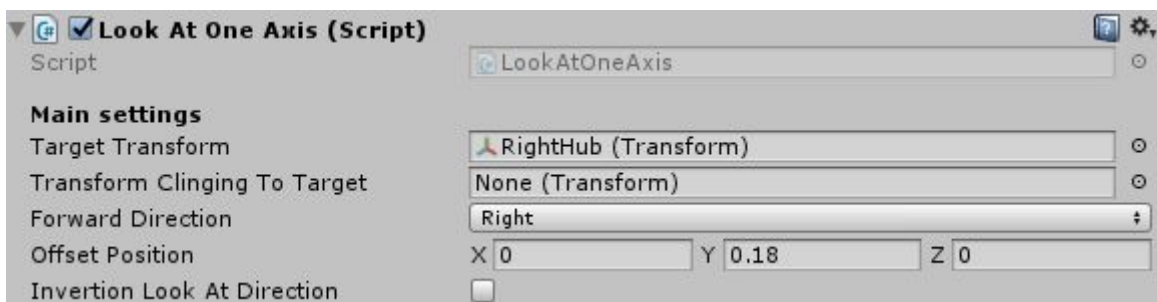
4. Arms:

Rotation occurs only on the x axis. Using a script "LookAtOneAxis.cs". Gizmos will help set the correct settings.

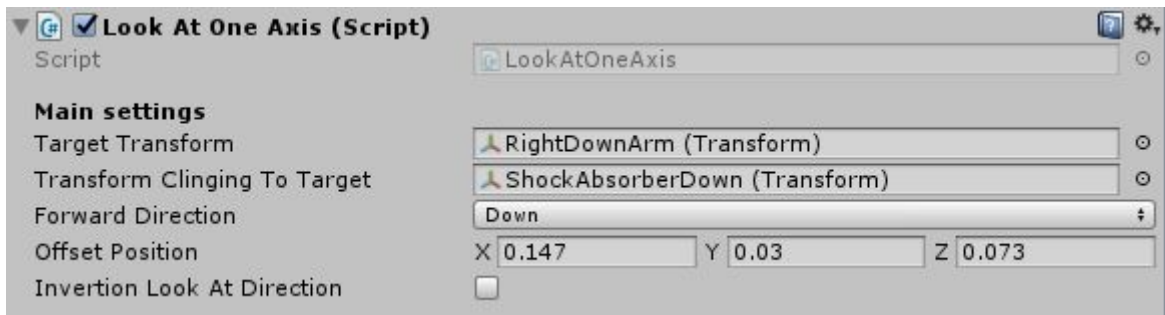


The upper arm and shock absorber is configured the same way, but with different offset.

upper arm:

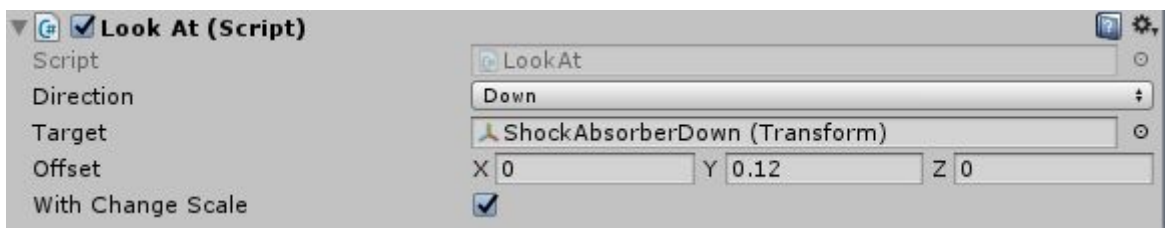
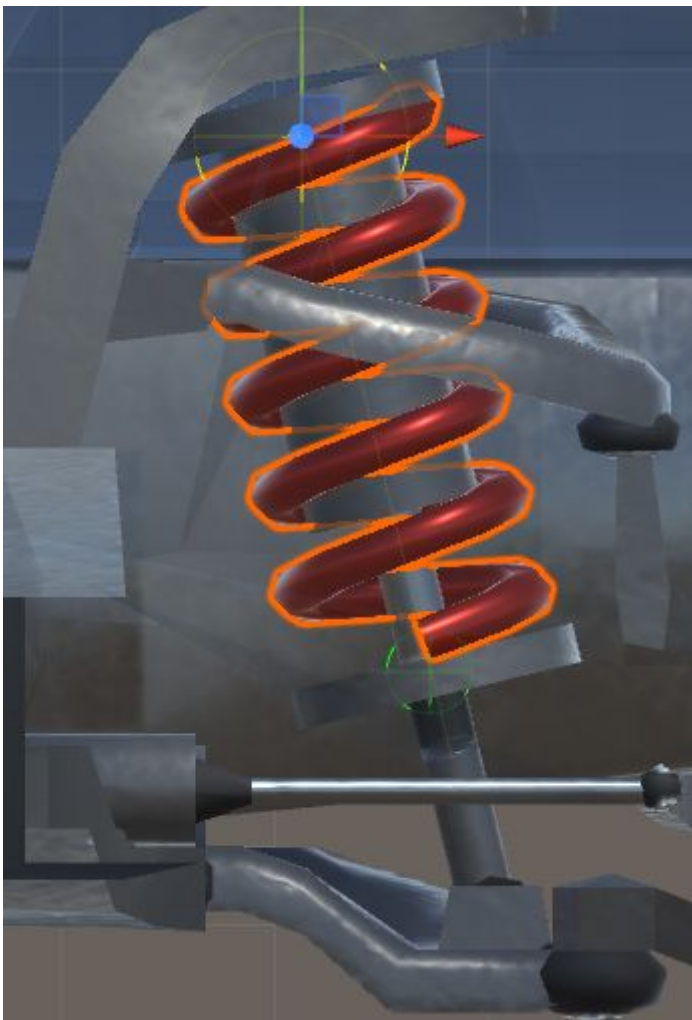


shock absorber:



5. Front spring:

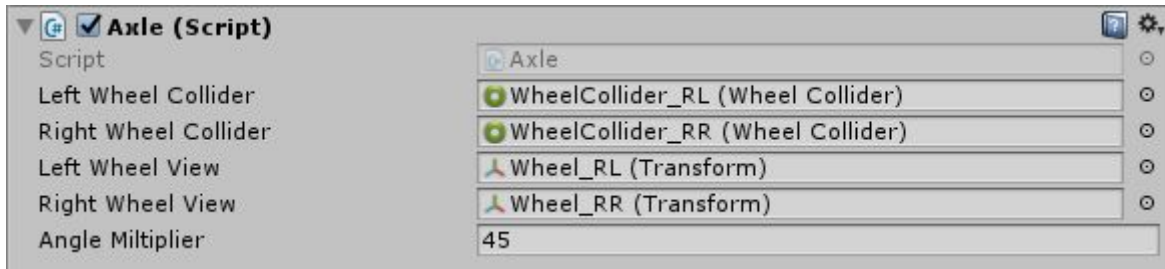
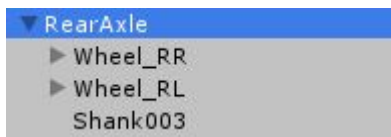
The length and direction of the spring is changed using a script "LookAt.cs"



The left wheel is configured the same way, only in some scripts you need to choose an inversion.

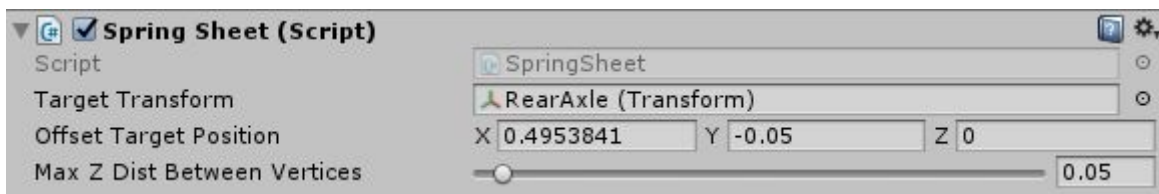
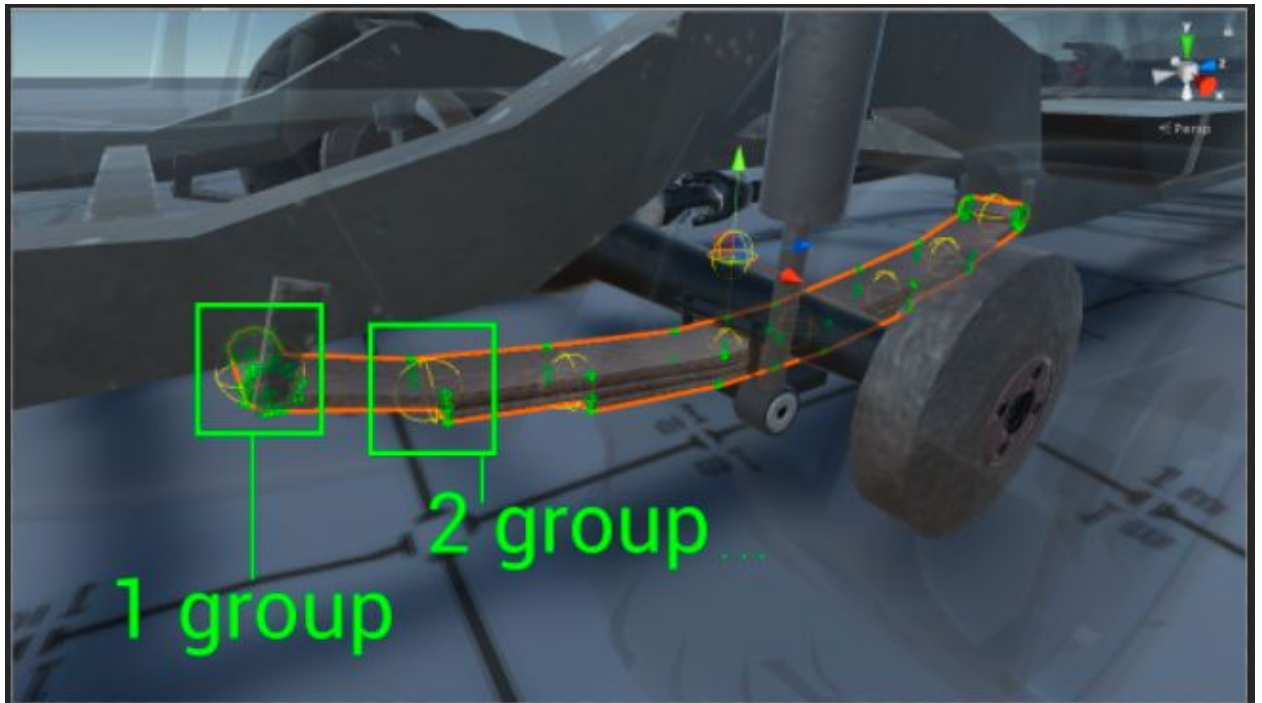
6. Axle:

Position and rotation are changed using the script “Axle.cs”. Also in this script changes the rotation of the rear wheels.



7. SpringSheet:

Geometry is changed by the script “SpringSheet.cs”.



Hierarchy all objects.

- ▼ Pickup_Opacity
 - ▶ View
 - ▶ WheelCollider_FR
 - ▶ WheelCollider_FL
 - WheelCollider_RR
 - WheelCollider_RL
 - ▼ Suspension
 - ▼ FrontRightWheel
 - ▶ Wheel_FR
 - ▼ RightHub
 - RightBreak
 - RightDownArm
 - RightUpArm
 - ▼ RightShockAbsorber
 - RightSpring
 - ShockAbsorberDown
 - ▶ FrontLeftWheel
 - ▼ RearAxle
 - ▶ Wheel_RR
 - ▶ Wheel_RL
 - Shank003
 - ▶ LeftBackShockAbsorber
 - ▶ RightBackShockAbsorber
 - LeftBackSpring
 - RightBackSpring
 - ▶ SteeringRack
 - Com

On all questions you can contact by e-mail:

war.90@mail.ru

Bugrov1990@gmail.com