Cars and Trucks Autodrive System v7.1.1 for Blender v2.49b.
A simple and efficient setup allowing automated animation for Cars and Trucks, on all kinds of roads and terrains.

Philippe ROUBAL. http://3d-synthesis.com

Cars and Trucks Autodrive System 7.0 for Blender v2.49b. 27/10/2009.

This setup is released under Creative Commons BY 2.0 License. France.

SPECIAL THANKS AND CREDITS :

To "Litteneo" (Jérôme MAHIEUX) For the wheelsrotation.py script, released under Copyleft license.
"littleneo" is the author of the famous Python script "Blended Cities", generating amazing procedural cities.
You can find this script on his Website : http://jerome.le.chat.free.fr

WARNING :

Please type Alt+P with the cursor in the text window below and run the animation from Frame 1 to activate the script.

FEATURES OF THIS SETUP :

- Altitude Sensors allowing driving on bumpy roads, with slopes and bridges. This setup is more versatile than previous versions, but the geometry of the chassis may not be so accurate on very bumpy terrains. For specific 4x4 animations, you can use instead my 4x4 Autodrive system, available on my website.

- Only one Empty to move to animate the car: Empty-Front. This Empty can be keyframes by hand, driven on a curve by using a Clamp To constraint, or parented to an other Empty driven by a Clamp To constraint, or even animated by a script (OpenSteer). In the blend file provided with this tutorial, the Empty-Front is parented to Empty-Guide, which is driven on the curve by a Clamp-To constraint.

- Speed control by an unique Ipo curve applied to the Empty-Guide.

- 4 independent wheels with dynamic suspension.

- Tires deformation.

- Automated Animation of the steering wheel.

- Automatic transversal slope in curves.

- Trajectory easily editable in Top view by editing a simple curve.

- Fine secondary animation of the car body available through the car body itself and the two empties Empty-WheelsDropRear and Empty-WheelsDropFront .

- Very light Setup :

  - 4 Empties (Empty-Guide is not necessary). Only 2 Empties If Secondary animation is not required.

  - 1 Lattice.

  - 9 solid objects. Only 7 If Secondary animation is not required on the rear wheels. Only 6 if you tires deformation is not required (in this case you can also remove the Lattice).

- Just add the visible objects : Car body, wheels, Steering wheel.

NOTE : The rear wheels are be parented to the objects "Rim-Rear.R" and "Rim-Rear.L", but they can also be parented to the object "Chassis", either directly to the object itself or by Vertex Parenting, using Vertex groups as reference.

The front wheels are parented to the objects "Steer-left" and "Steer-Right". The rear wheels copy the Z location of the external rear vertices of the object "Sensor", and the "Steer-left" and "Steer-Right" objects copy the Z location of the external front vertices of the object "Sensor" (vertices in the four corners).

LAYERS :

Layer 1 : Car rig and wheels.

Layer 2 : Car body and steering wheel

Layer 3 : Road and road curve.

Layer 4 : Bridge.

Layer 5 : Lamps and cameras.

Layer 11 : Character, Helmet and glasses.

Layer 12 : Armature of the Character.

Layer 13 :

Layer 14 :

SOME EXPLANATIONS ABOUT THE SETUP :

- The purple object named Steer-Pointer has two vertices in two groups. They are used as tracking targets to control the transversal slope of the Suspension object and the rotation of the steering wheel.

- Select the inboard camera to see the rotation of the steering wheel (can be adjusted by the influence factor of the Tracking constraint).

USE OF THE Empty-Steering-Target :

This empty has a very important function. It is used to control the steering. The object name "Steer-Pointer" is locked on the Z axis tracking the Empty-Steering-Target.

The location of the Empty-Steering-Target can be defined in two manners. In the current blend file, this Empty is parented to the chassis of the car, and affected by a negative Time Offset (-1.5 frame). As the distance between the chassis and the empty vary with the speed this Time Offset has to be ajusted each time you make an important modification in the speed of the car. If you use a curve as a trajectory guide for the car, you can apply to the Empty-Steering-Target a Clamp To constraint, as you do for the Empty-Guide, or the Empty-Front. By moving the Empty along the Curve, you can adjust the distance between the chassis and the Empty. As they are driven by the same Loc Ipo, this distance will remain constant independently of the Speed.

USE OF Empty-Guide and Empty-Front :

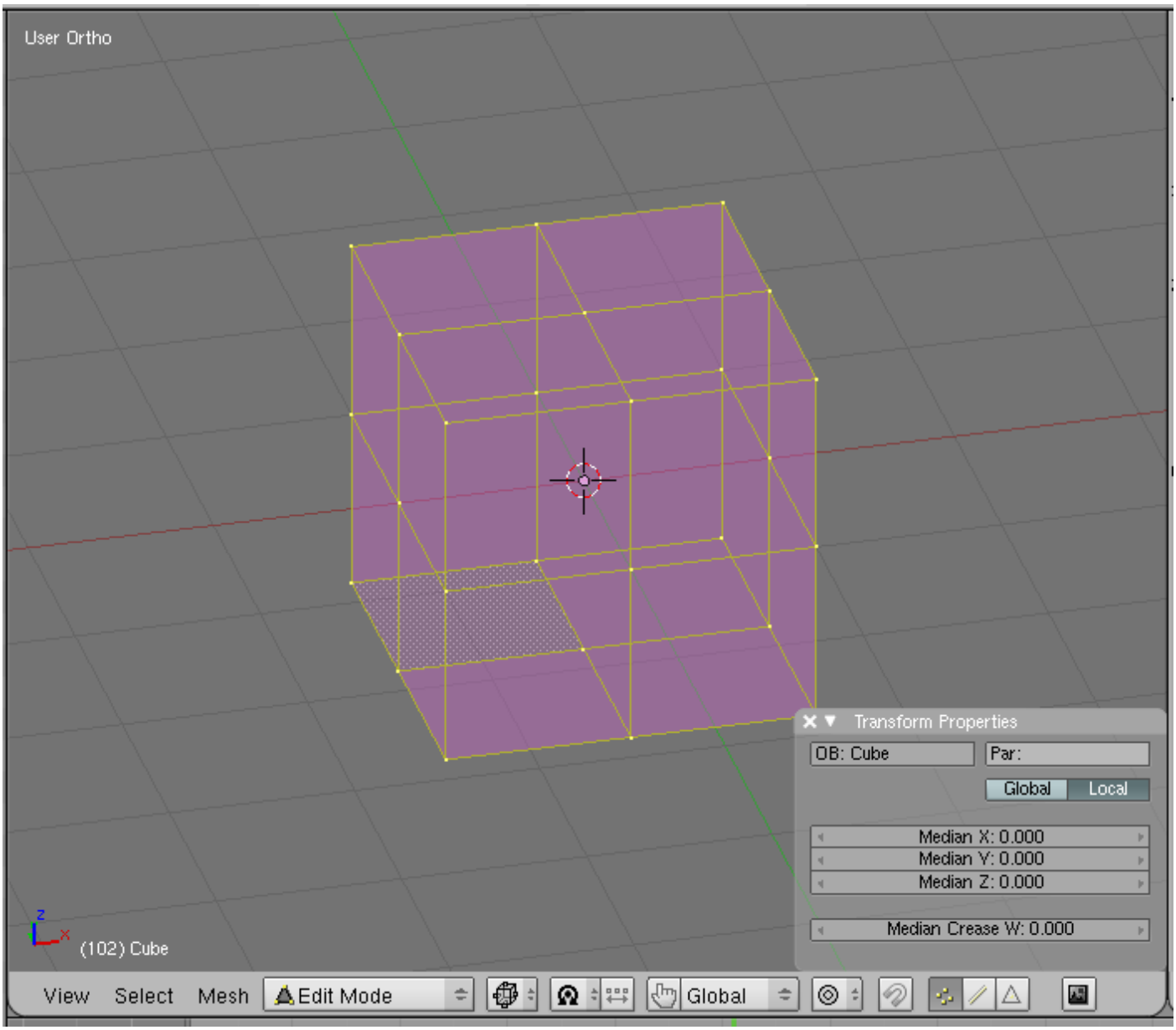
Empty-Front can be used alone to move the car. It can be either keyframed or driven on a curve through a Clamp To constraint.

For my own, I prefer to use one more Empty. So I have added Empty-Guide. Empty-Front is parented to Empty-Guide. The goal of Empty-Guide is to act as a "relay". This Empty gives more freedom when animating. This "relay" allows to drive several cars on the same curve for example, allowing them to have different behaviours and trajectories, although sharing the same Curve and Ipo.

CONSTRUCTION OF THE CAR RIG :

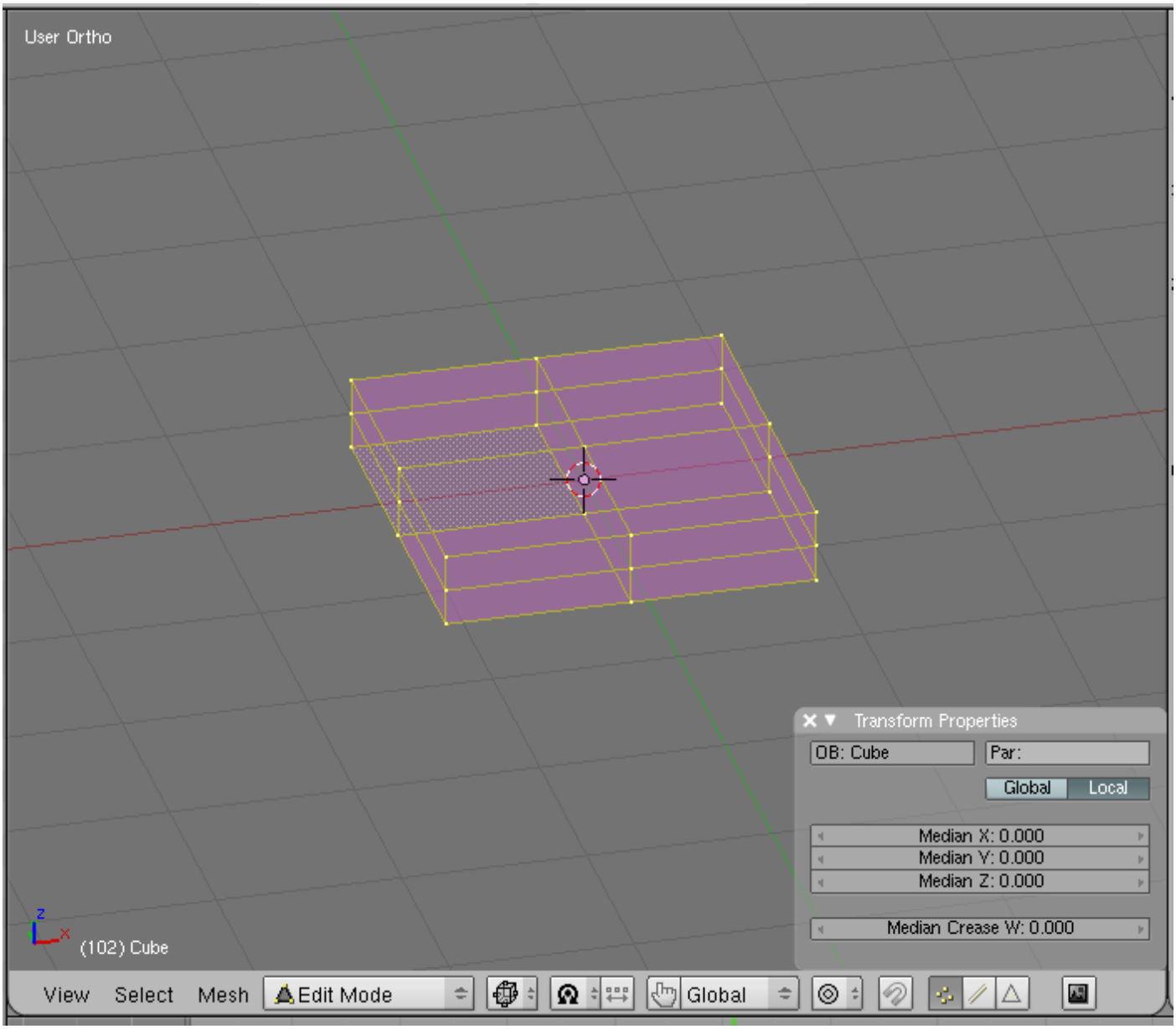
Step 1 : Chassis following an Empty.

- First, Add a cube and name it Chassis. Subdivide it one time.



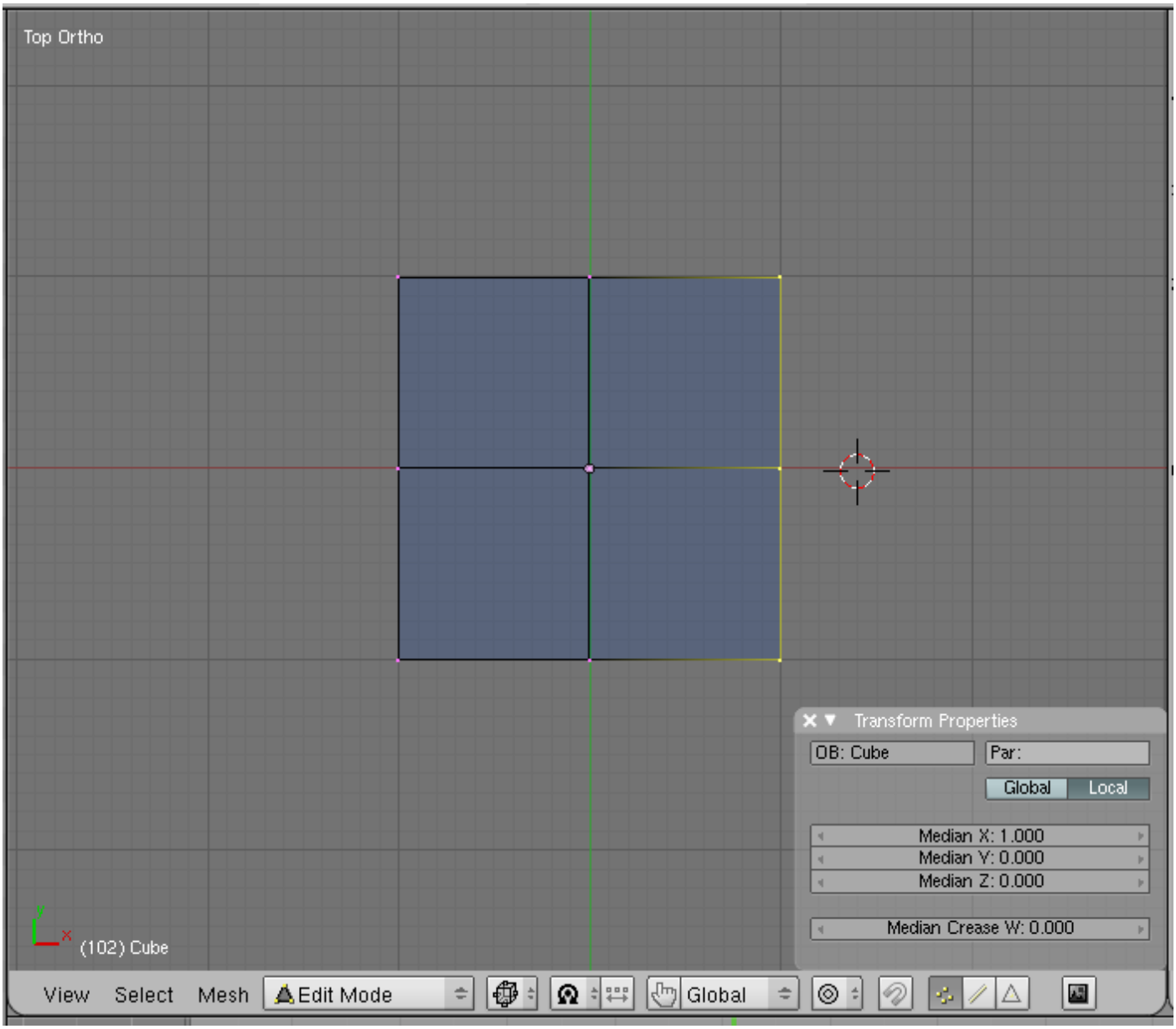
#001

- Then, Scale it in Z direction : SY0.2.



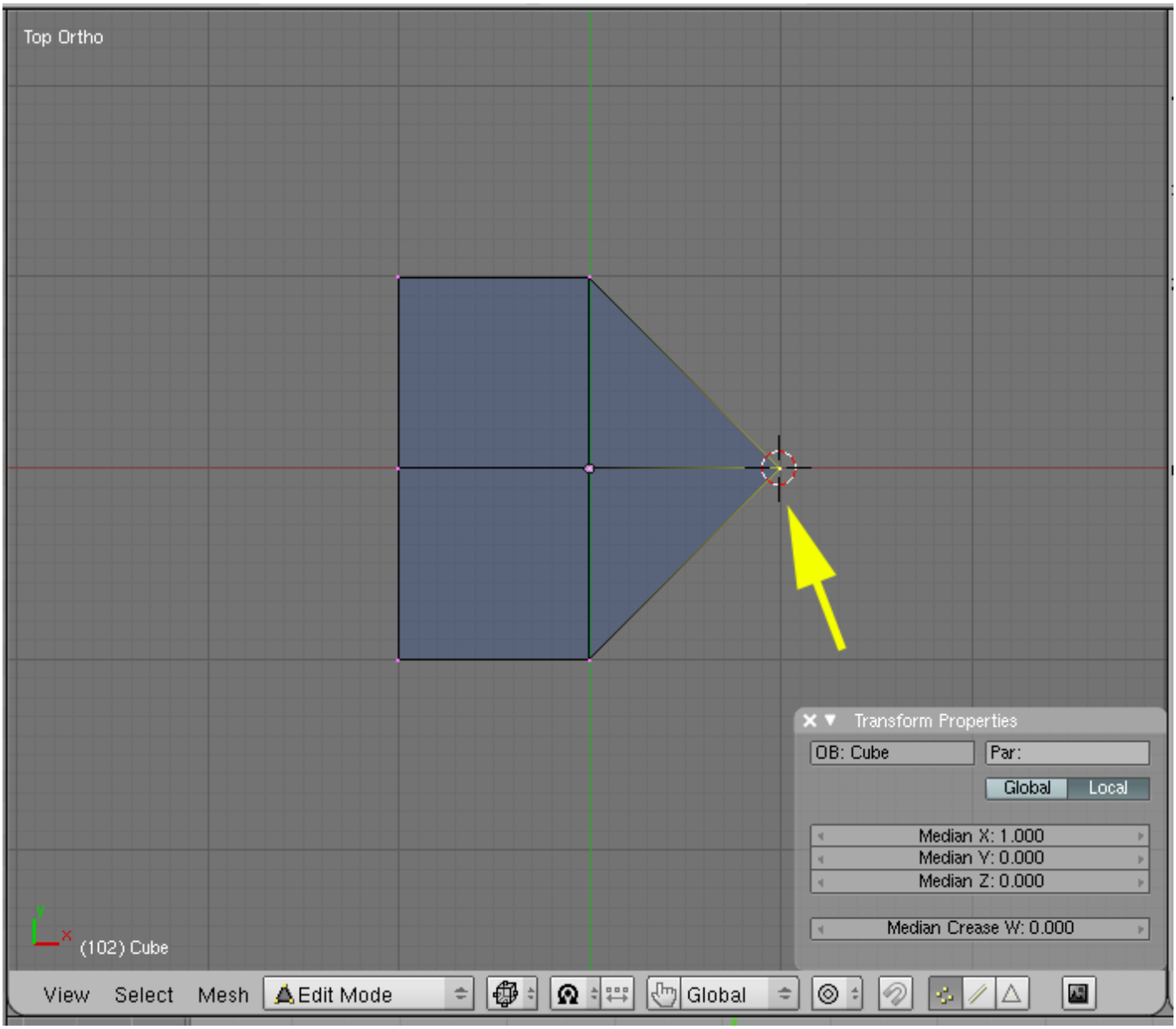
#002

- In top view, select all the vertices of the cube on the right side of the screen.



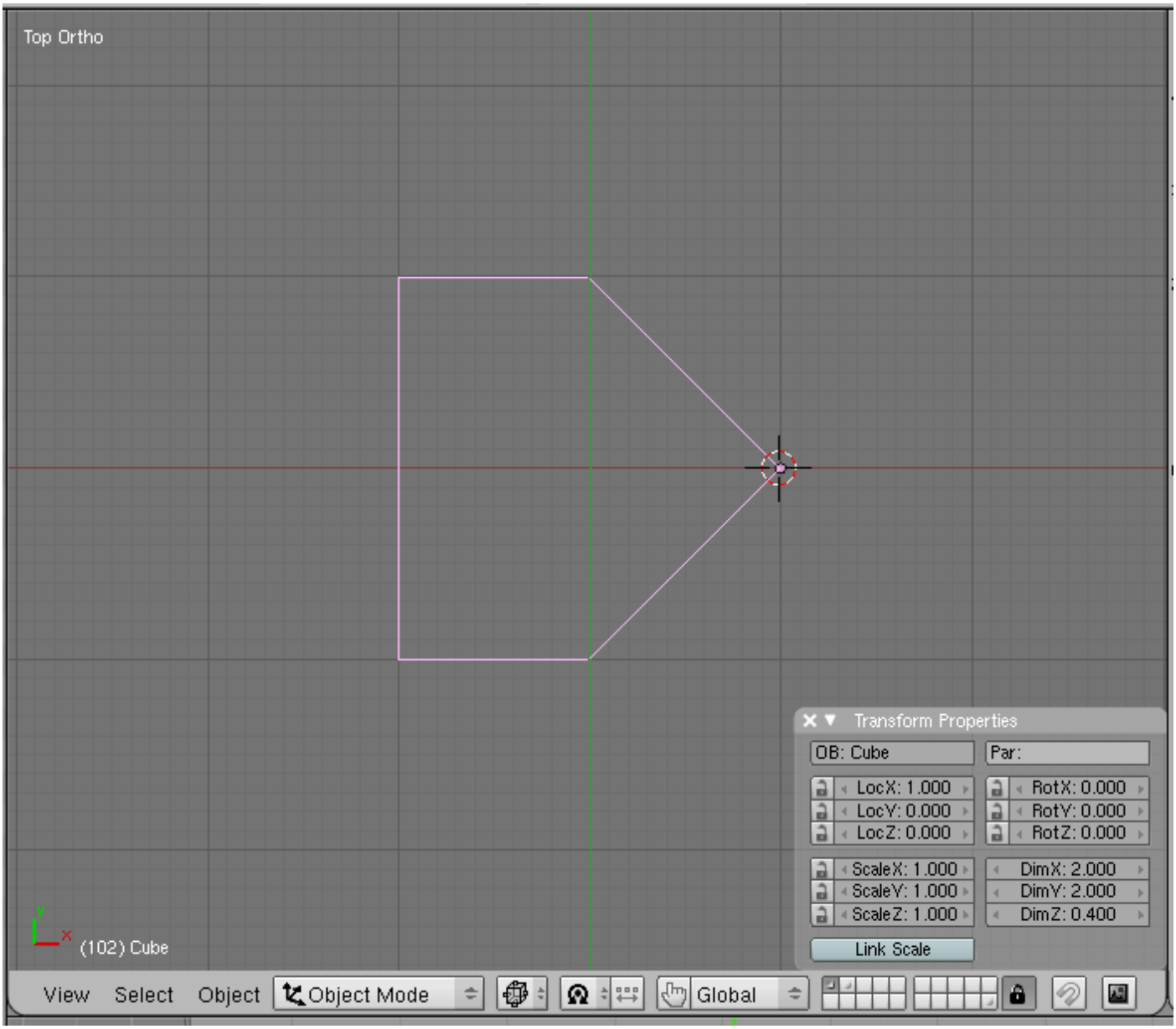
#003

- Scale to 0 in Y direction and remove doubles.



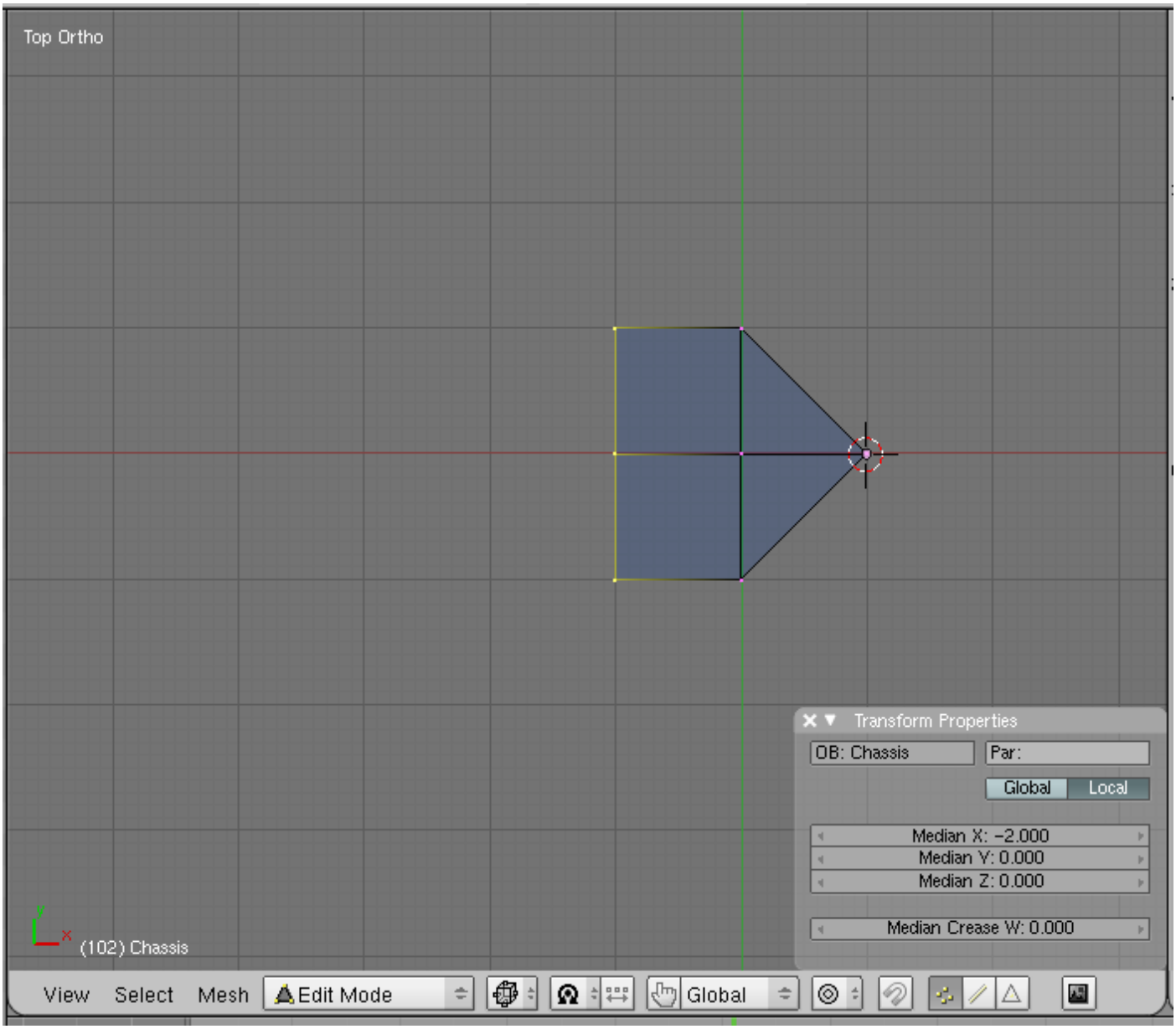
#004

- In Object Mode, bring the center of the Chassis object on the scaling center.



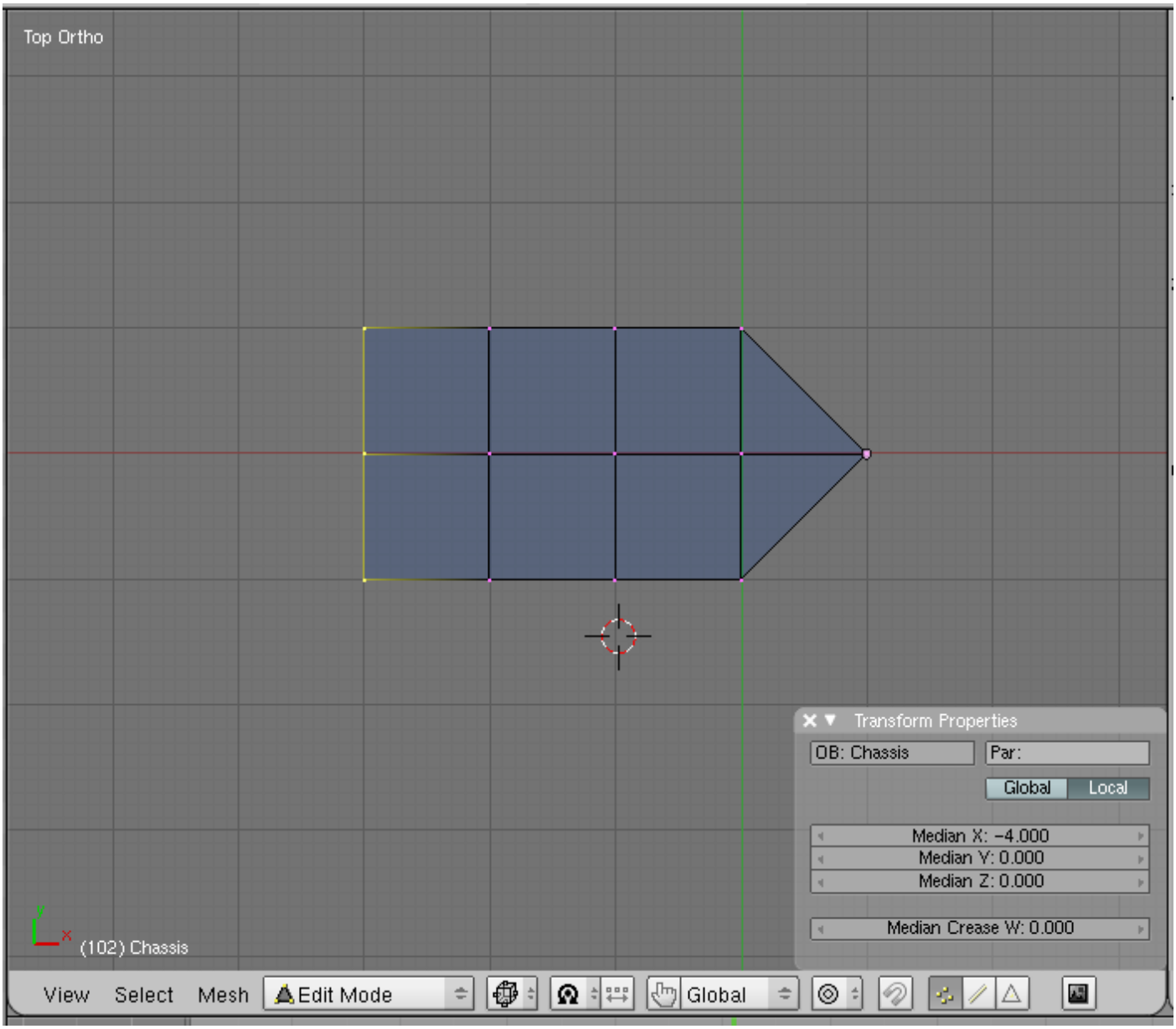
#005

- In Edit mode, Select the vertices on the left of the screen.



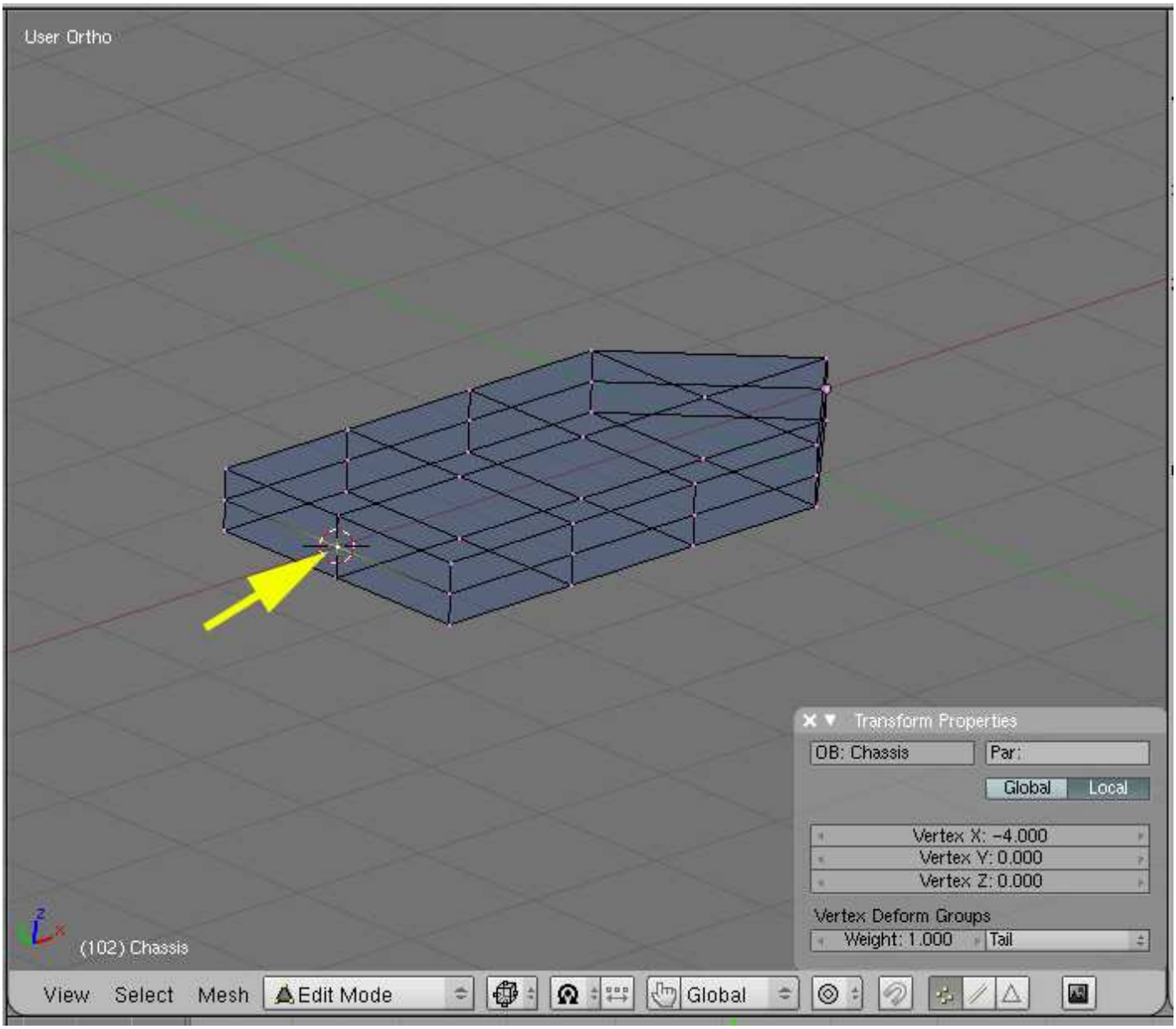
#006

- Extrude two times of -1 Blender Unit.



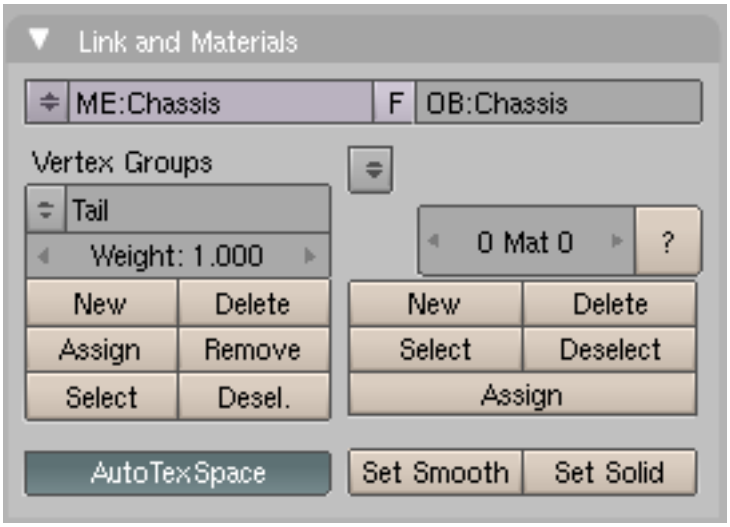
#007

Now, the Chassis object has the shape of an arrow head.
- Select the center Vertex at the rear side of this arrow, in the Vertex layer set at half height of the mesh.



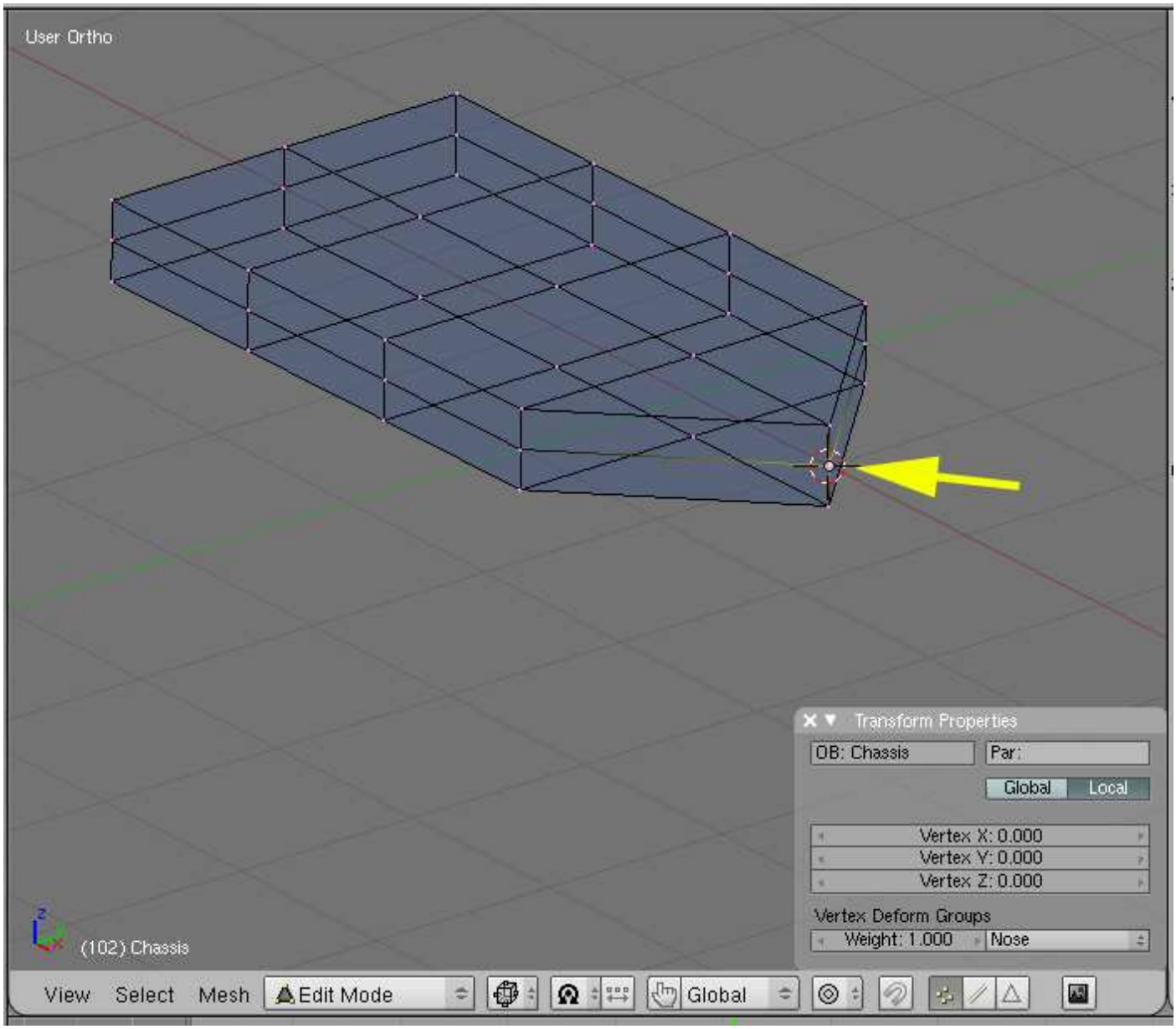
#008

- Create a vertex Group including only this Vertex, give it a Weight of 1.0, and name it "Tail".



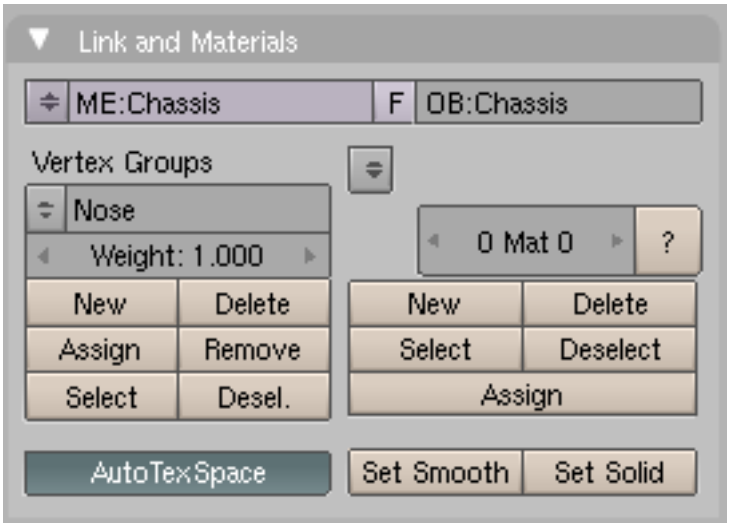
#008B

- Select the Vertex at the opposite side of the arrow. This Vertex is located at the location of the object center.

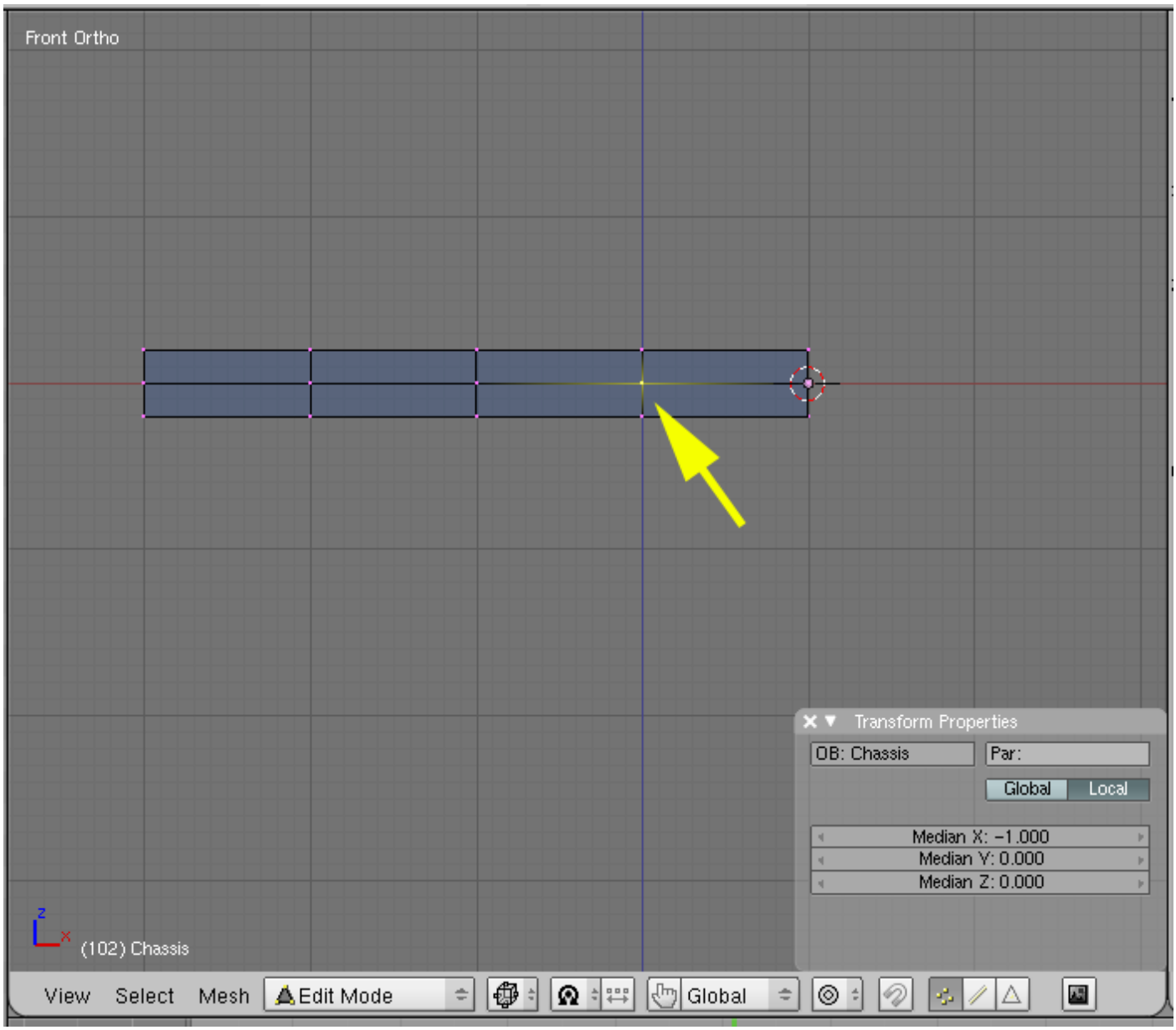


#009

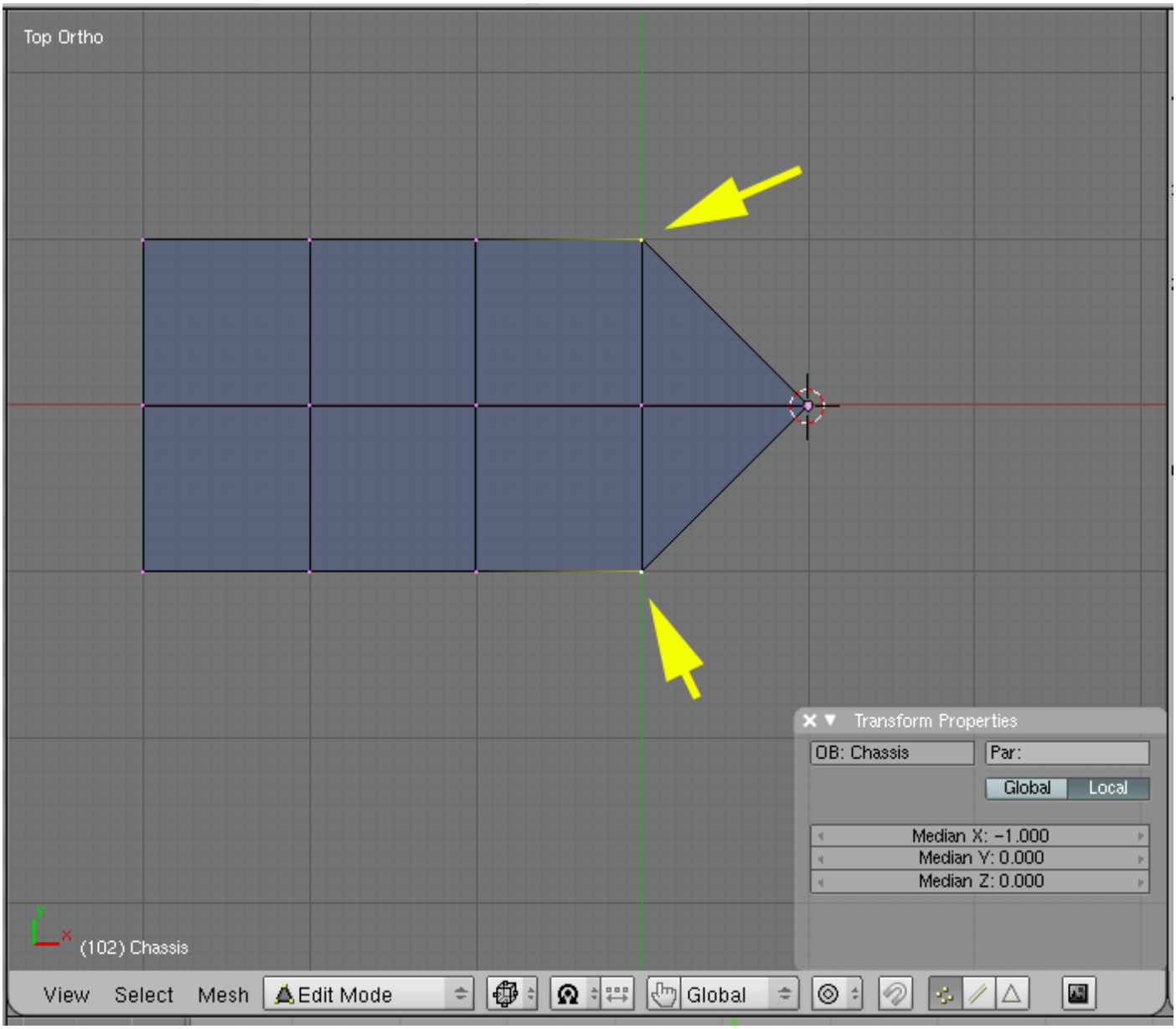
- Create a Vertex Group including only this vertex and name it "Nose".



#009C
- In Front Ortho view, select the two vertices in the middle height of the mesh, just before the center object location.

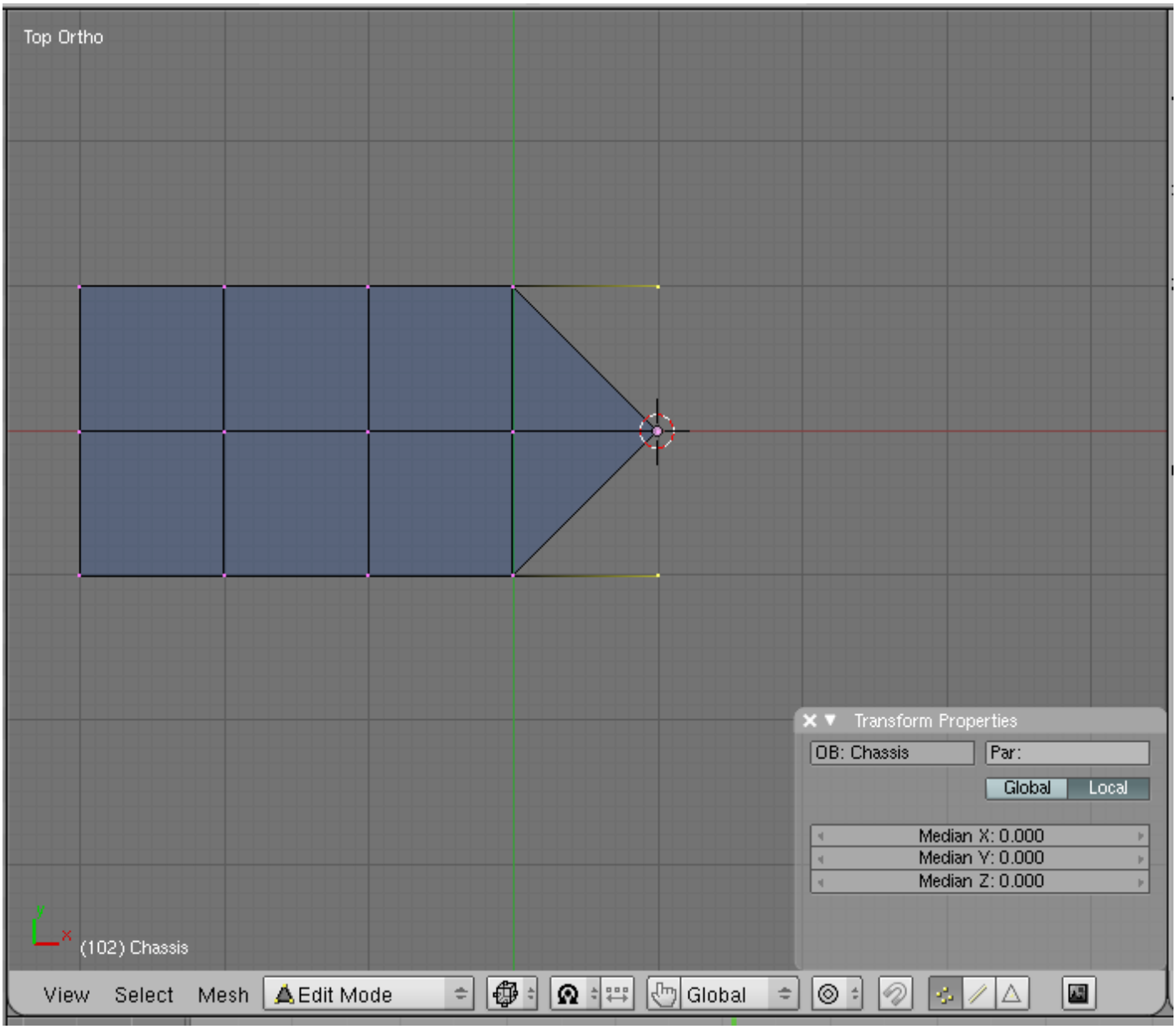


#010


Top Ortho
Transform Properties
OB: Chassis
Par:
Global
Local
Median X: -1.000
Median Y: 0.000
Median Z: 0.000
(102) Chassis
View
Select
Mesh
Edit Mode
Global

#011

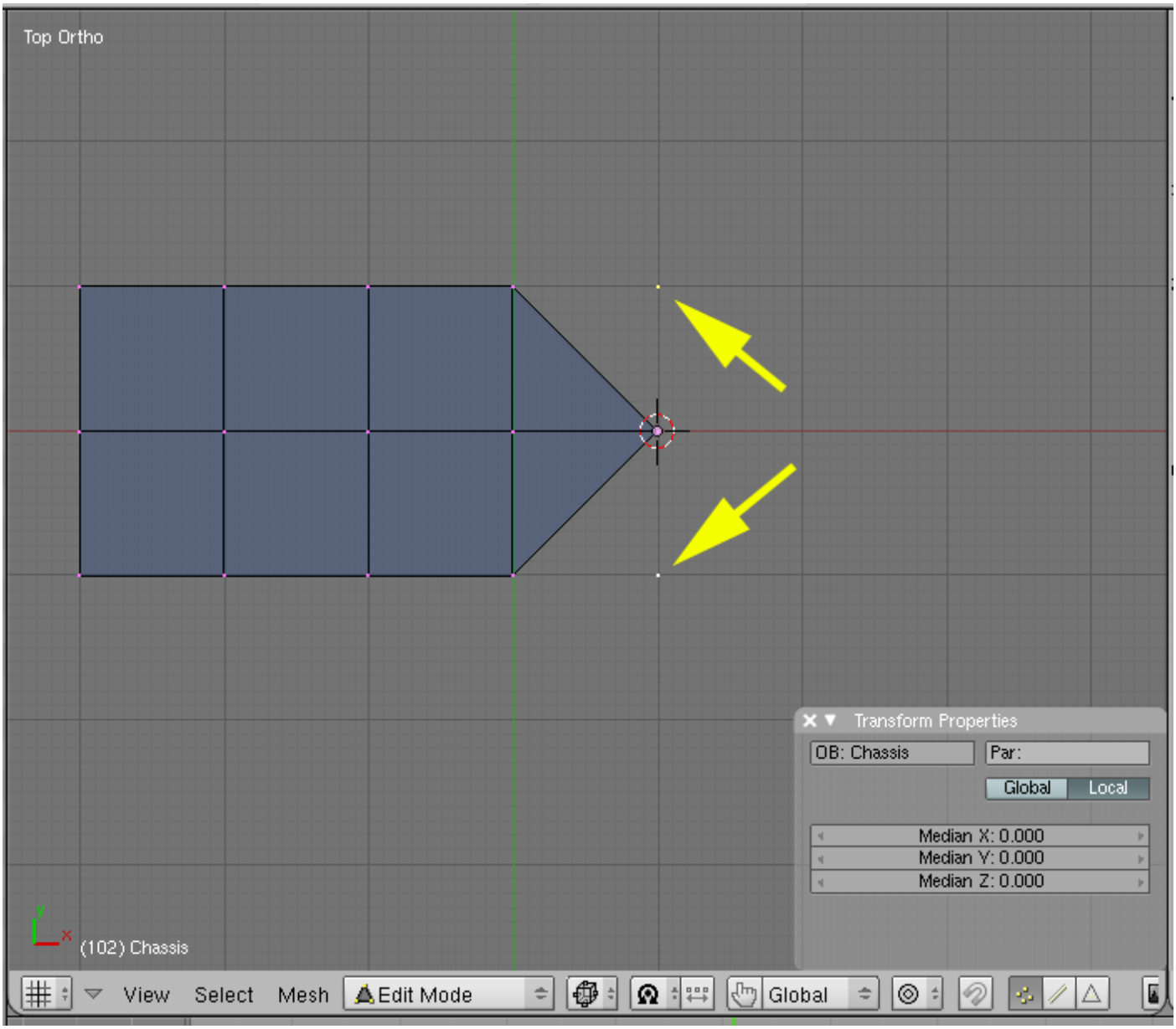
- Extrude on the right of 1 Blender Unit in the direction of the object center.



#012

NOTE : These two vertices are only useful if you create a car rig that will be used on a flat ground. If you are going to use the rig on a bumpy terrain, we will use the corners of the Sensor plane for the wheels location.

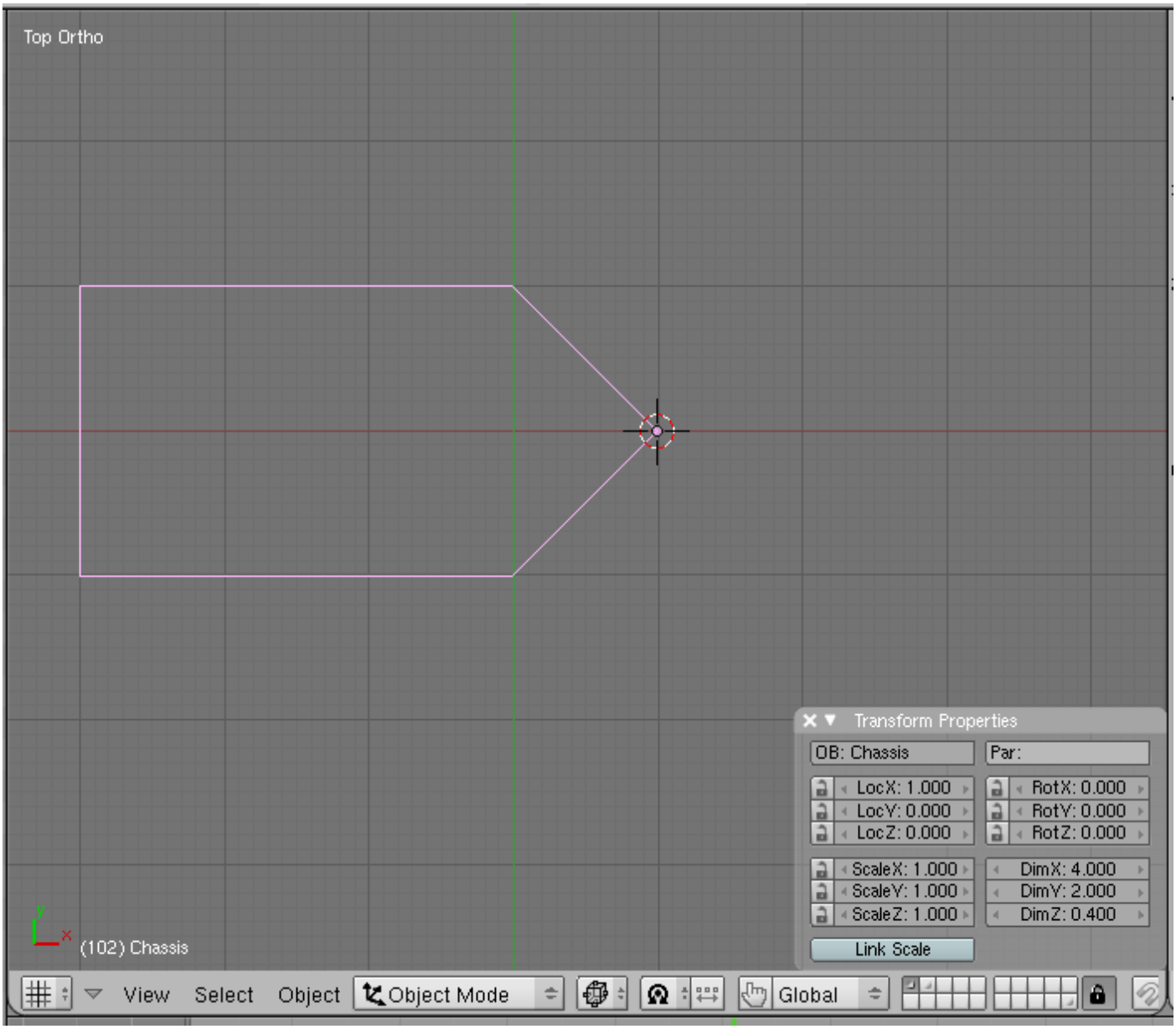
- As we need only the vertices, you can delete the two edges that you have just extruded.



#013

NOTE : Don't forget to remove these vertices later if they are not used or no more needed.

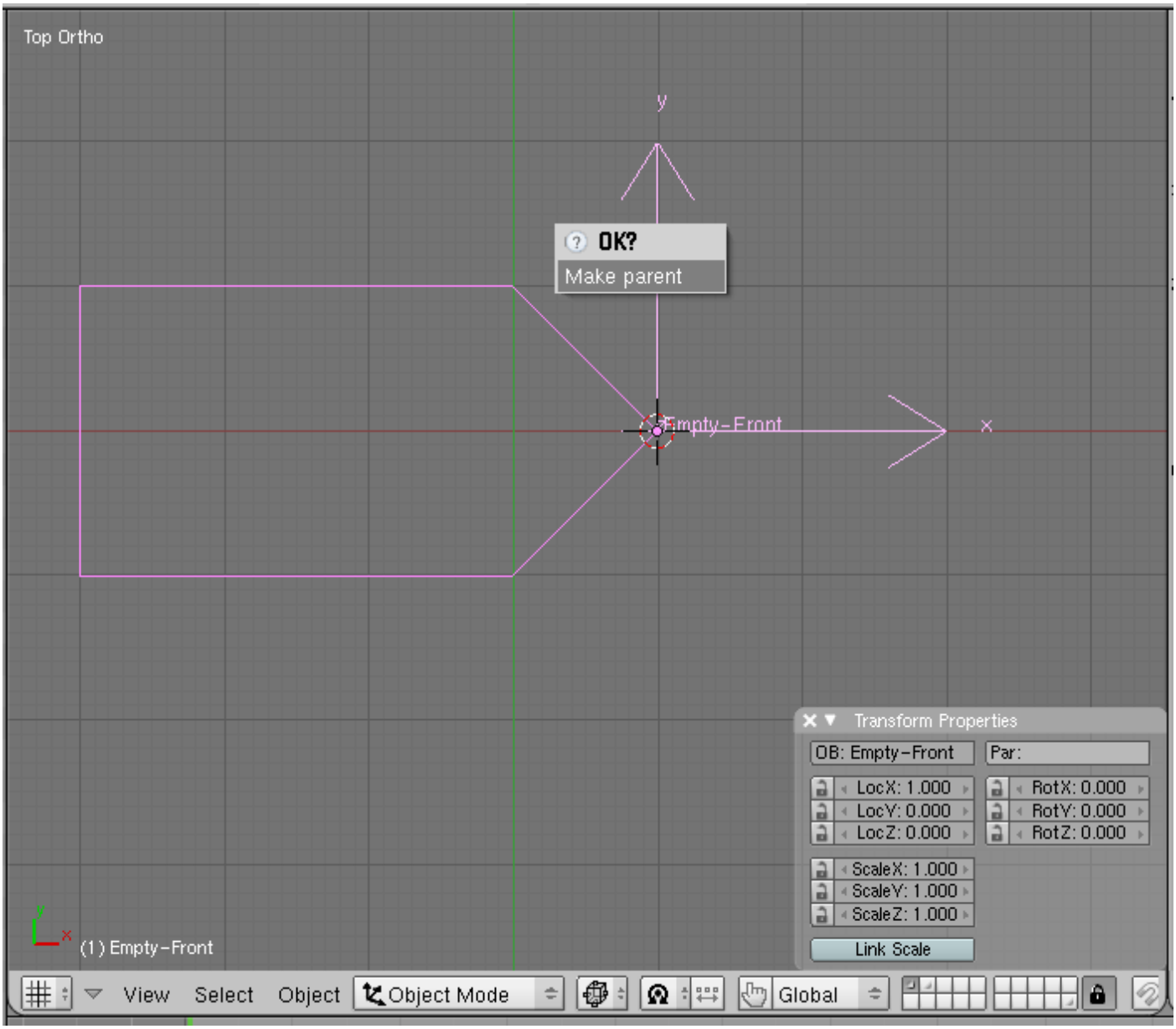
- In Object Mode, select the Chassis and bring the 3D Cursor on its center.



#014

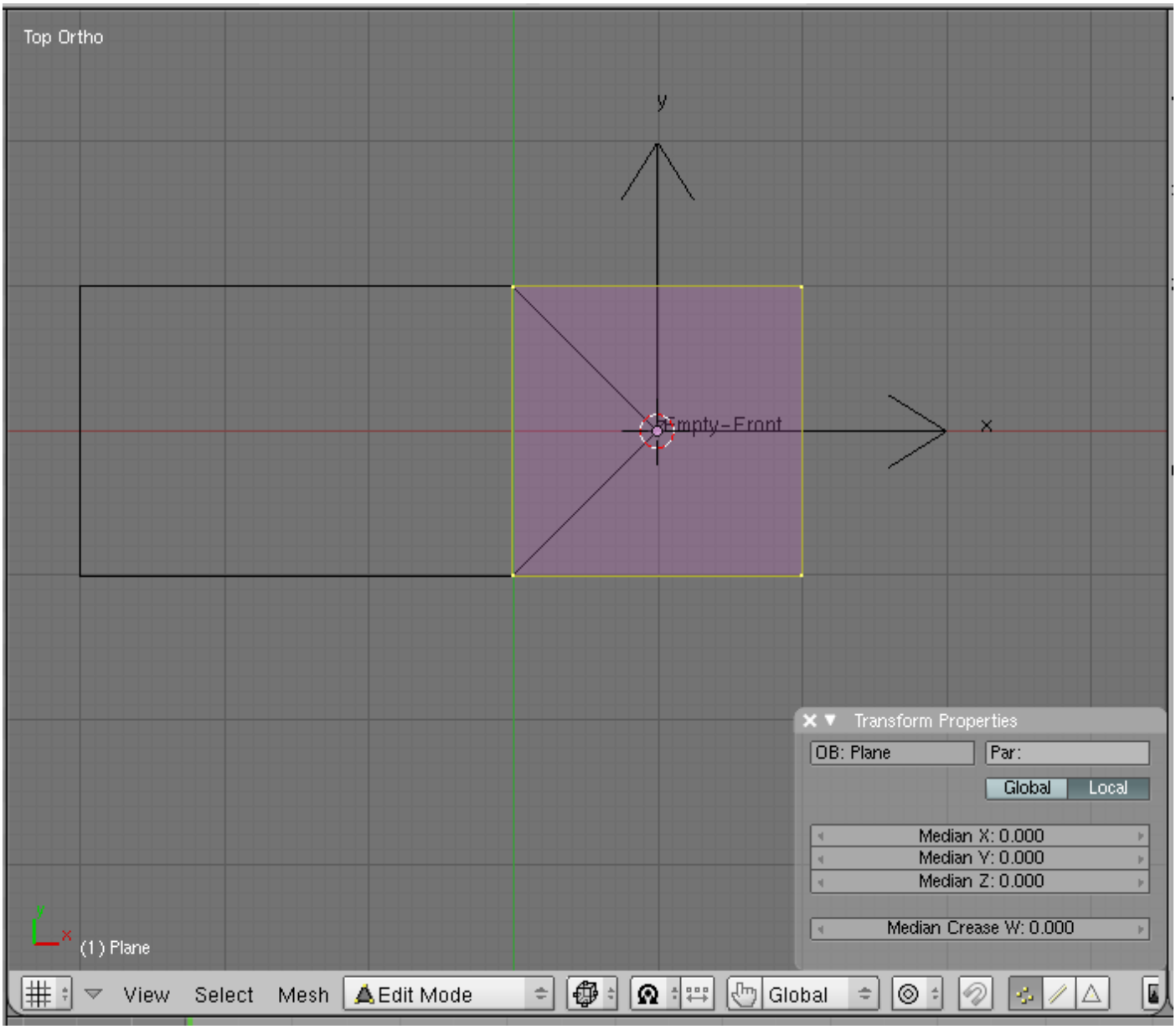
- Add an Empty. Name it Empty-Front. And give it a size of 2 Blender Units.

- Make the Empty-Front the parent of the Chassis.



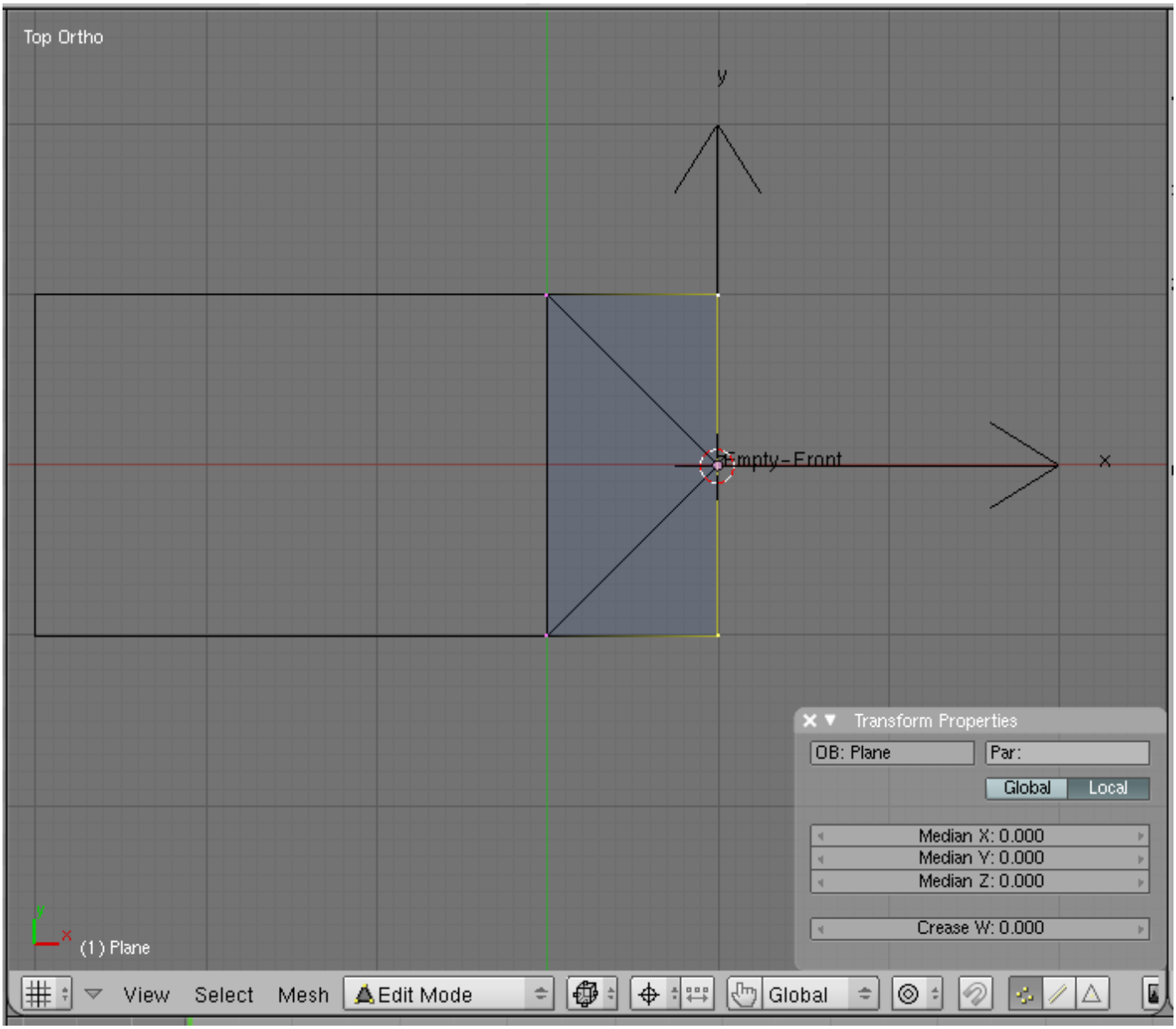
#015

- In top view, add a plane centered on the center of the Chassis object.



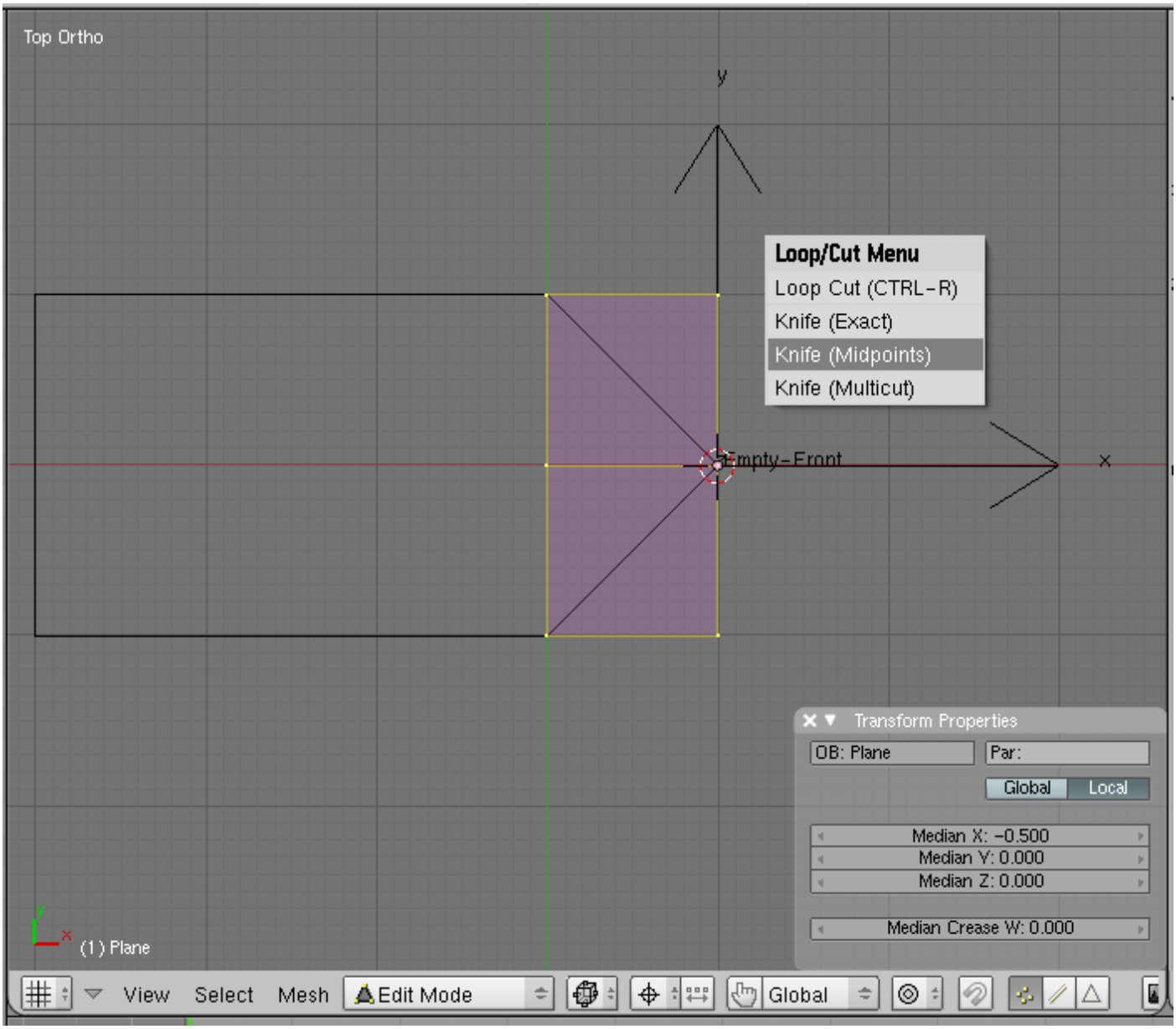
#016

- Scale the right side of the plane to bring the edge on the Empty-Front location.



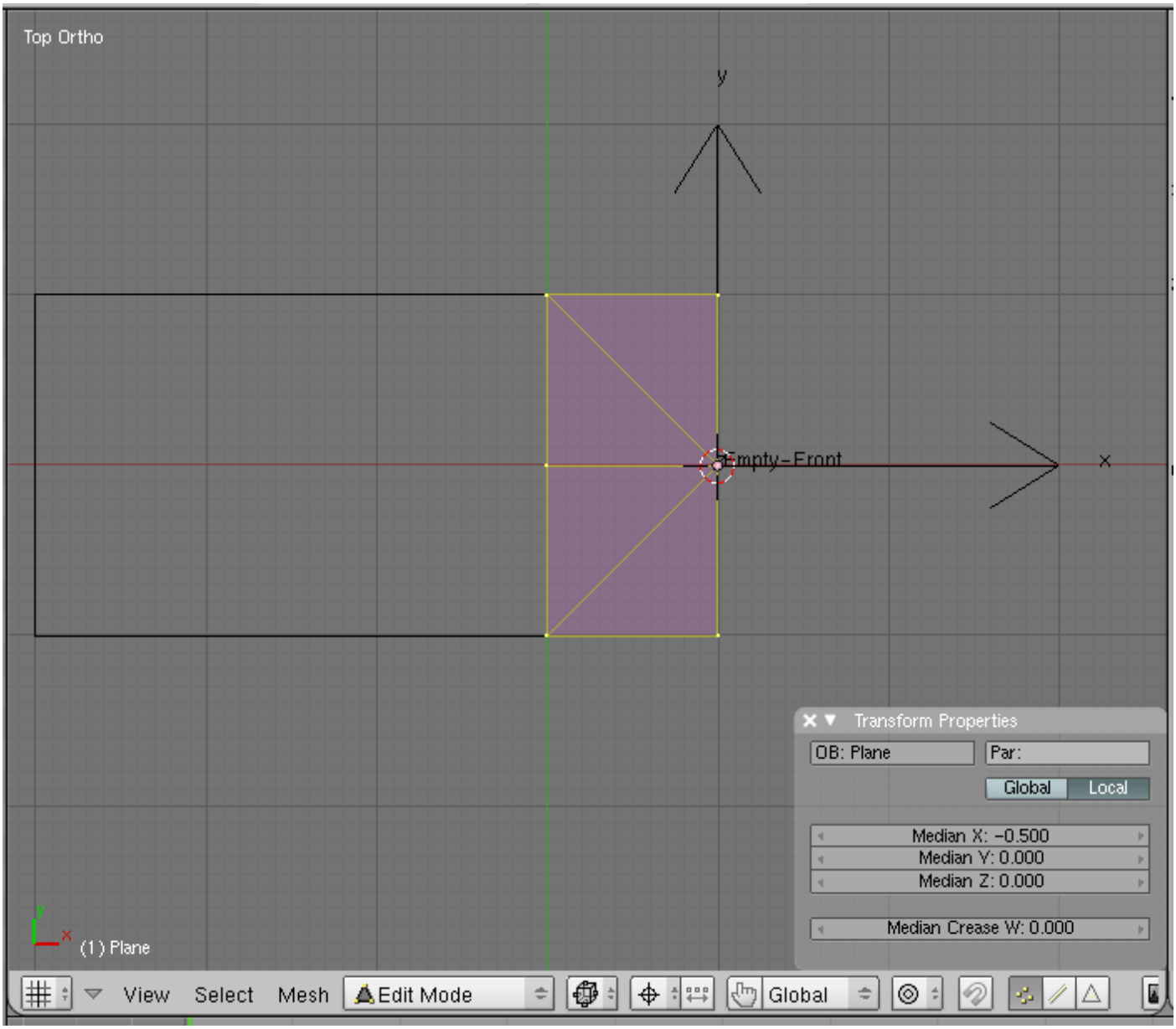
#017

- Typing K, cut the plane in the middle.



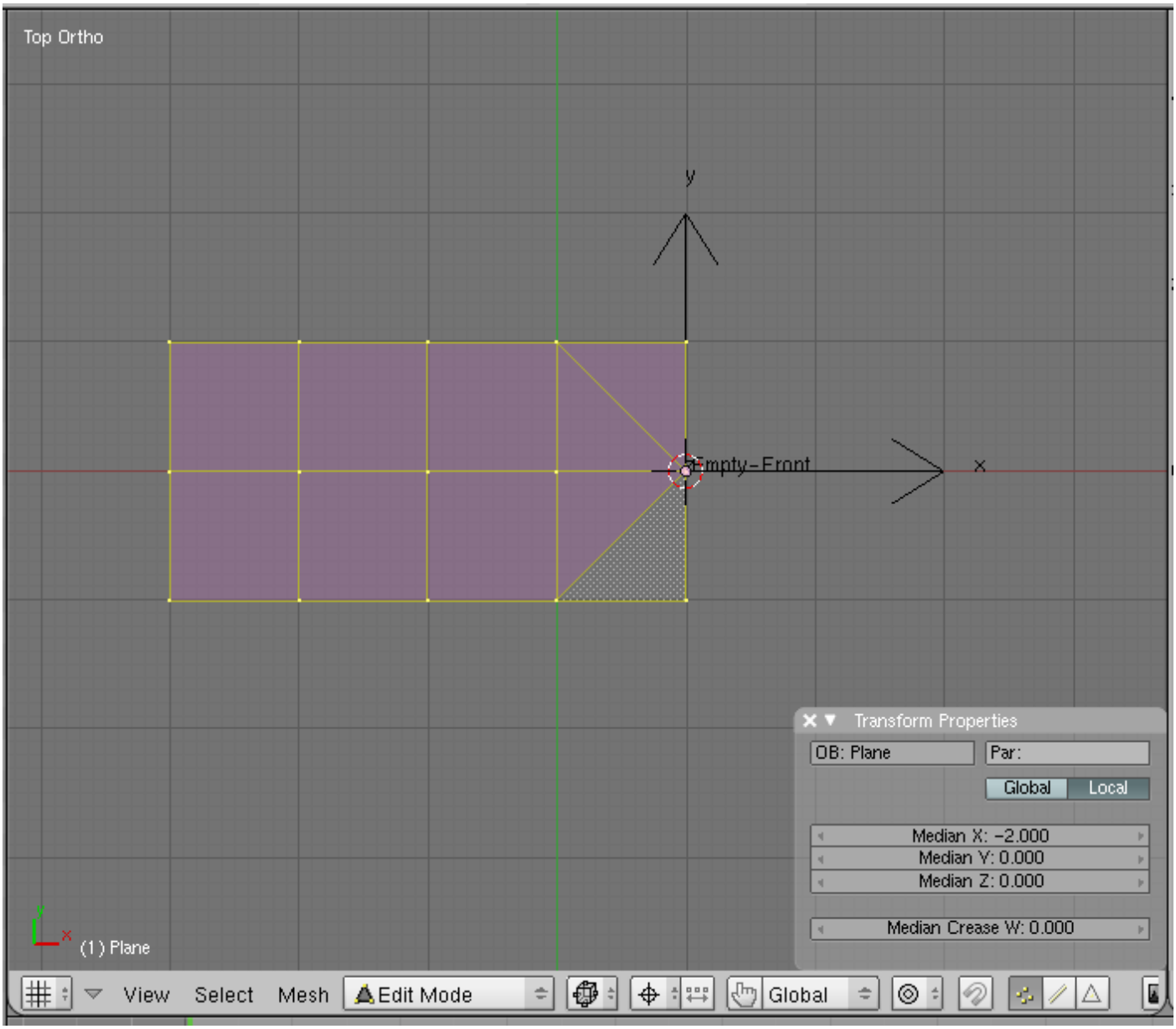
#018

- Typing Ctrl+T and Ctrl+Shift+F, divide the 2 faces into 4 triangles. The edges separating the triangles make an arrow head in the direction of the Empty.



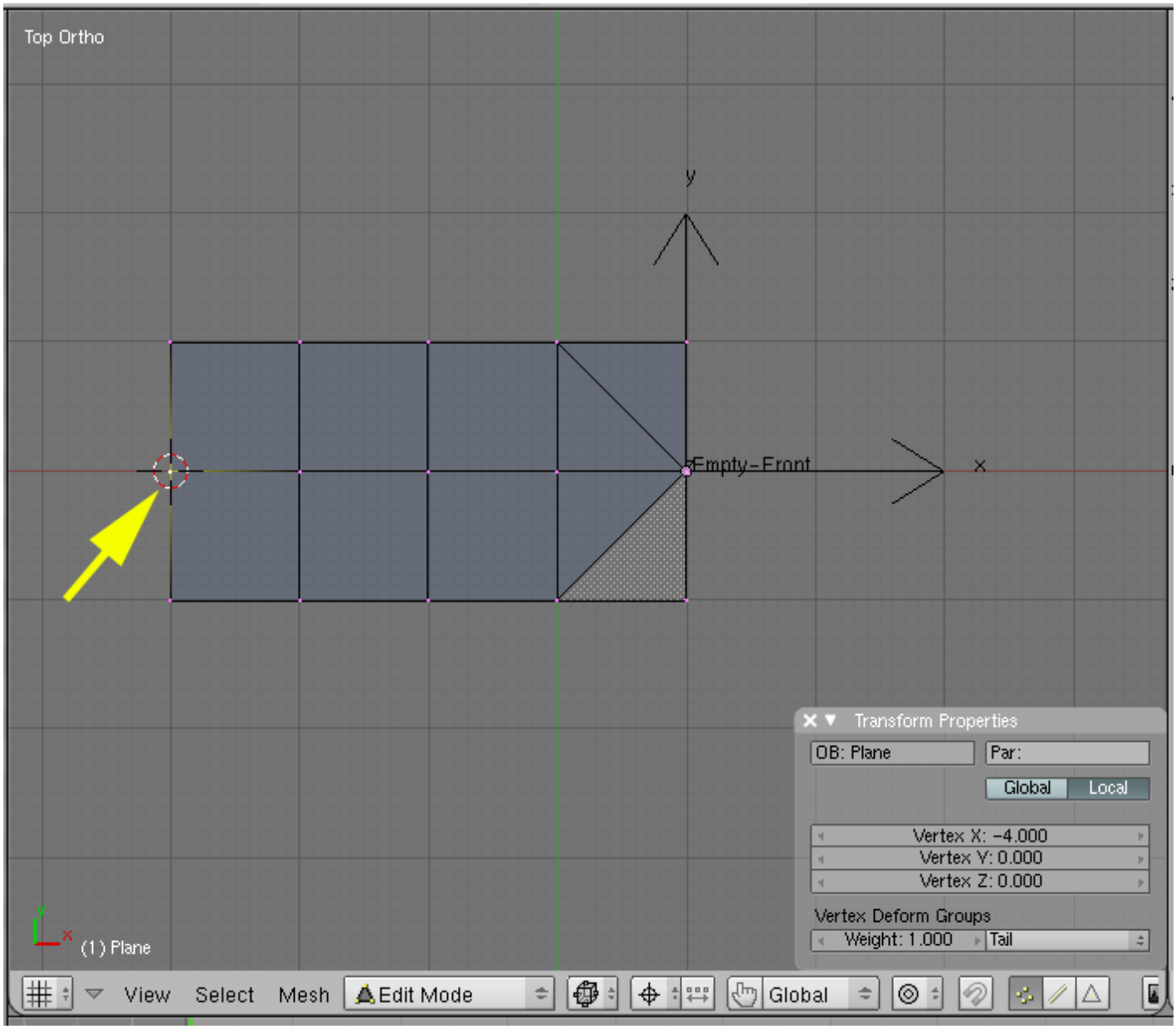
#019

- Extrude the left side of the plane 3 times of 1 Blender Unit each time.



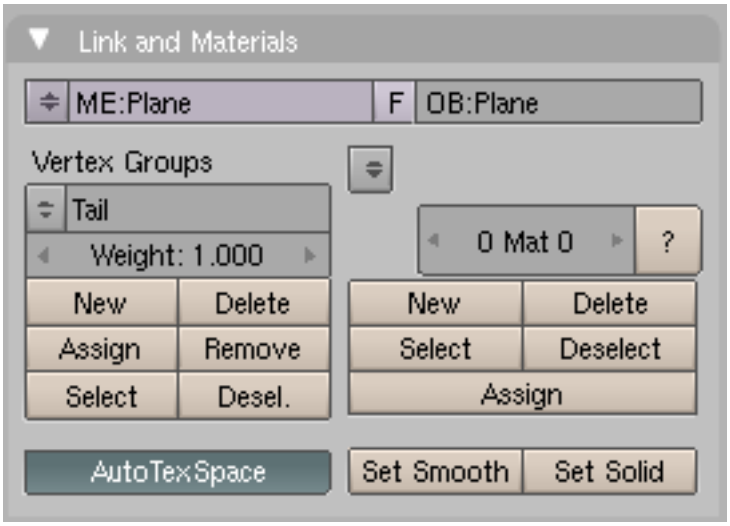
#020

- Select the center Vertex at the rear side of this plane (opposed to the Empty-Front).



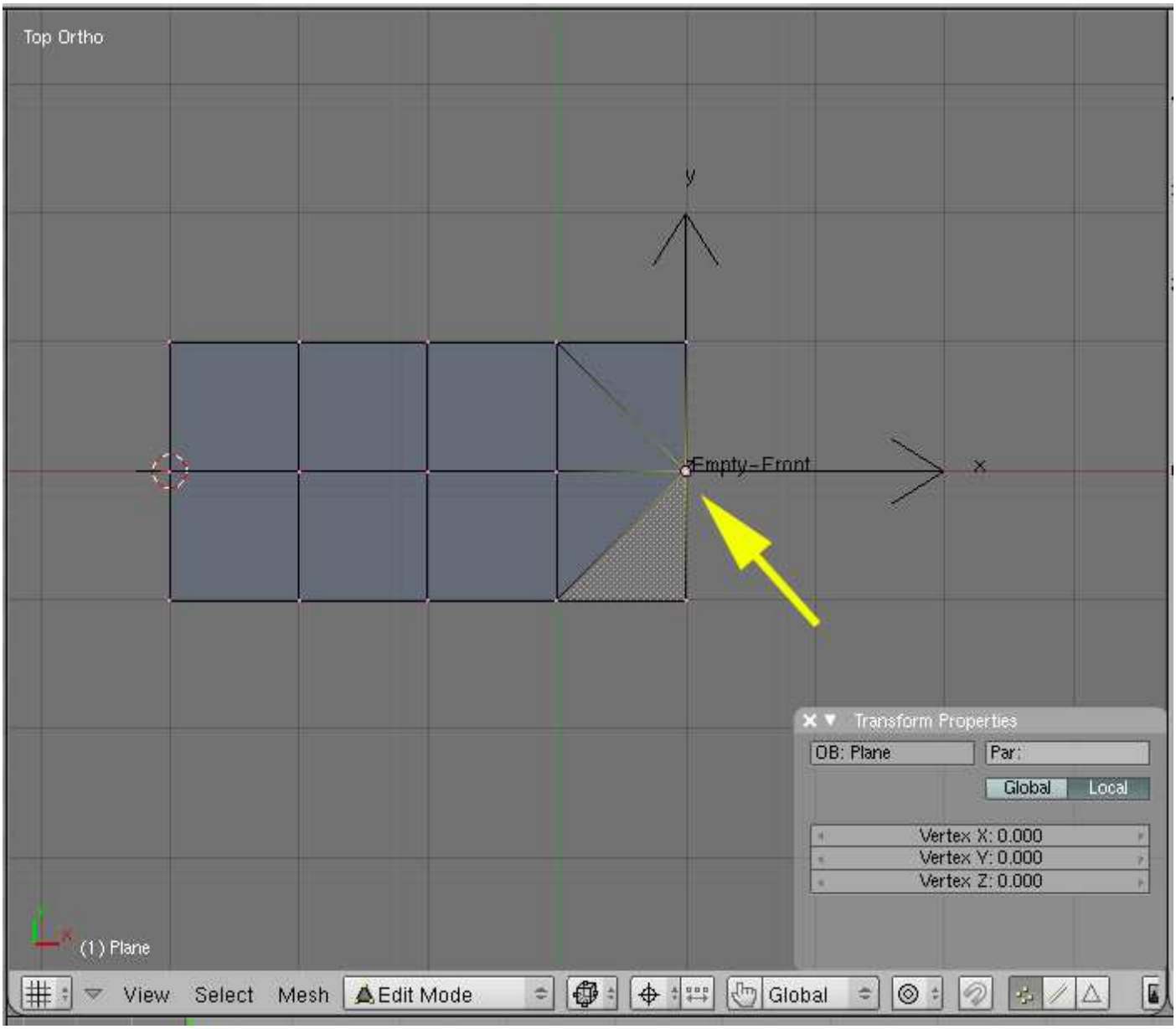
#021

- Create a vertex Group including only this Vertex, give it a Weight of 1.0, and name it "Tail".



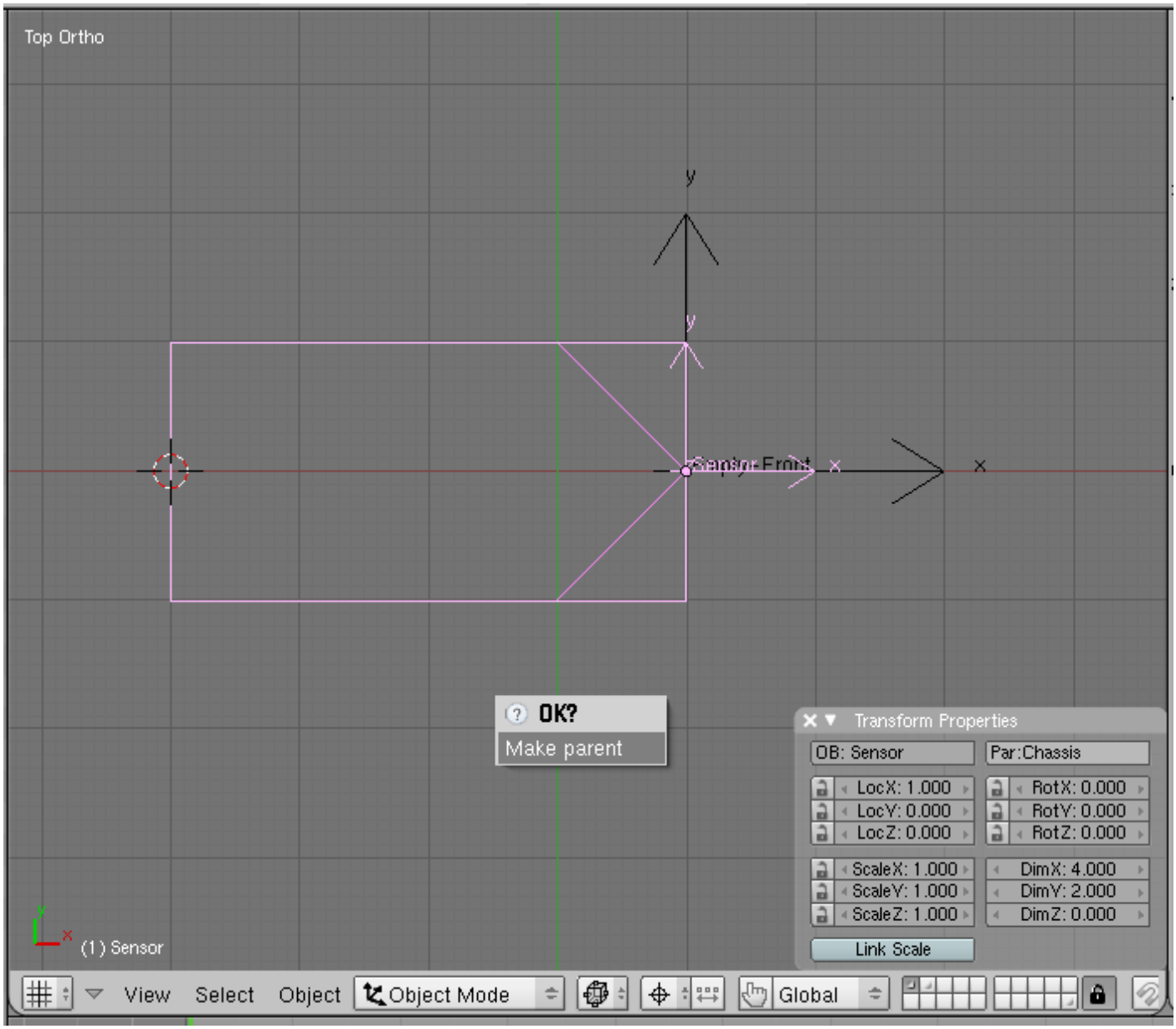
#021B

- Select the center Vertex at the front side of this plane (At the Empty-Front location).

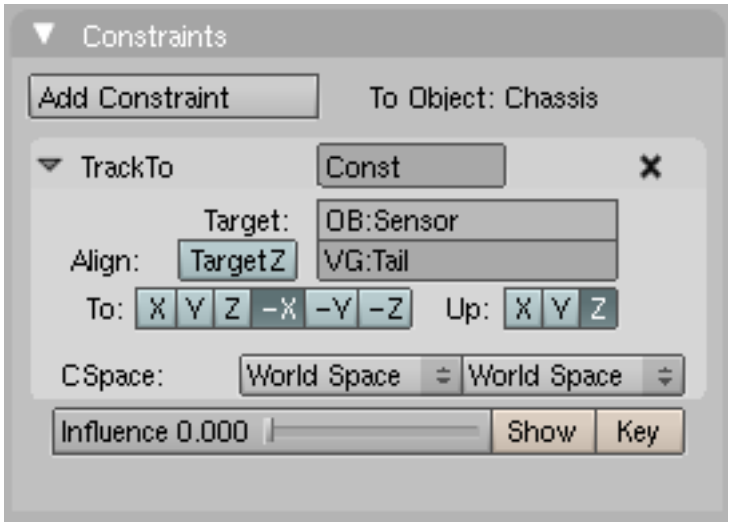


#022

- Create a vertex Group including only this Vertex, give it a Weight of 1.0, and name it "Nose".
- In Object mode, make the Chassis the Parent of this plane.

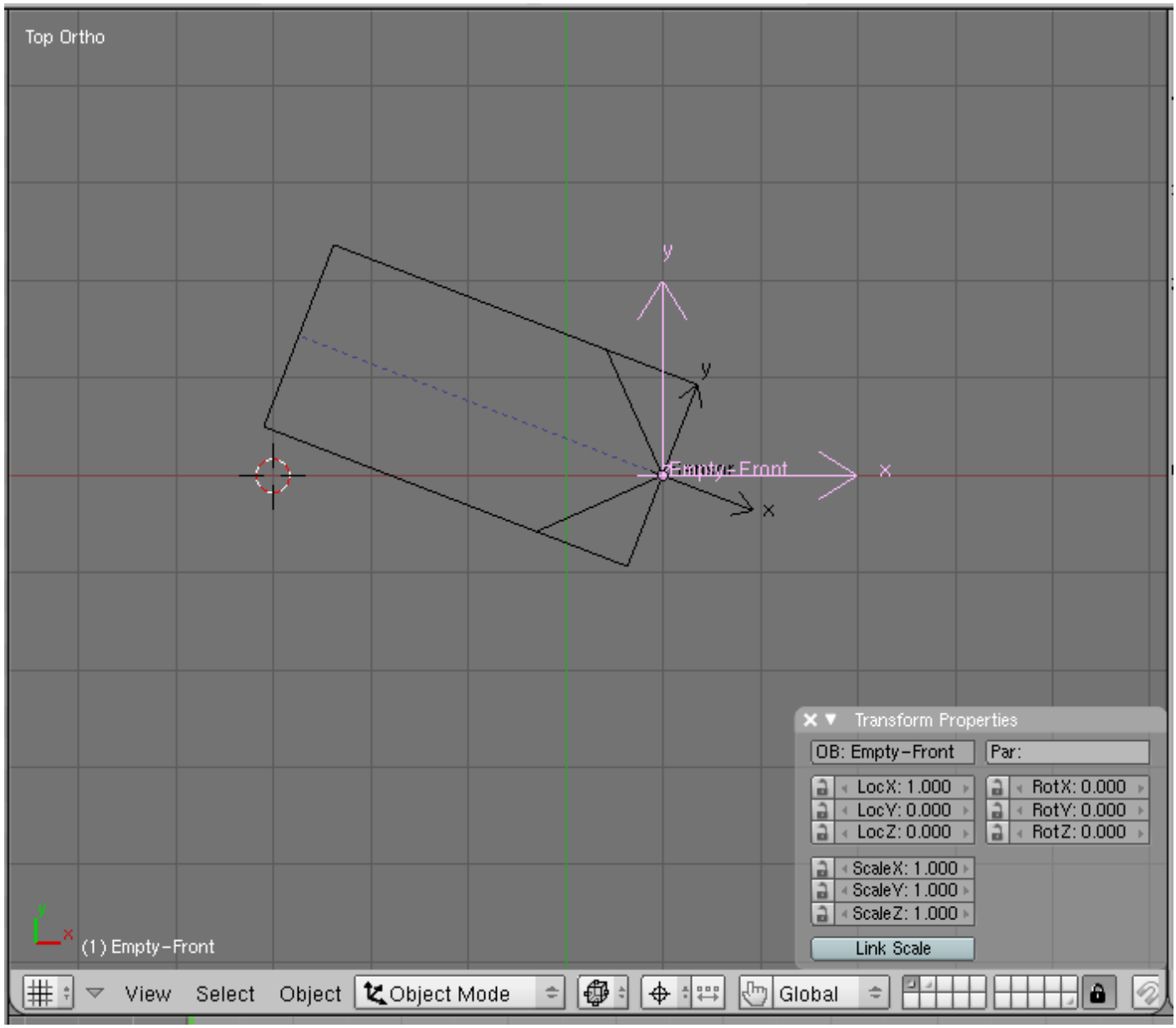


#023
- Name the Plane "Sensor". Select the Chassis, and add a Track To constraint. In the file Target, type Sensor. in the field VG (Vertex Group), type Tail. To = -X, Up=Z. Leave World Space in the two CSpace choices.



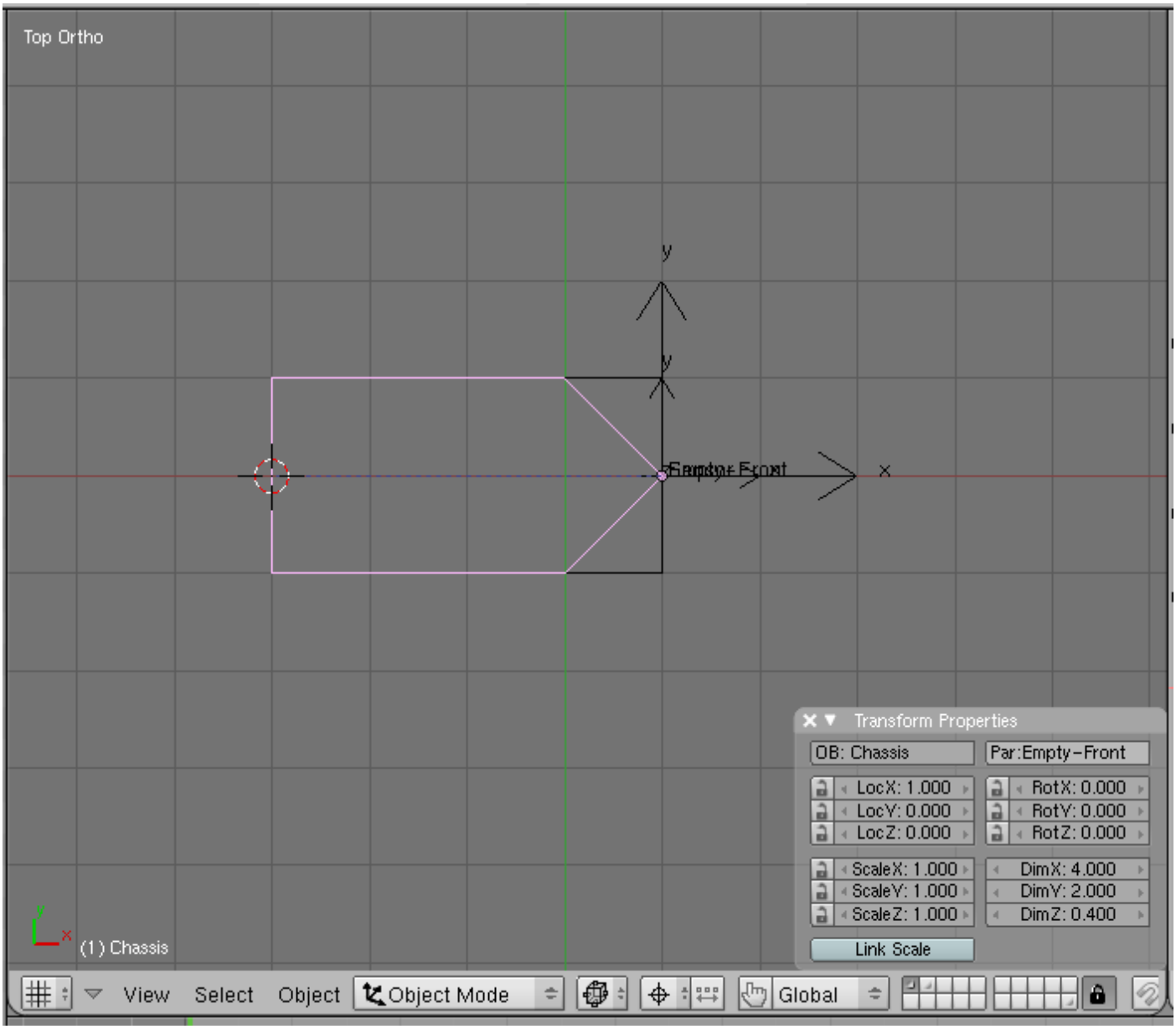
#023B

- Select the Empty-front and move it around. You will notice that the Chassis and the Sensor plane are following gently the Empty-Front. You will also see that when dropping the Empty, the system doesn't come back to its initial location.



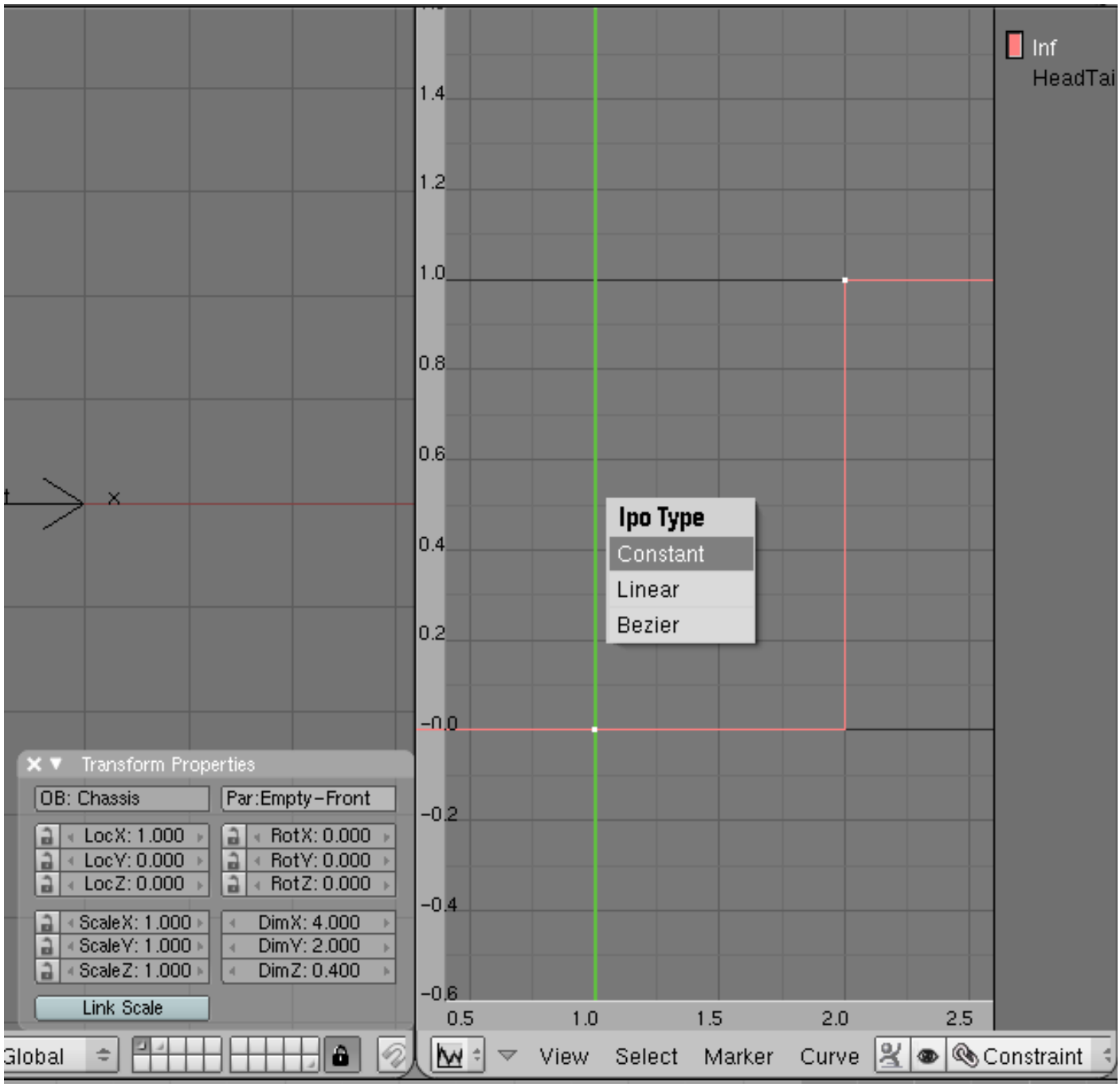
#024

- No problem : Just set the influence cursor of the chassis to 0.000.



#025

- Open the Ipo Constraint window.
- Select the Chassis, and at Frame 1 with the influence cursor of the constraint set to 0.000, Hit the Key button in the constraint panel. A constraint Inf Ipo is created, showing one keyframe at frame 1.
- Go to Frame 2 and set the influence value to 1.000. Then, Hit the Key button in the constraint panel. A second keyframe is added in the inf Ipo curve. Select the Ipo curve and by Hitting T key, choose Constant for the Ipo Type.

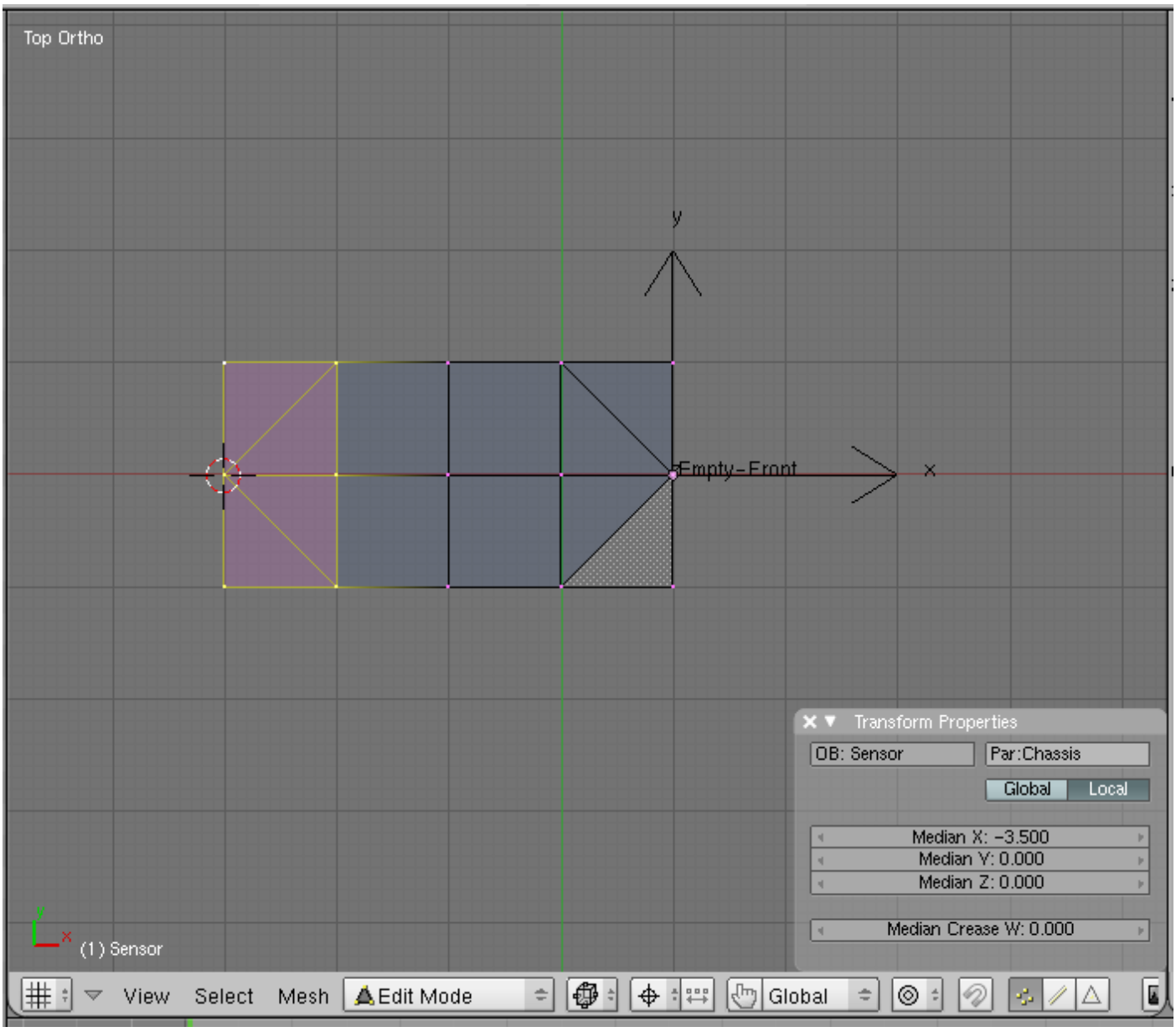


#025B

- Now, each time you will come back to Frame 1, the setup will be in its initial position.
- For testing purpose, go to frame 3. Select and move the Empty-Front and insert a Loc keyframe. Go some frames further, move the Empty-Front and insert a new Loc keyframe... etc. When playing the animation, you will see that the rear of the chassis nicely follows. The Empty can be moved by at least 3 methods : By direct keyframes like you just tested it, by parenting it to an other empty, by applying to it a Clamp to constraint using a Curve as trajectory guide. If you use a second empty as parent it can be keyframed or driven as well by a Clamp To constraint. You could aso use a path constraint, but it would not be a good choice, because keyframing on a path constraint is not easy.

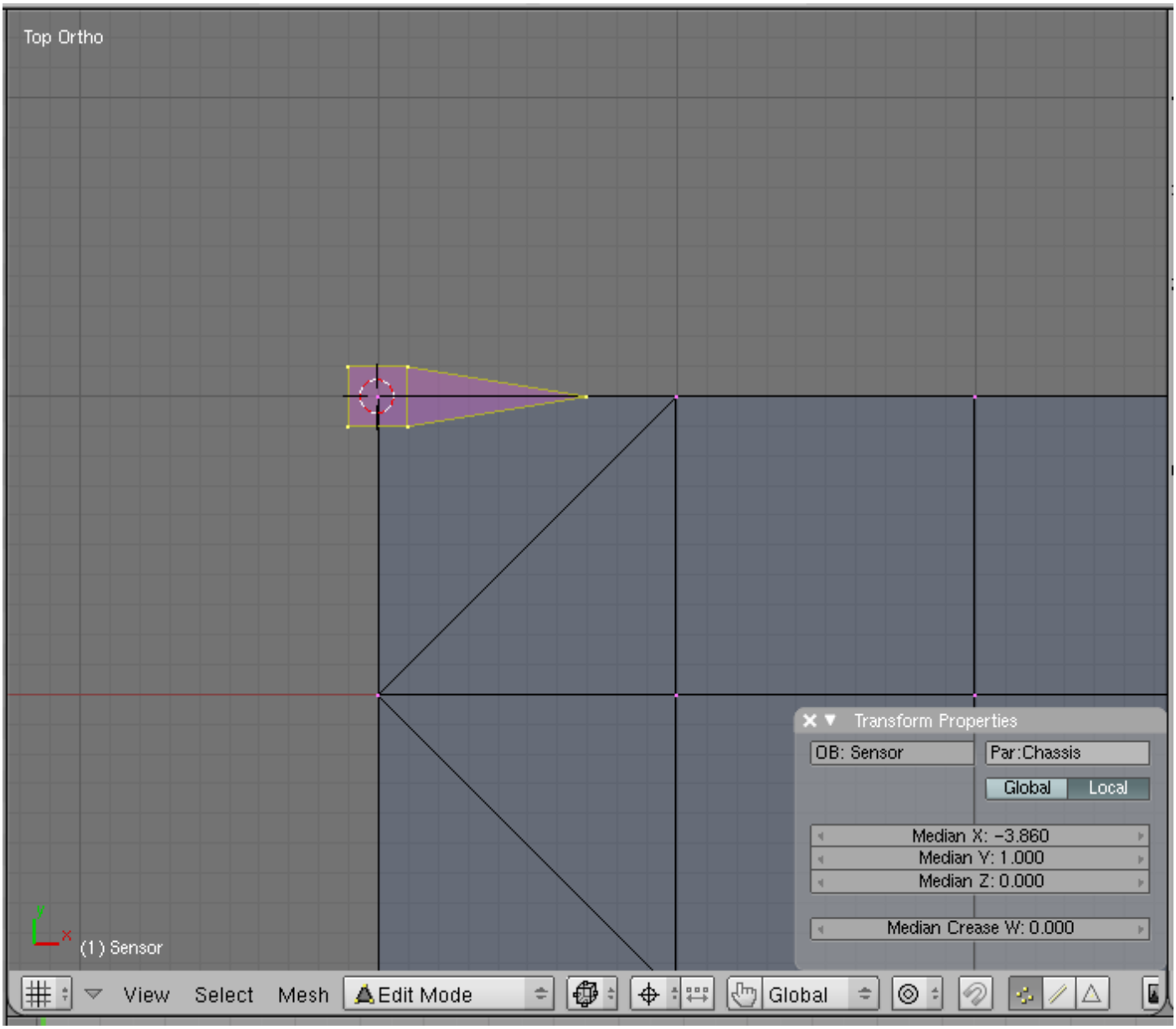
Step 2 : Adding the wheels supports.

- Select again the plane named "Sensor" and in top view, select the two rear faces. By typing Ctrl+T and Ctrl+Shift+F, divide the 2 faces into 4 triangles like you did previously for the front part of this plane. This time, the edges separating the triangles make an arrow head in the direction of the Tail Vertex.



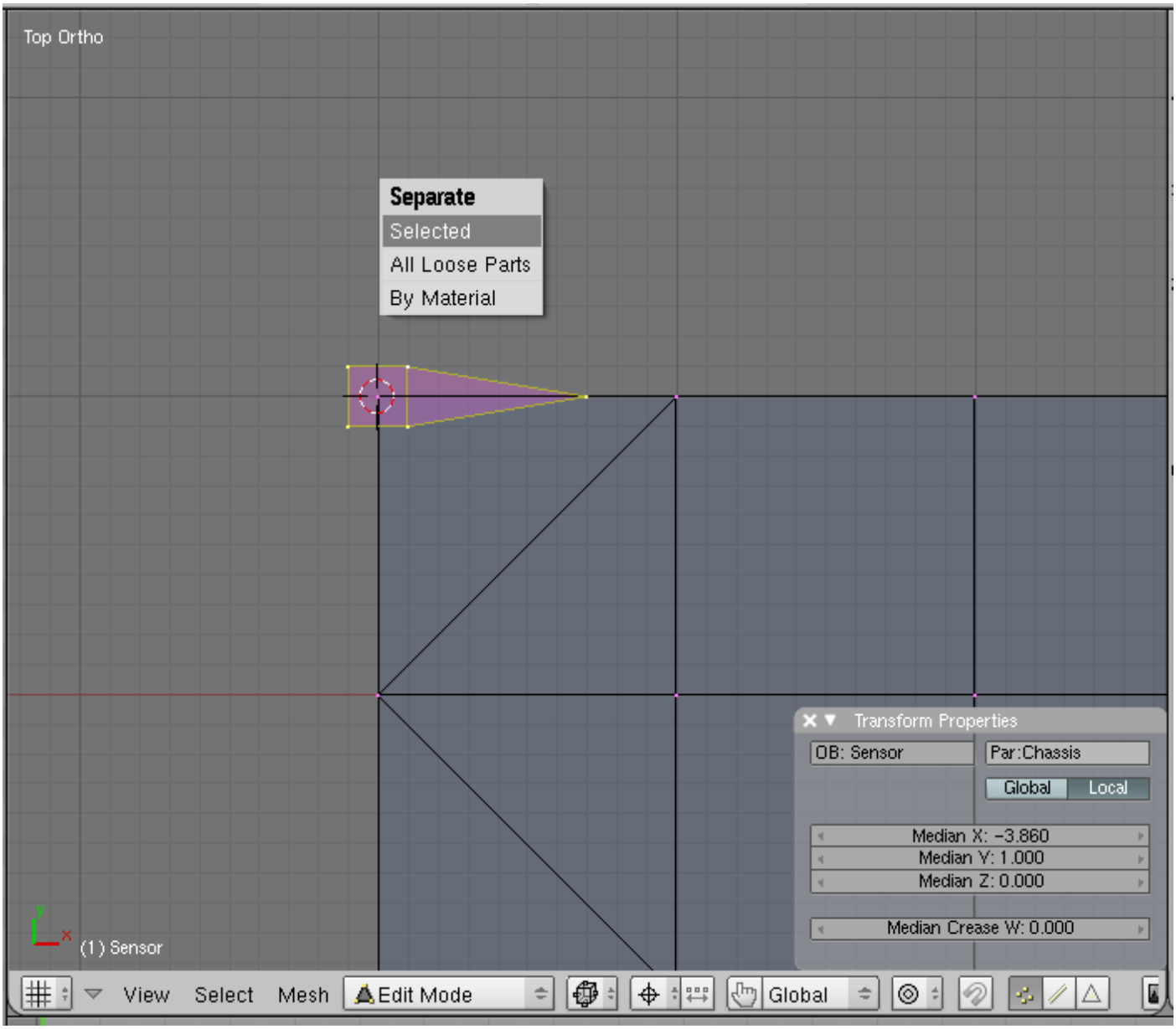
#026

- Select a Vertex in a corner of the plane, and bring the 3D Cursor on it. Then, add a Cube. Scale the cube by 0.1.
- Select the right part of the cube and extrude it on the right of 0.6 Blender Unit (for example. This value has few importance).
- Scale the last extruded Vertex to 0 in Y direction, and remove doubles.



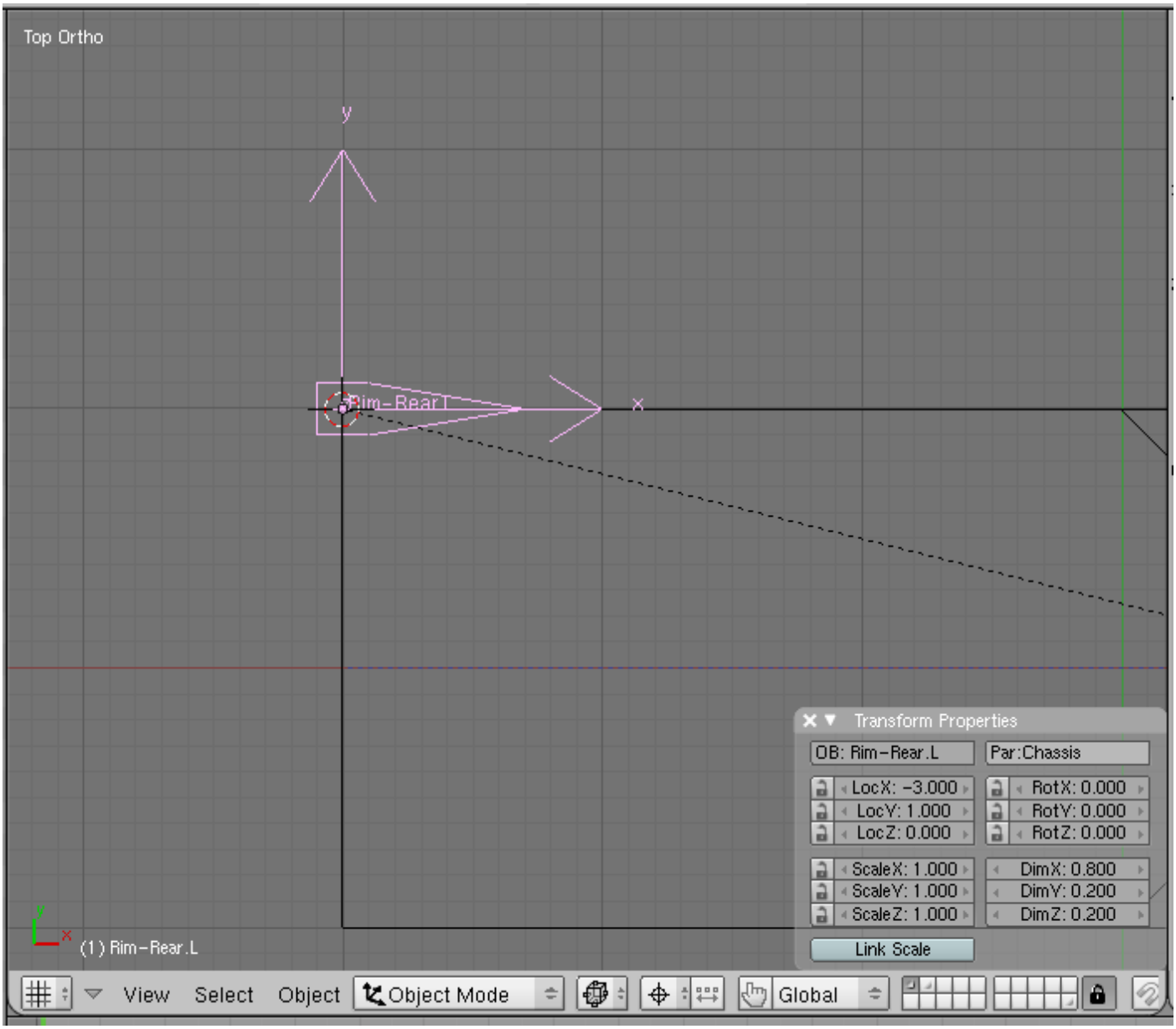
#027

- Select all the vertices of the group made from the added cube, and separate it from the Sensor Plane by hitting P key.



#028

- Put the center of the object at the center of the cube you added to create the object (at the location of the Vertex that you had selected before adding the cube).

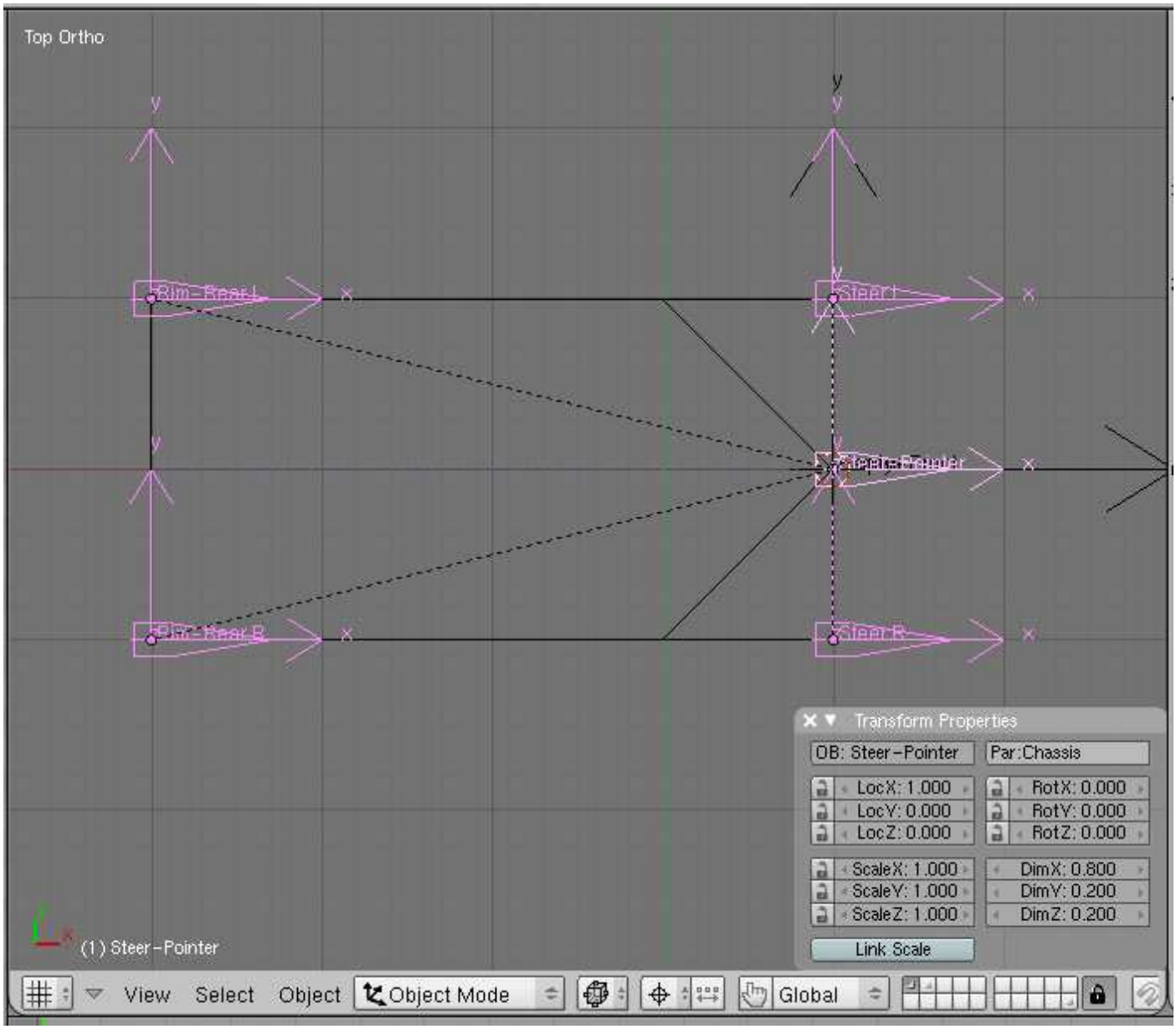


#029

- Duplicate this object an put a copy at the location of each corner of the Sensor Plane. This can easily be done by selecting one by one the vertices in the corners of the Sensor plane.

- Name the two rear objects Rim-Rear.L and Rim-Rear.R. These objects will support the rear wheels.

- Name the two front objects Steer.L and Steer.R. These objects will support the front wheels.



#030

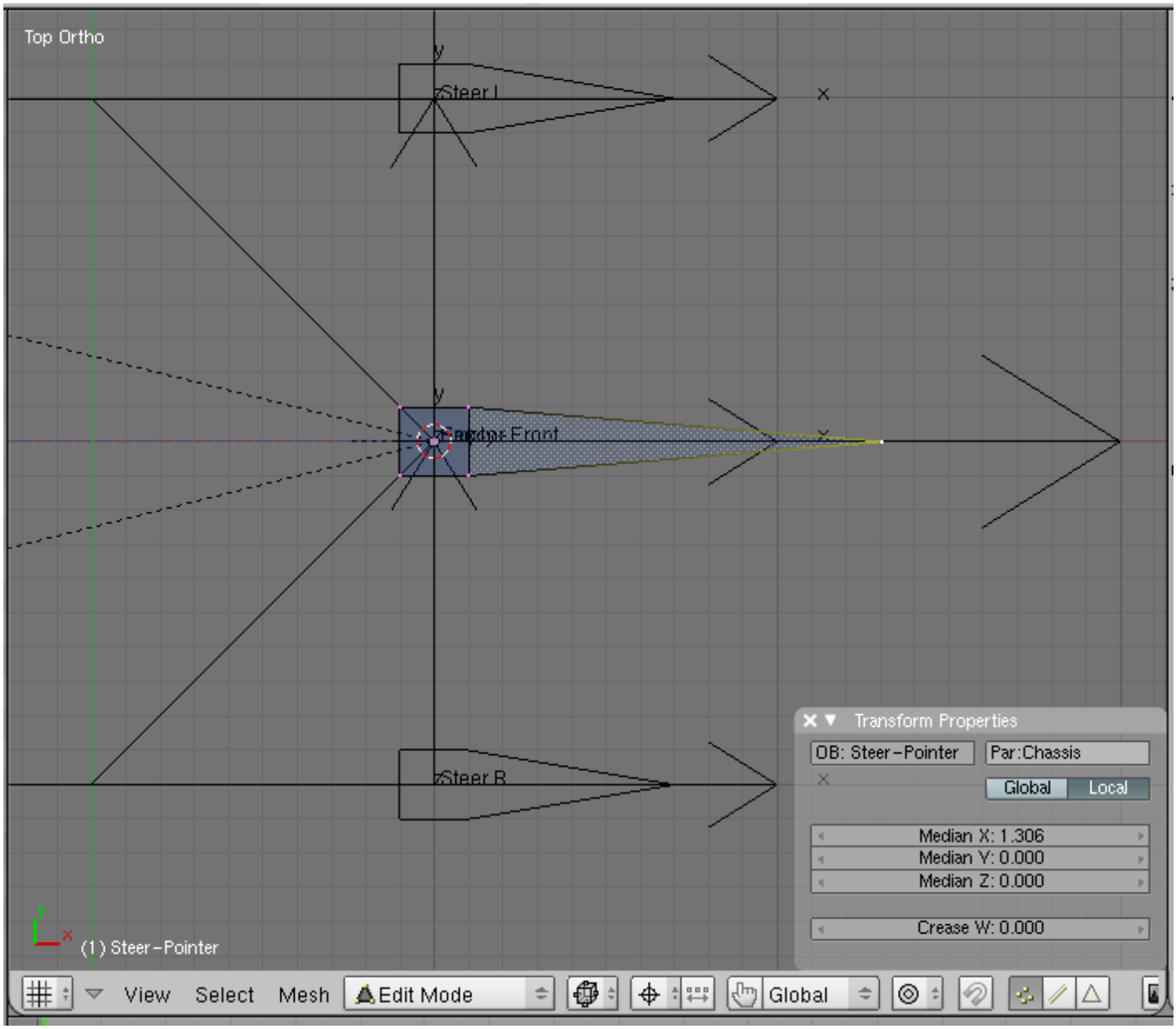
Step 3 : The steering Pointer.

- Now, select the Empty-Front and add at its location one more copy of the created object. For example a copy of Steer.R.

- Name this copy Steer-Pointer.

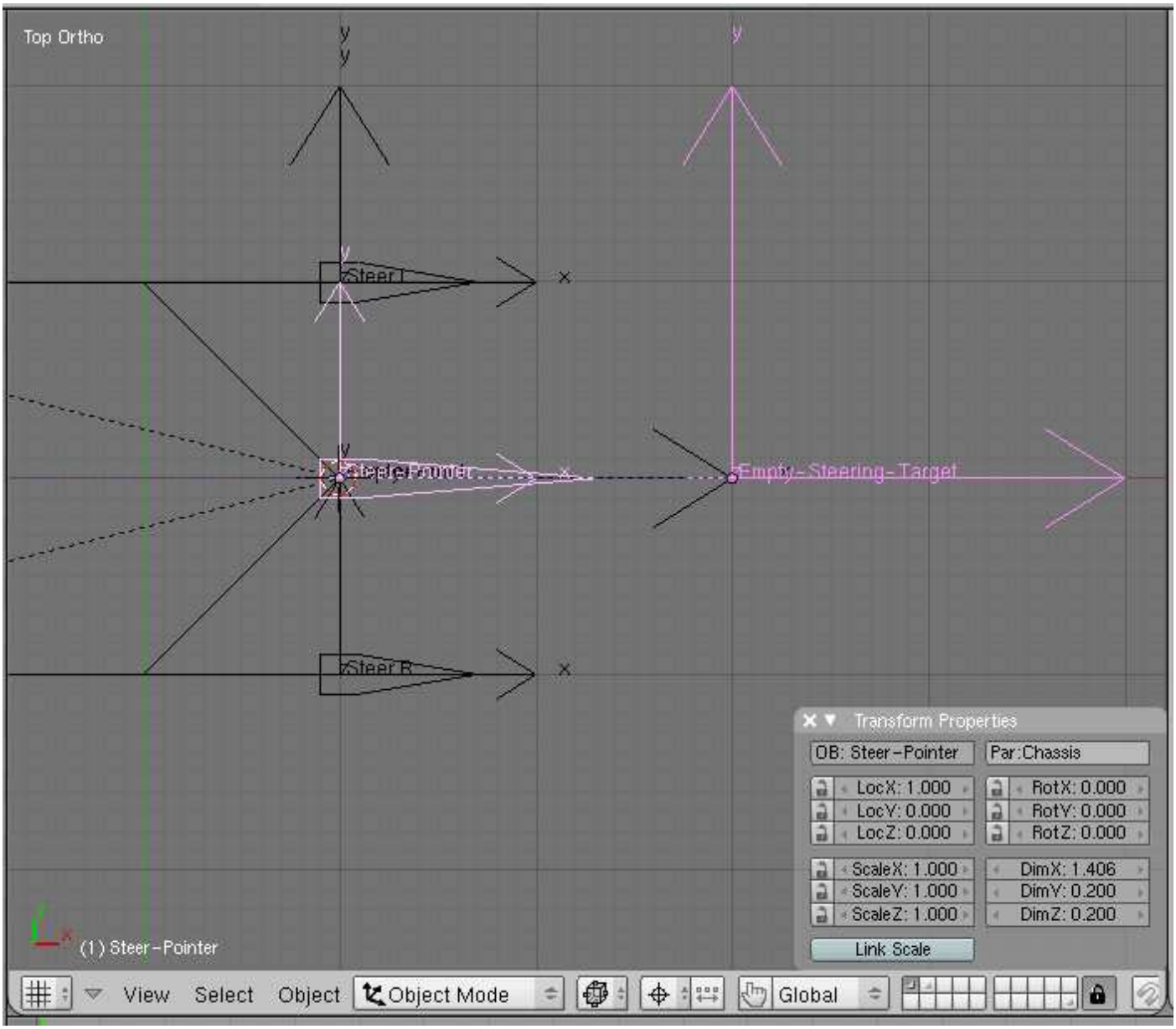
As all these objects have been separated from the Sensor plane which is parented to the Chassis, they should be all parented to the Chassis as well. Just check it by moving around the Empty-Front of the chassis.

- As the object Steer-Pointer will be given a special function, we want to make it more visible, so select it and in Edit Mode, just increase the length of its arrow.



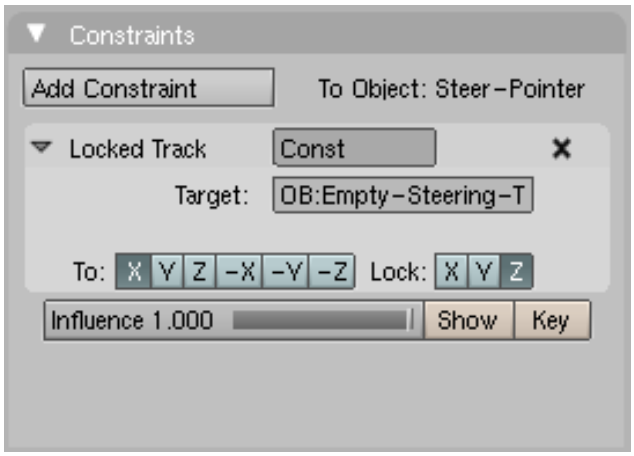
#031

- Duplicate Empty-Front and rename it Empty-Steering-Target.



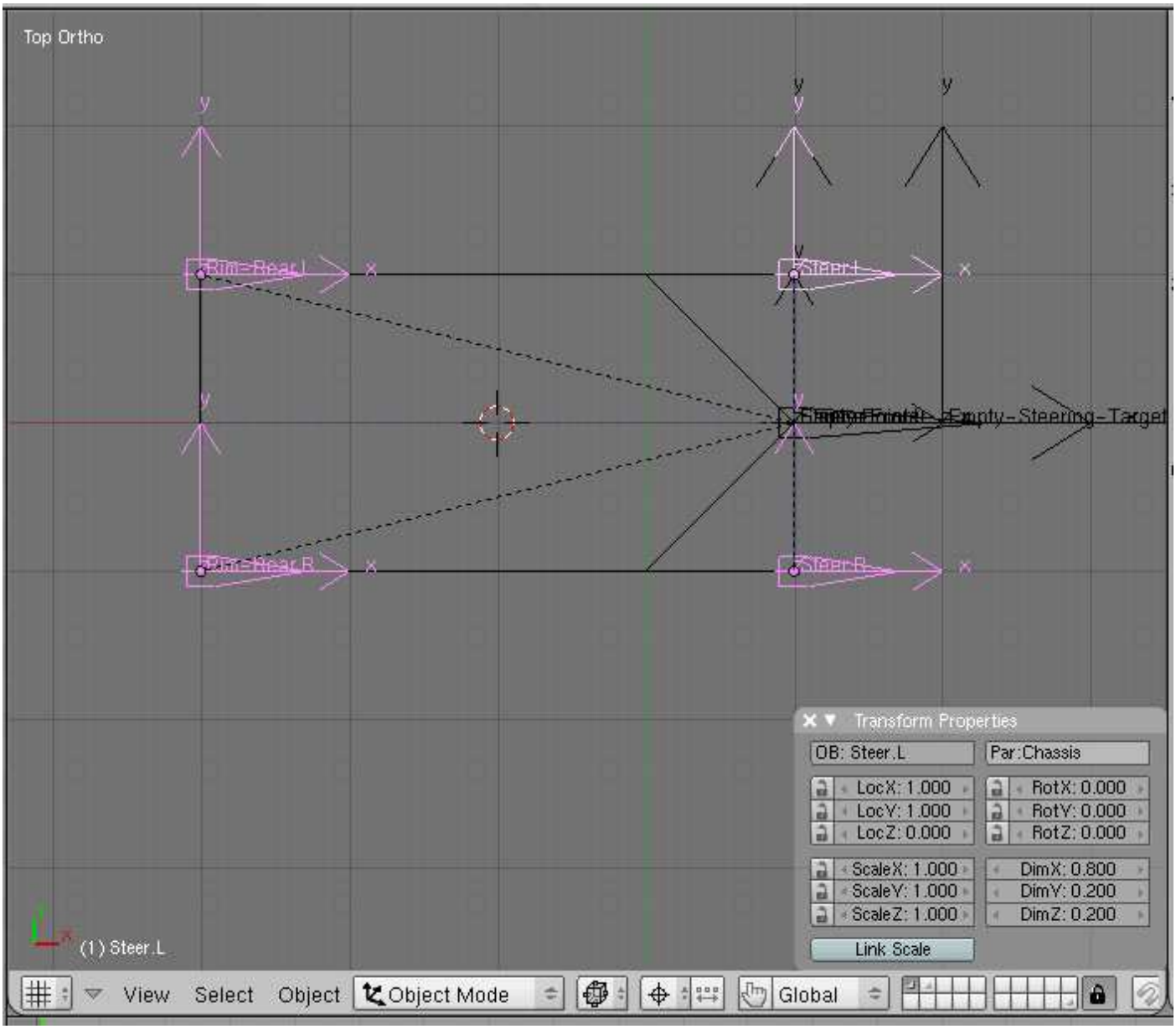
#032

- Select the object named "Steer-Pointer" and add it a Locked Track constraint. The To direction of the constraint is X, and the rotation is locked on its Z axis. The Steer-Pointer will track the Empty-Steering-Target.



#032B

- Now, select the Steer.L object and add it a Copy Rotation constraint. It will copy the Z rotation of Steer-Pointer.



#034 (#033 is not used)

- Select the Steer.R object the same way and add it a Copy Rotation constraint. It will also copy the Z rotation of Steer-Pointer.

- Add a Limit Rotation constraint to the two objects Steer.L and Steer.R, to limit their rotation in a range of -45° to +45° arount their local Z axis.

Doing that you will avoid exagerated torsions of the wheels in your animations. Later, these objects will support some more constraints. Some constraints will be necessary to Limit their distance from the chassis, to preserve the geometry of the car and some others to allow secondary animation.

- Now, if you move the Empty-Steering-Target, you will see that the front wheels supports rotate and follow the direction of the empty.

- But if you move the chassis or the Empty-front, you will notice that the Empty-Steering-Target doesn't follow the rest of the setup.

- There are two methods to control the behaviour of Empty-Steering-Target. One method is to apply to it Clamp To constraint and to drive it on a curve. This method can be used when the Empty-Front is itself driven on a curve by a Clamp To constraint. It is easy to move the empties on the curve to adjust the distance between them. When you don't use a curve (when you prefer to animate Empty-Front by keyframing), there is an other possibility : parent the Empty-Steering-Target to the chassis, and apply to it a Time Offset (For example -1.5) in the Anim Settings panel. As the distance between the two empties will vary according to the speed of the vehicle, you may have to modify this value.

- Now, if you move the Empty-Front and insert some keyframes on the time line, you will notice that the whole setup will follow the Empty-Front, and that the front wheels are rotating on their Z axis, following the direction of the car. Don't worry about the wheels themselves. We are going to add them, but we will not have to do much things for their rotation, as it will be done by a nice script written by Jérôme MAHIEUX, better known as "littleneo", the author of the amazing Python script "Blended Cities", allowing to create huge procedural cities.

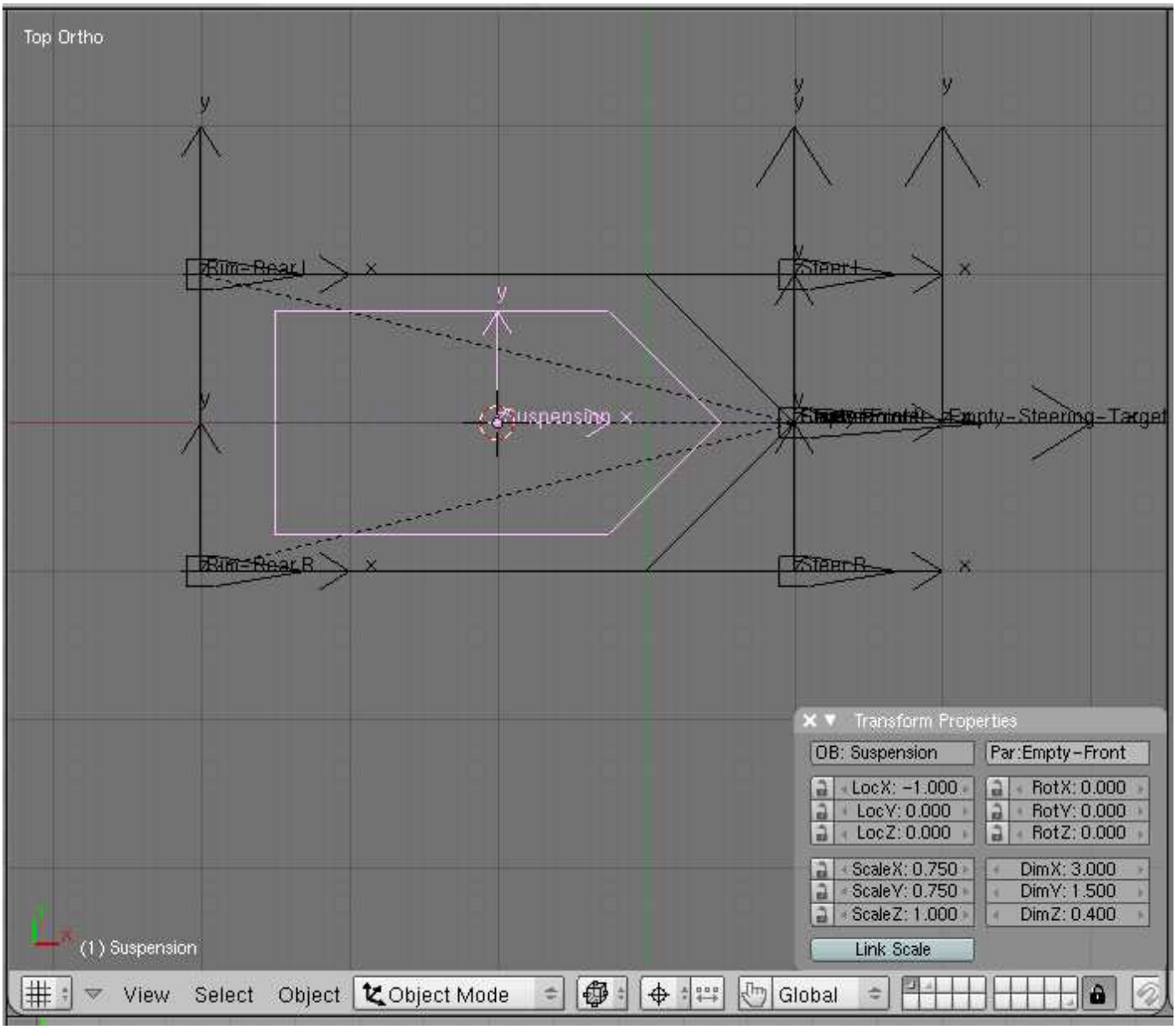
Step 4 : Adding the suspension.

The suspension Object is a "floating object". Its is located at the average location of the four corners of the "Sensor" mesh, and is Copying the Global (World) rotation of the chassis.

- The suspension will be centered, so in a first time select the four wheels supposts, and bring the 3D Cursor on this selection, by Shift+S Cursor-> Selection.

- Select the Chassis, and duplicate it by Shift+D. Name this copy "Suspension", and move its center to the location of the 3D Cursor, using the Center Cursor button in the Mesh panel.

- Scale the suspension object down, except in the Z direction : S+0.75+Shift+Z.



#035

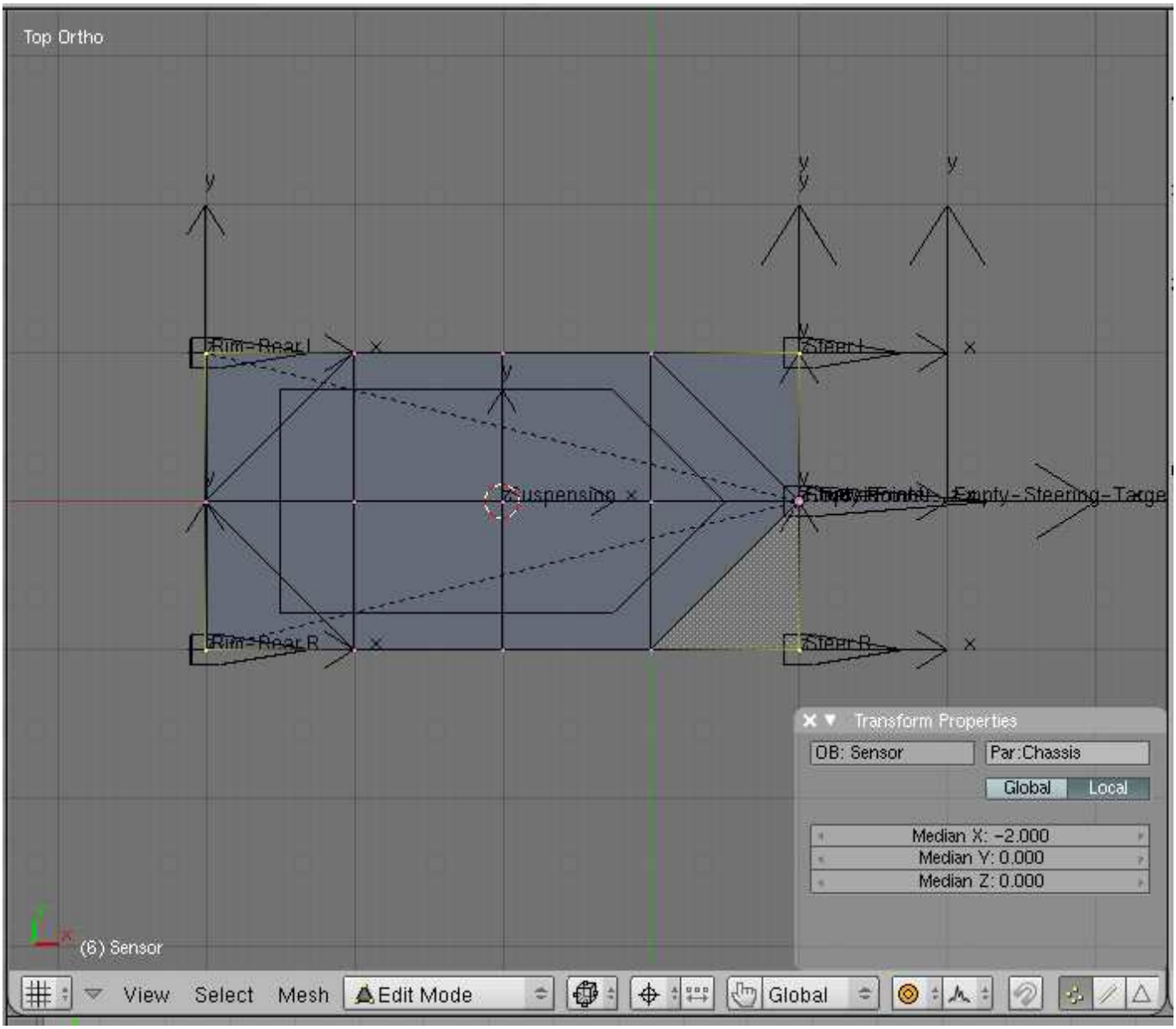
- Before going further with the Suspension, we will need some elements of the Sensor plane used for the Ground level detection.

Step 5 : Ground level detection.

Now, we are going to activate the Sensors.

CAUTION : Sensors can be hidden (or/and set to not renderable), but they must be put on visible layers. Otherwize they are not active and the animation is broken. Do not forget that Shrinkwrap Modifiers work that way !

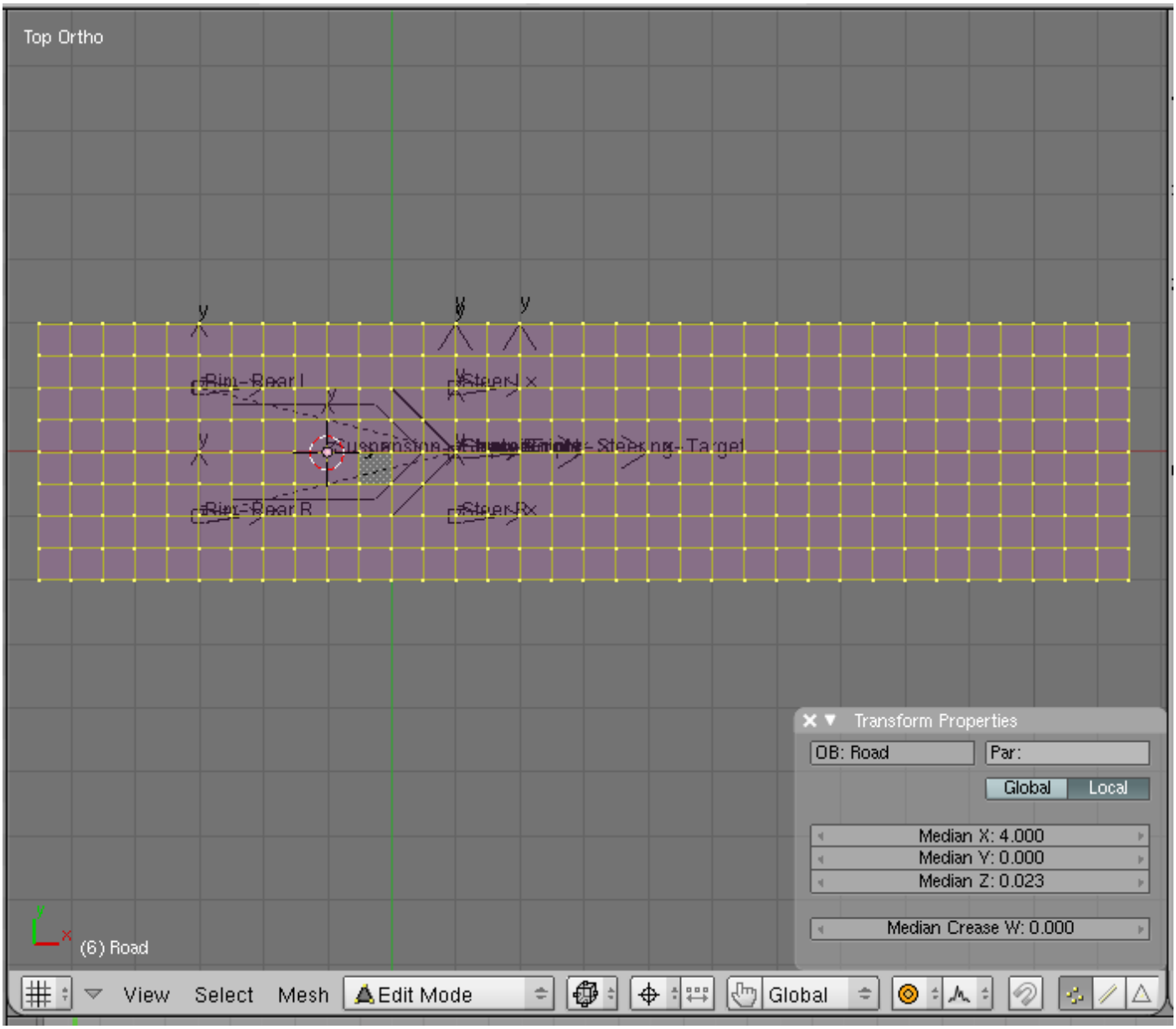
- Select the Sensor plane and In Edit Mode, create a Vertex Group named "Sensors", including the four vertices in the corners of the plane.



#036

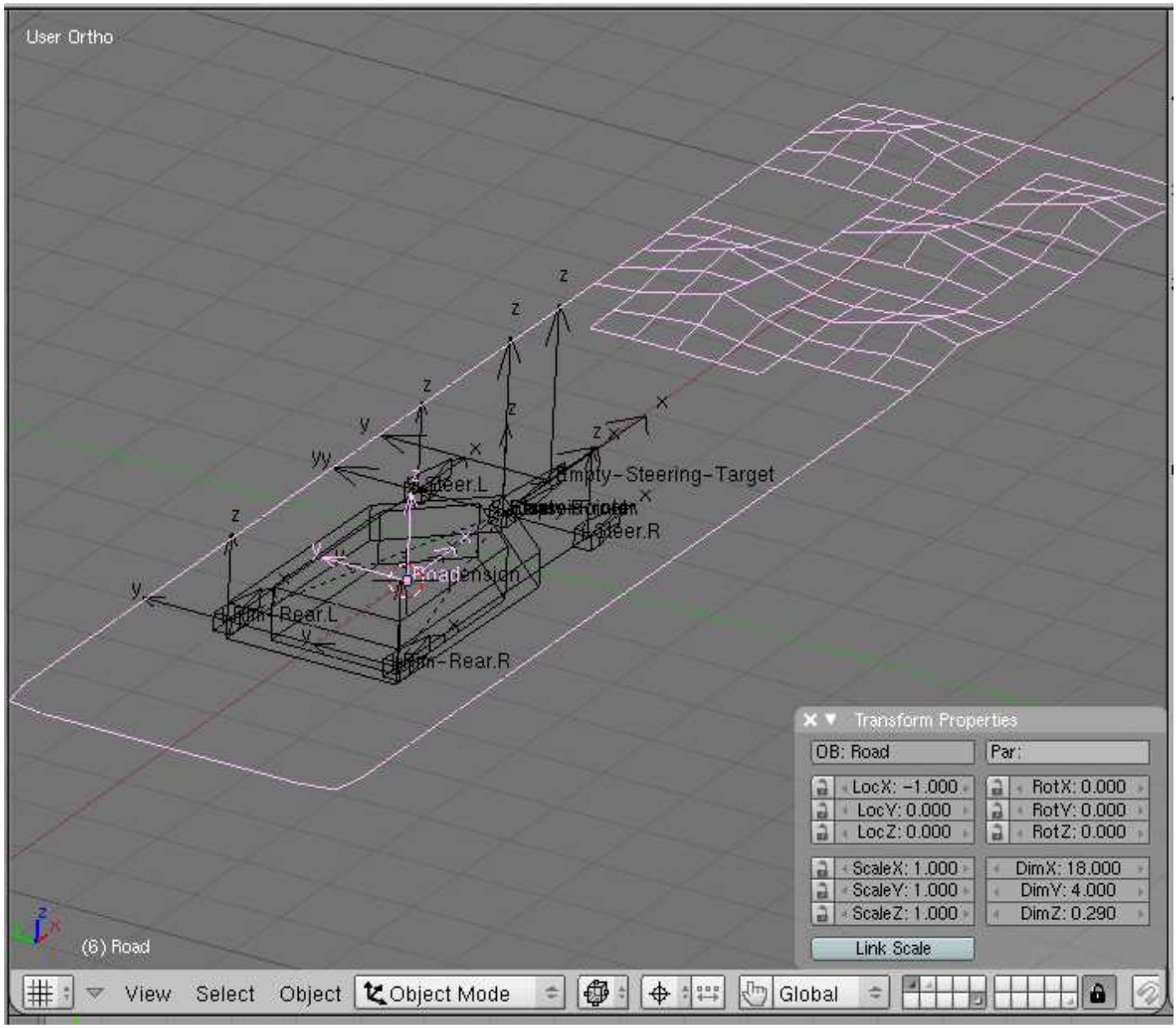
- Select also the four vertices one by one and create for each of them an other Vertex Group. We will name them F.L,F.R,R.L,R.R. These groups will be used soon by the Suspension, through Location constraints.

- Add a plane and name it Road. Subdivide it a couple of times and extrude it to get something like a short part of a road.



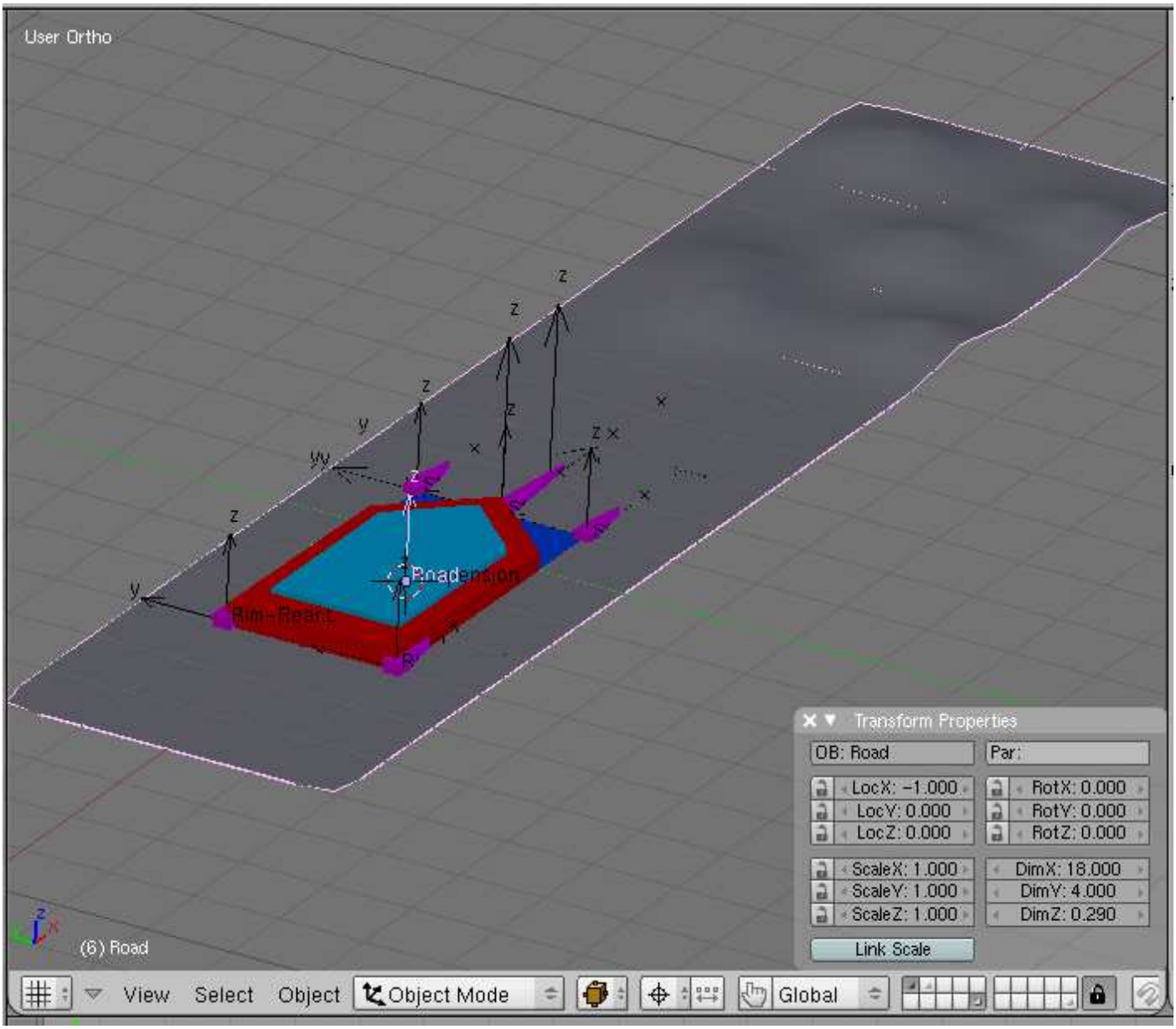
#037

- Using the proportional edition tools, add some bumps on the area of this "Road" before the car rig. By shifting this plane under the car rig, we will be able to test the altitude Sensors. It is also useful to add a subsurf level of at least 1.



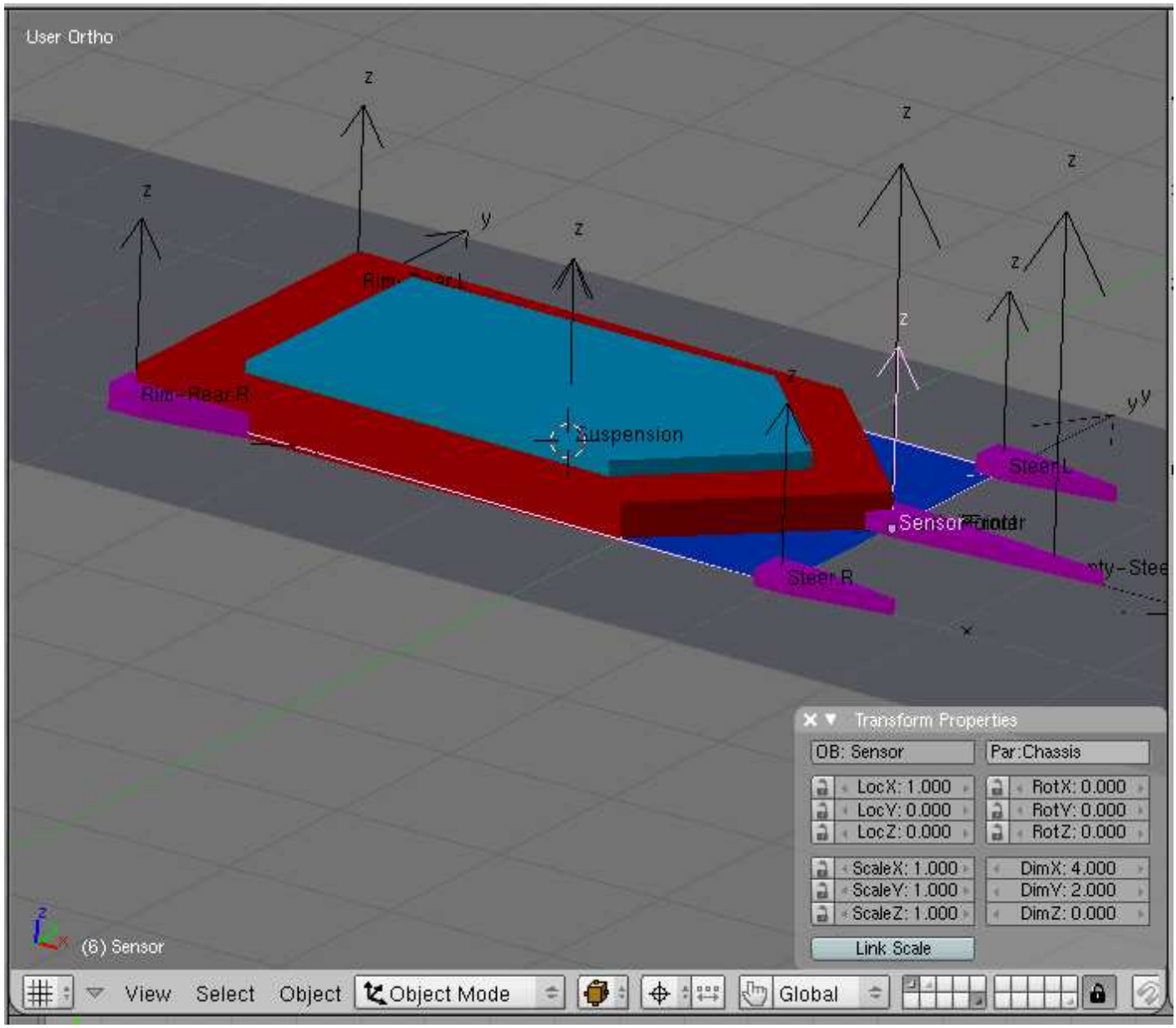
#038

- Adding colors to the materials will be helpful in future operations.

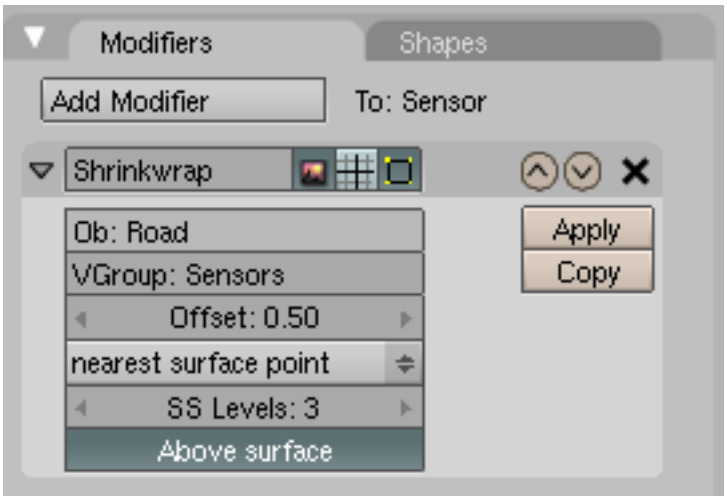


#039

- Select the Sensor plane, and add to it a Shrinkwrap Modifier. Type Road in the Ob field, and Sensors in the VGroup field. Choose Nearest surface point for the mode. Set the Subsurface Level to 3, and enable the Above surface button.



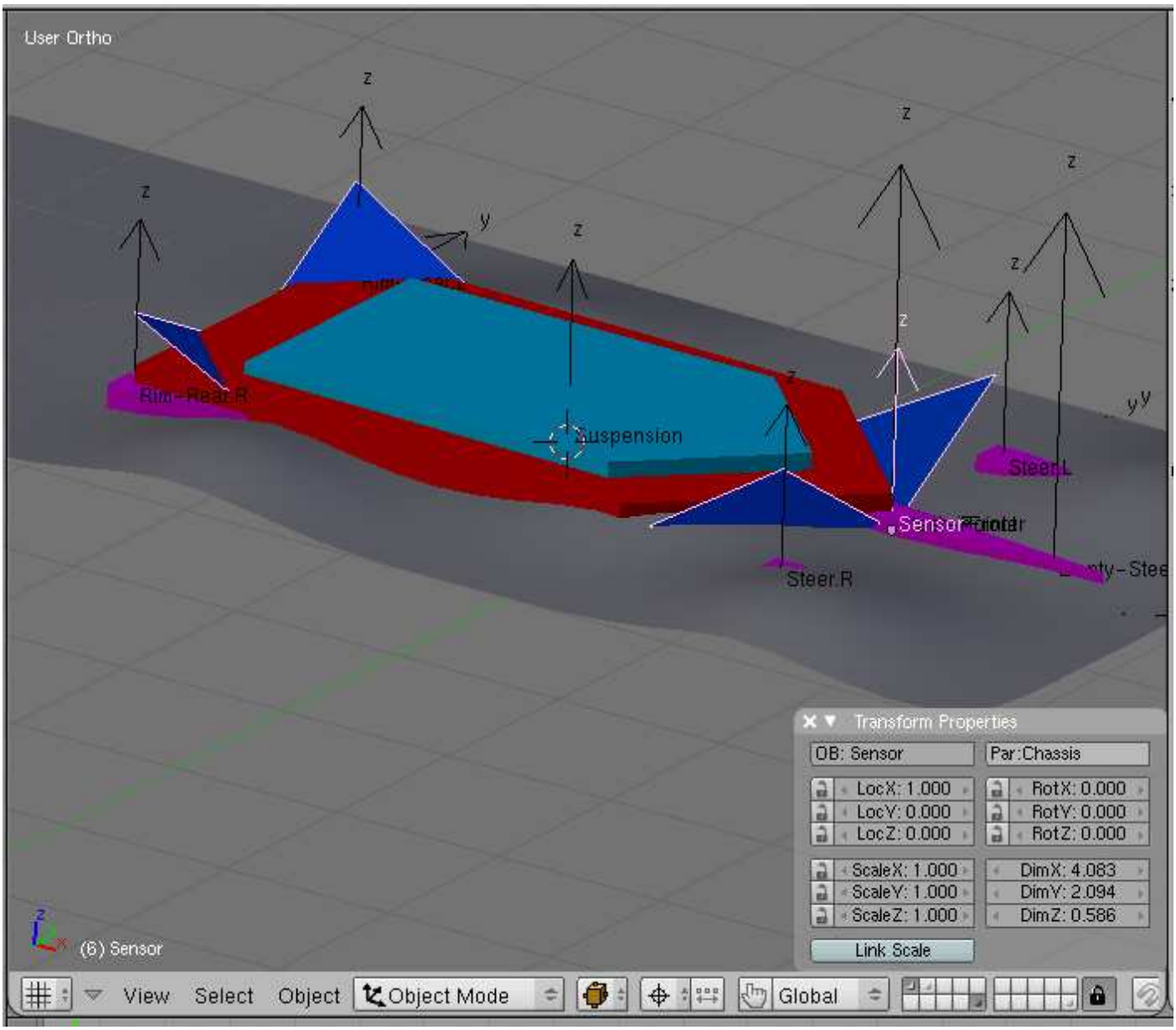
#040



#040B

Now, shift the Road plane in the X direction, and bring the bumpy arear at the car rig location. You can notice that four corners of the Sensor plane (the vertices used as Sensors) move.

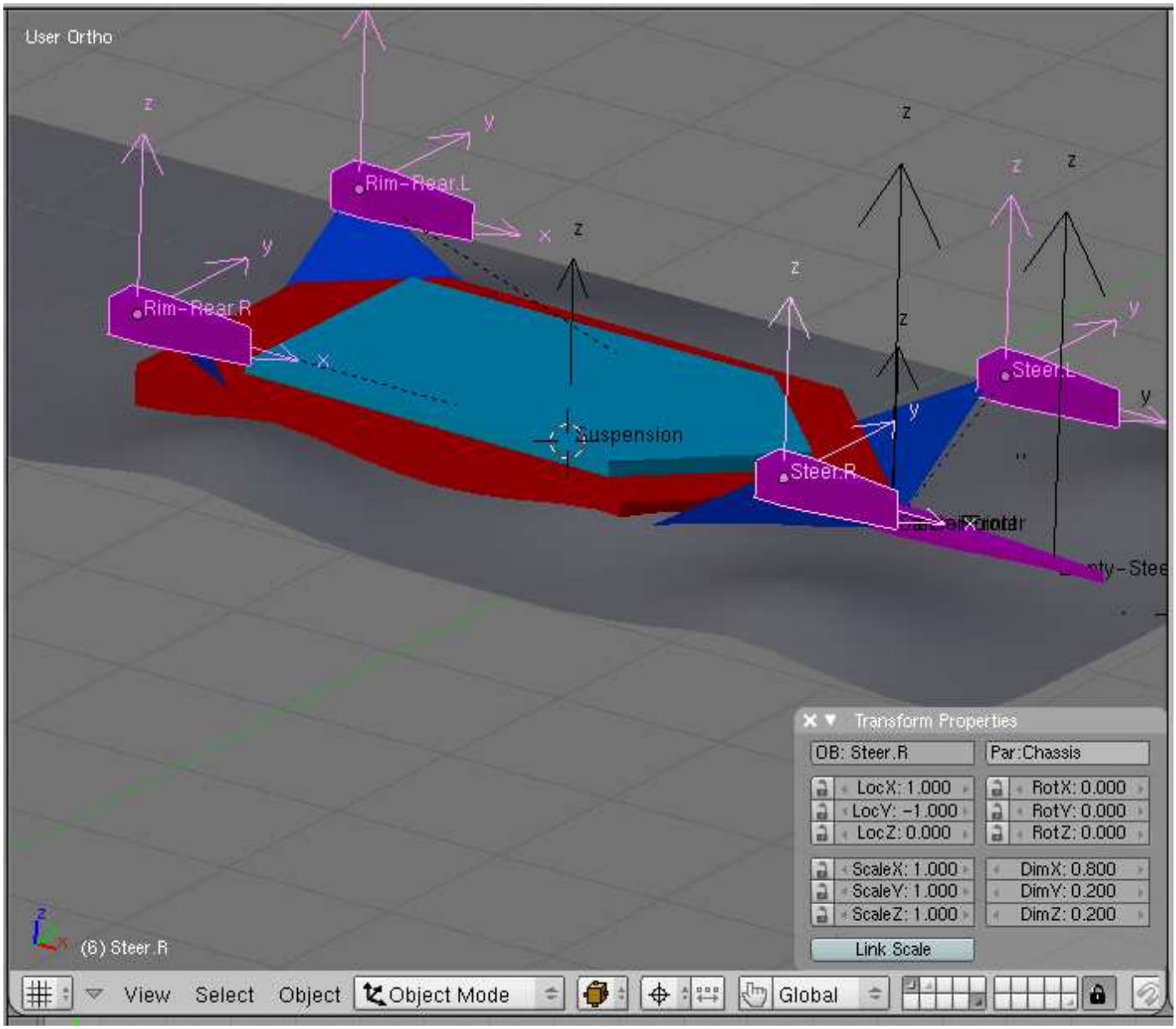
- Select the Sensor plane and adjust the Offset value of the Shrinkwrap modifier to 0.50. The four corners will rise up.



#041

- They also move a bit in X and Y direction. This is an unwanted effect that may be the cause of a slight lack of accuracy, but it is not really noticeable in most cases, because we will use only the Z information of each sensor.

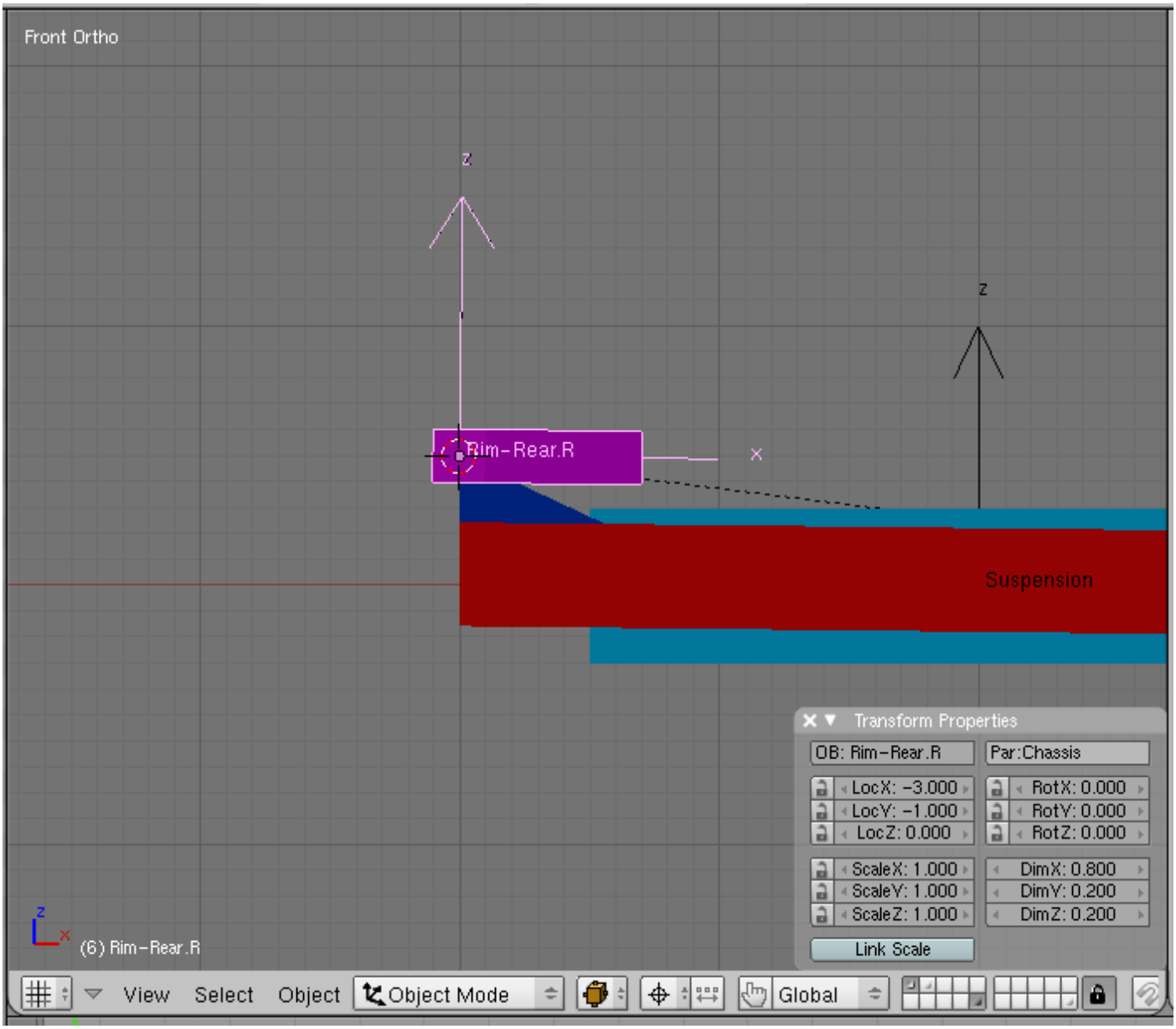
- The four wheels supports (objects of purple color) are parented to the chassis, and each of them has a Copy Location constraint using as target one of the four Vertex Groups used as sensors. So, by adjusting the Offset value of the Shrinkwrap modifier, you adjust the altitude of the center of the wheels over the Road level. It is very useful and easy to adjust this altitude with precision when you change the diameter of the wheels.



#042

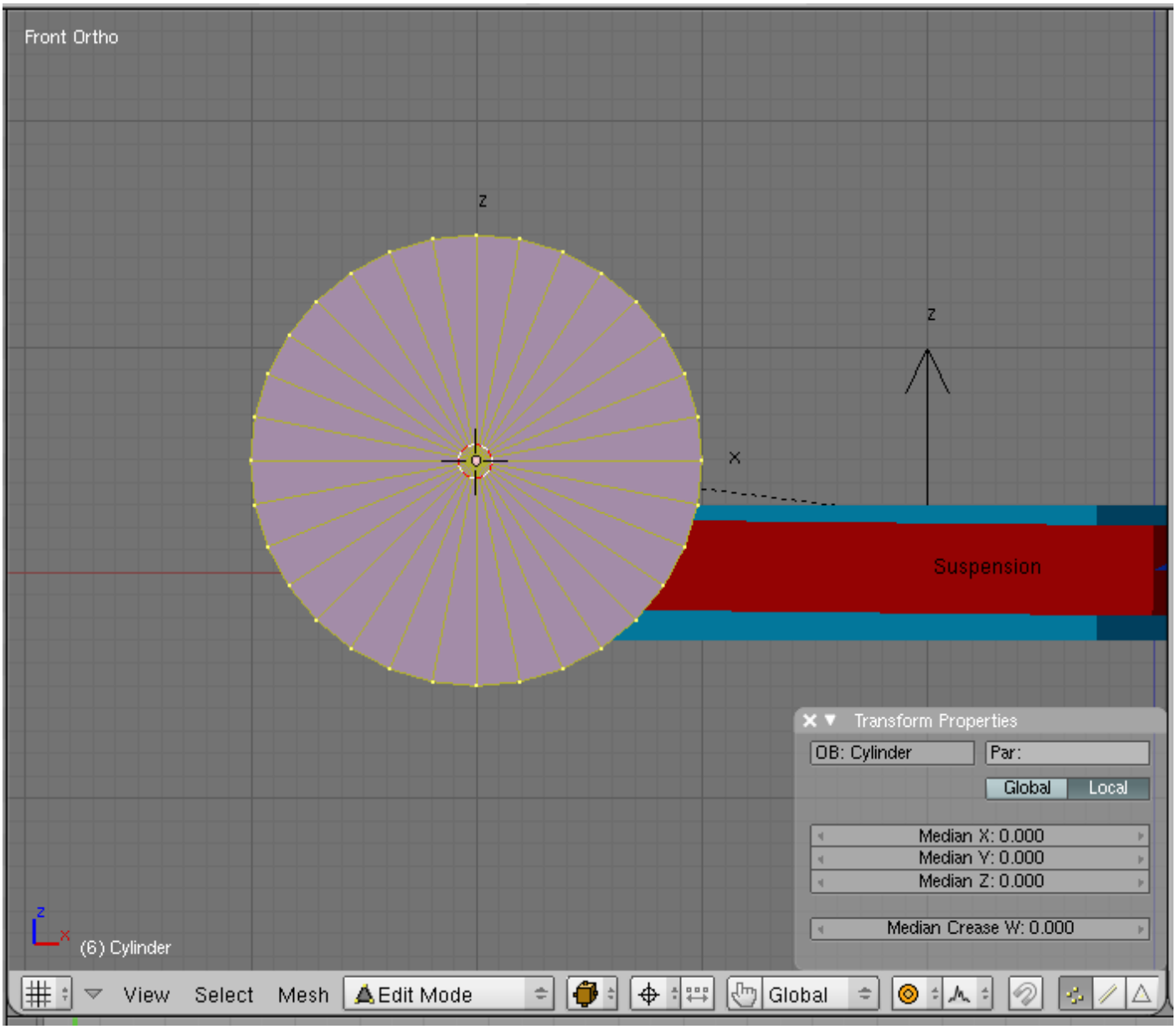
- Move the road on the X axis to displace the bumpy part before the car rig, to avoid deformations of the rig.

- Select the Rim-Rear.R object, and bring the 3D Cursor on it.



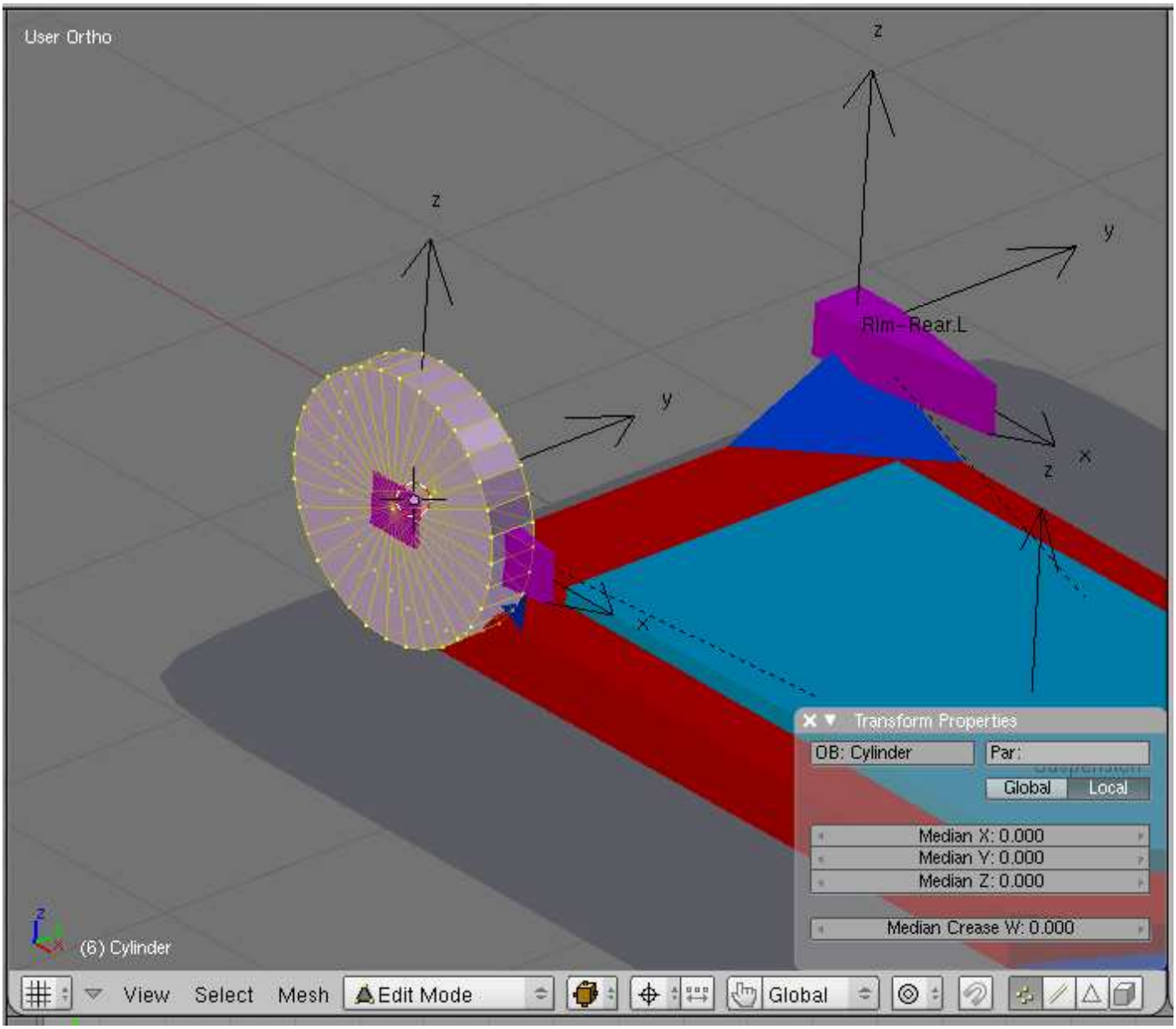
#043

- Add a Cylinder object.



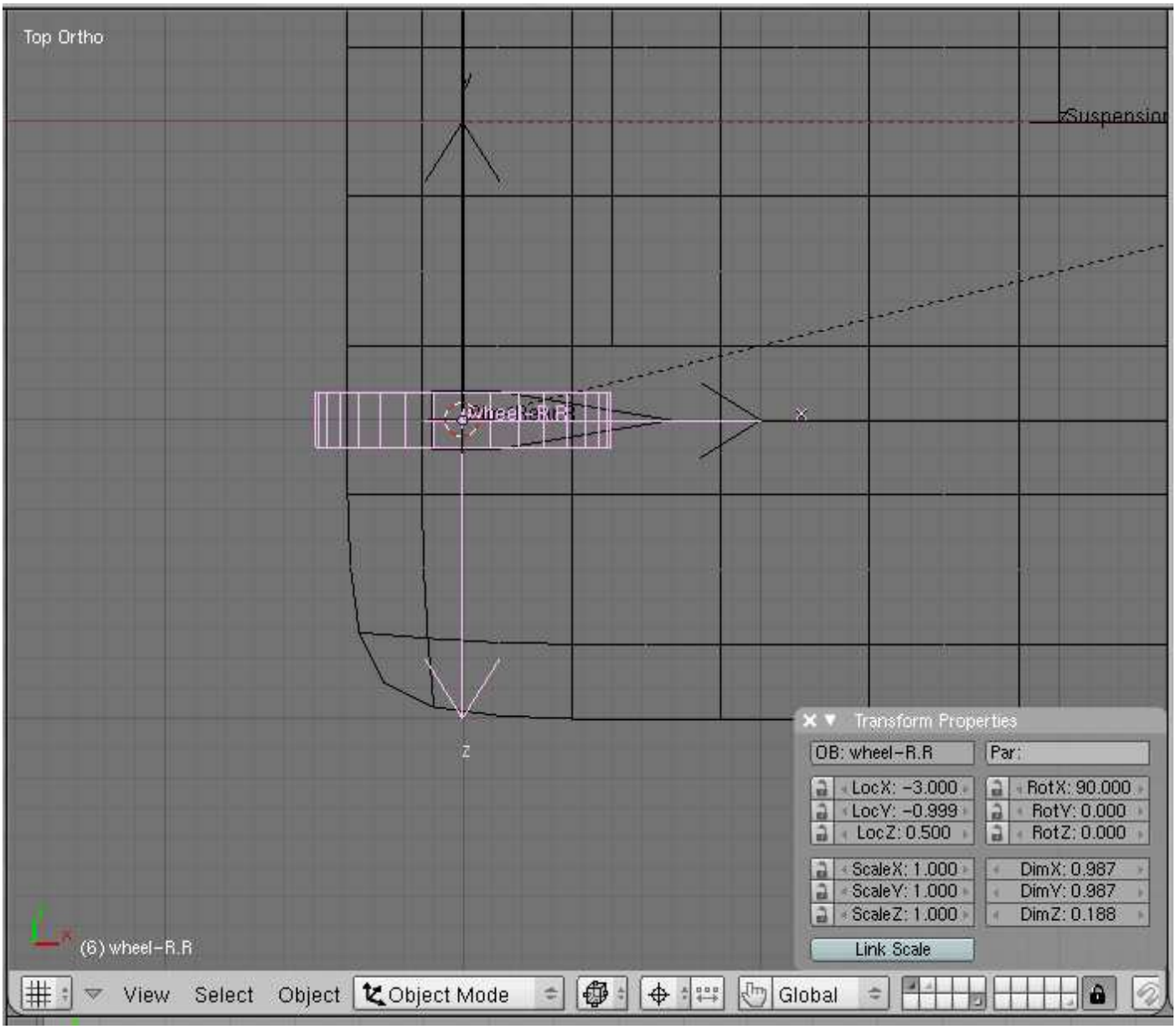
#044

- Scale it down in Y direction and scale it again to get something with the proportions of your wheels template.



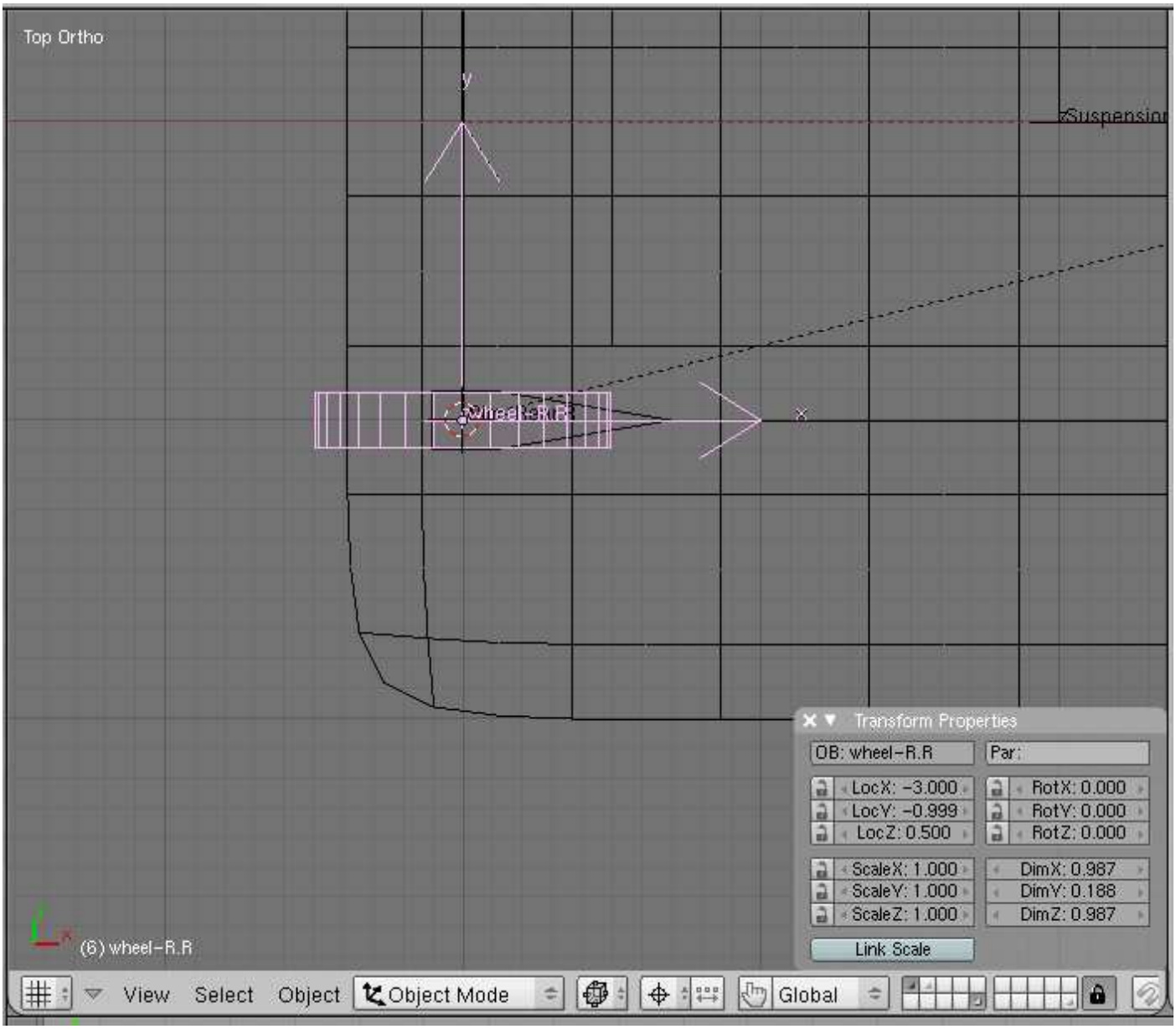
#045

- In top view, look at the axis of your wheel.



#046

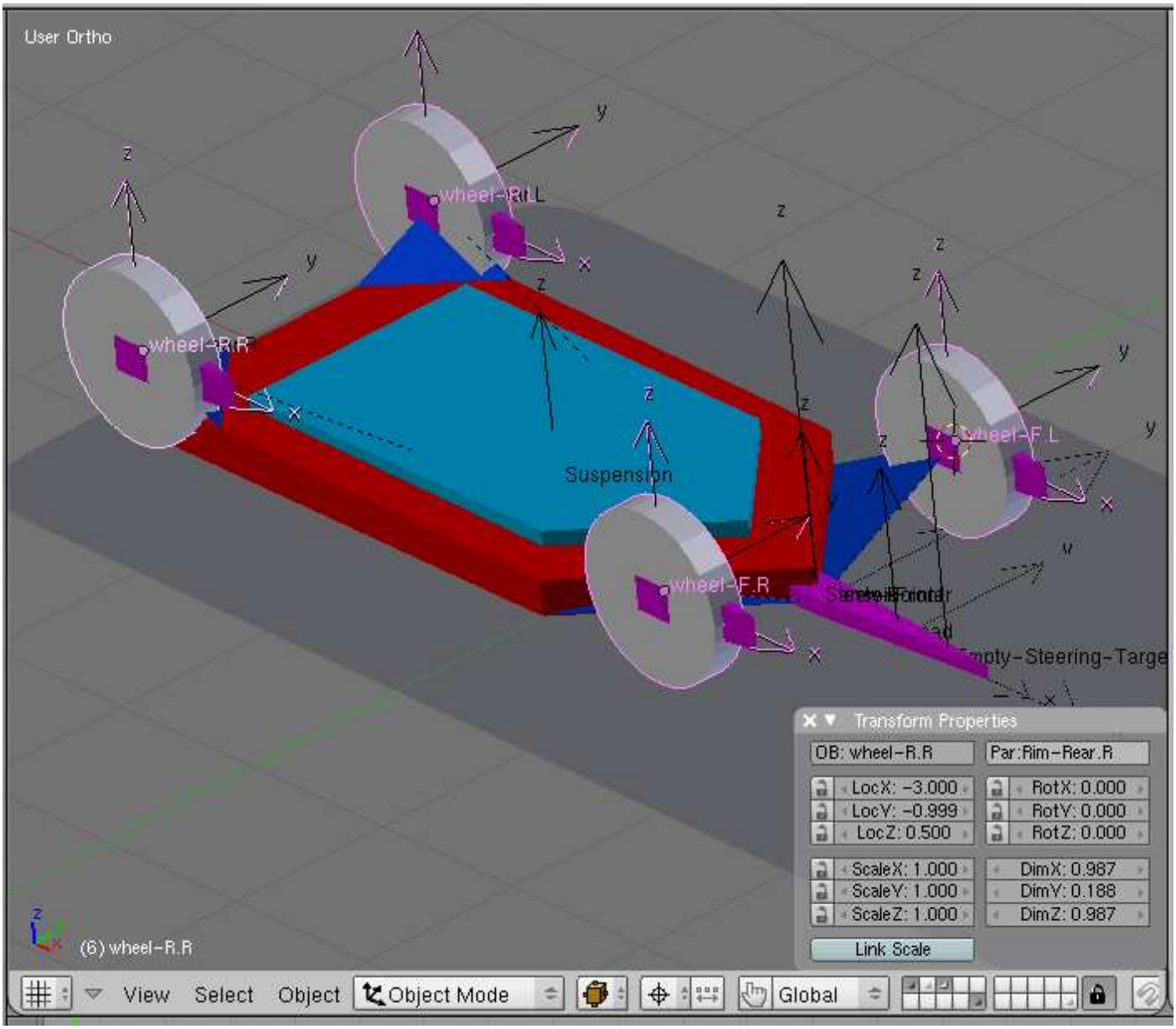
- To get a normal behaviour when we will use the wheelsrotation.py script, the Y axix of the wheels must all point to the left part of the car. So, apply Ctrl+A.



#047

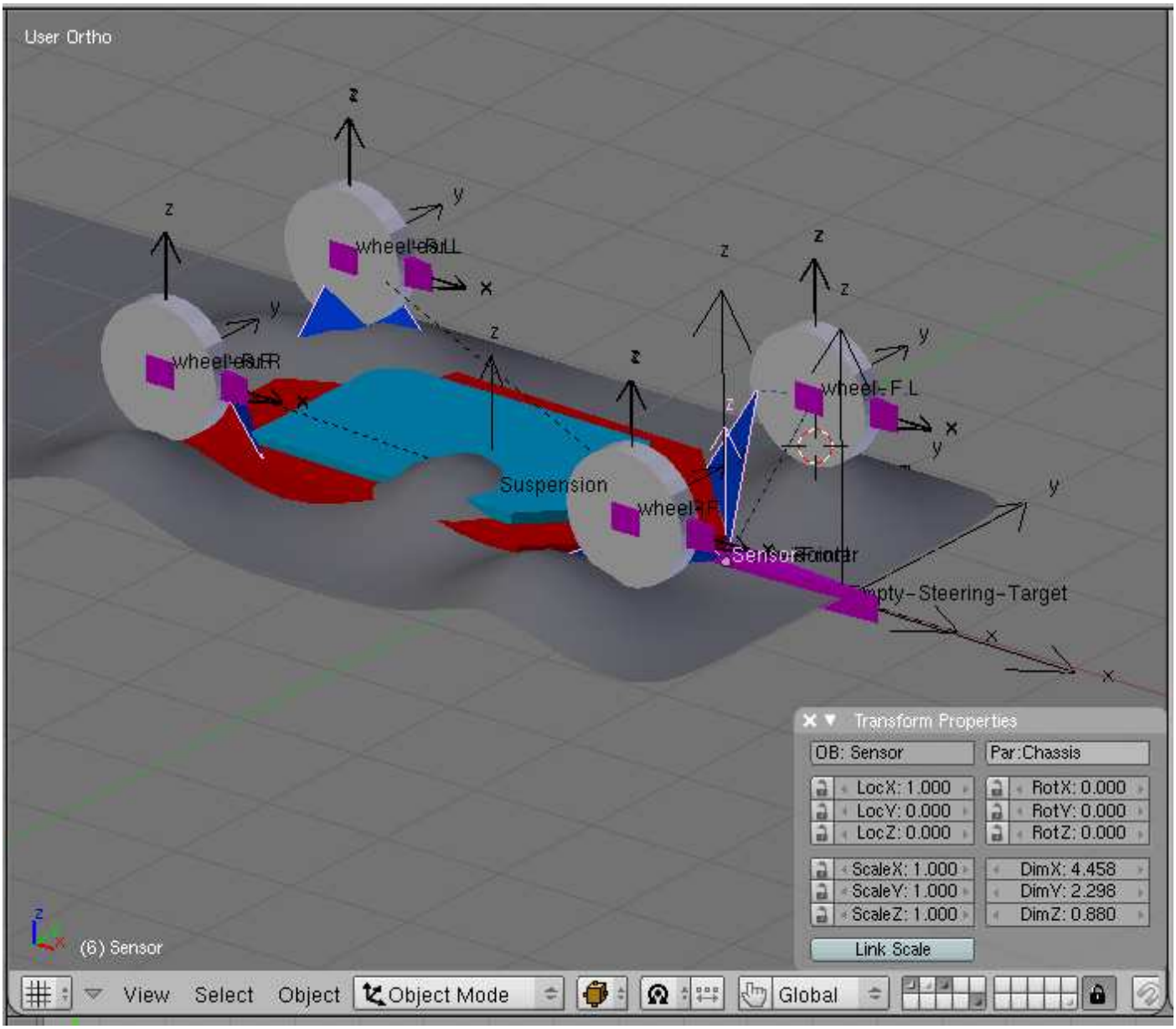
- Now, parent the wheel to the wheels support Rim-Rear.R.

- Do the same thing for the 3 other wheels, and name your wheels. To allow the script rotating your wheels, it is important to name them wheel... (more details in the script file itself).



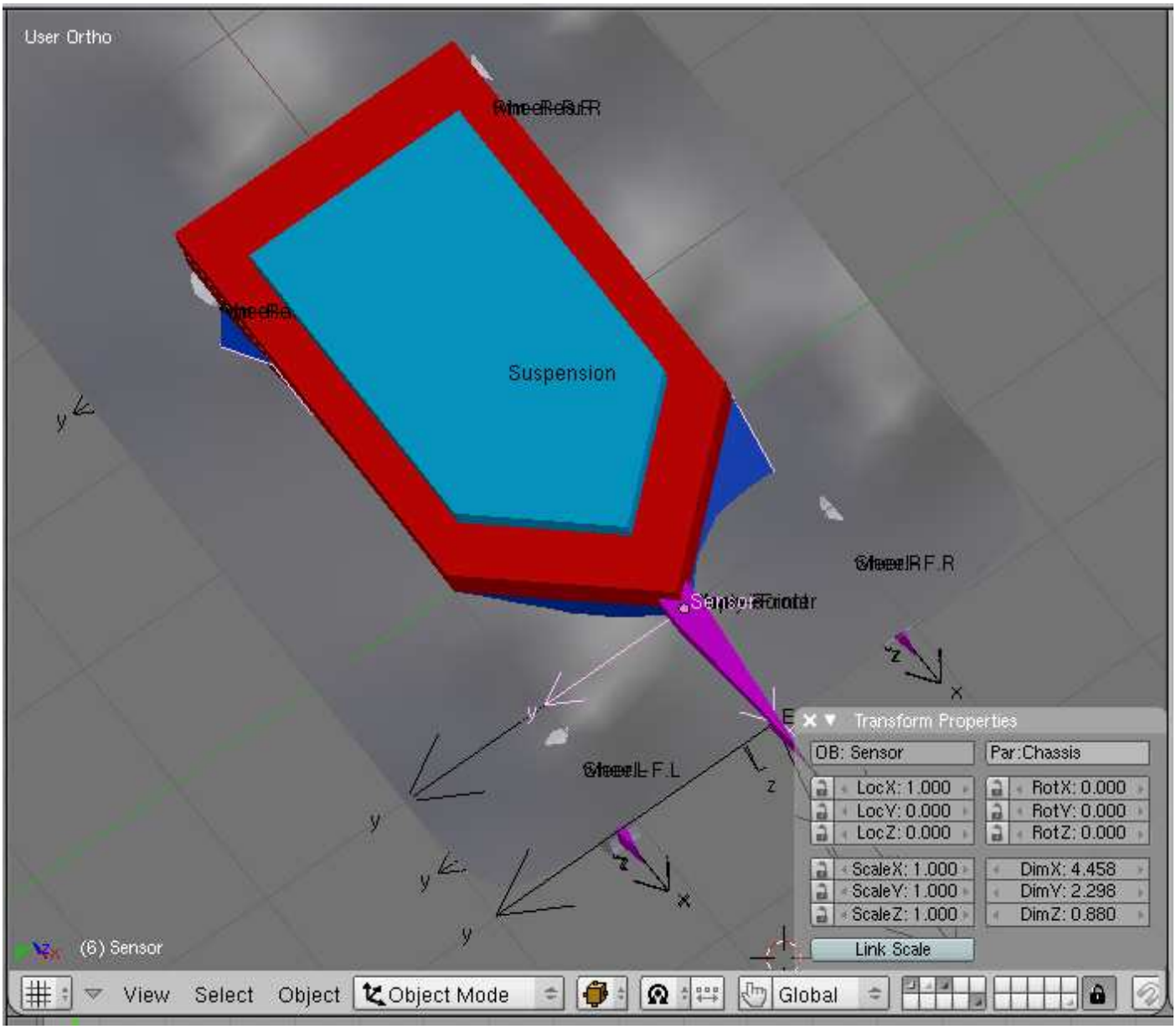
#048

- Now, if you move again the Road mesh under the car rig, you will see that the wheels are following the relief of the road.



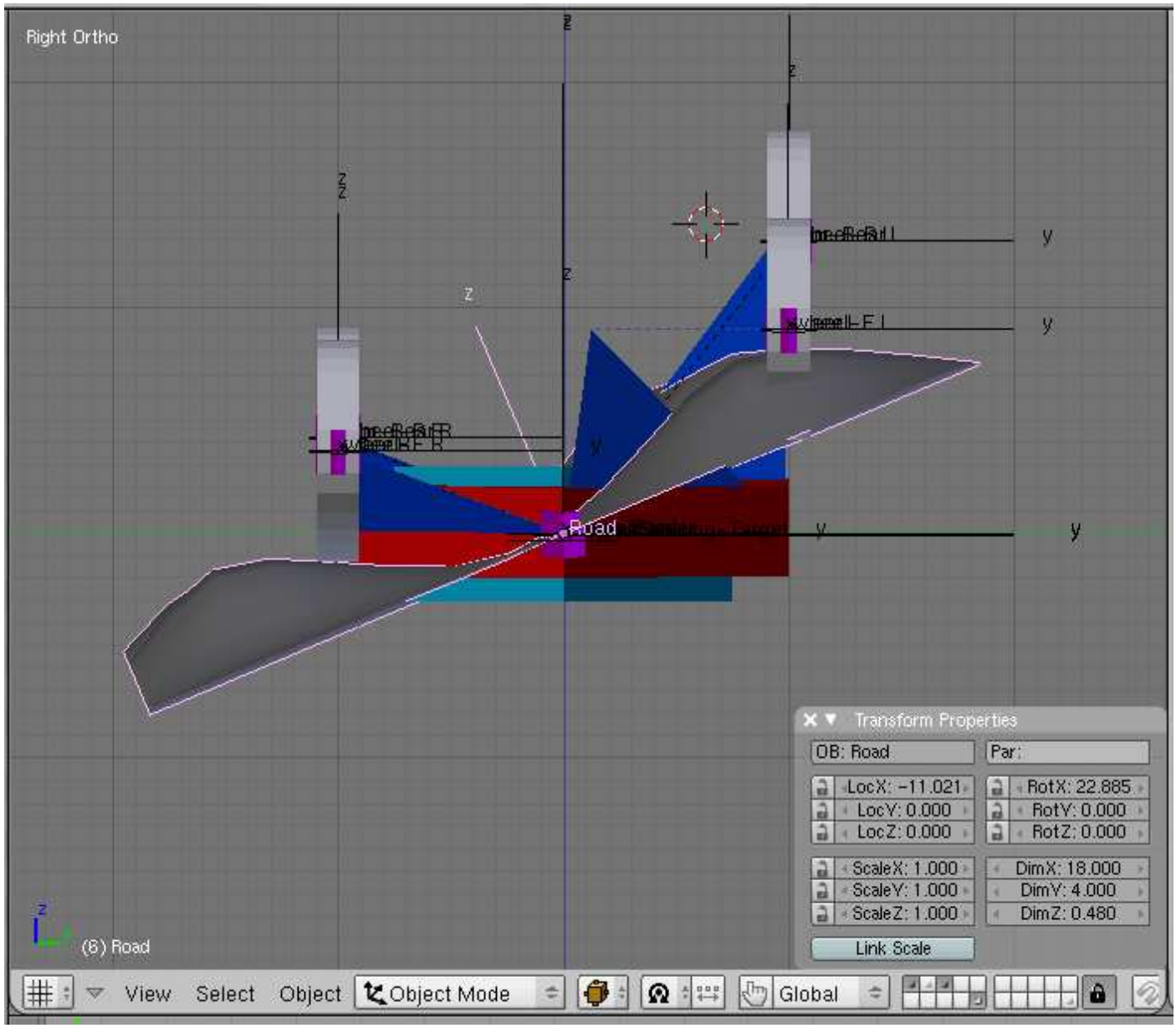
#049

- You can look under the road, and adjust the Offset value of the Shrinkwrap modifier while looking to the wheels. They must be just passing through the road a little bit.



#050

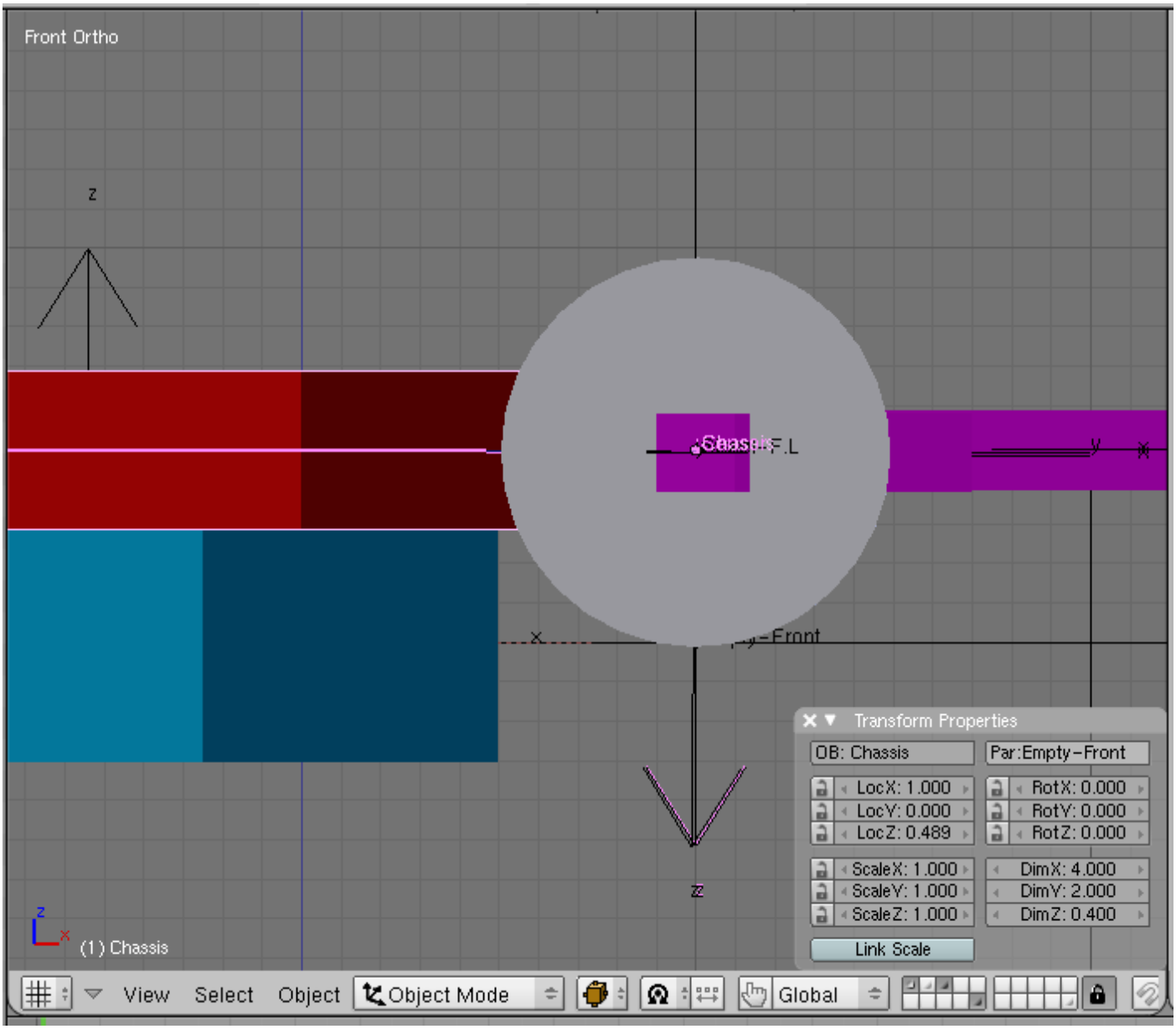
- Looking to the front of the rig, if you rotate the Road mesh to get a transversal lope, you will notice a bad thing : The wheels are following the altitude of the terrain, but the chassis doesn't follow, and the wheels don't follow the direction of the normal of the terrain.



#051

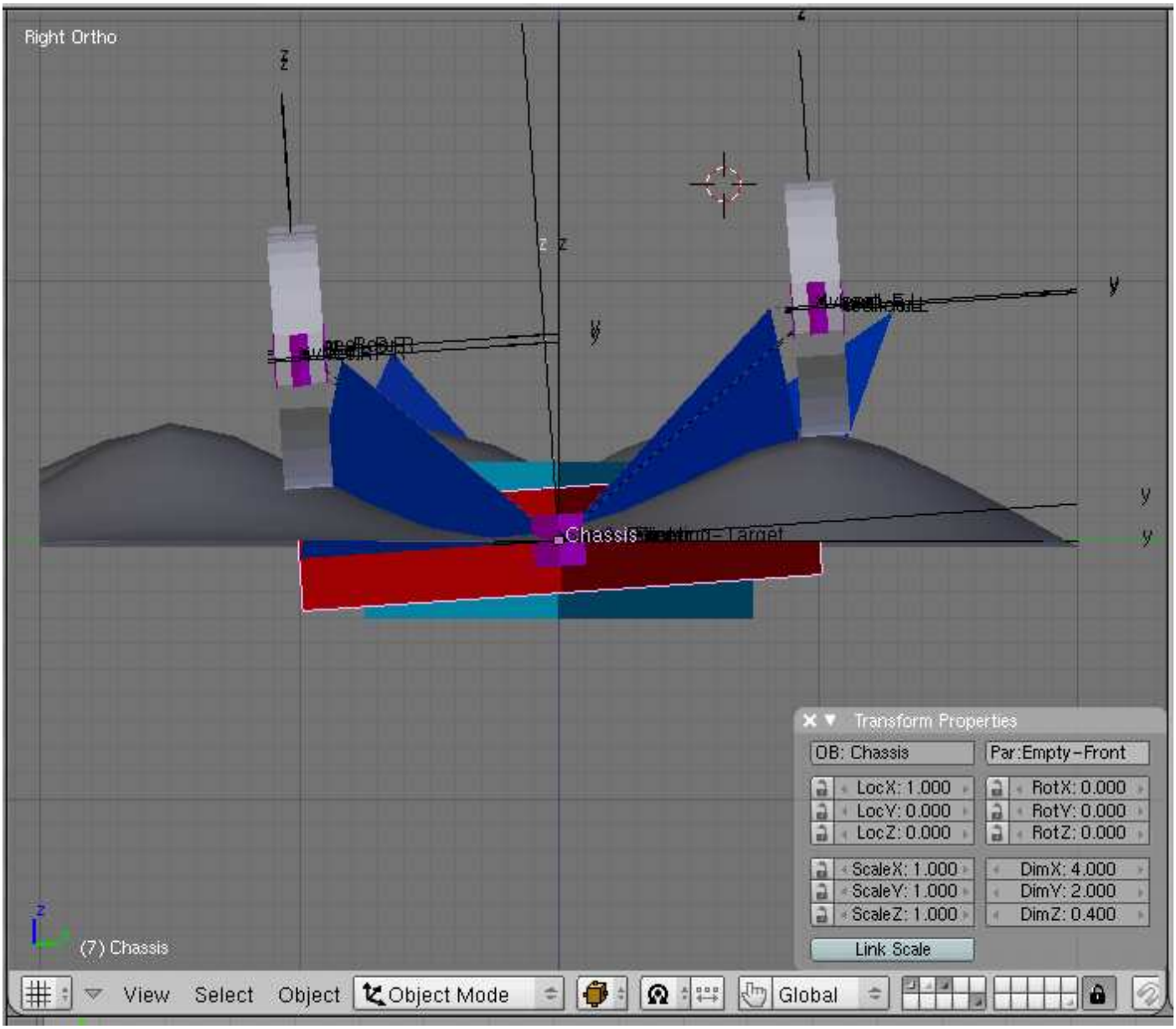
- First, we have to correct a bad parallax error. As the Chassis and the default position of the Sensor Mesh are not at the altitude of the wheels axles, the whole rig is affected by some weird deformations.

- On a flat road (move the Road mesh if necessary), Select the Chassis and the Sensor mesh, and looking at the side of the rig, just move up the two object until the center of the Chassis is at the height of the front wheels axles.



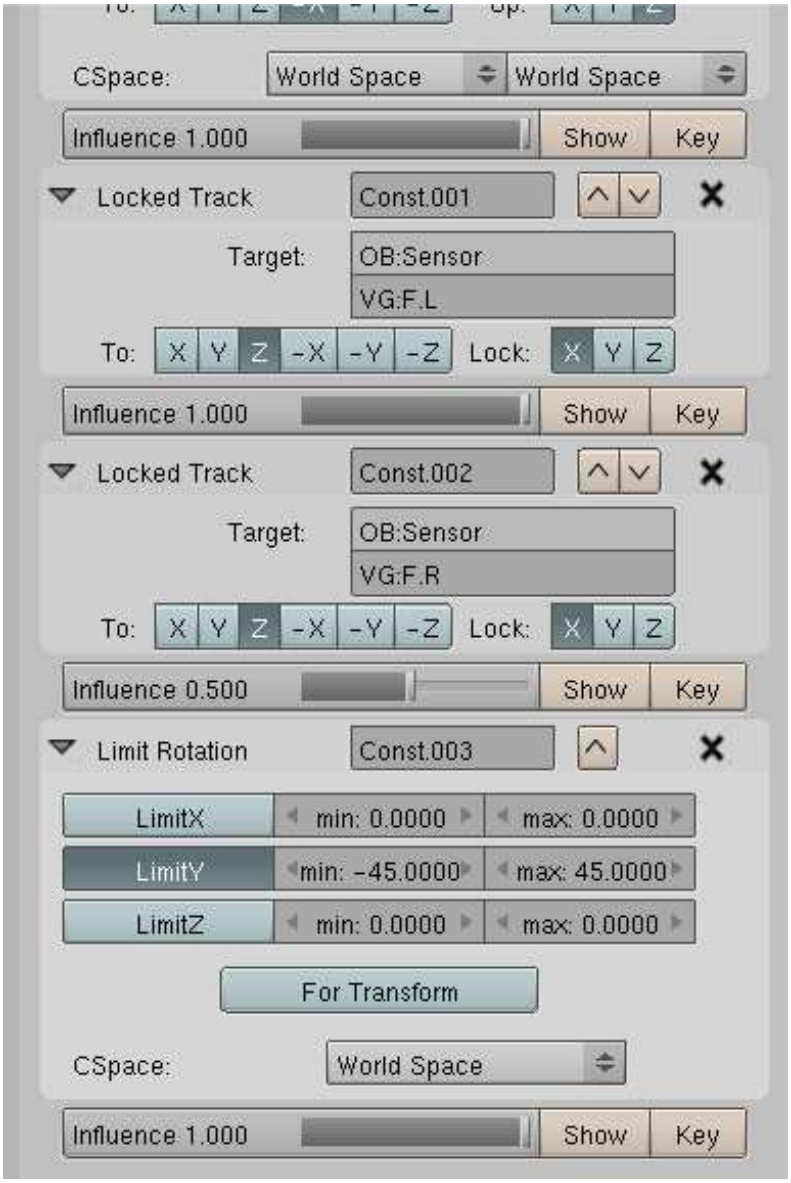
#053

- The rear of the rig doesn't follow immediately. It does when you go forward few frames and come back to frame 1, to reset the constraints.

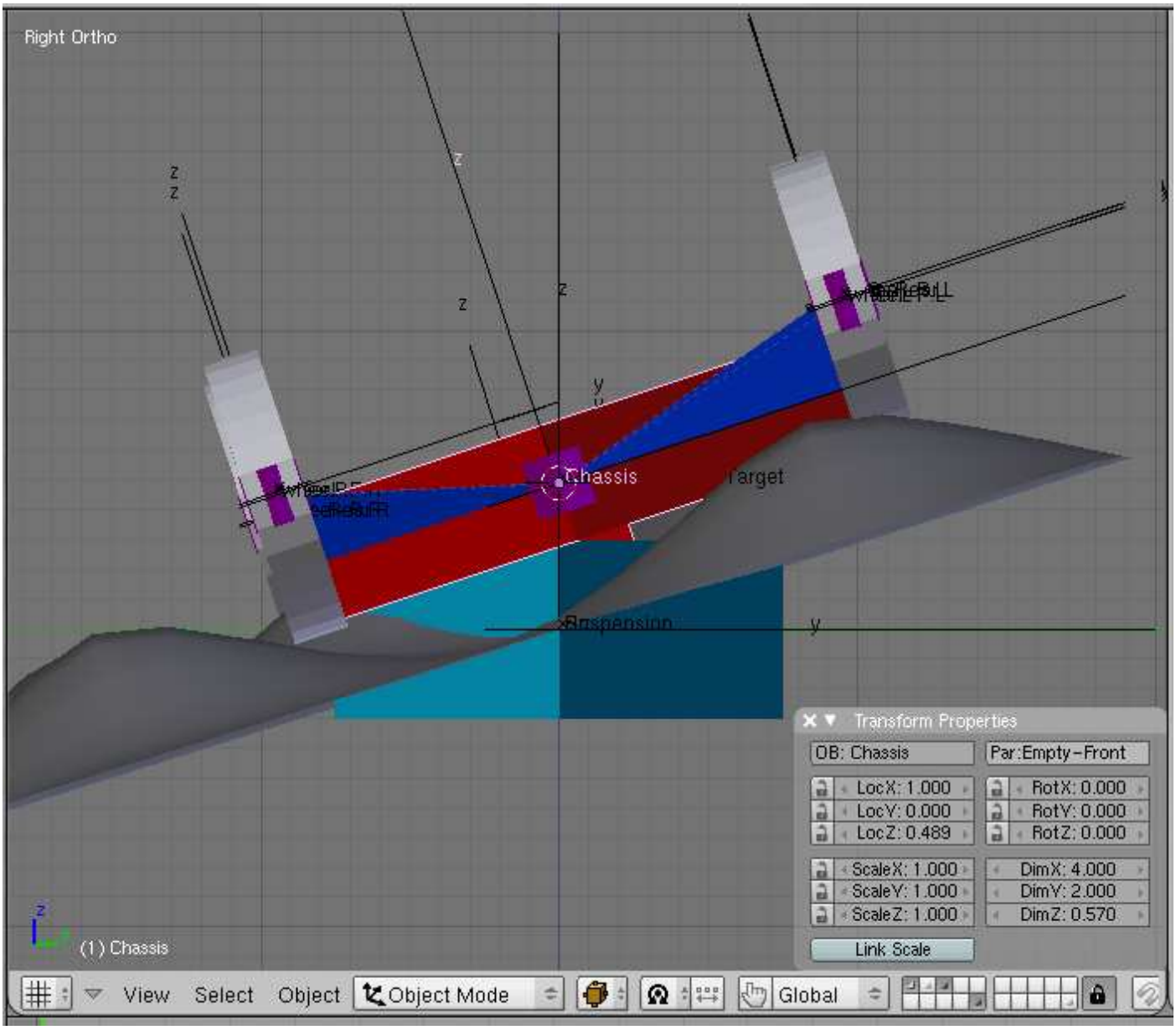


#052

- By adding two Locked Track Constraints (Locked on X axis) to the Chassis, Tracking the F.L and F.R Sensors, the problem is solved.

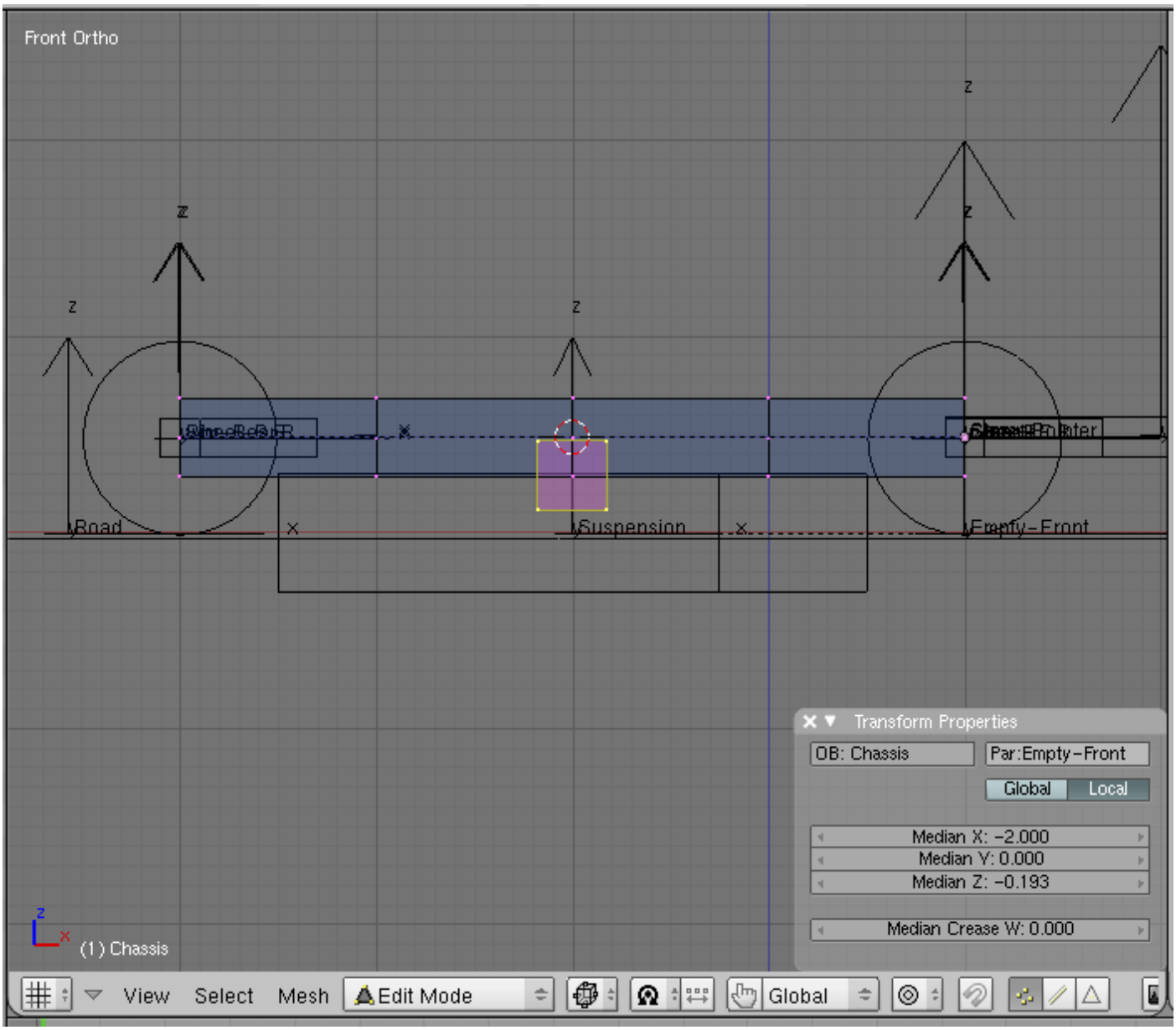


#053B


Right Ortho
Chassis
Target
Transform Properties
OB: Chassis
Par:Empty-Front
LocX: 1.000
LocY: 0.000
LocZ: 0.489
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
DimX: 4.000
DimY: 2.000
DimZ: 0.570
Link Scale
(1) Chassis
View
Select
Object
Object Mode
Global

#054

- You may notice that I have added a cube under the Chassis. It is a visual clue, allowing to detect easily the flip of the chassis when playing with tracking constraints. To avoid the Chassis flipping by accident in some extreme conditions, its rotation around its X axis is limited to the range of - 45° to + 45°.

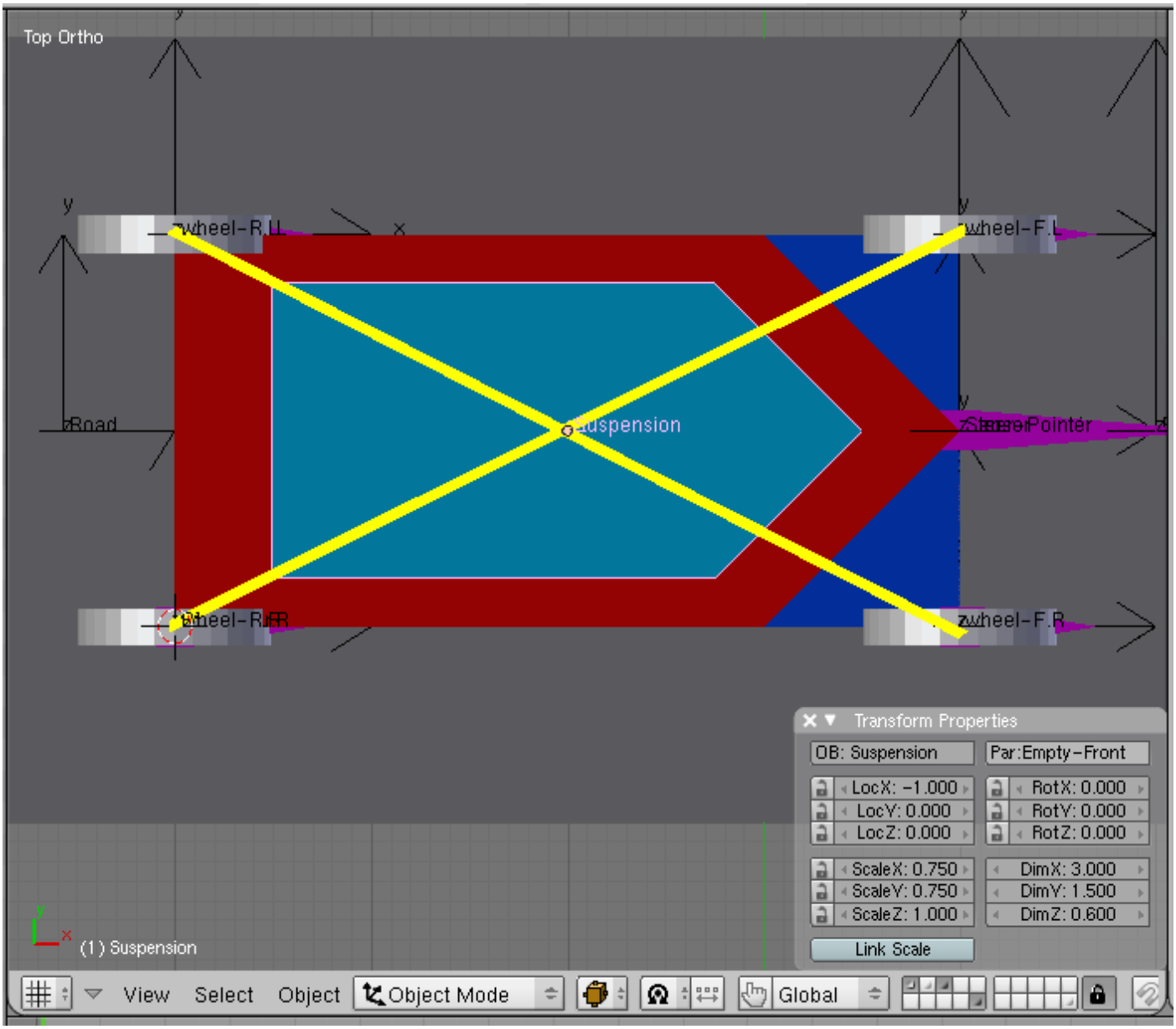


#056

Step 6 : Activating the Suspension.

- Select the Suspension and add four Copy Location constraints. Their Target is Sensor, and they are pointing to the four Sensors Groups F.L,F.R, R.L and R.R.

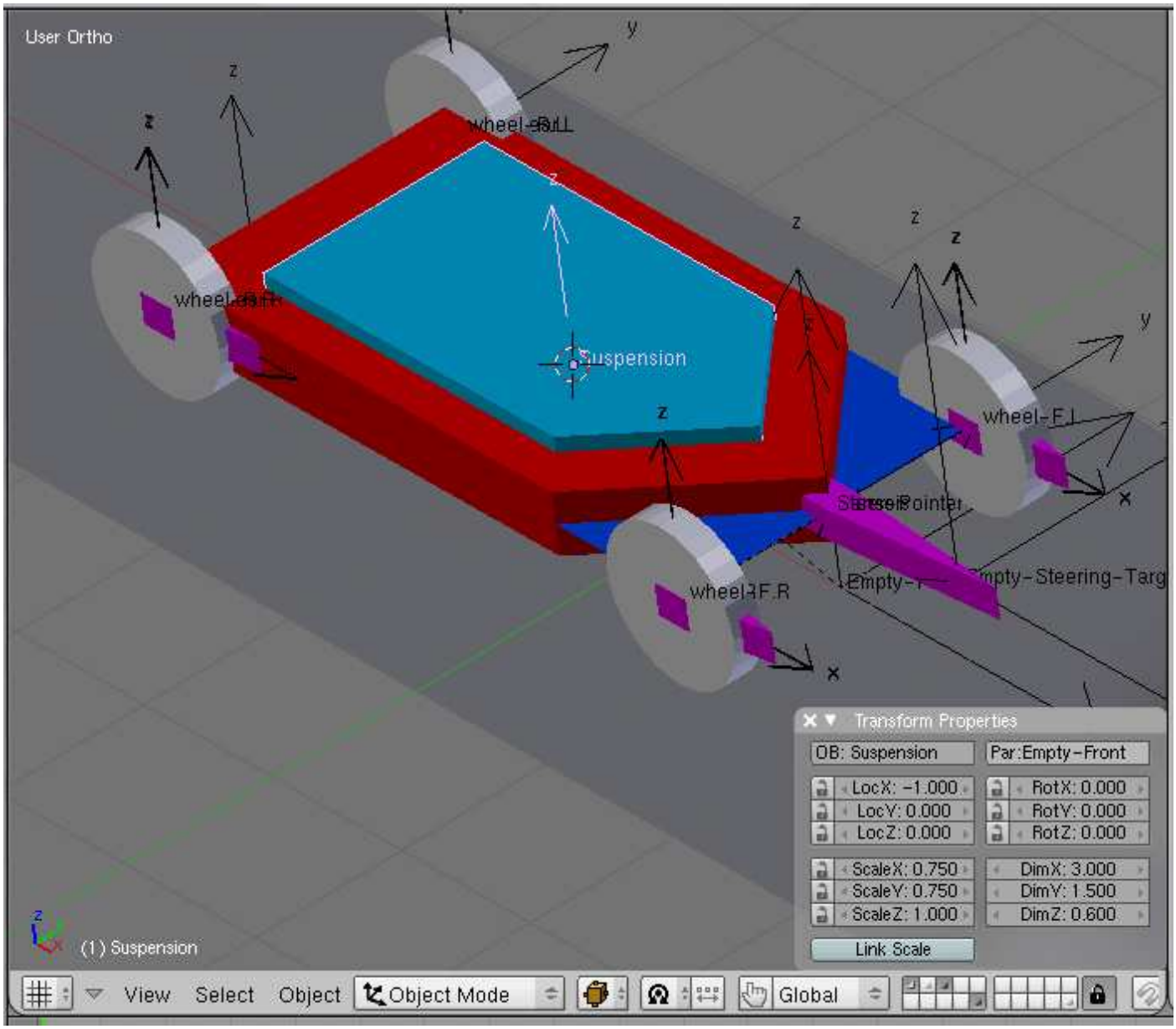
- It is important to set the Copy Location constraints using a "diagonal crossing" method : F.L with Influence = 1.000 and R.R with Influence = 0.500. Followed by F.R with Influence = 1.000 and R.L with Influence = 0.500.



#057

- If you mix the order of the constraints, you will have to make endless corrections and tweakings without getting the accurate location of the Suspension object.

- Now, the Suspension is centered between the wheels location. Its center point will be floating along the Z axis.

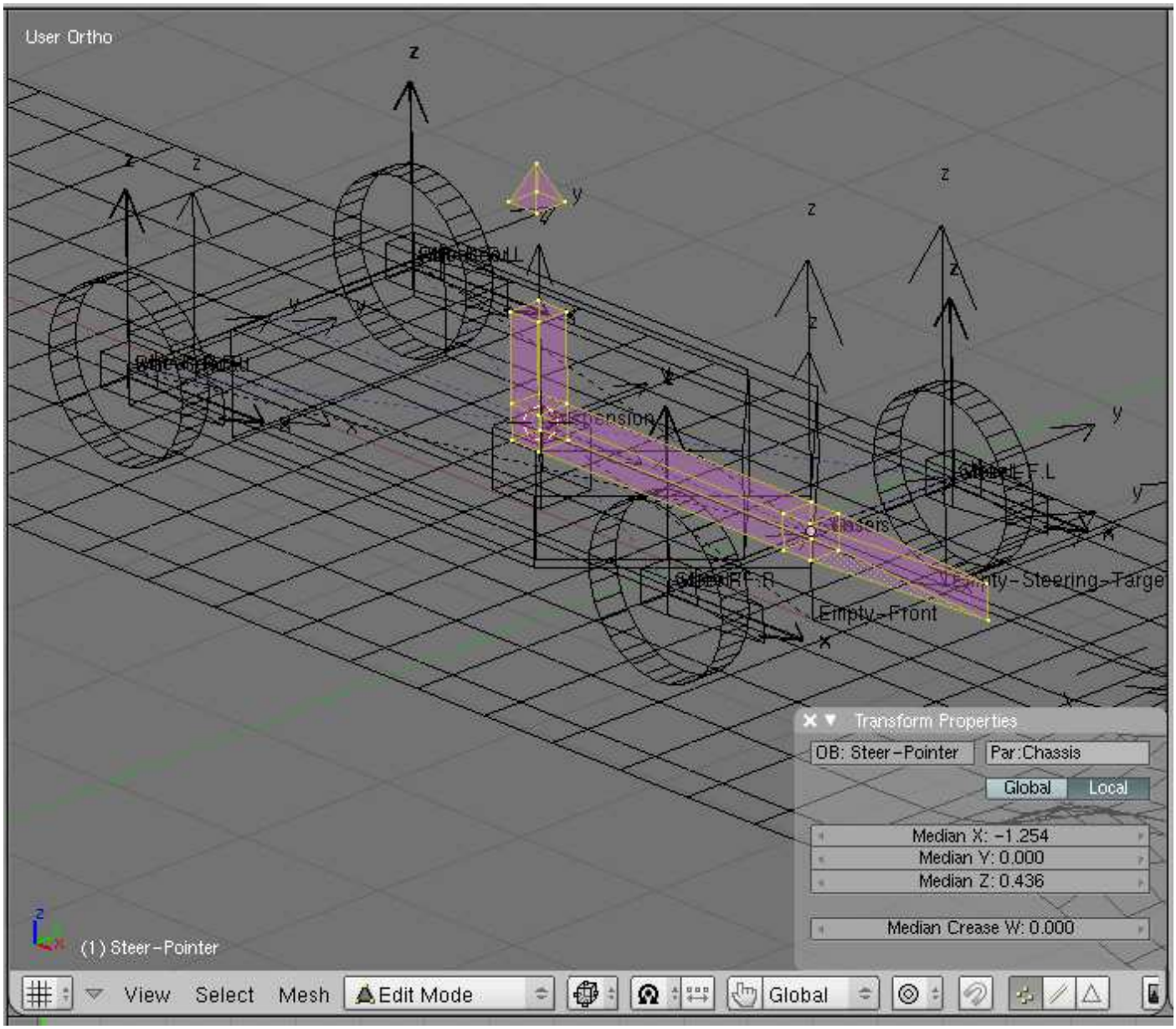


#058

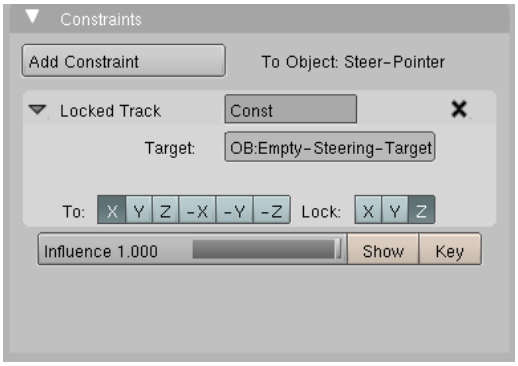
- This Suspension object will be the Parent of the car body. It will bounce over the wheels, and the wheels will never be able to pass to trespass its level.

Step 7 : Using the steering to control the Suspension.

- To achieve a realistic animation, it is important that the car simulate the effect of gravity in the curves. So, we are going to add this effect by using one more constraint to the Suspension object. It will be a Locked constraint (Locked on the local X axis), using as a target a new Vertex Group (Transversal-Slope) including only one vertex, and located at the upper tip of an extension of the Steer-Pointer object.

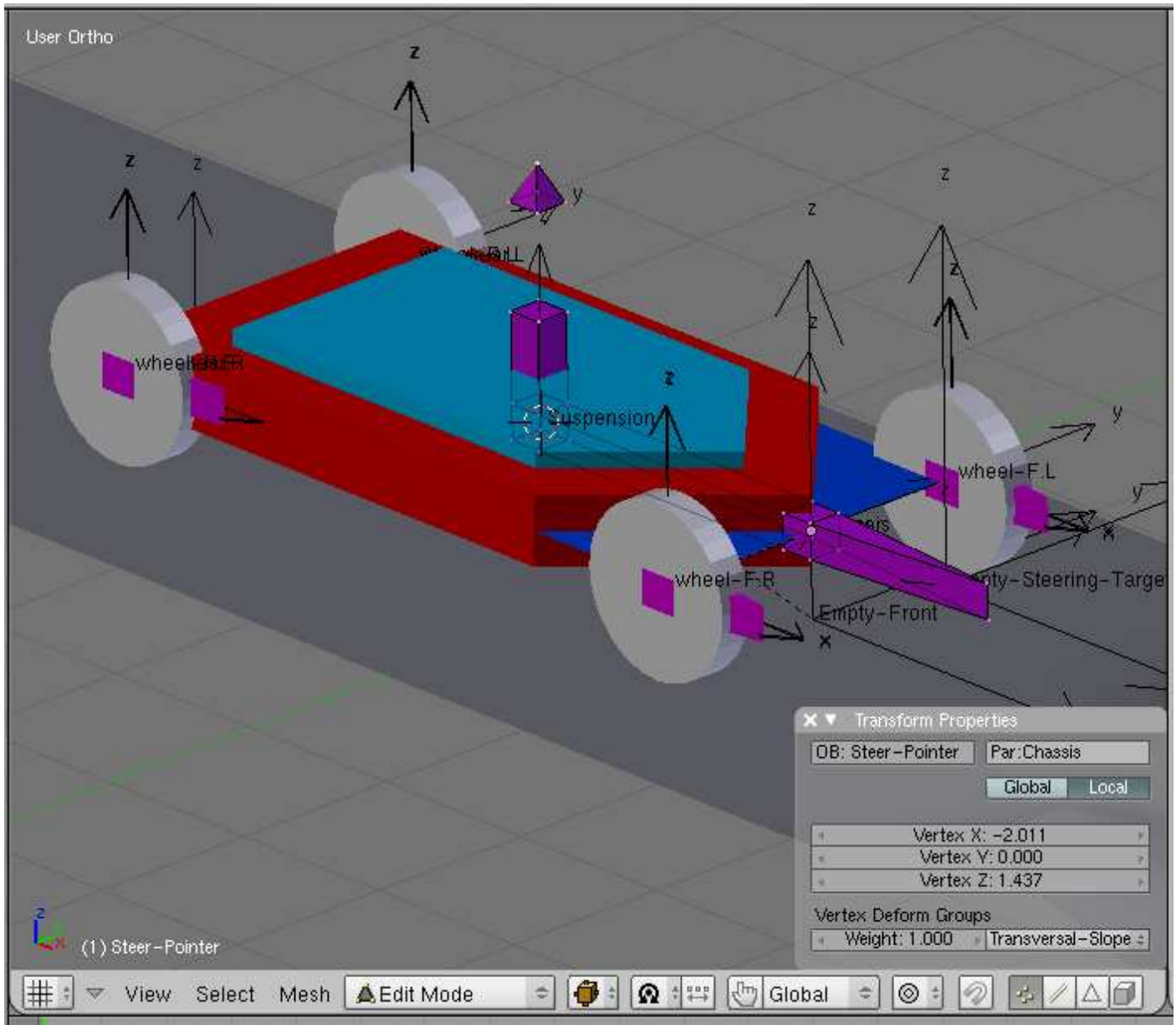


#059

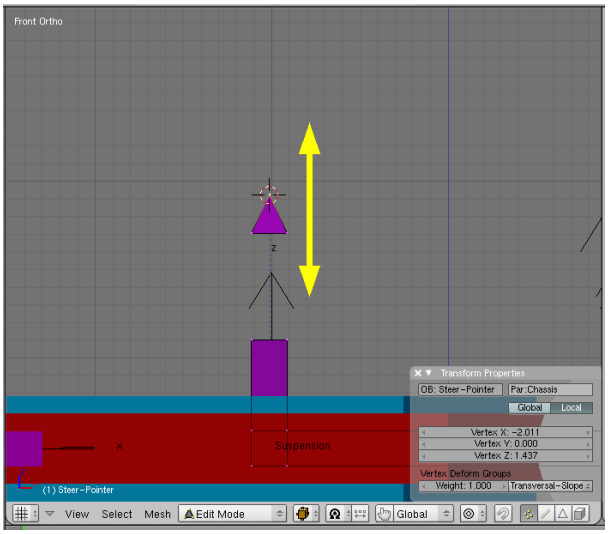


#059B

- The influence of the curve angle on the transversal slope can be adjusted in two manners : by adjusting the Influence factor of the constraint or by moving up or down in Edit Mode the Vertex Group of the Steer-Pointer used as a target for the constraint.



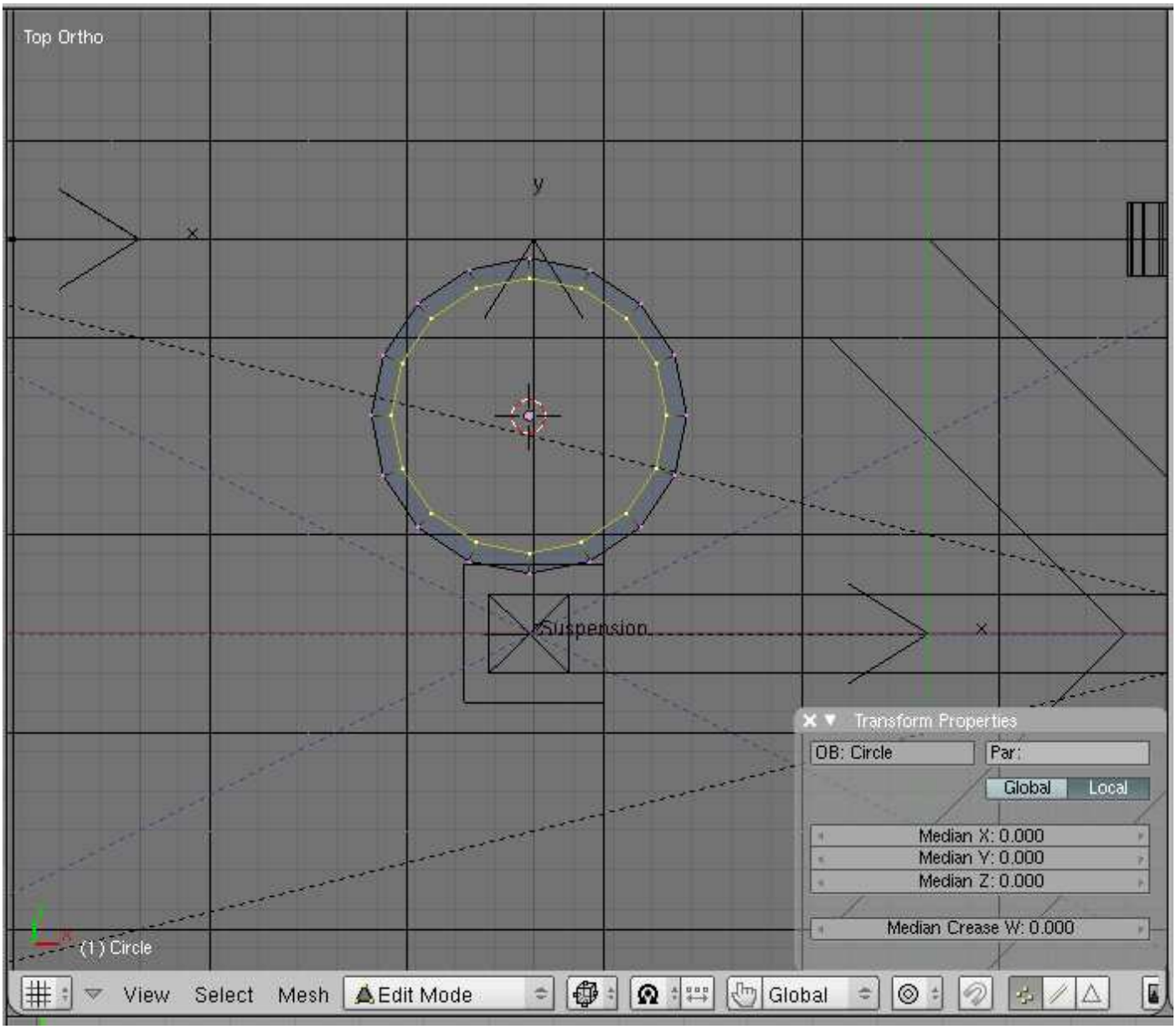
#060



#061

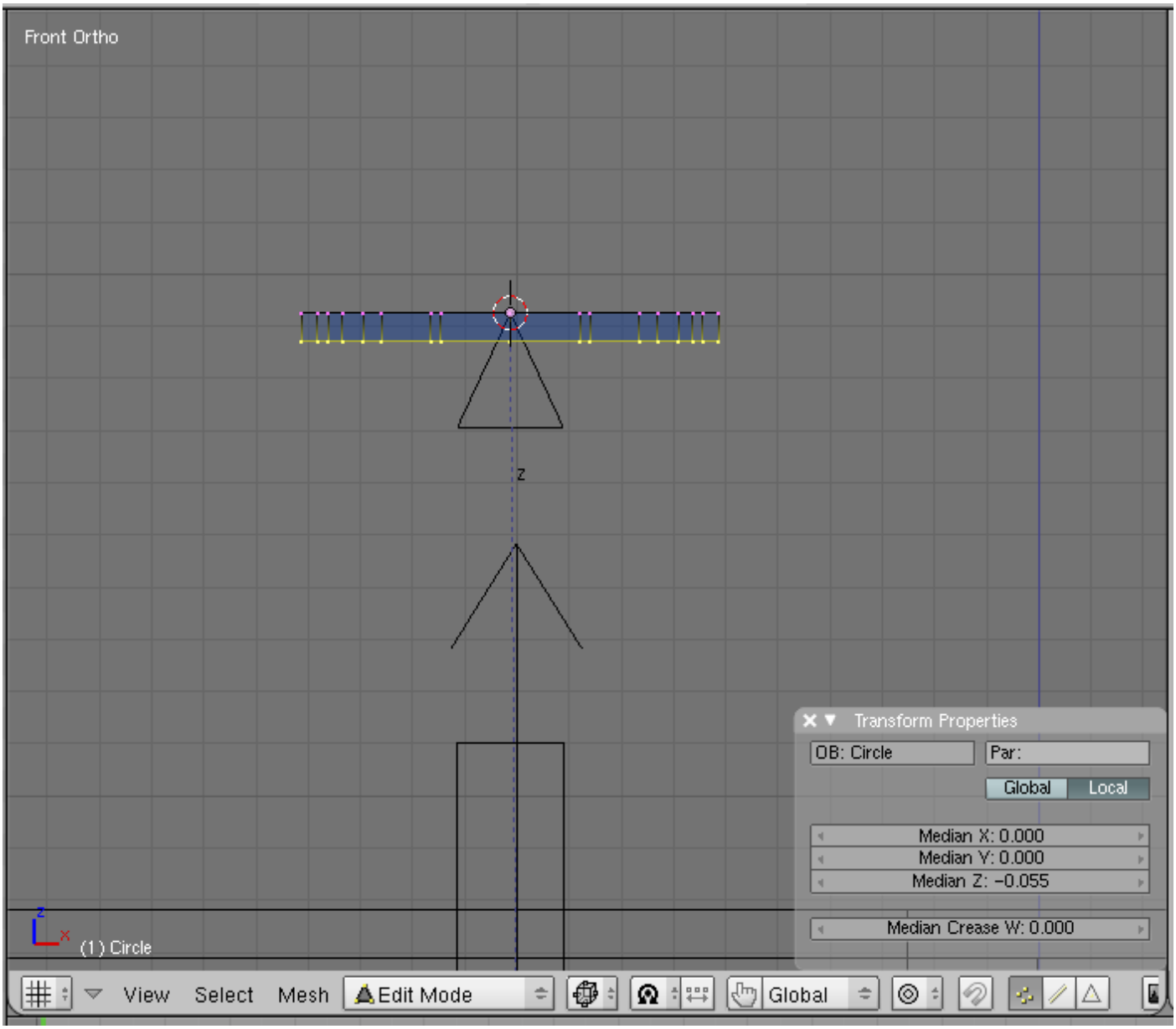
Step 8 : Using the steering to control the steering wheel.

Let us add a Steering-Wheel.
- In Top view, add a circle. Estrude and scale it down a bit.



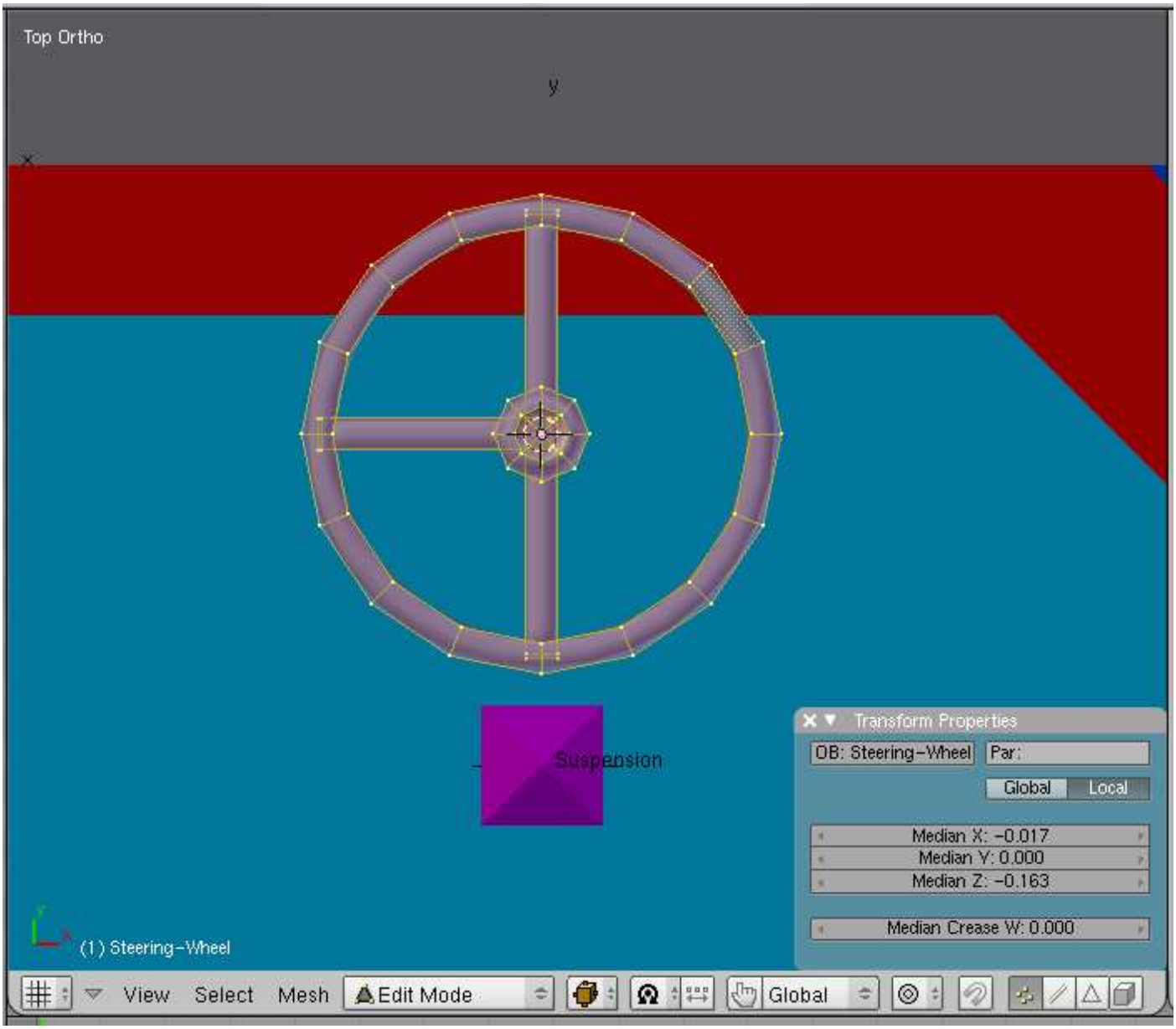
#062

- Select all the vertices and extrude again in Z direction.



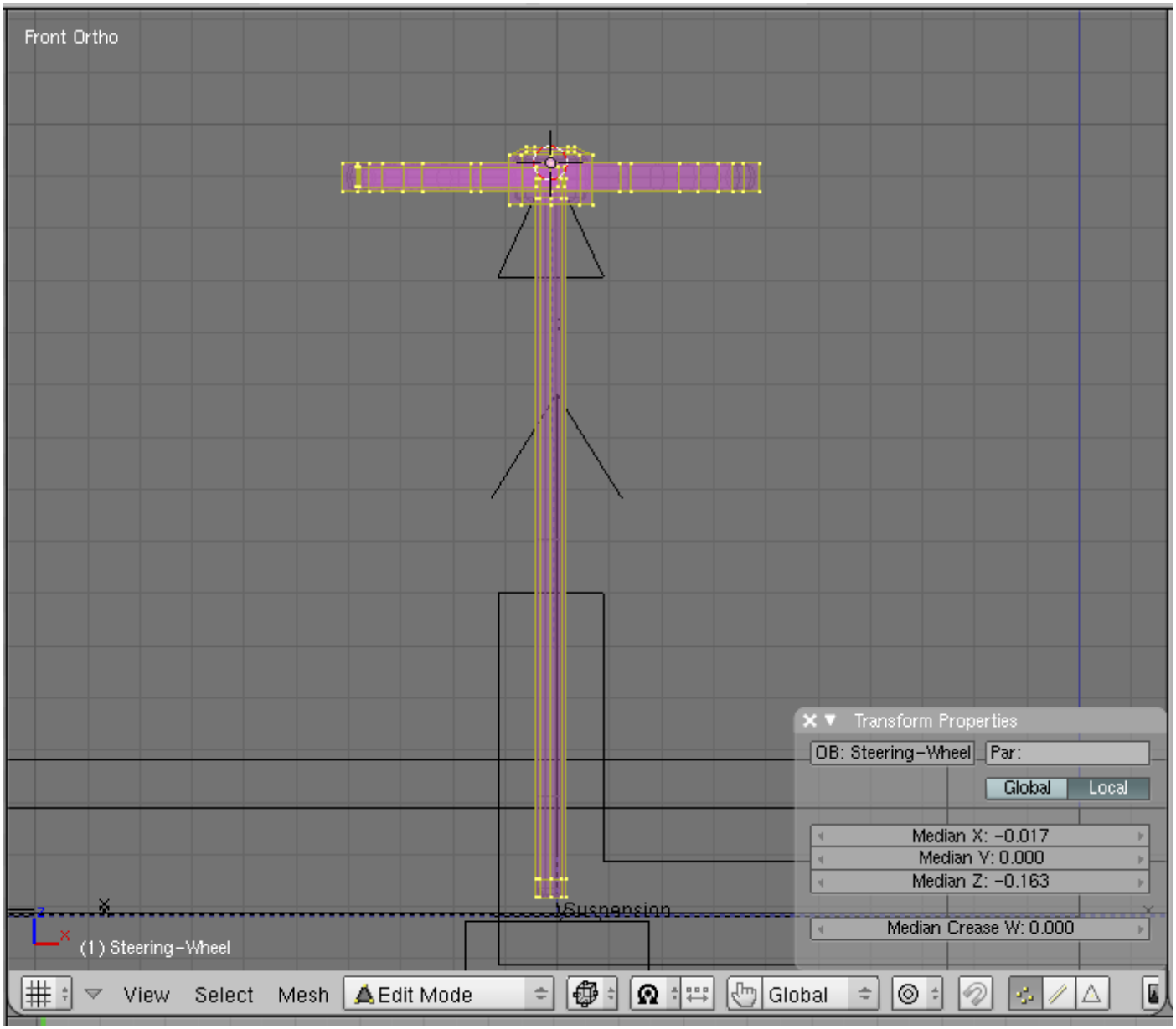
#063

- Add two small cubes, scale them in X and Y direction to create the branches.

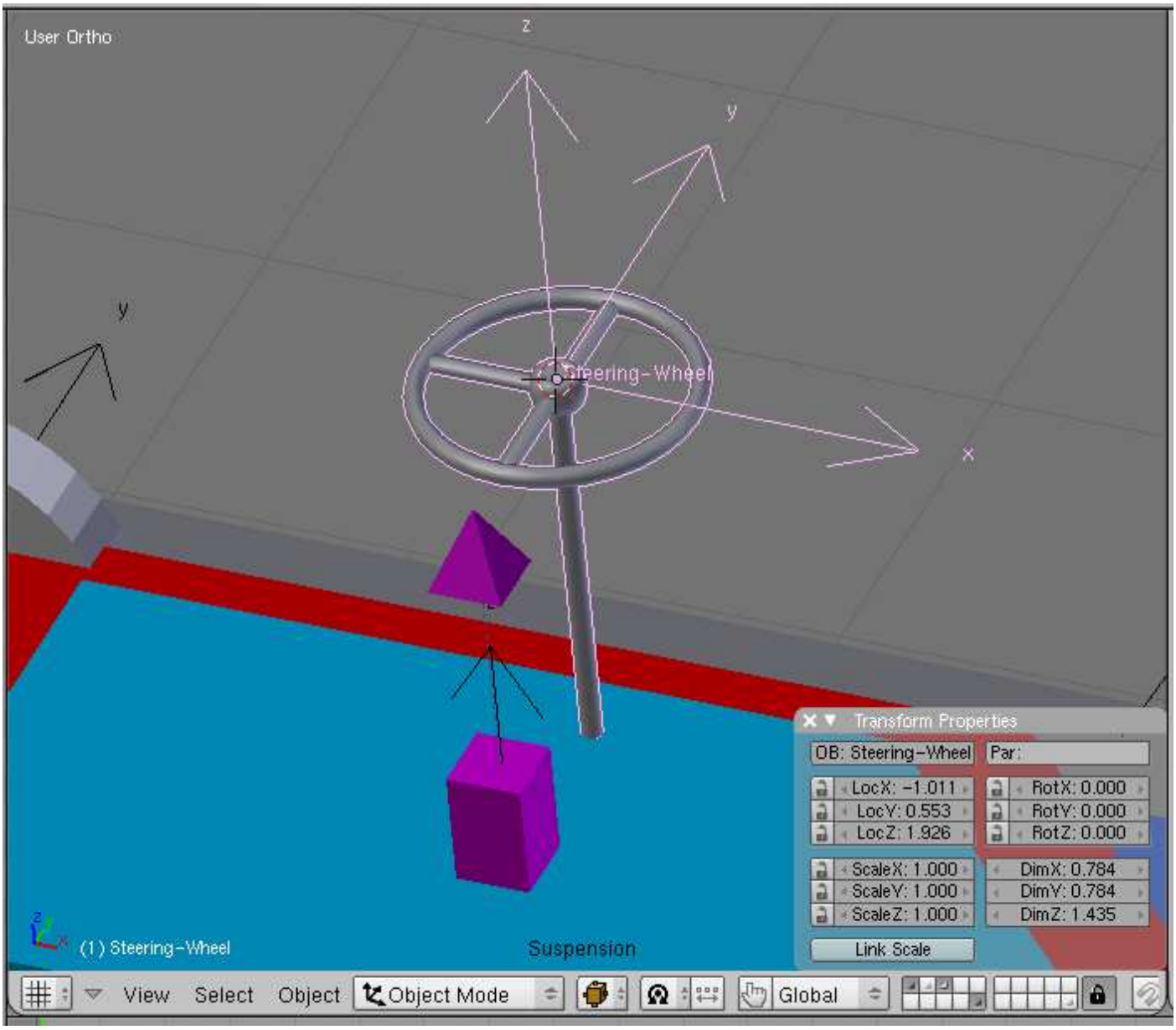


#064

- Add a cylinder, extrude and scale it to add the axle of the Steering-Wheel. When it is finished, you should have your Steering-Wheel with its Z axis vertical.

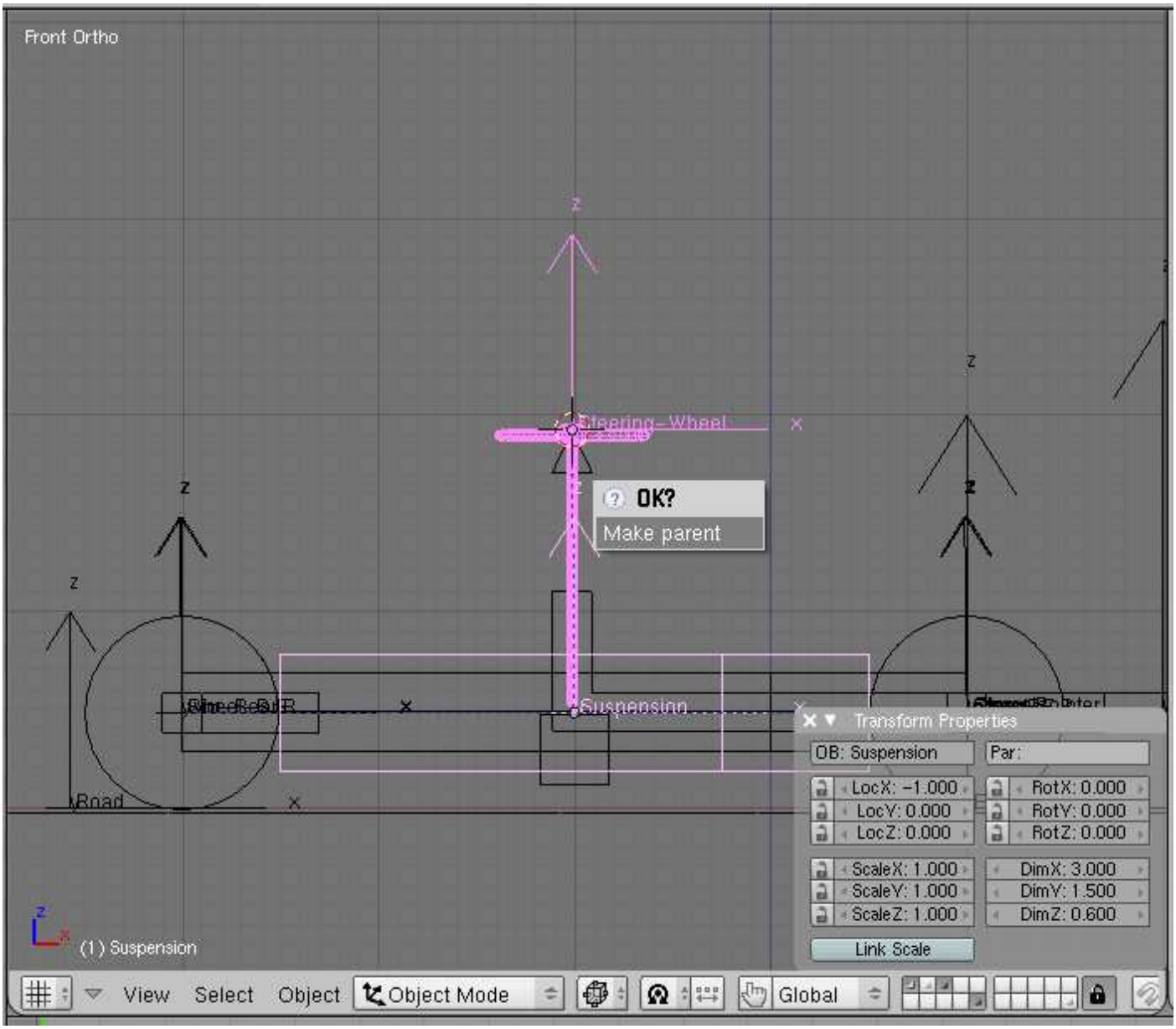


#065


User Ortho
z
y
y
x
Steering-Wheel
Transform Properties
OB: Steering-Wheel
Par:
LocX: -1.011
LocY: 0.553
LocZ: 1.926
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
DimX: 0.784
DimY: 0.784
DimZ: 1.435
Link Scale
(1) Steering-Wheel
Suspension
View
Select
Object
Object Mode
Global

#066

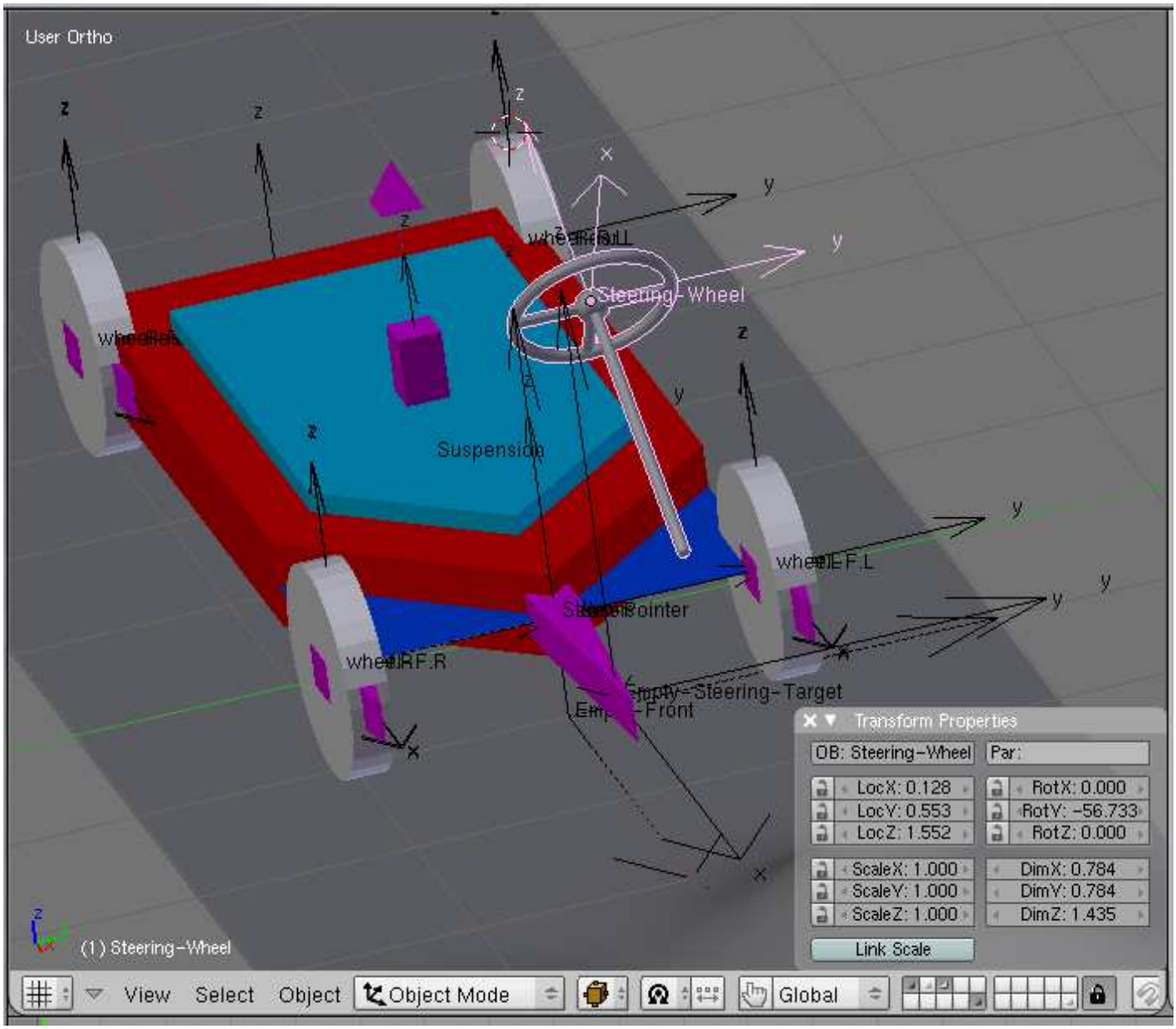
- The Steering-Wheel object will be parented to the body of the car. The body of the car will be for itself parented to the Suspension. Animating directly the body of the car you will allow fine secondary animation.



#067

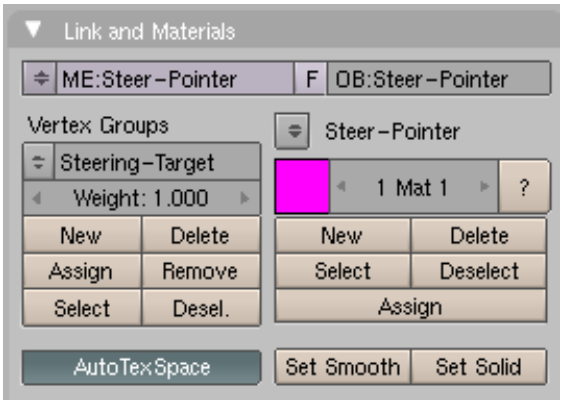
- In this example, I will not add yet the body of the car, to allow looking inside the setup, so we will parent the Steering-Wheel to the Suspension, instead of the car body.

- Now, we are going to give a slope to the Steering-Wheel axle, to put it in its usable position.

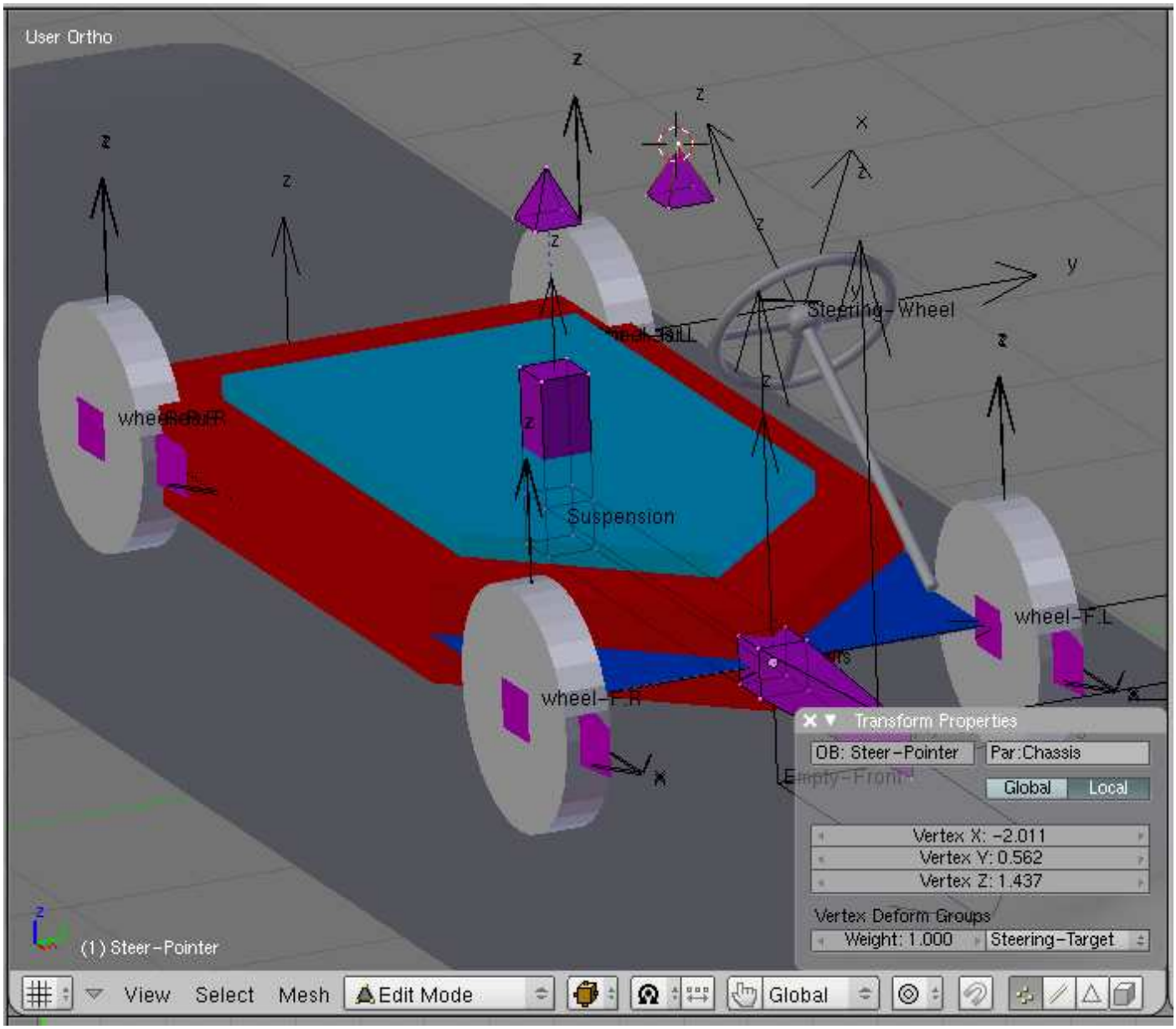


#068

- Create a new Vertex Group aligned on the direction of the Steering-Wheel axle, or duplicate the small pyramidal Vertex Group including the Vertex used as target for the Suspension constraint, and remove its vertices from the original group it belongs to. Select the Vertex at the tip of the pyramid, and add it to a new vertex Group named "Steering-Target".

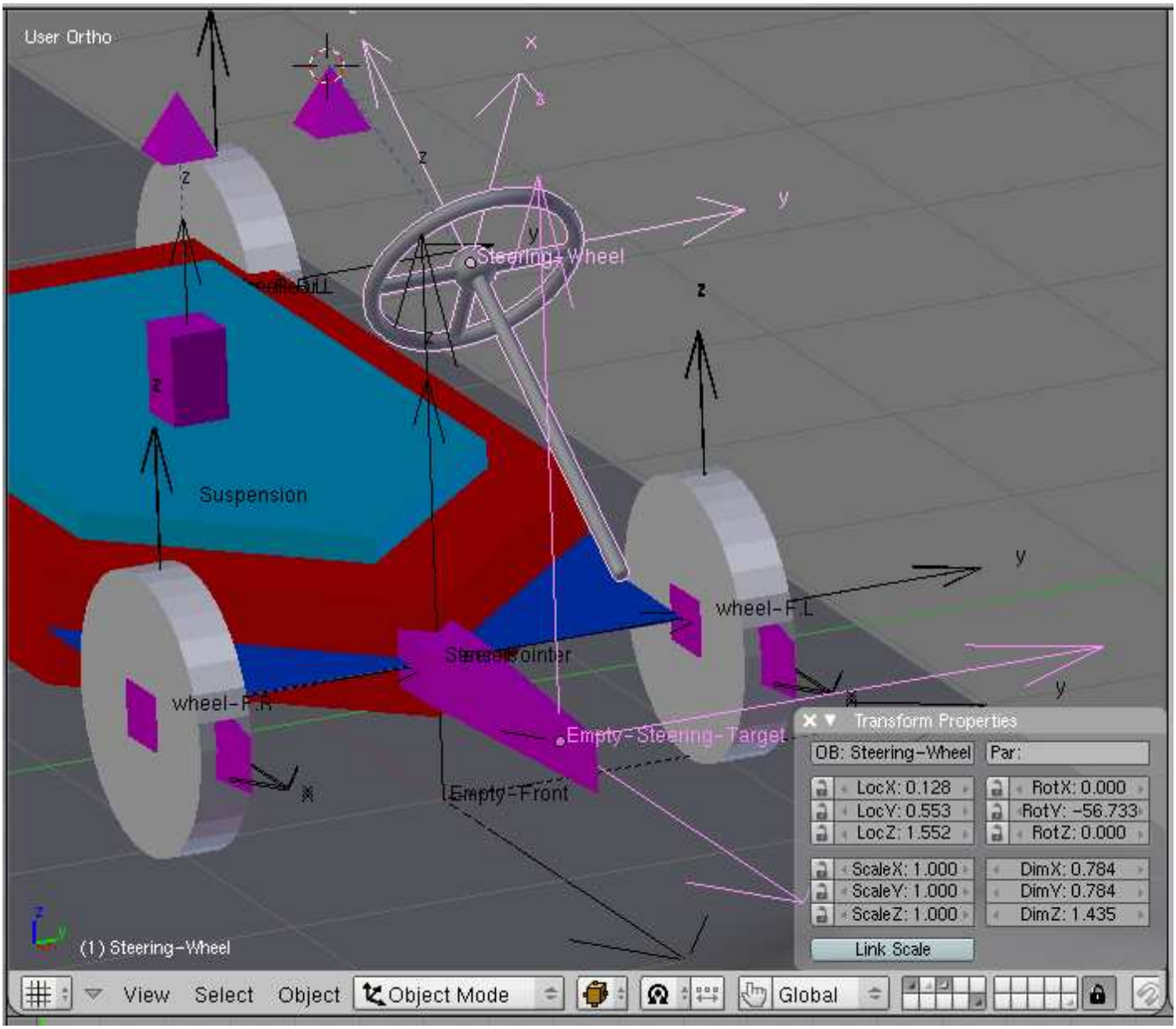


#068B

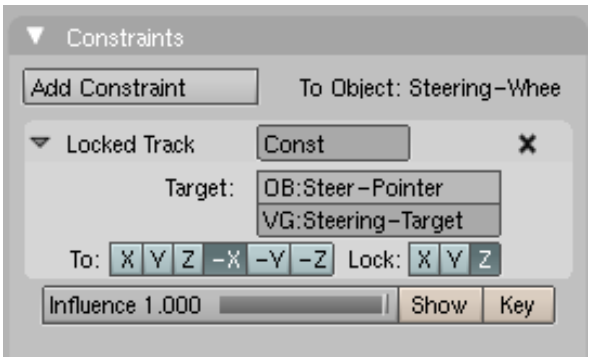


#069

- Select the Steering-Wheel and add a Locked Track constraint. The Steering-Wheel will be locked on its Z axle (local Z axis). Its -X axis7 will track the Steering-Target Vertex group of the Steer-Pointer object.

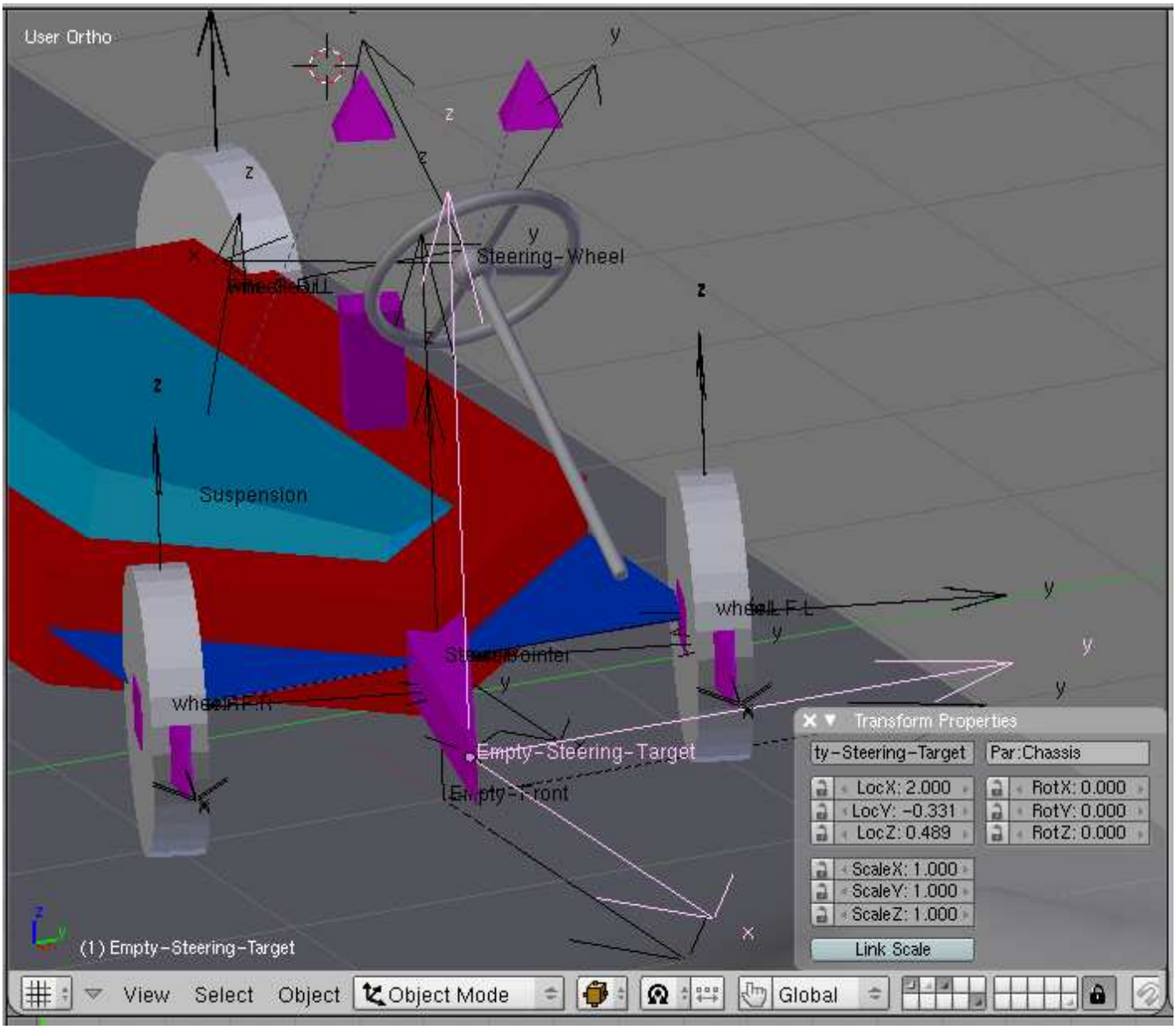


#070



#070B

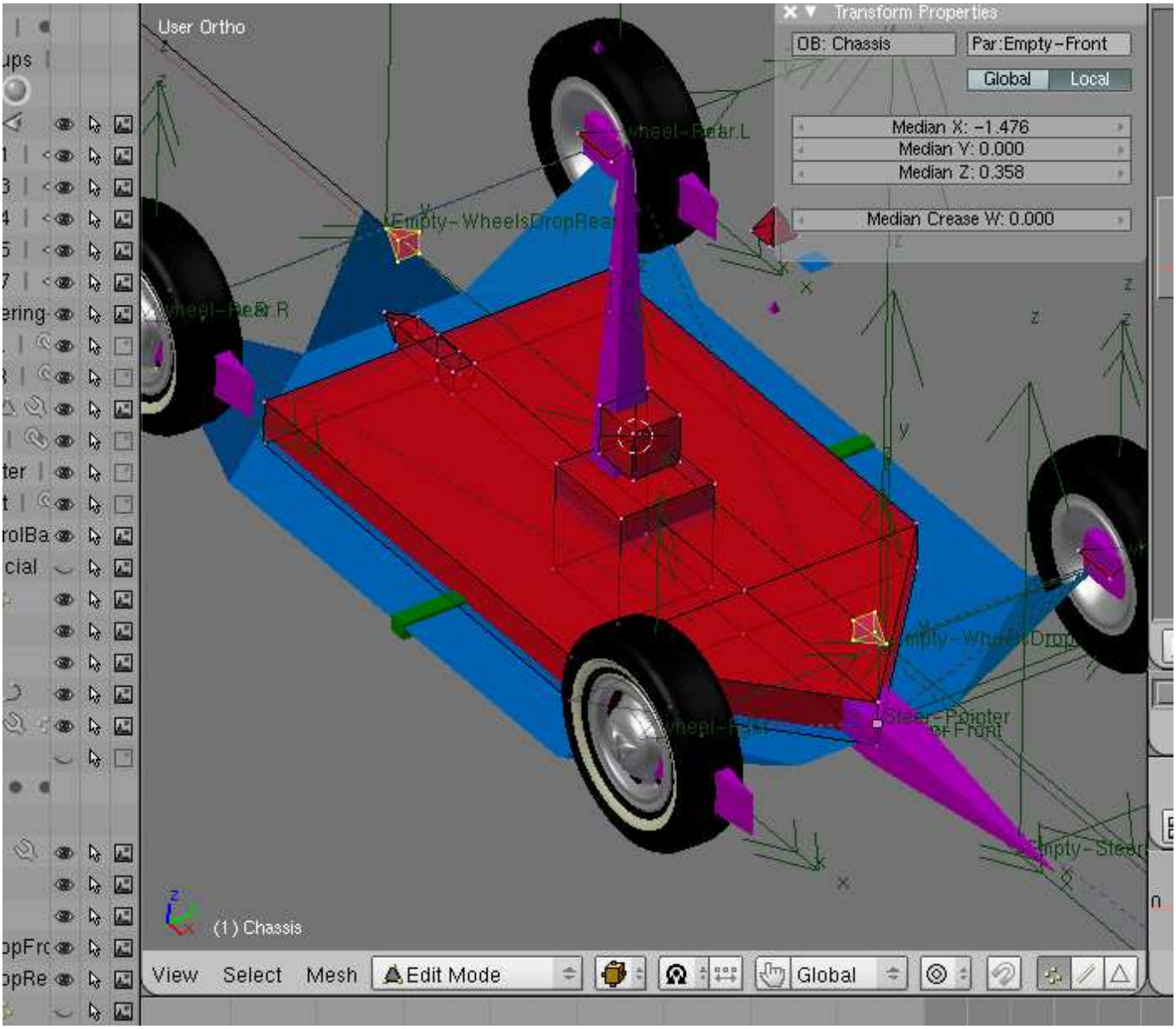
By moving the Empty-Steering-Target in Y direction, you will see the effect of this new constraint on the Steering-Wheel.



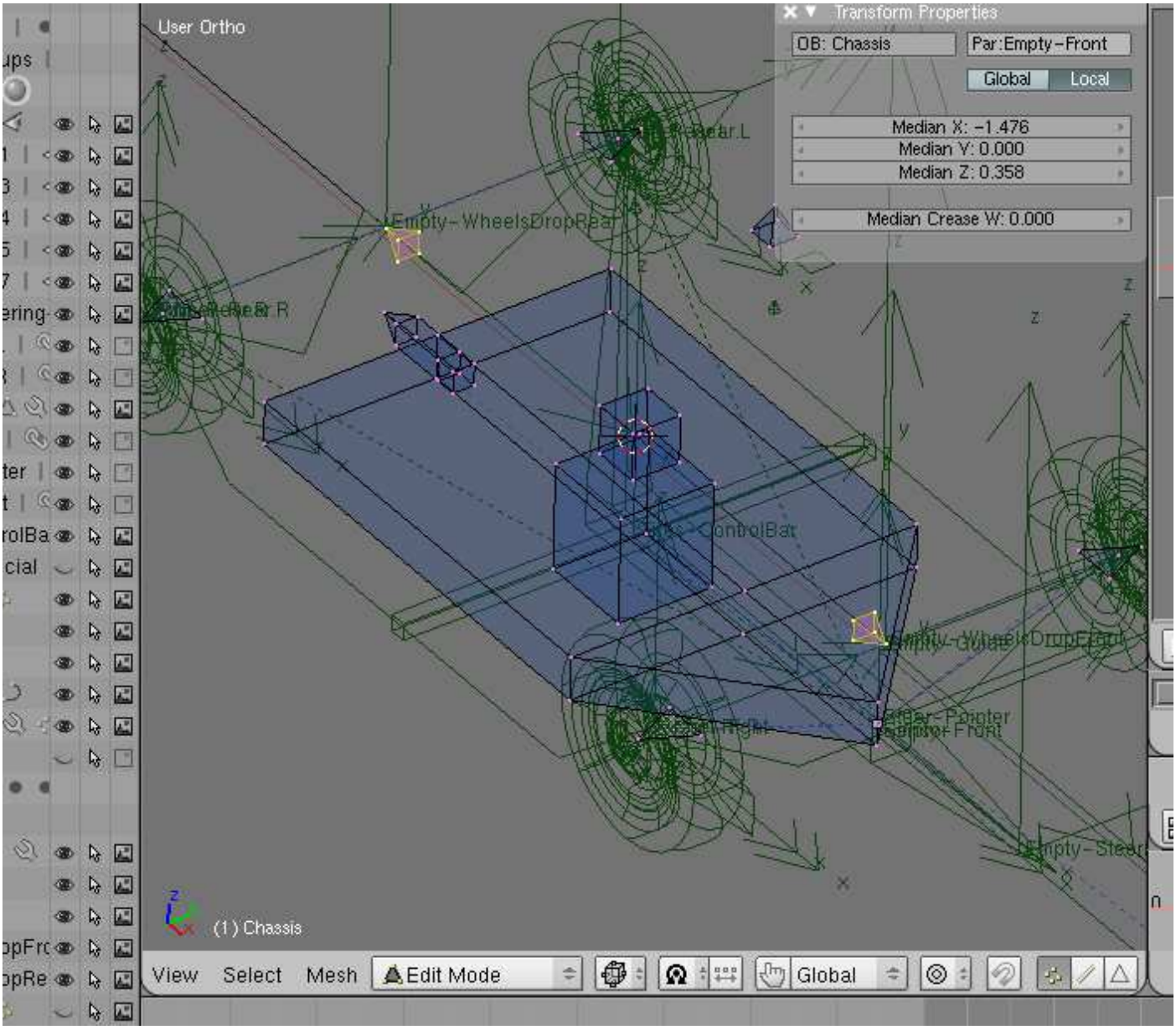
#071

Step 9 : Improving the Geometry of the wheels rig.

In some circumstances, depending mainly on whether you left the Chassis at the ground level or not, you may notice some deformations or abnormal location of the wheels on a very bumpy terrain. This can be corrected by adding to the wheels, or to the wheels supports a Limit Distance constraint, using as references the Front and Rear Vertex Groups previously created in the Chassis object, and named "Nose" and "Tail", or new Vertex Groups located at the level of the wheels axles (if you left the body of the car at ground level).



#072


User Ortho
Transform Properties
OB: Chassis
Par:Empty-Front
Global
Local
Median X: -1.476
Median Y: 0.000
Median Z: 0.358
Median Crease W: 0.000
Empty-WheelsDropRear
ControlBar
(1) Chassis
View
Select
Mesh
Edit Mode
Global

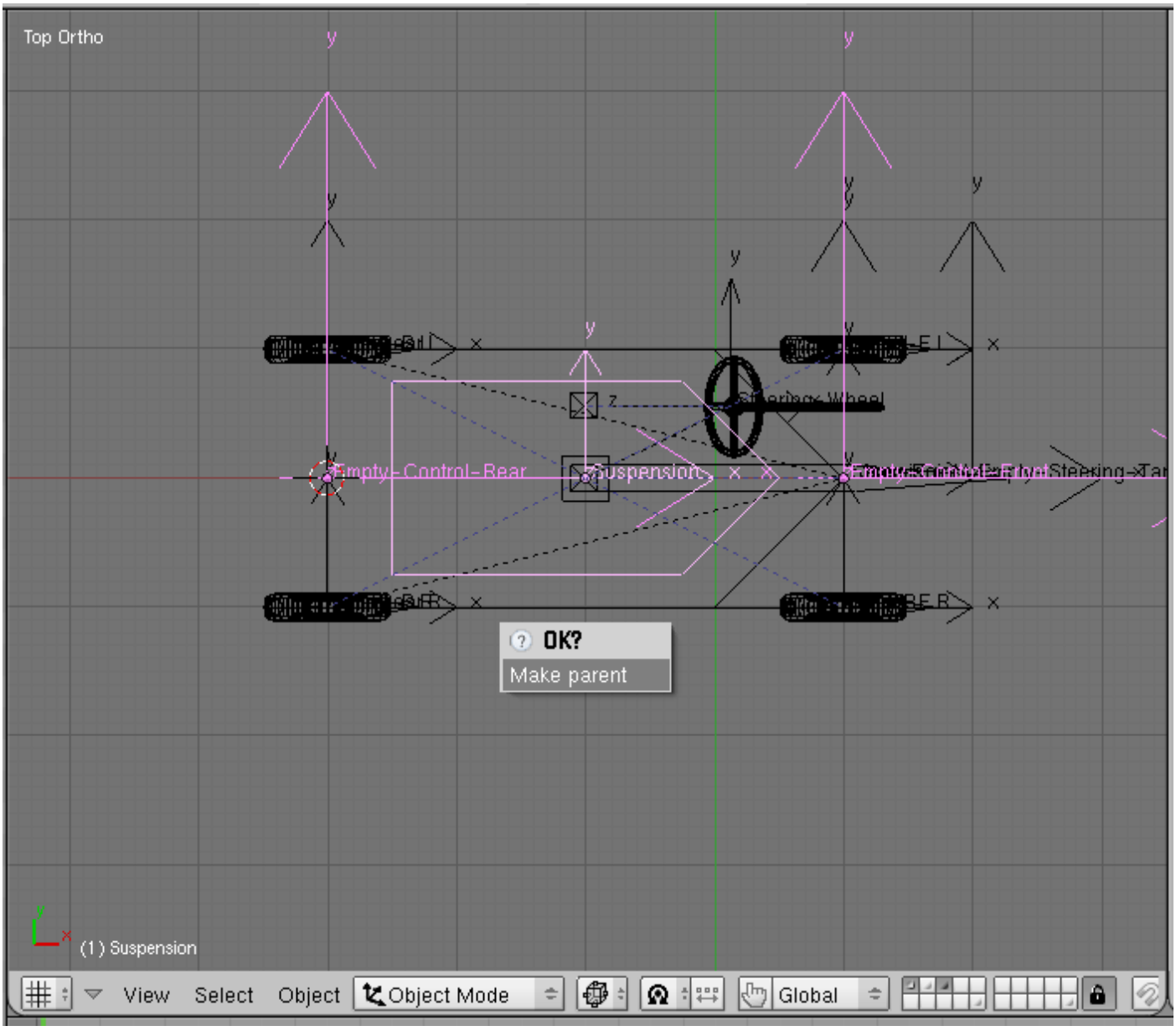
#073

Step 10 : Improving the suspension by adding controls for secondary animation.

In some situations, espescially when the car takes off, like when jumping after a bump or over a broken bridge, we may need more freedom and more control on the wheels, to simulate the effect of the gravity on them when they are no more supported by the ground.

This can be easily done by adding two Empties that we will use as handles, to ad secondary animation on the wheels. In fact, we will not directly act on the wheels themselves, but rather on their supports. I mean on the objects Steer.L, Steer.R, Rim-Rear.L and Rim-Rear.R.

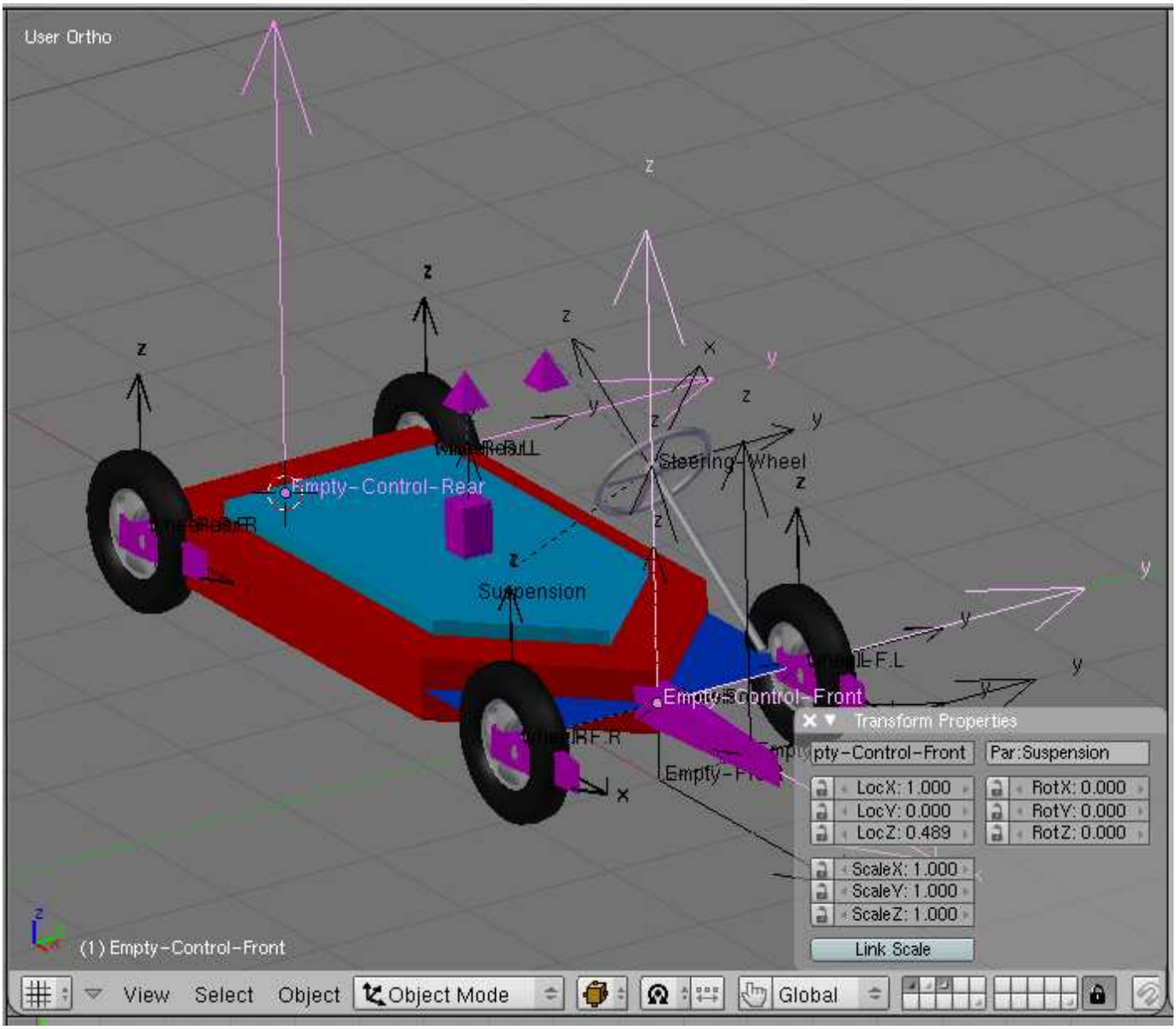
- At the location of the Chassis center, add an Empty. Name it Empty-Control-Front. Add an other Empty at the location of the Tail Vertex of the Chassis. It is exactly in the middle between the two rear wheels. Select the two new empties, and parent them to the suspension object.



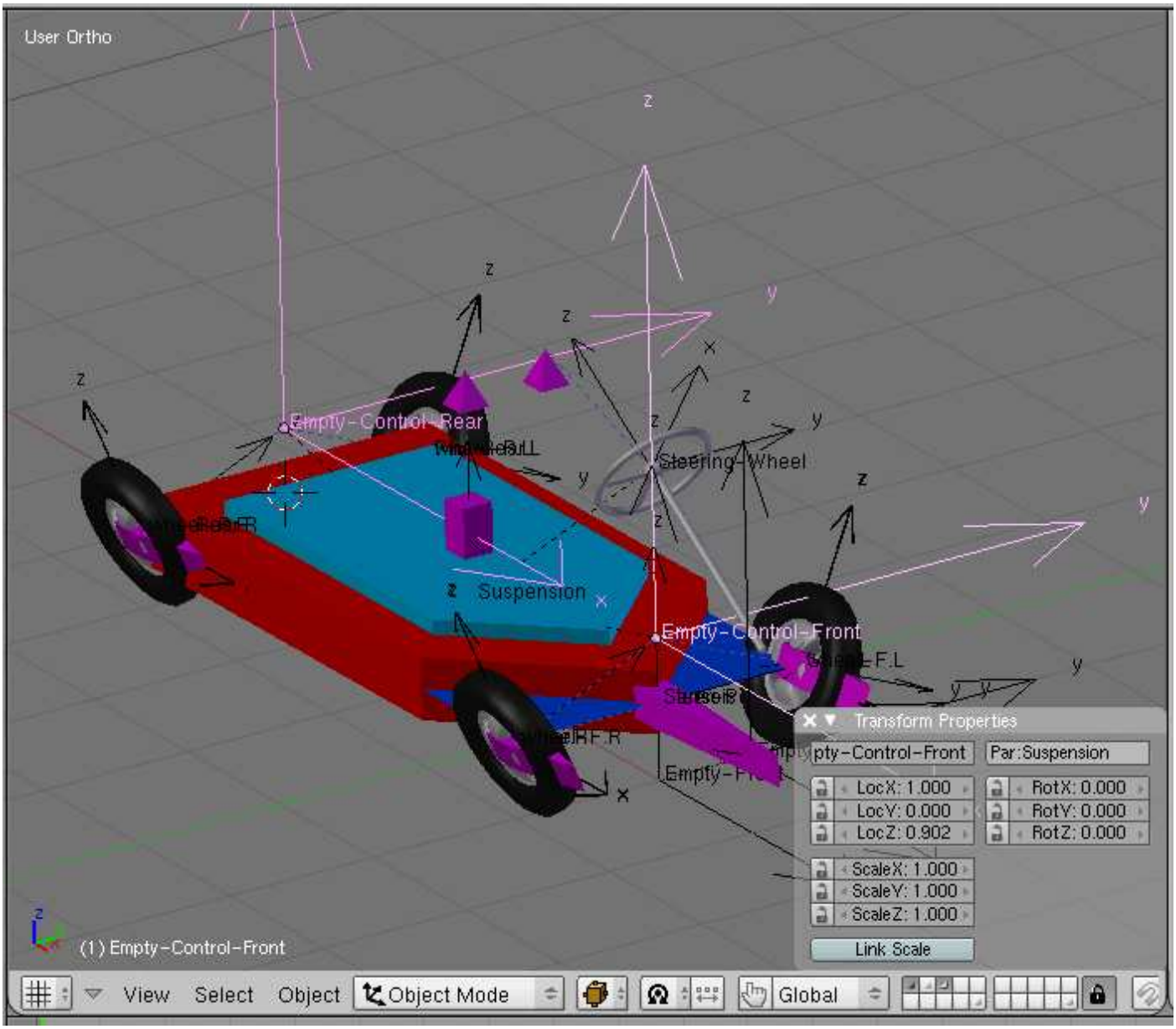
#074

- Now, select one by one the four wheels supports Steer.L, Steer.R, Rim-Rear.L and Rim-Rear.R. And add to each of them a Locked Track constraint. Each wheel support will be locked on its local X axis, and the two front object will track the Empty-Control-Front, while the rear objects will track the Empty-Control-Rear.

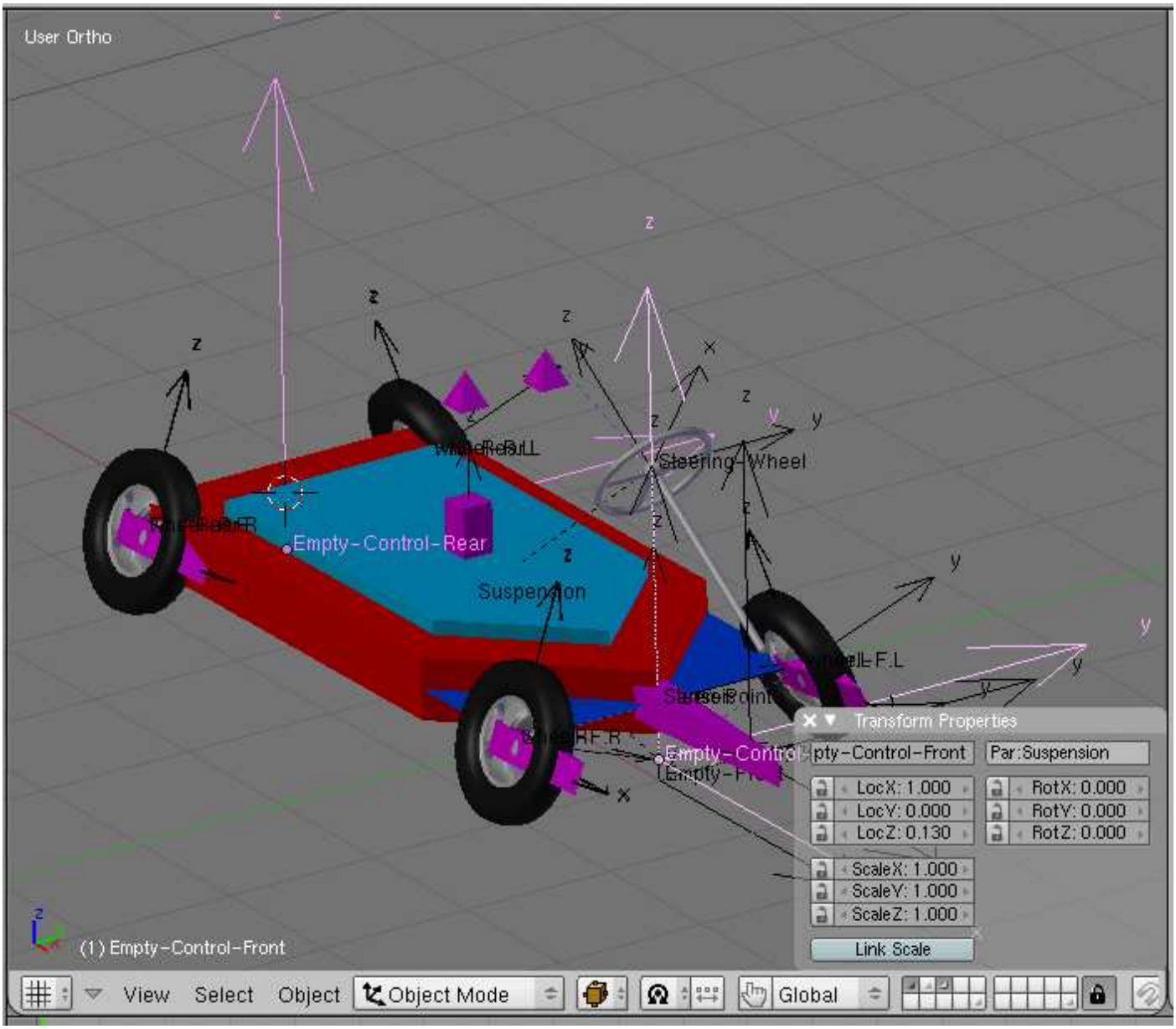
- The two empties will not only allow to add secondary animation by hand : as they are parented to the Suspension, you will notice that the slope of the Suspension in the direction of the trajectory will have an influence on the transversal slope of the wheels. Weight displacement will be automatically simulated and the angle of the suspension around its local Y axis will modify the transversal slope of the wheels.



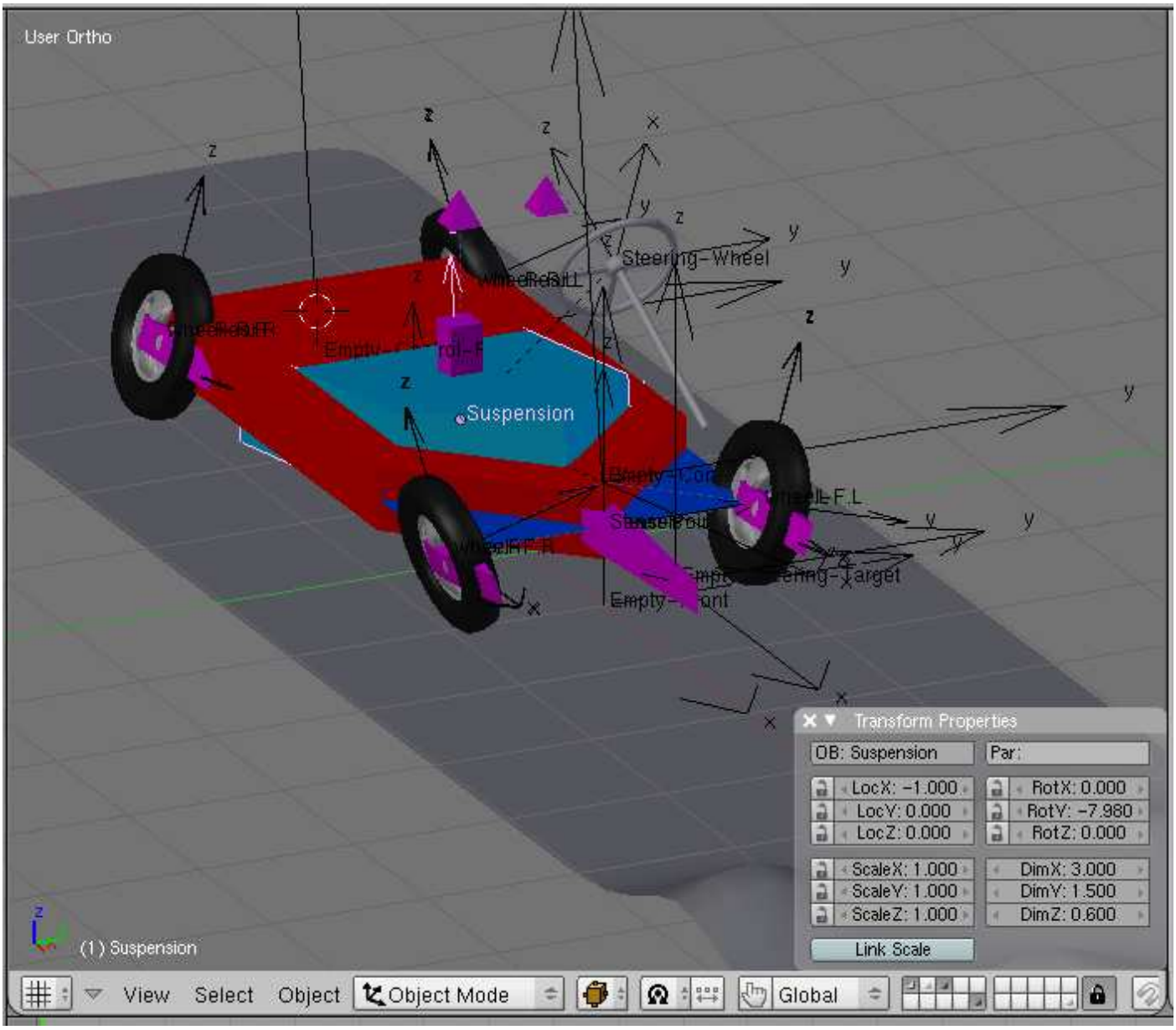
#075


User Ortho
Empty-Control-Rear
Steering-Wheel
Suspension
Empty-Control-Front
Transform Properties
pty-Control-Front
Par:Suspension
LocX: 1.000
LocY: 0.000
LocZ: 0.902
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
Link Scale
(1) Empty-Control-Front
View
Select
Object
Object Mode
Global

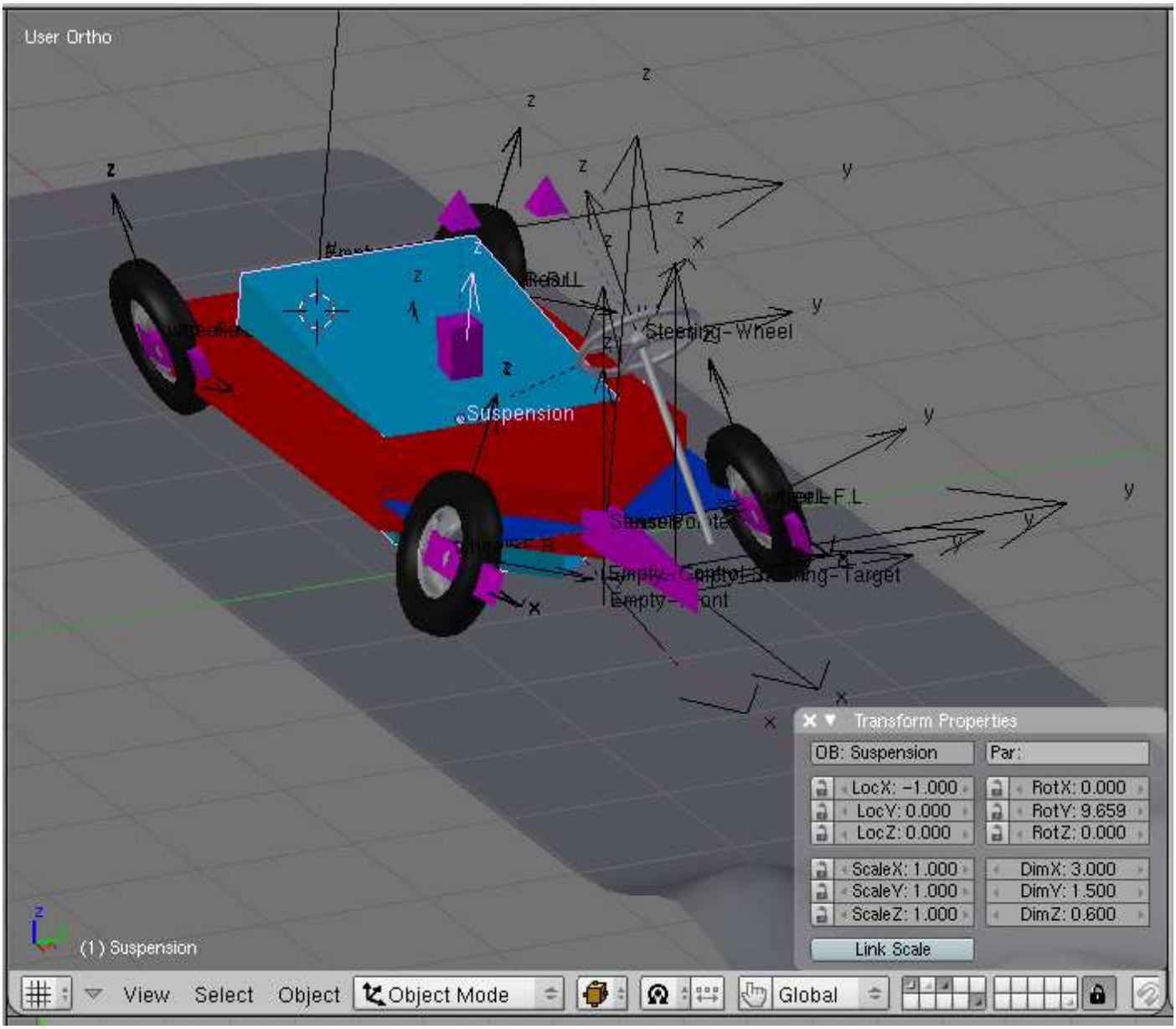
#076


User Ortho
Steering_Wheel
Empty-Control-Rear
Suspension
Transform Properties
Par:Suspension
LocX: 1.000
LocY: 0.000
LocZ: 0.130
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
Link Scale
(1) Empty-Control-Front
View
Select
Object
Object Mode
Global

#077


User Ortho
Steering-Wheel
Suspension
Transform Properties
OB: Suspension
Par:
LocX: -1.000
LocY: 0.000
LocZ: 0.000
RotX: 0.000
RotY: -7.980
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
DimX: 3.000
DimY: 1.500
DimZ: 0.600
Link Scale
(1) Suspension
View
Select
Object
Object Mode
Global

#078

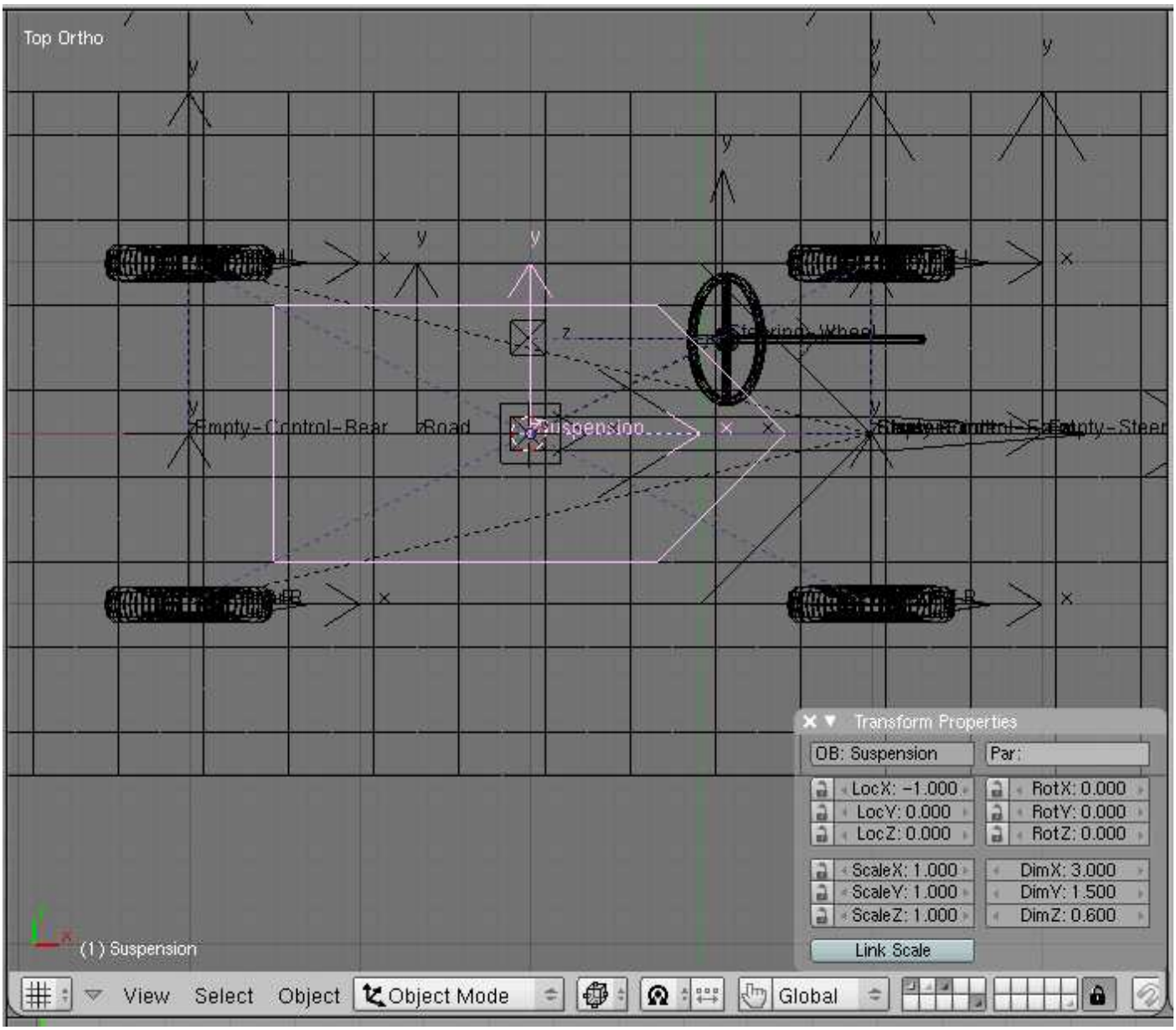

User Ortho
Steering-Wheel
Suspension
Transform Properties
OB: Suspension
Par:
LocX: -1.000
LocY: 0.000
LocZ: 0.000
RotX: 0.000
RotY: 9.659
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
DimX: 3.000
DimY: 1.500
DimZ: 0.600
Link Scale
(1) Suspension
View
Select
Object
Object Mode
Global

#079

Step 11 : Tires deformation.

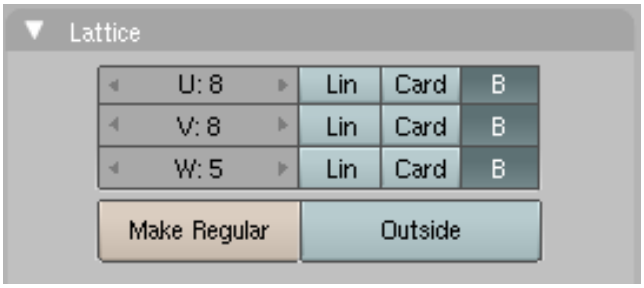
The tires deformation will be done by using a Lattice. In the past, in a former version of this car setup, I used one lattice per wheel. In this new system, only one is necessary.

- Select the Suspension object, and bring the 3D Cursor on its center.

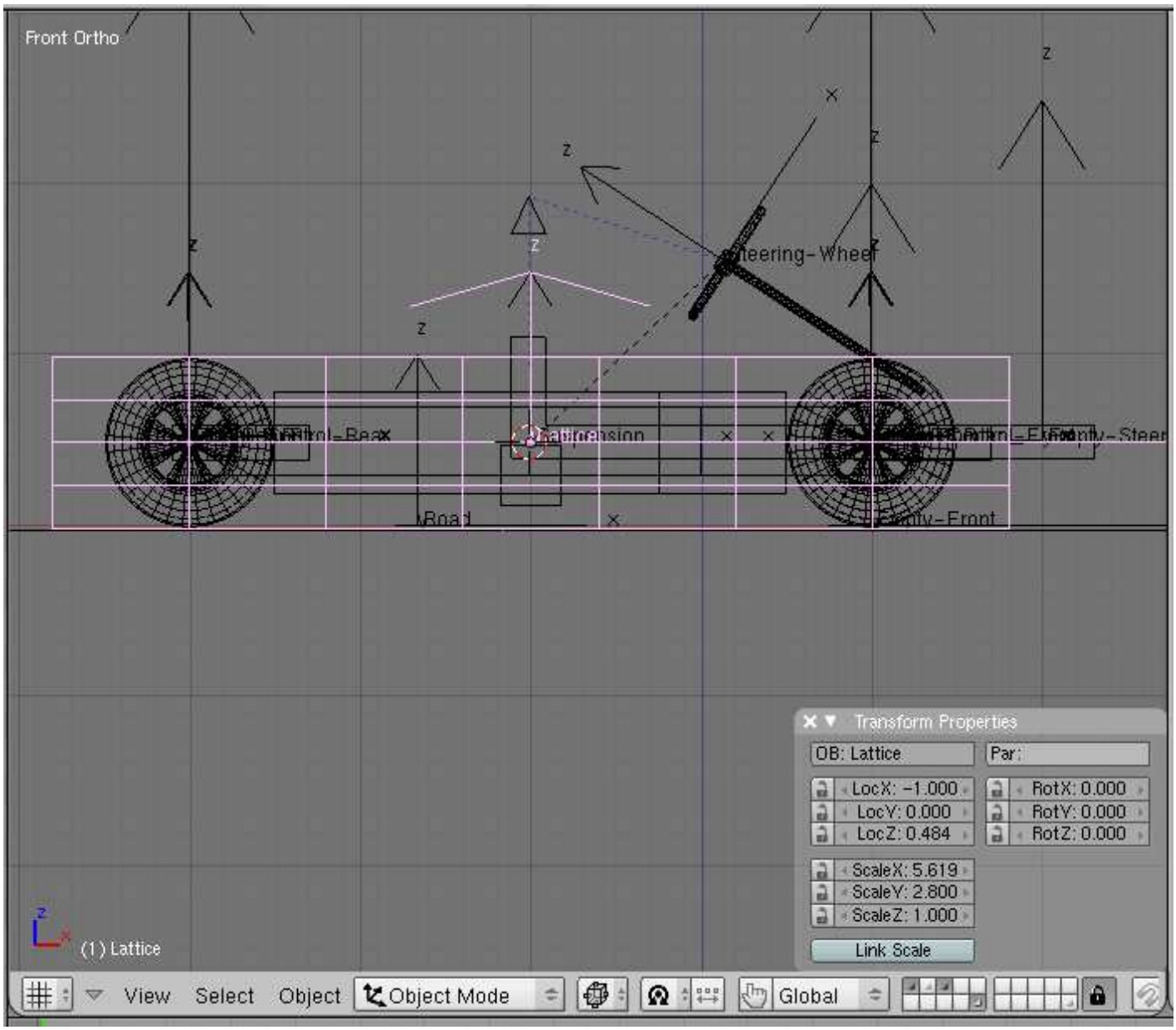


#080

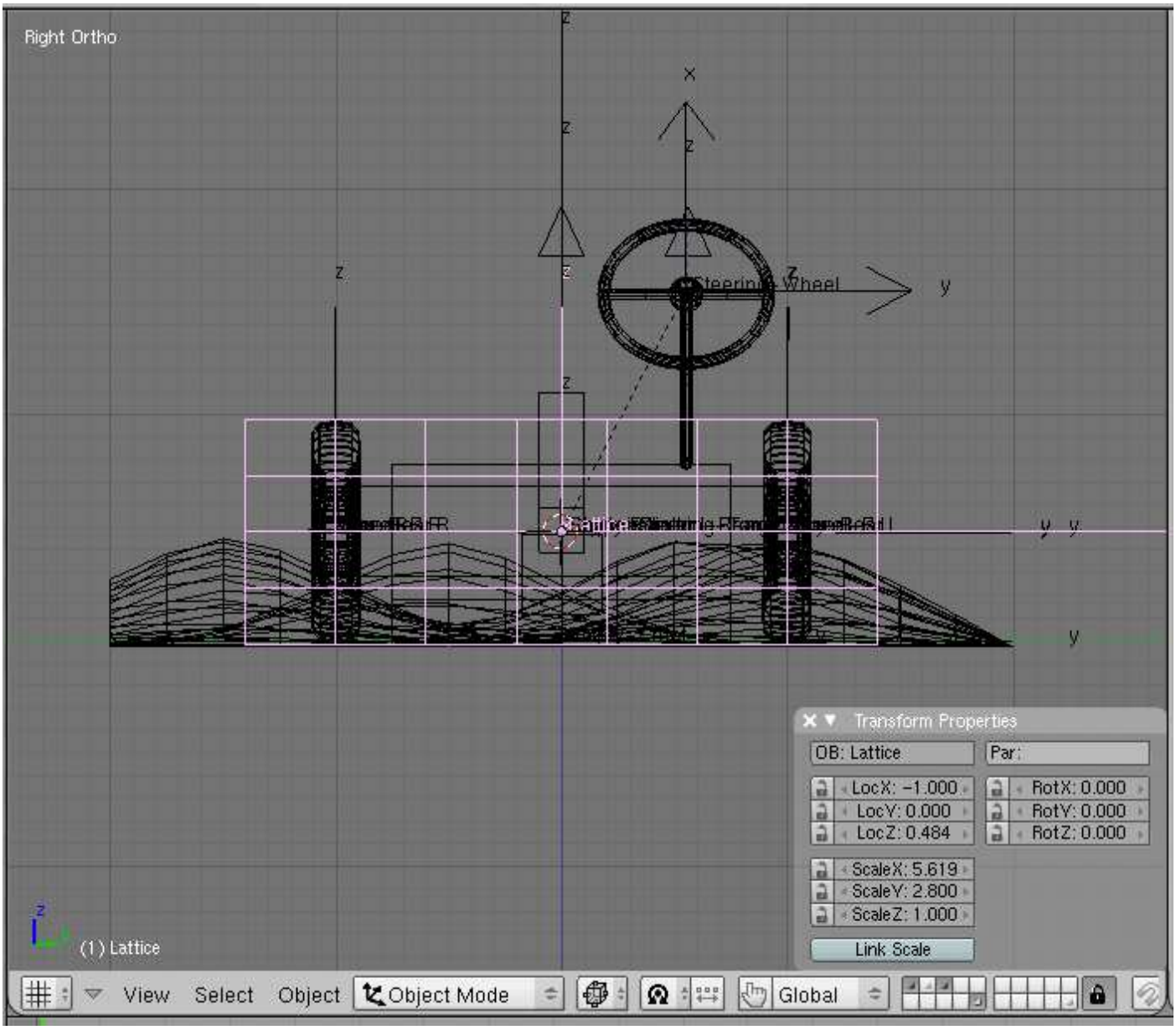
- Add a Lattice. Scale the lattice and divide it o get a grid with lines centered on the four wheels in the different views. U=8,V=8 and W=5 in the example. Note that the directions of the axis can be different. It has no importance.



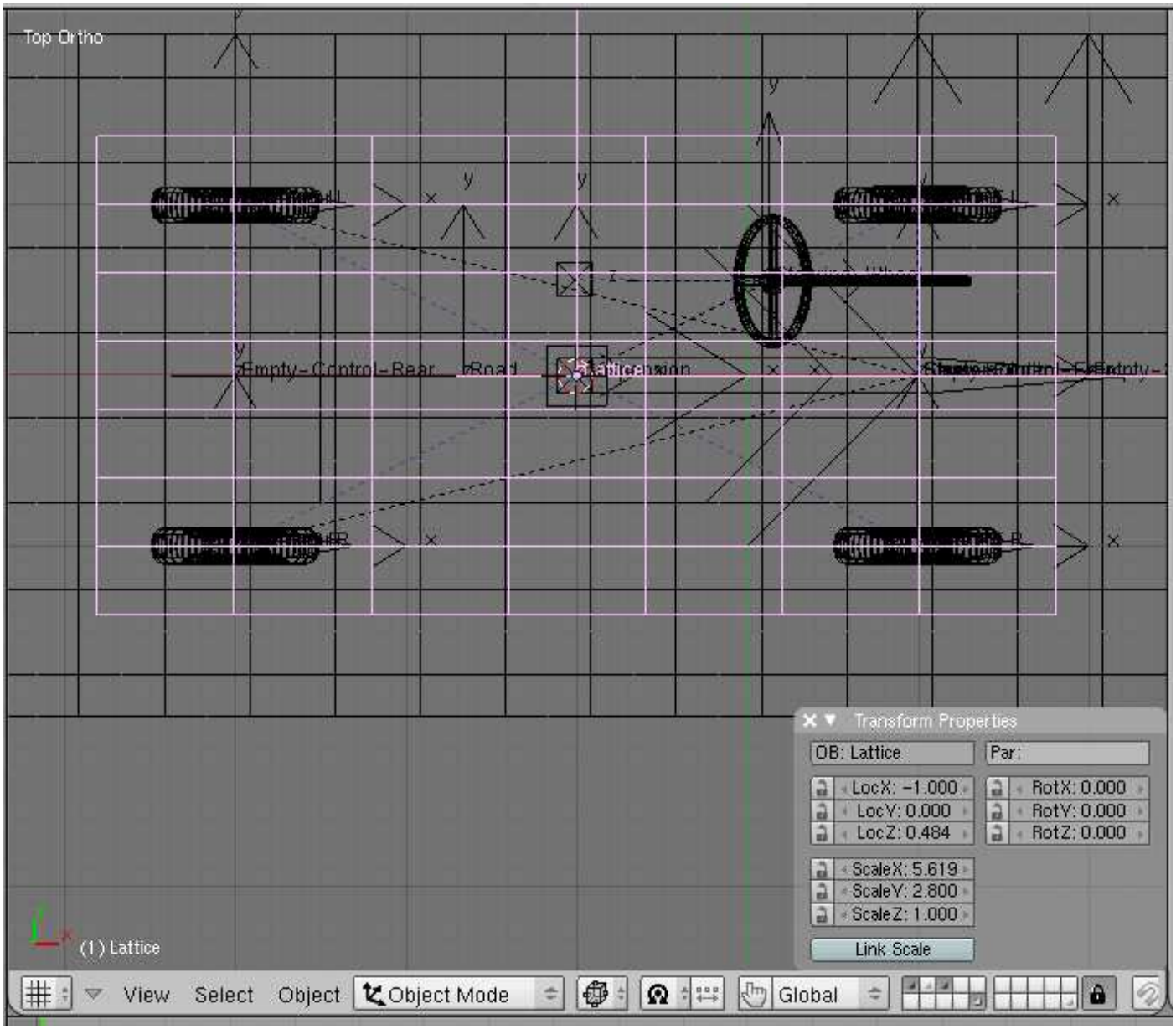
#80B



#081


Right Ortho
Steering Wheel
Transform Properties
OB: Lattice
Par:
LocX: -1.000
LocY: 0.000
LocZ: 0.484
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 5.619
ScaleY: 2.800
ScaleZ: 1.000
Link Scale
(1) Lattice
View
Select
Object
Object Mode
Global

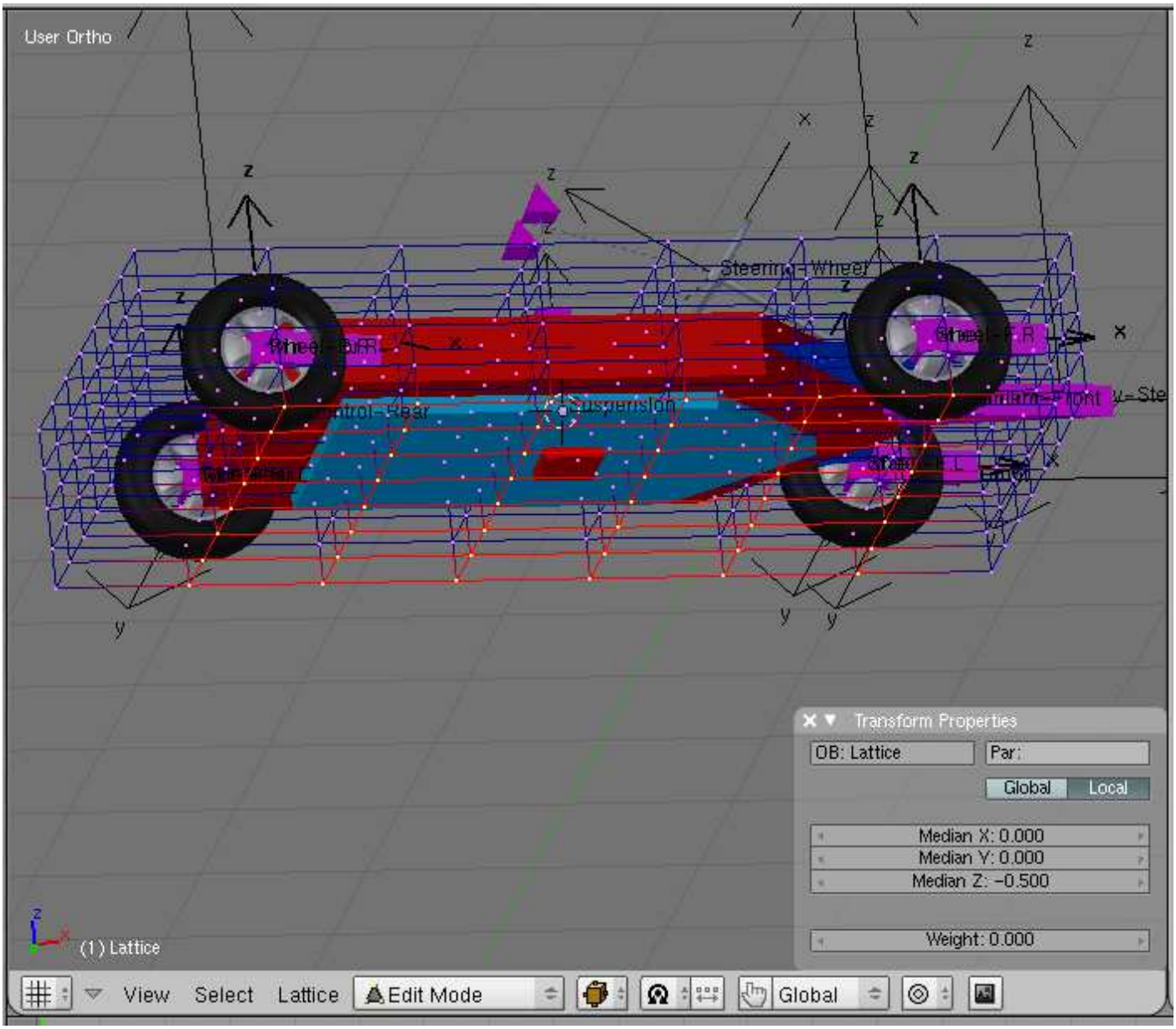
#082


Top Ortho
Empty-Control-Rear
Transform Properties
OB: Lattice
Par:
LocX: -1.000
LocY: 0.000
LocZ: 0.484
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 5.619
ScaleY: 2.800
ScaleZ: 1.000
Link Scale
(1) Lattice
View
Select
Object
Object Mode
Global

#083

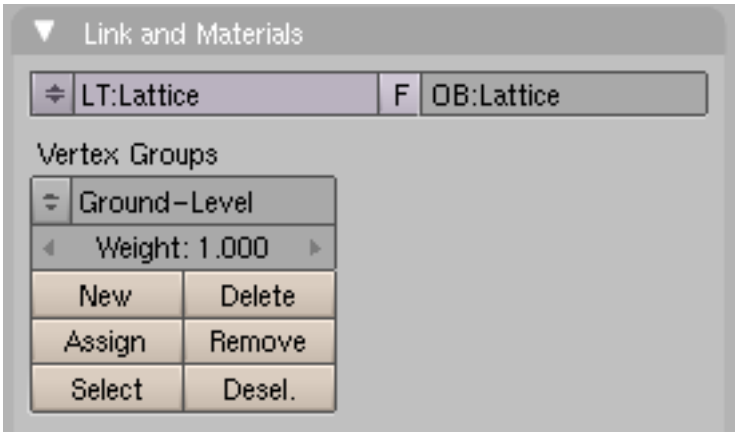
- Parent the Lattice to the Suspension.

- Select the Lattice and in Edit Mode, add it a Vertex Group named for example "Ground-Level".



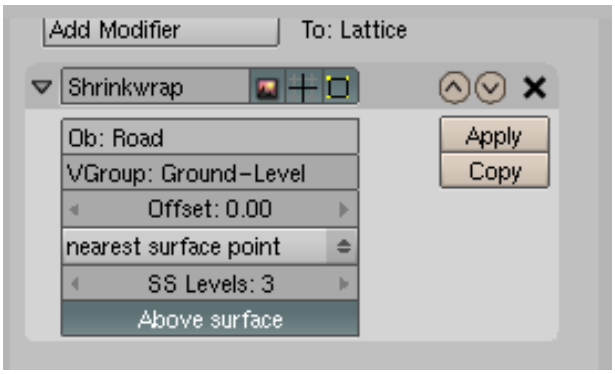
#084

This Group will include all the vertices set at the level of the road, except the most Front and Rear row.



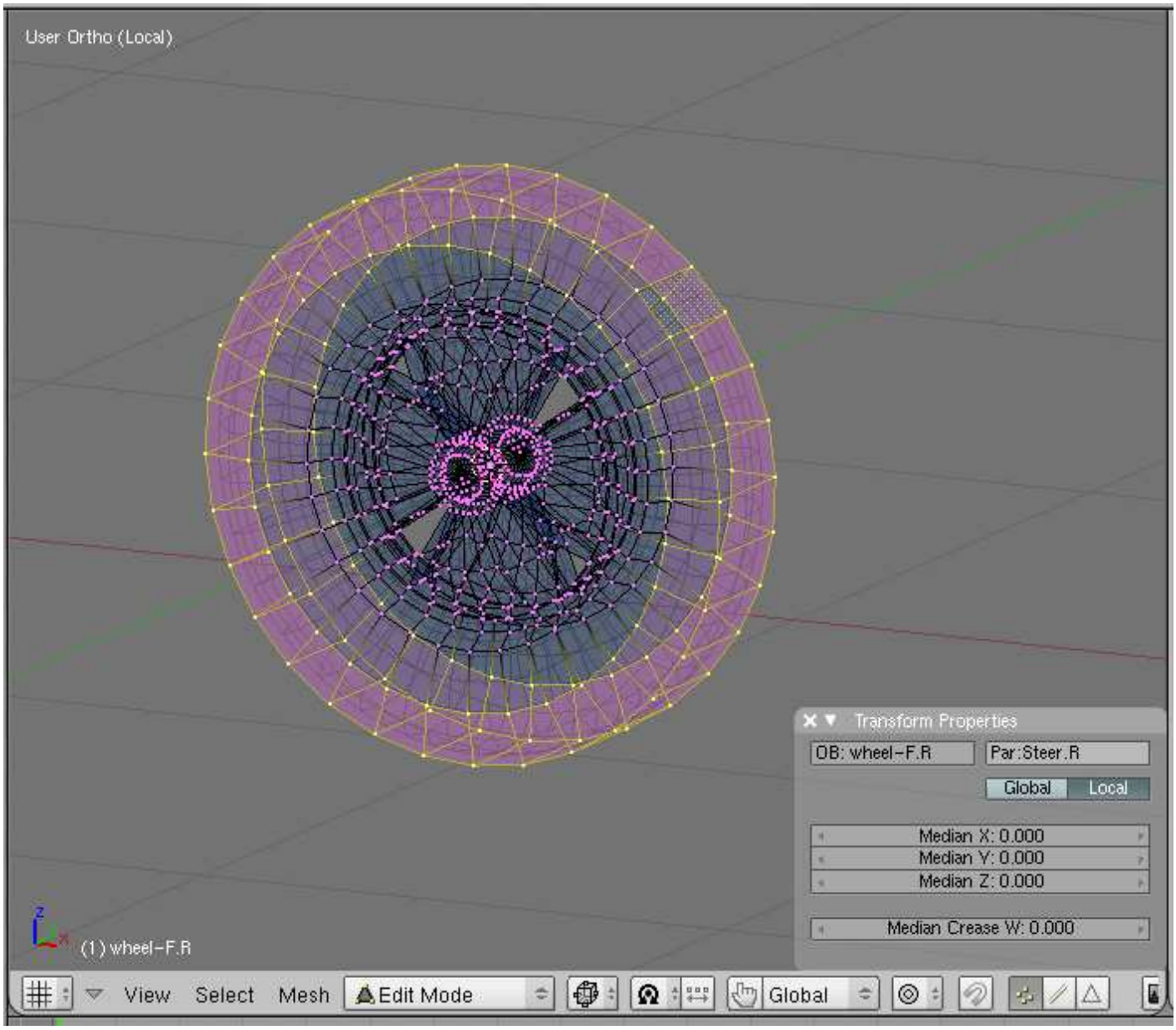
#084B

- Add a Shrinkwrap Modifier to the Lattice, and use the Road as target in the Object field. Type also Ground-Level in the VGroup field, to limit the deformation to the selected points of the grid. Use "nearest surface point" as Mode. Set the SS levels to 3 and enable the "Above surface" button Offset will be kept to 0.00.



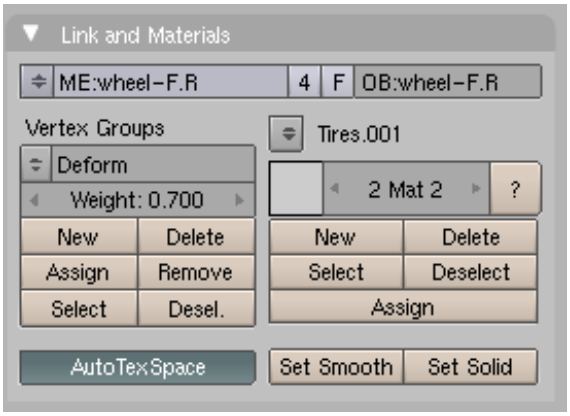
#084C

- Now, we have to take care of the tires themself. We want to limit the deformation to the outer ring of vertices, without affecting the rims of the wheels.

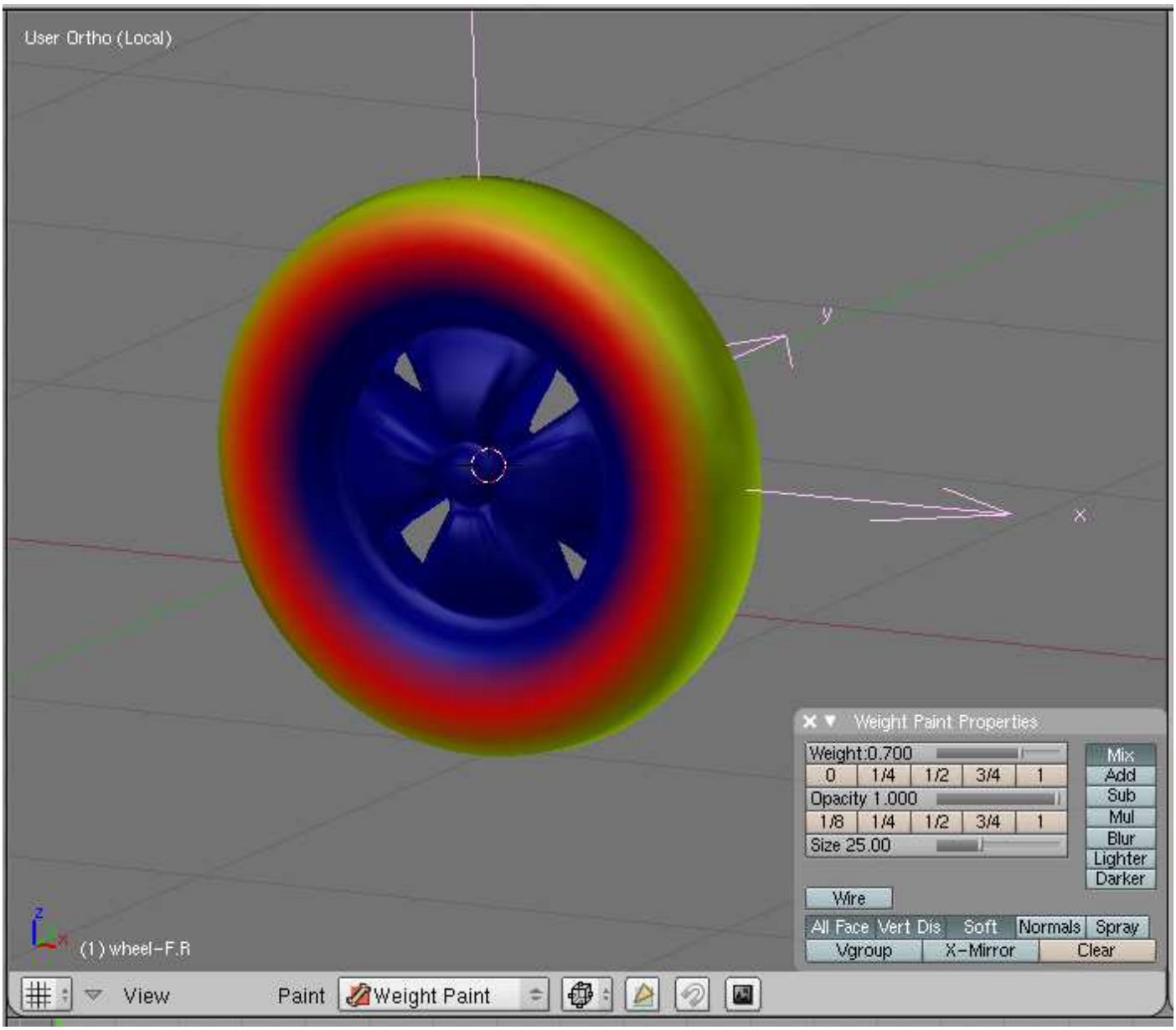


#085

- So, we have to select the outer part of the wheel, and create a Vertex Group named "deform". As the deformation has to be mostly visible on the side of the tire, the inner loops of vertices will be set with a weight of 1.0 while the outer loops will be set with a weight of only 0.7.

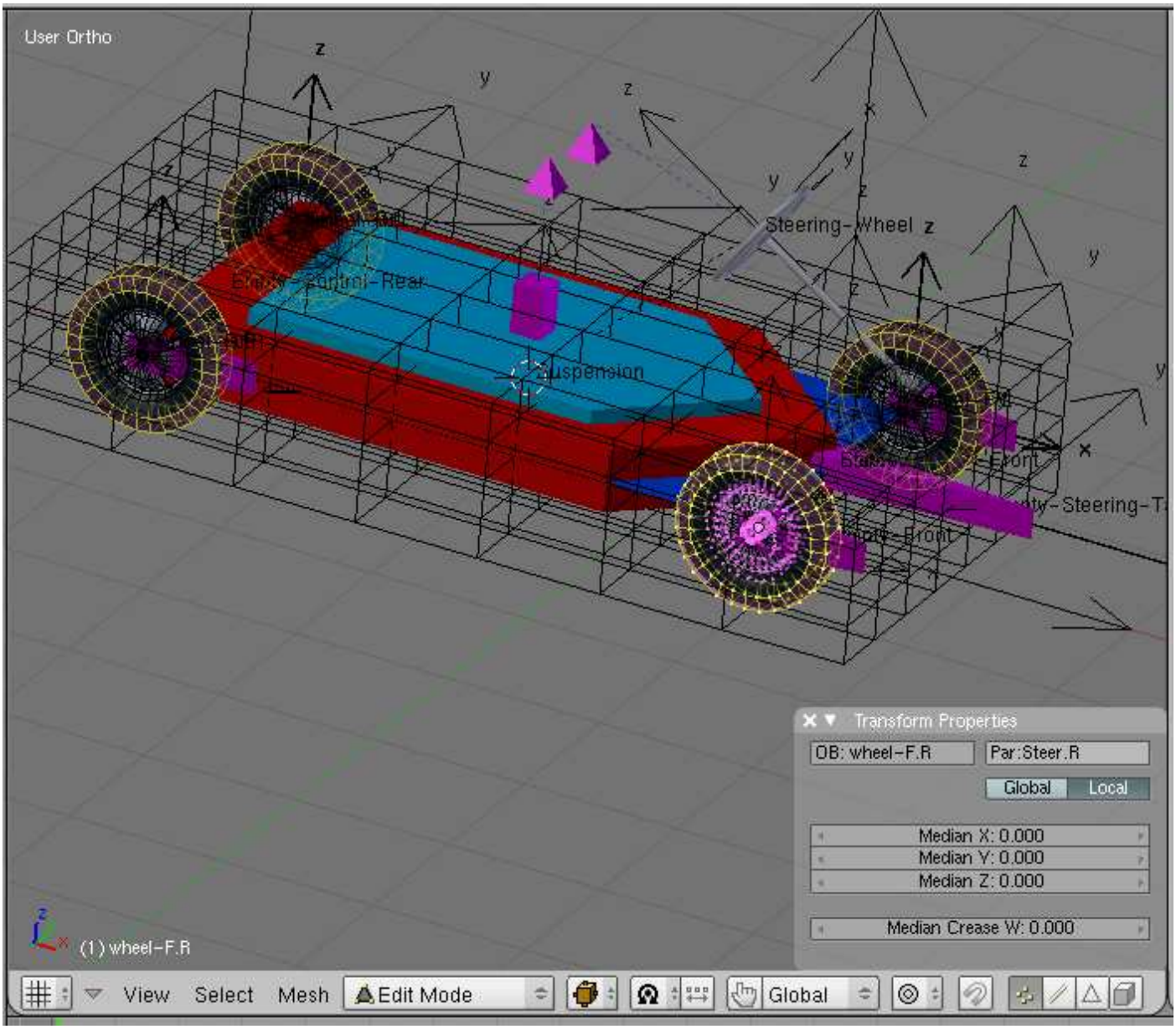


#085B



#086

- You will have to repeat the operation for the four wheels, Or simply duplicate the modified wheel.

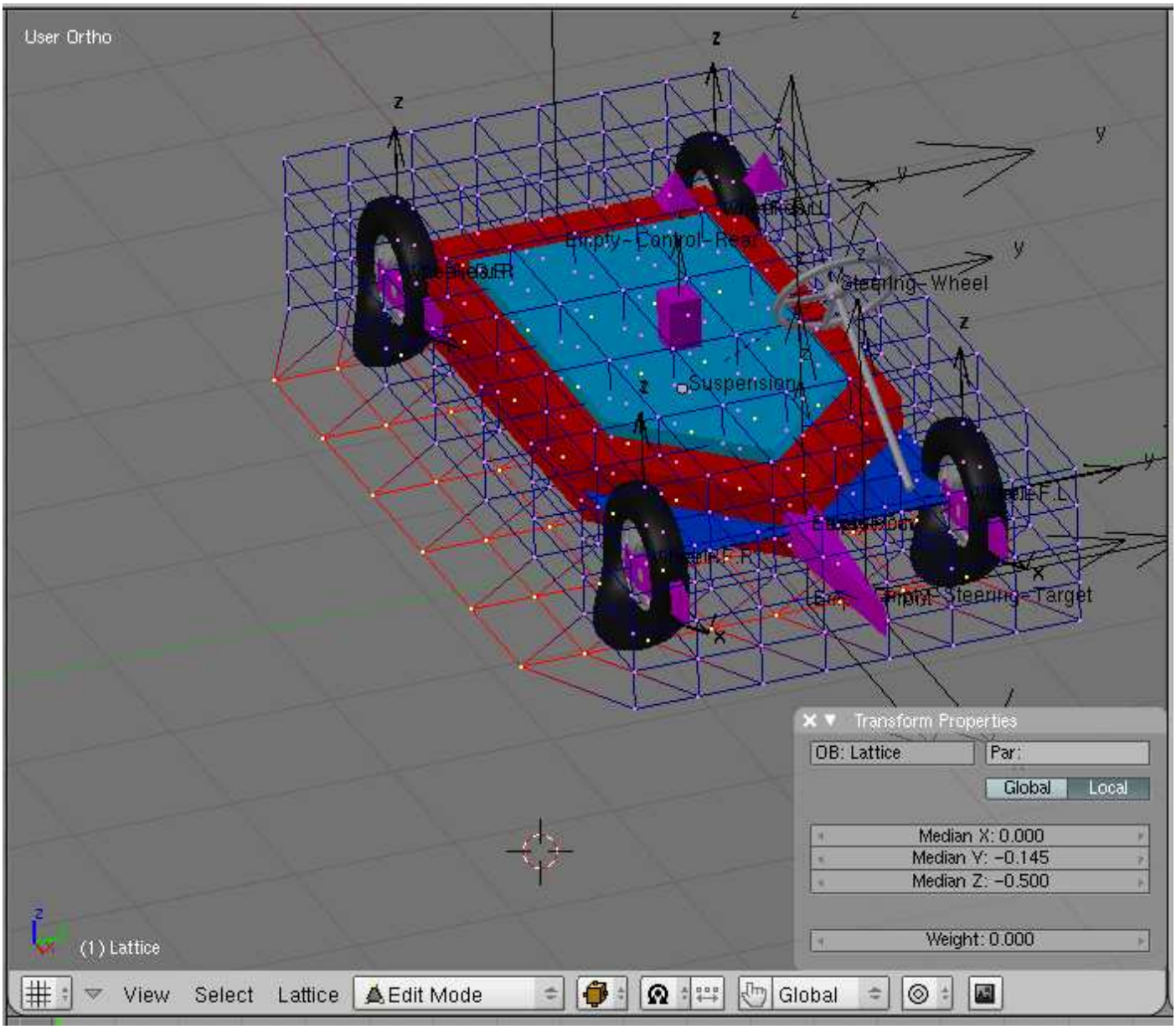


#087

- Now, select your wheels one by one and apply to each of them a Lattice Modifier.

- Fill the Ob field with the name of the Lattice, and limit the deformation of the wheel to the Vertex Group named Deform. In the Modifiers Stack, it is better to put the Lattice modifier first and the Subsurf modifier after it, if you use one.

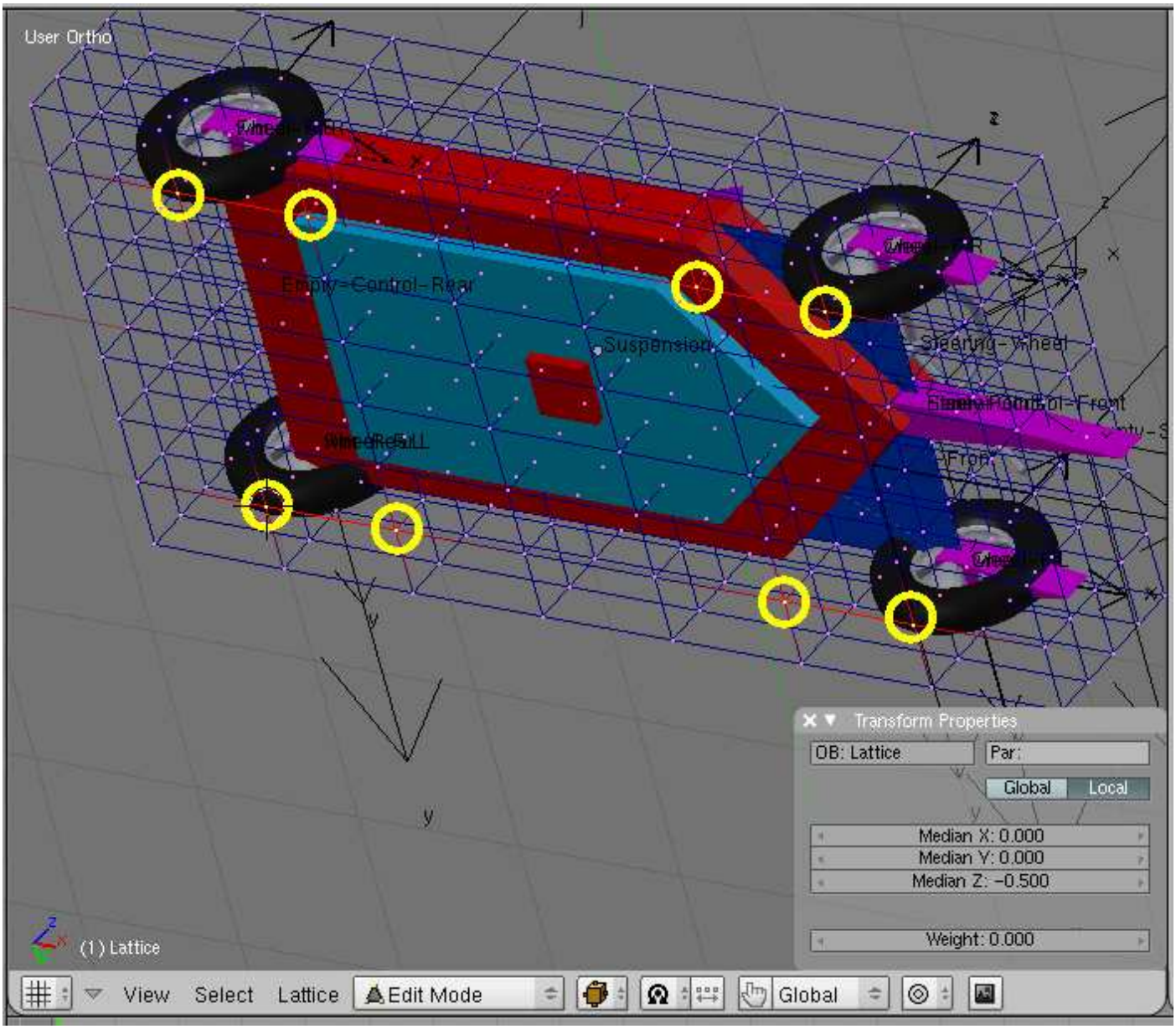
- Just move the Ground-Level Vertex Group of the Lattice in Edit Mode, to check the action of the Lattice on the tires.



#088

- In fact, the operations that we have just done only allow the vertical compression of the tires. For the transversal deformation of the tires in curves, we have to add one more object and set some more constraints.

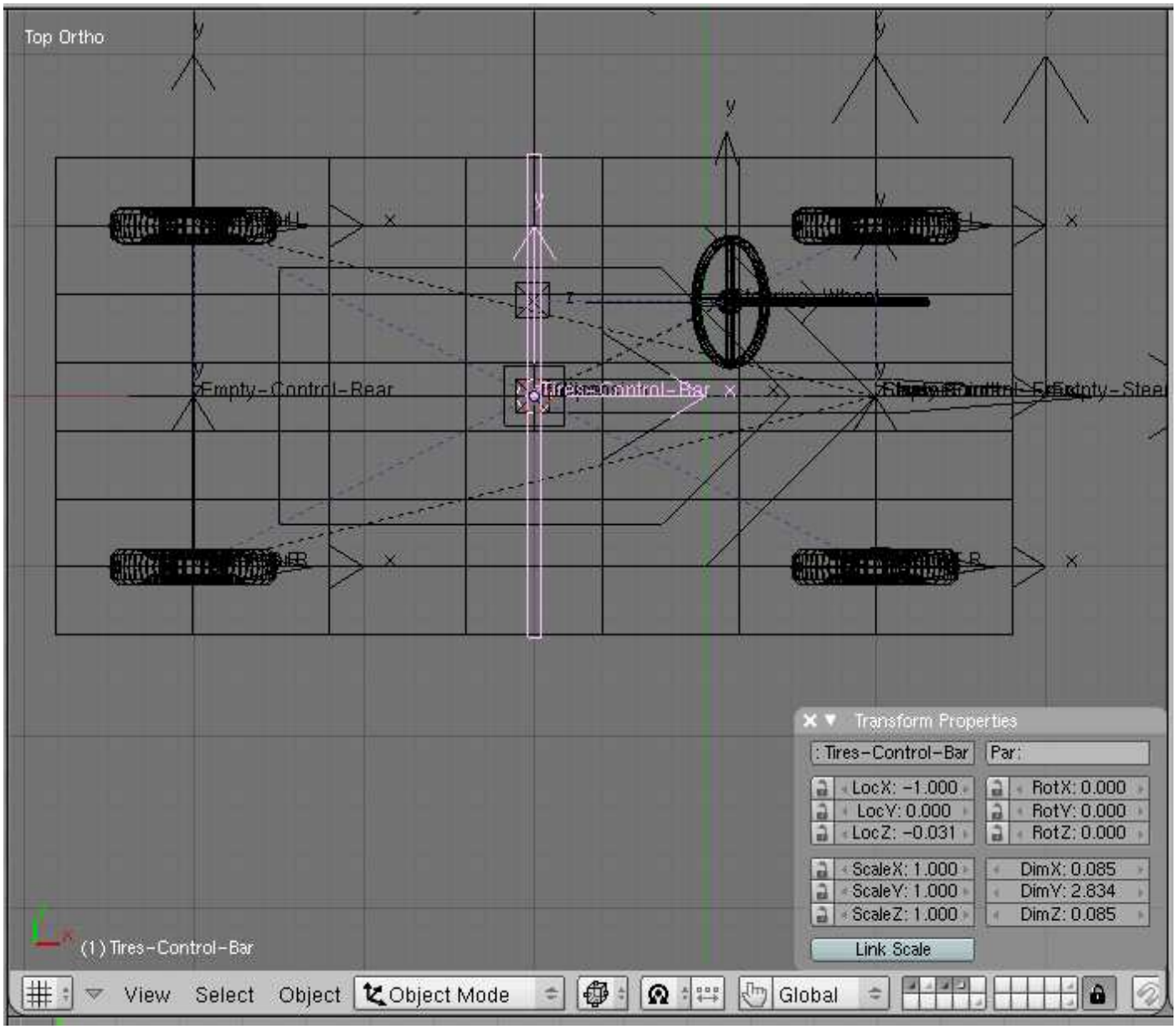
- Still in Edit Mode, add a new Vertex Group to the Lattice. This Group will include the vertices right under the wheels, Plus one vertex before for each rear wheel, and one vertex after for each front wheel. These vertices are shown inside the yellow circles on the image below.



#089

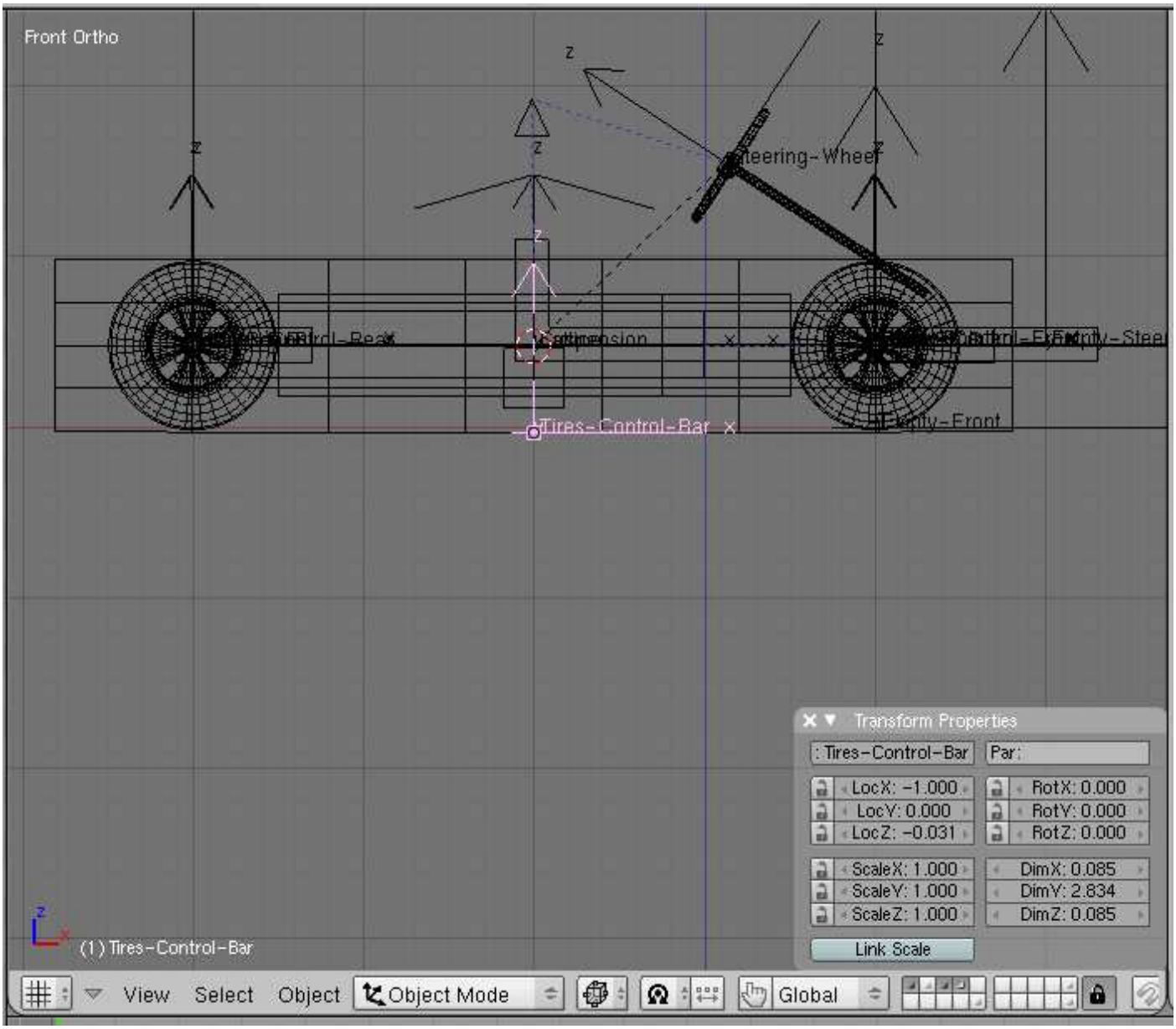
- Select the Suspension object and bring the 3D Cursor on its center.

- In Object Mode, add a cube, and scale it down. Name this cube "Tires-Control-Bar" Then, scale it in the Y direction to create a bar. Its width must be equal to the width of the Lattice.



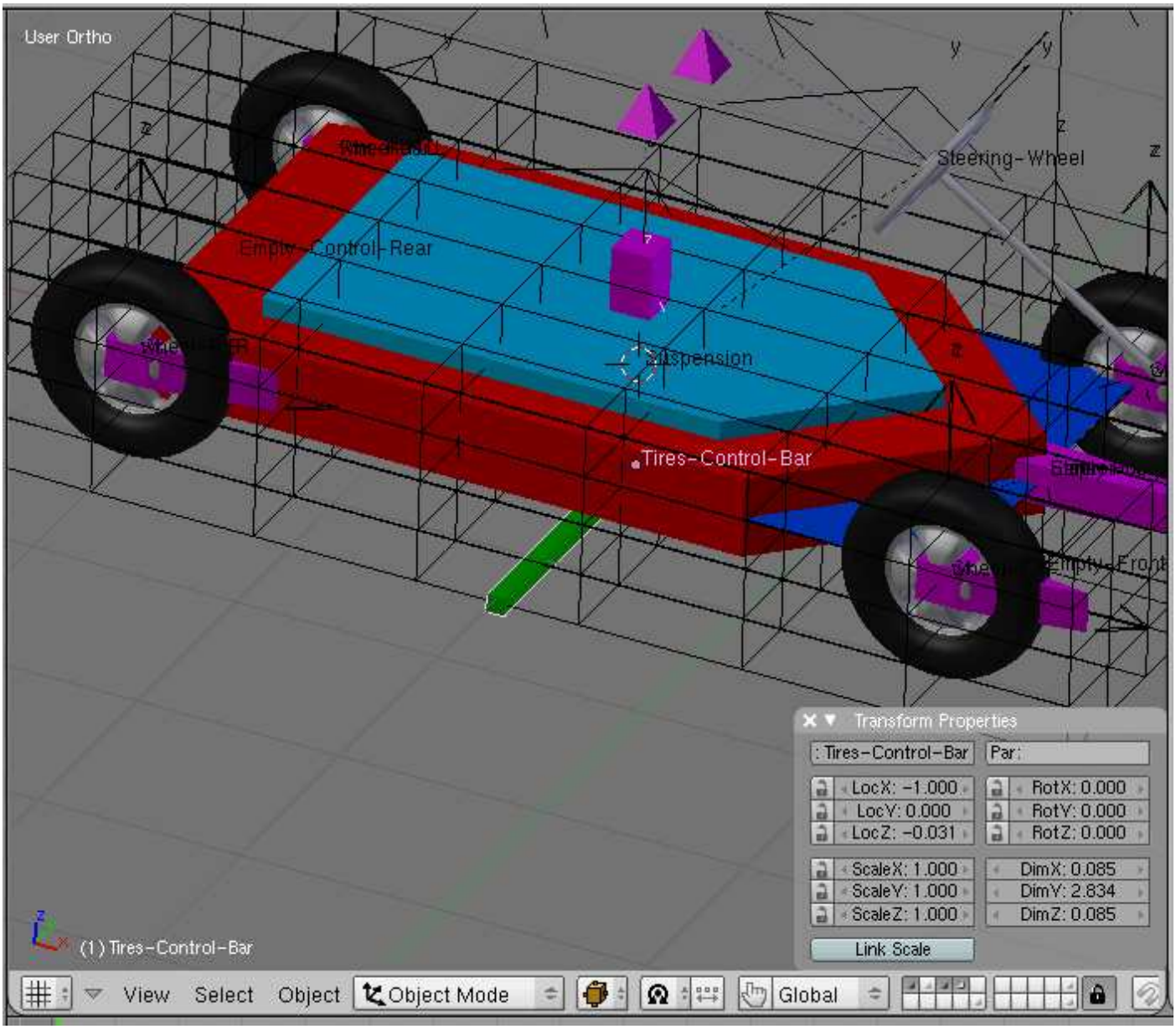
#090

- In side view (related to the car rig) , bring the bar at the height of the lower vertex layer of the lattice (the Ground Level).



#091

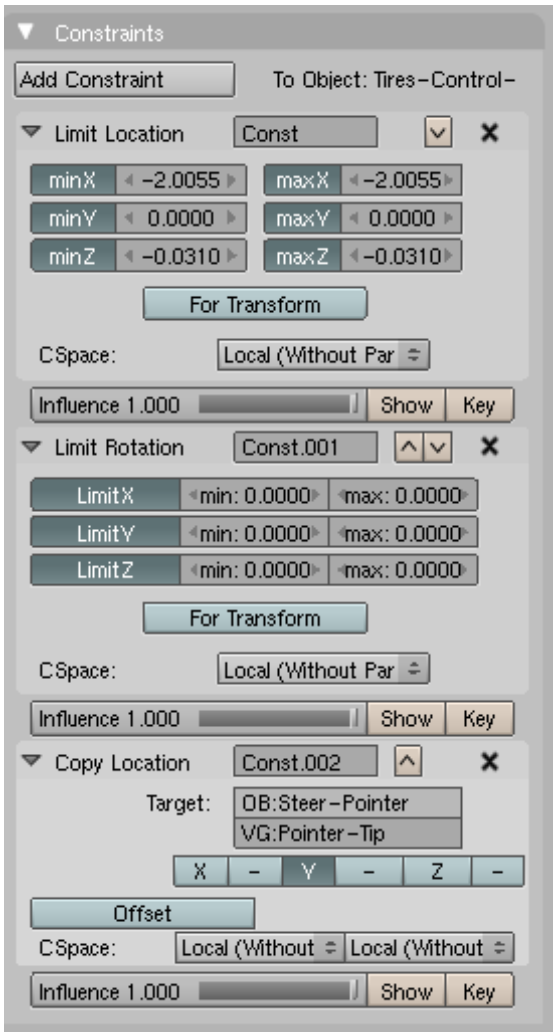
- Parent the Tires-Control-Bar to the Chassis.



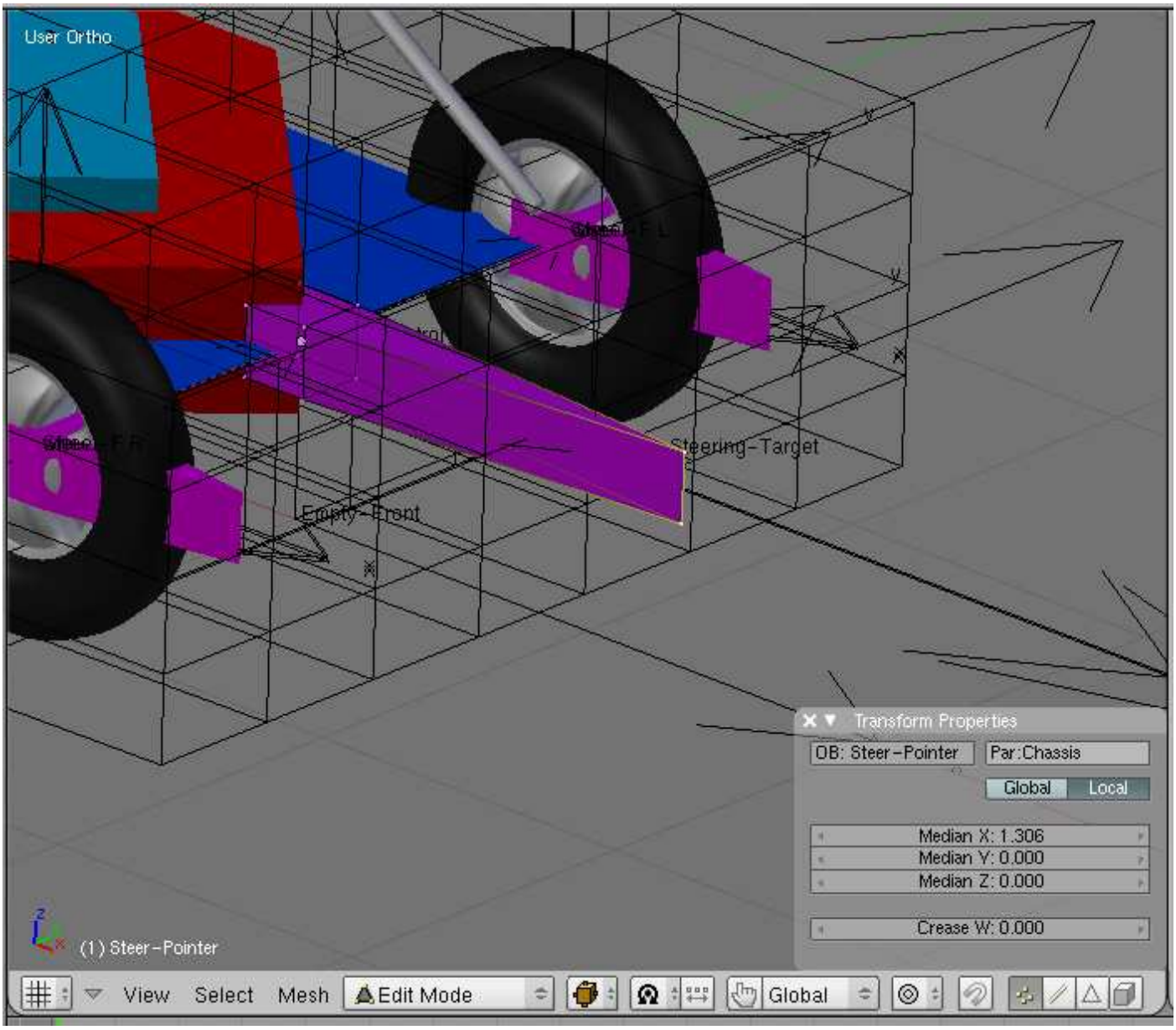
#092

- Select the Tires-Control-Bar again and apply to it a Limit Location constraint. The simplest way to apply X, and Z coordinates to this constraint is to add an Empty at the location of the object itself. Then copy one after the other the values of the different fields of the Transform Properties Panel of the Empty in the fields of the Limit Location constraint of the Bar. Select Local for the CSpace mode. Do not limit the Bar displacement in local Y direction, because it is the direction that we are going to use to control the lattice deformation.

- Add to the Tires-Control-Bar a Limit Rotation constraint on all Local axis, with min and max values to 0.0000, because we don't want this object to rotate.

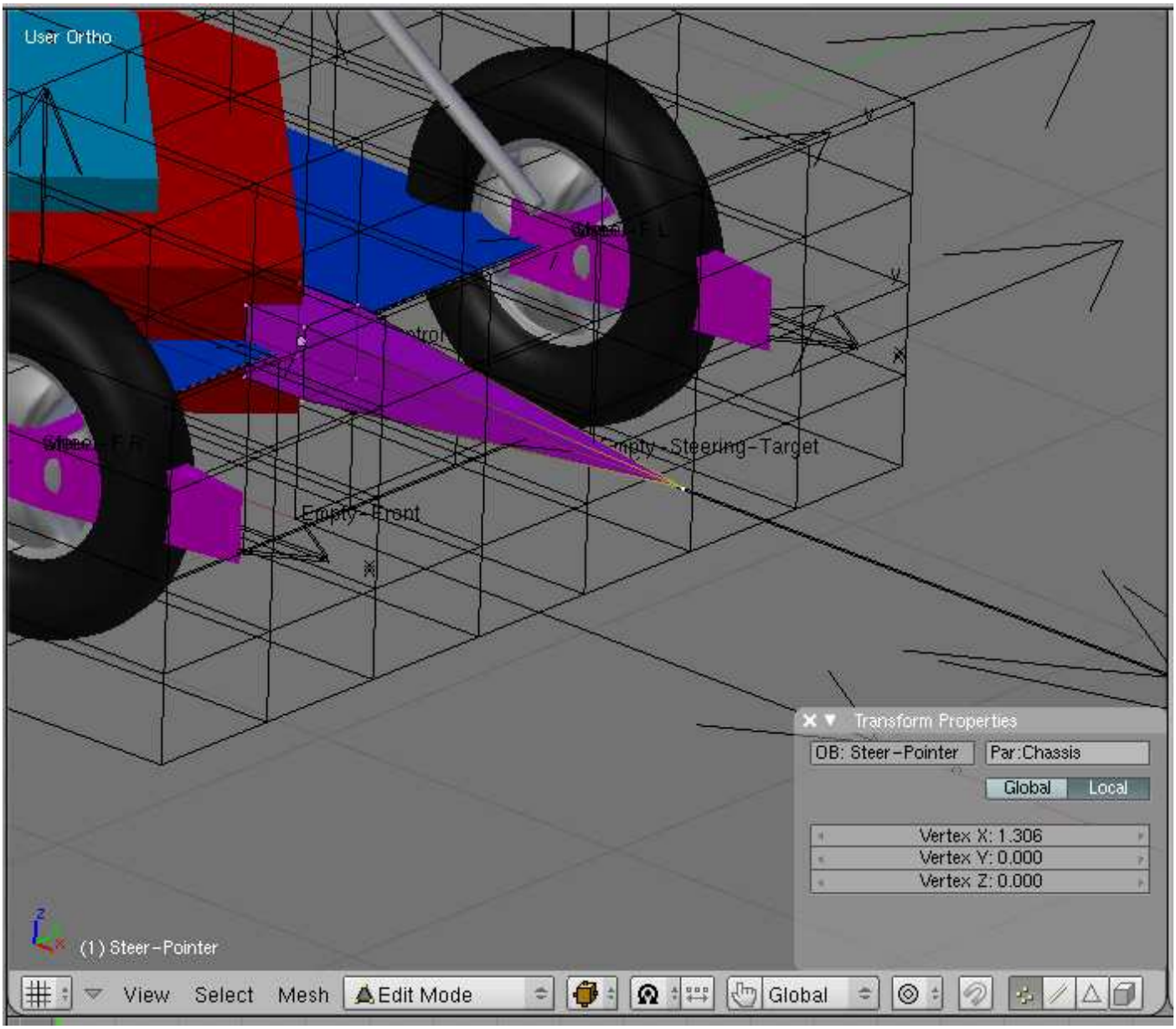


#092B


User Ortho
Steering-Target
Empty-Front
Transform Properties
OB: Steer-Pointer
Par:Chassis
Global
Local
Median X: 1.306
Median Y: 0.000
Median Z: 0.000
Crease W: 0.000
(1) Steer-Pointer
View
Select
Mesh
Edit Mode
Global

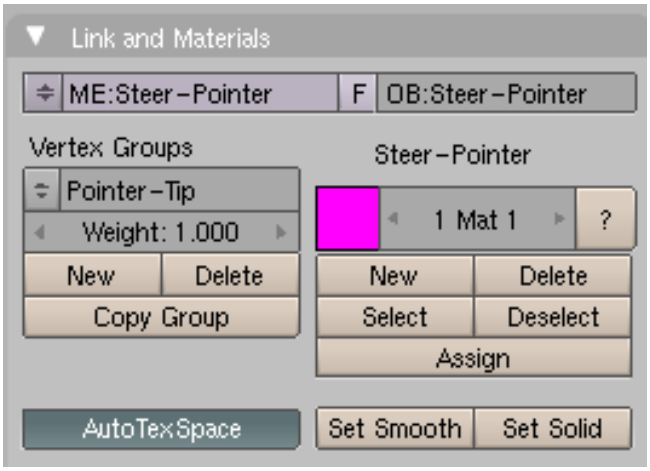
#093

- Now, select the Steer-Pointer object and in Edit Mode, select the two vertices at the most front tip. Scale them to Zero in the Z direction, or merge at center, in order to get only one vertex.



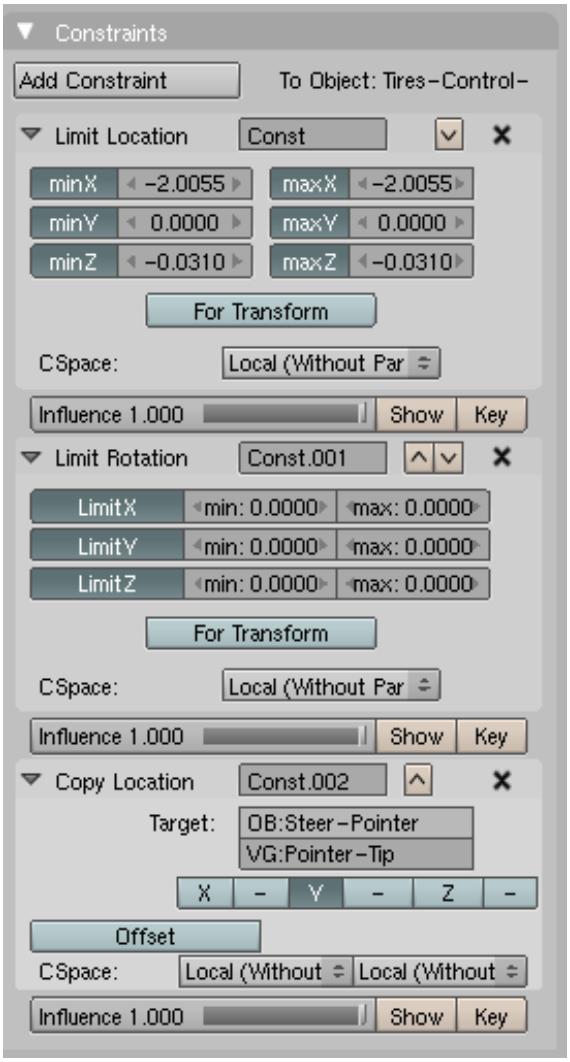
#094

- Create for it a Vertex Group named "Pointer-Tip".



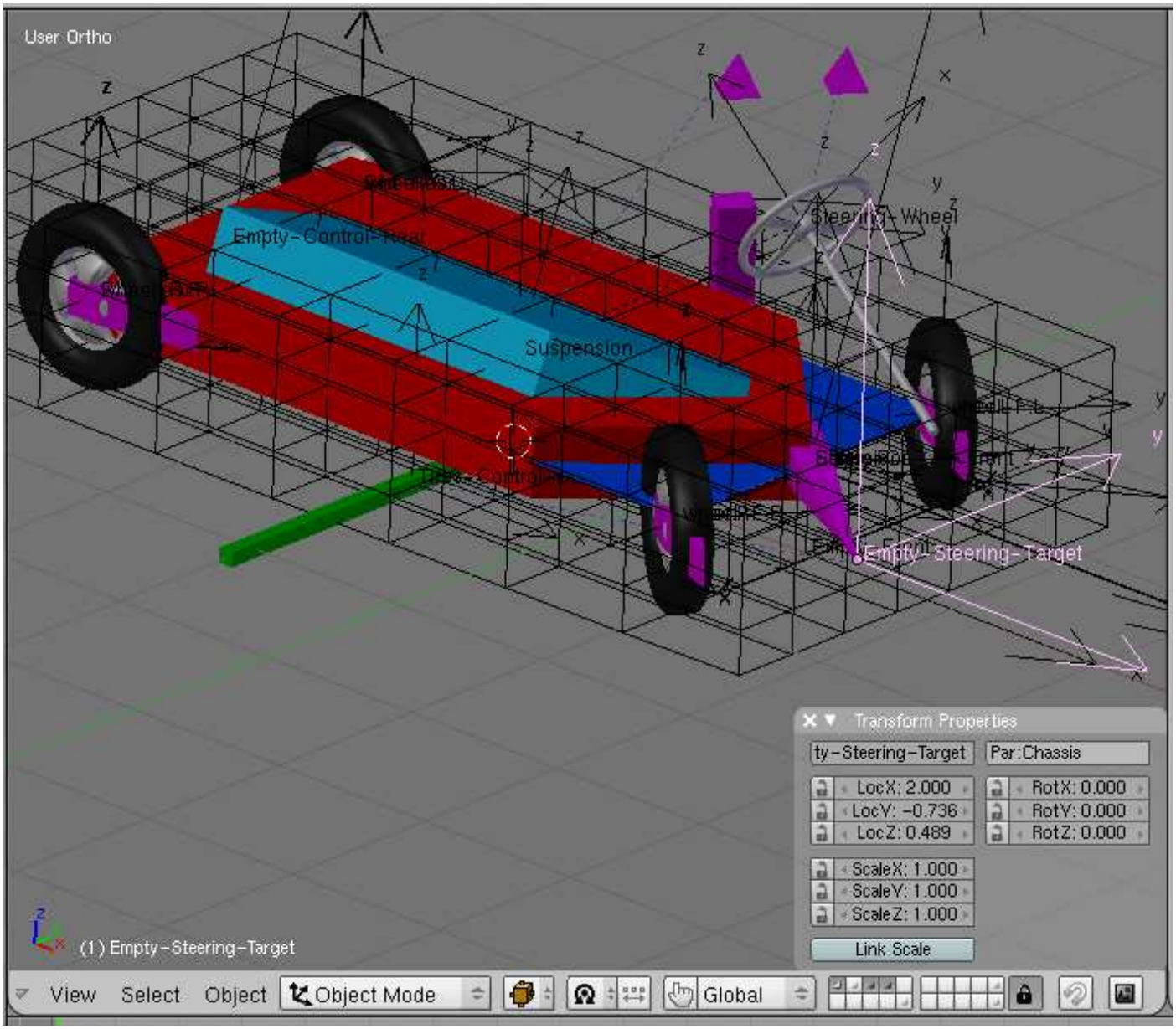
#094B

- In object mode, Select the Tires-Control-Bar and add to it a Copy Location constraint. In the Target fields, type Steer-Pointer and the name of the Vertex Group that we have just created : Pointer-Tip. Enable the Y direction only, and choose Local for the two CSpace fields. leave the influence to 1.000.

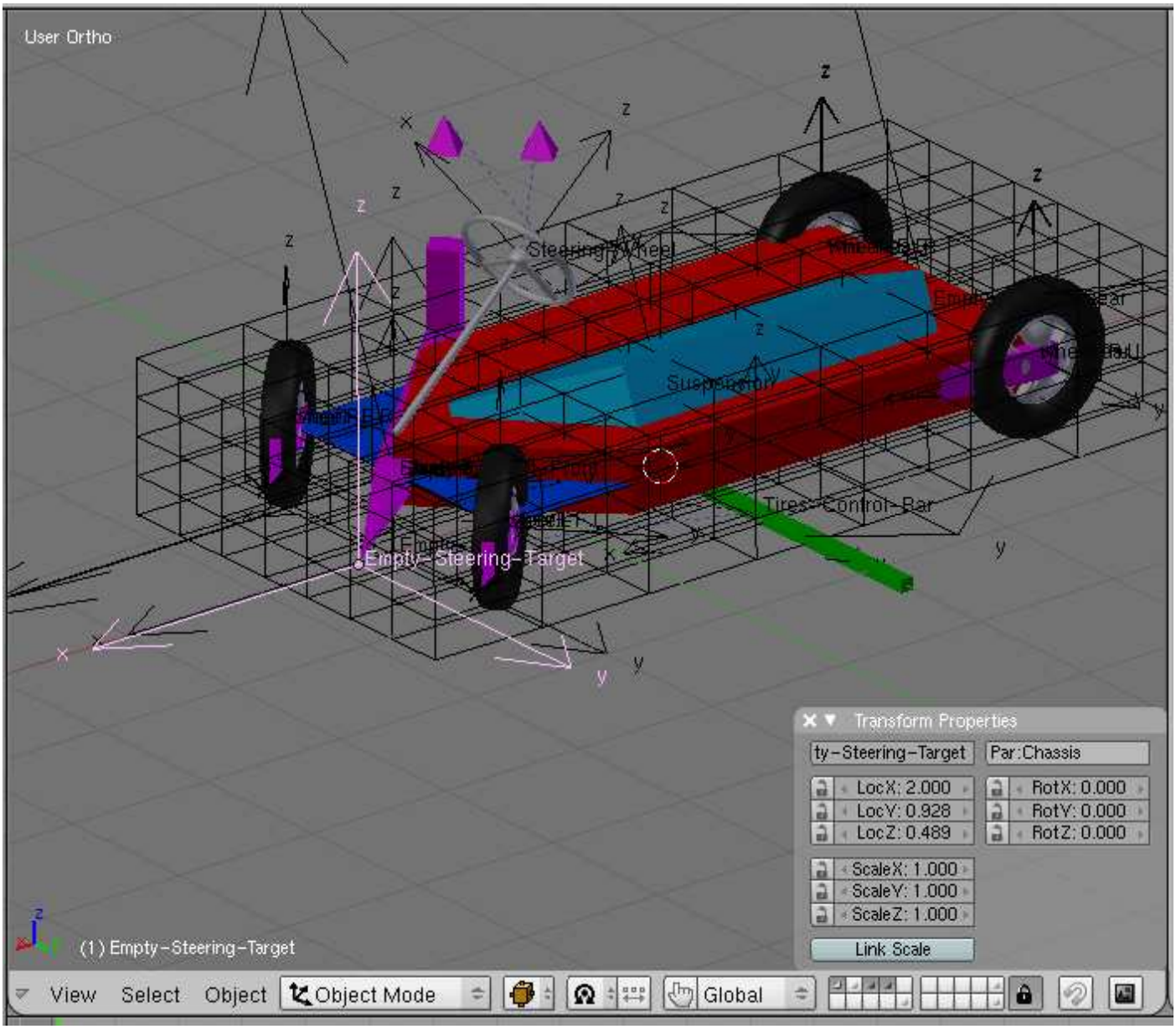


#092B

- Now, if you move the Empty-Steering-Target in the Y direction, you will notice that the Tires-Control-Bar moves also in the Y direction.

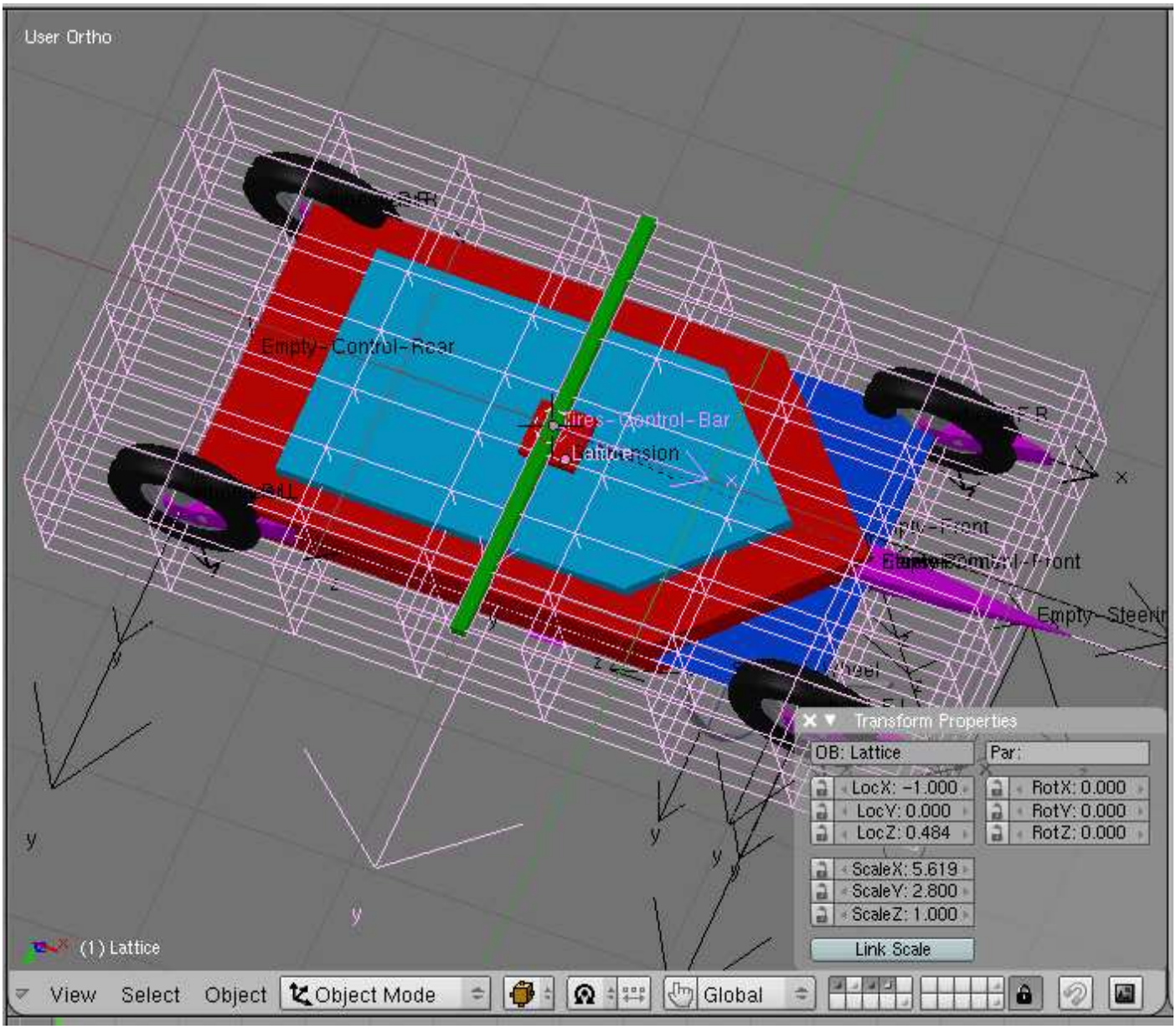


#095


User Ortho
Steering_Wheel
Suspension
Tires_Control_Bar
Empty_Steering_Target
Transform Properties
ty-Steering-Target
Par:Chassis
LocX: 2.000
LocY: 0.928
LocZ: 0.489
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
Link Scale
(1) Empty-Steering-Target
View
Select
Object
Object Mode
Global

#096

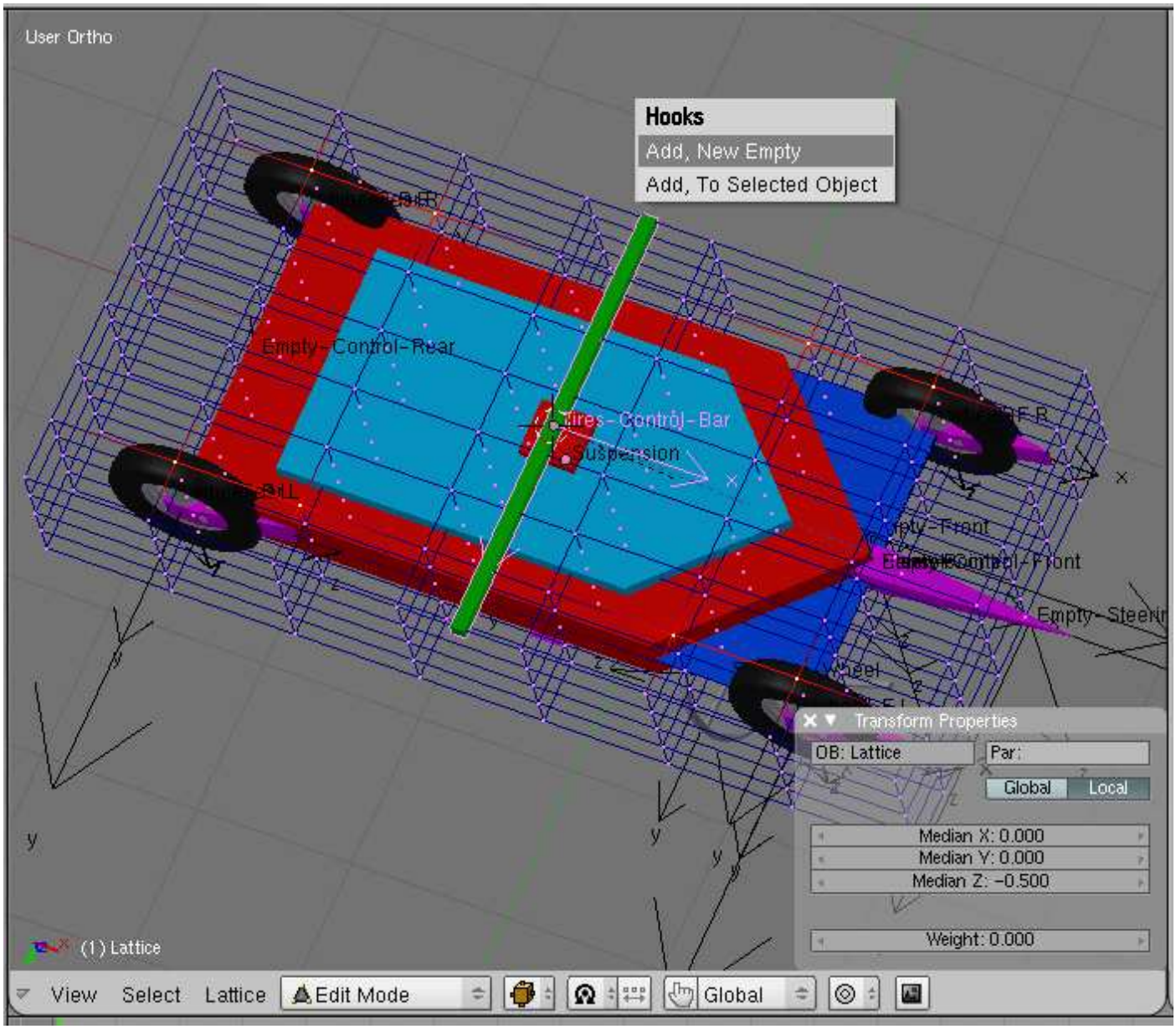
- Looking under the setup, select the Tires-Control-Bar and the Lattice (Shift+Select) and enter in Edit Mode.



#097

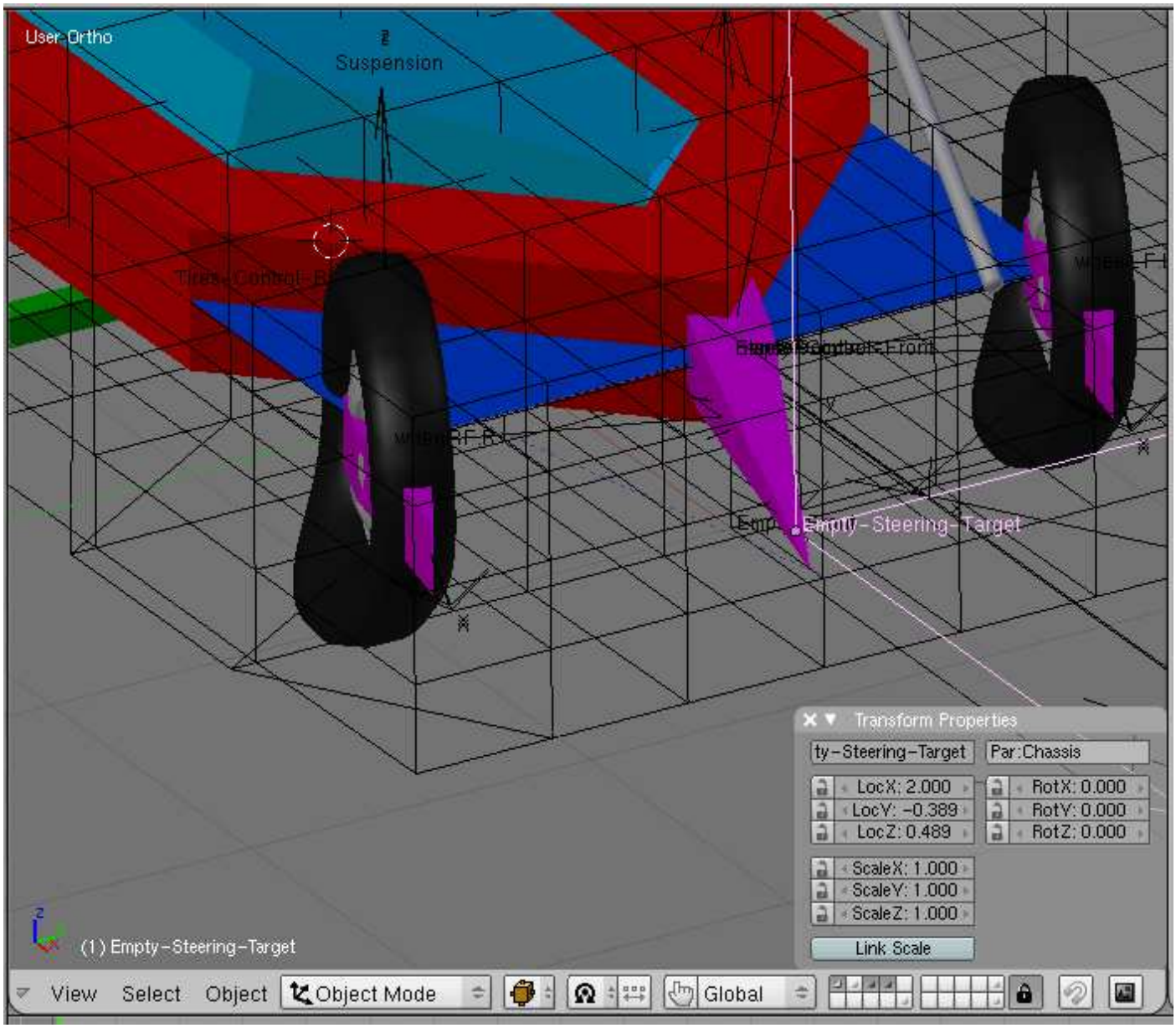
- If the Vertex Group named "Control-Bar" is not selected, then select it.

- Type Ctrl+H and in the small Hooks pop up window, select Add, To Selected Object.



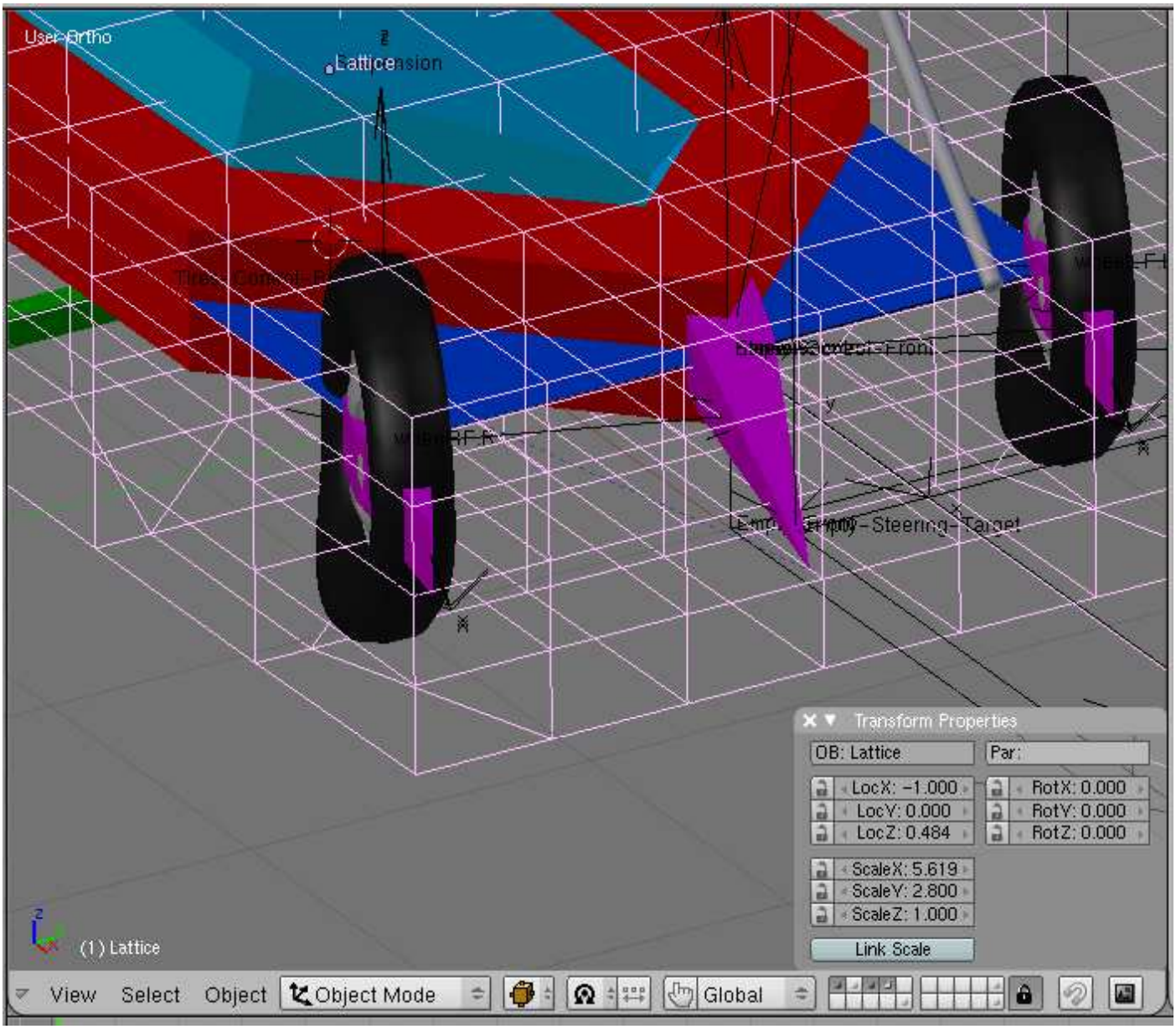
#098

- Now, when the Empty-Steering-target moves in Y direction, the tires are deformed. This said, the deformation is exagerated.



#099

- Just reduce the Force value to 0.5 or lower for the Hook Modifier of the Lattice, and the result will be much more realistic.



#100

Step 12 : Importing and Exporting the Setup.

- To import or export the Setup , it is better to include all necessary objects in a Group. In the future, this setup could be provided with "Blended Cities", a powerful procedural system based on Python code, to create big cities. In this context, the animation of the car would be driven by a script requiring that the Empties used as guide have the same name as the vehicle Group. In this example, if the name of the group is "Car-Rig", Empty-Front will be renamed "Car-Rig" as well.

- If you need to move the car rig in the 3D space to define its initial location and direction, you will have to select both the Empty-Front and the road, so it is better to include a "sample mesh" featuring a small part of road in the Group.

- Once the group is imported, just select the "sample mesh" and the real Terrain or Road. Apply Ctrl+L and select Mesh Data to convert the small sample into the real terrain.

- When the setup is imported and the real terrain or road as well, you can easily duplicate the Car-Rig, by duplicating everything included in the group ,except the ground sample that can be deleted.

On the links below, you can download two blend allowing much easier import of the setup, in a Full and A Light version.

http://3d-synthesis.com/tutorials/Cars-AutodriveSystem-V7-1-1_Full.blend

http://3d-synthesis.com/tutorials/Cars-AutodriveSystem-V7-1-1_Light.blend

More informations here :

http://3d-synthesis.com/tutorialsenglish.html

Step 13 : Editing the trajectory.

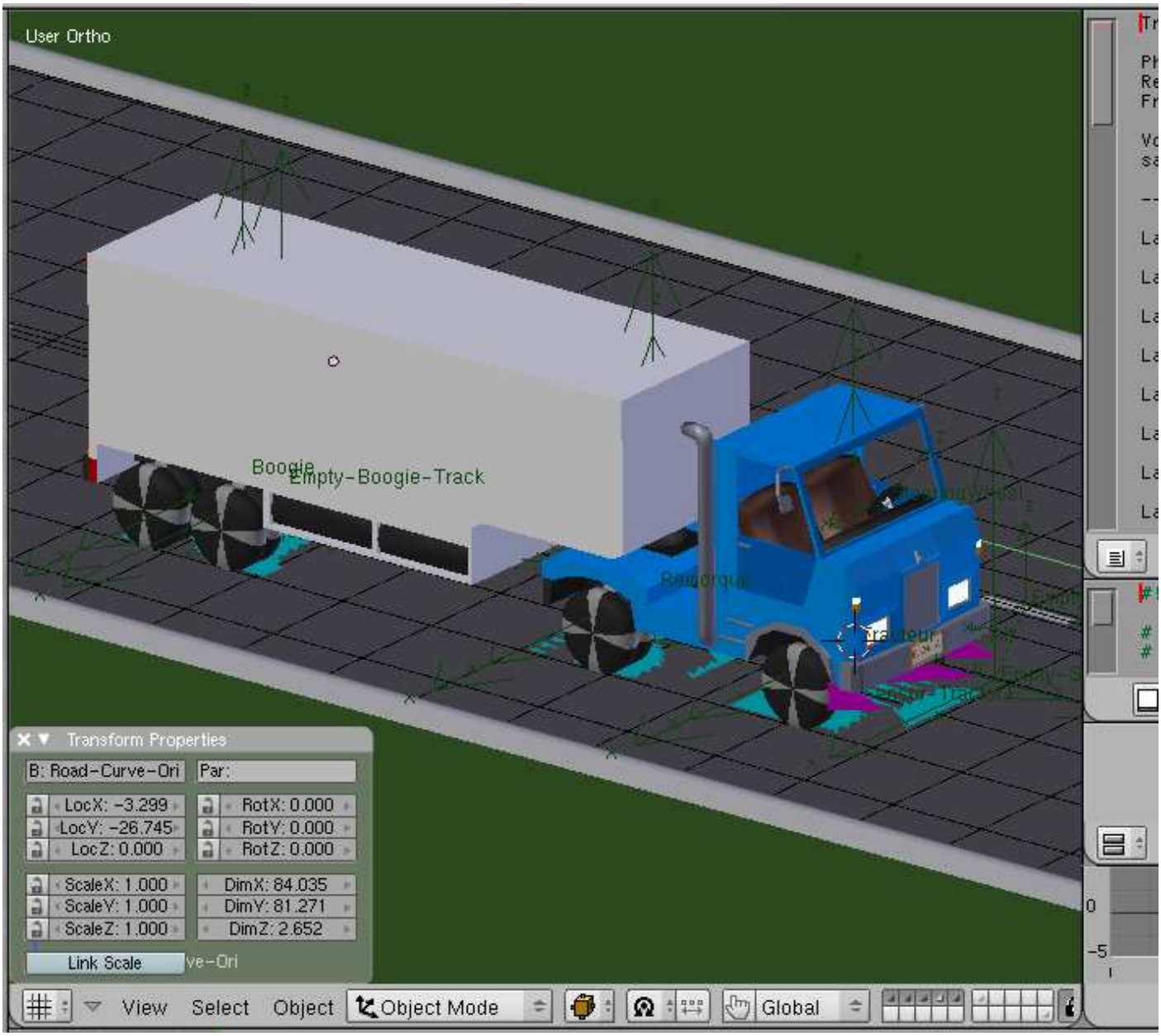
- If you have choosen to not use a curve, the trajetory of the car is only determined by the location of the Empty used as guide for each car, and that you will keyframe like any ordinary object.
- If you have used a curve, the Empty used as guide, will be either directly linked to the curve by a Clamp To constraint, or parented to an other Empty that will be itself linked to the curve by a Clamp To constraint (What I will name the undirect method).
- For my own, I prefer to use a curve and I also choose the undirect method, because it allows to keyframe the Empty-Front, separately of the Empty drirectly driven by the curve.

This way of doing things allows much more flexibility and freedom. Using this method, it is possible to use only one trajectory curve and one Ipo for several cars, while allowing each car to have an individual trajectory through the keyframing of their own empties.

An other big advantage of using a curve instead of only keyframing an empty is that the trajectory is visible in the 3D view. It is also easy to use a copy of the trajectory curve as a guide to create an unvisible road mesh (not rendered) supporting the car-rig through the sensors, and allowing to jump over broken bridges and other stunt effects.

Step 14 : Trucks : Adding a trailer.

Animating a truck in ordinary situations is quite easy. In most case we will show them on a flat road, and so, sensors and suspensions can be very simplified. This said, as now you know how to rig a car for all terrains, you could do the same for trucks if required.

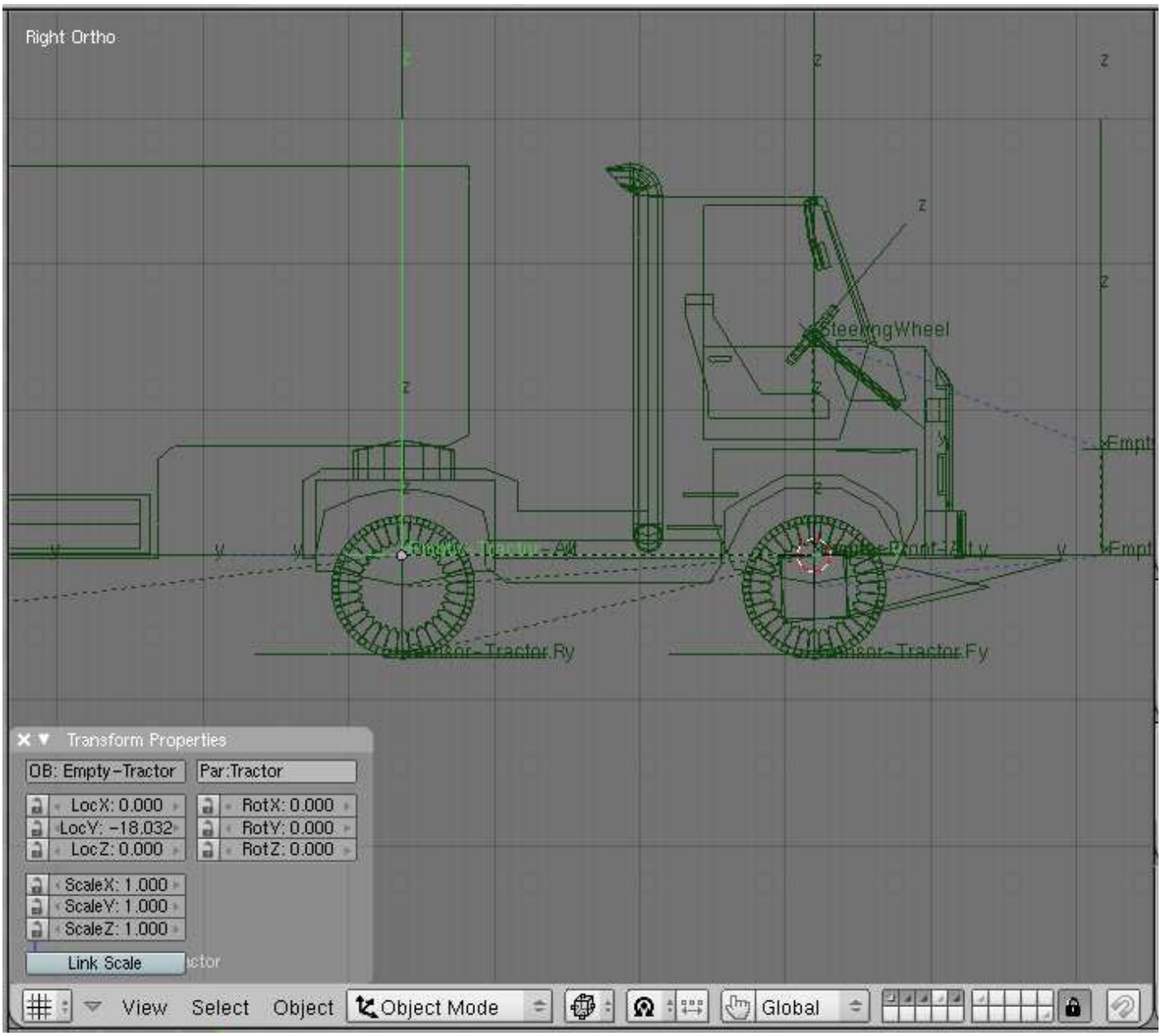


#101

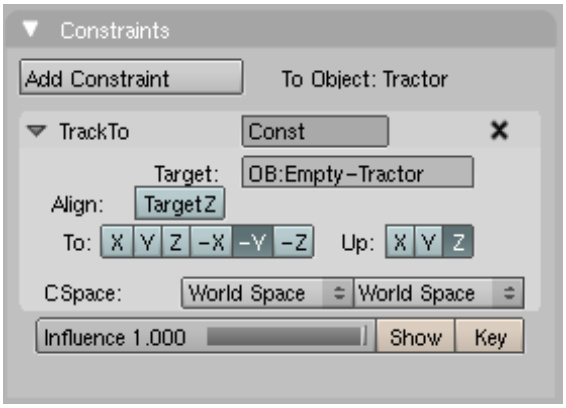
A blend file is available here :

http://3d-synthesis.com/tutorials/Trucks-AutodriveSystem-V2.blend

- A truck tractor is rigged like a car. The tractor is tracking an Empty named Empty-Tractor, that is located between the two rear wheels, a bit above their axle in this example, and that is parented to the Tractor mesh. This relationship between the tractor and its child Empty-tractor can be named self tracking, as the location of the empty depends on the location of the tractor.

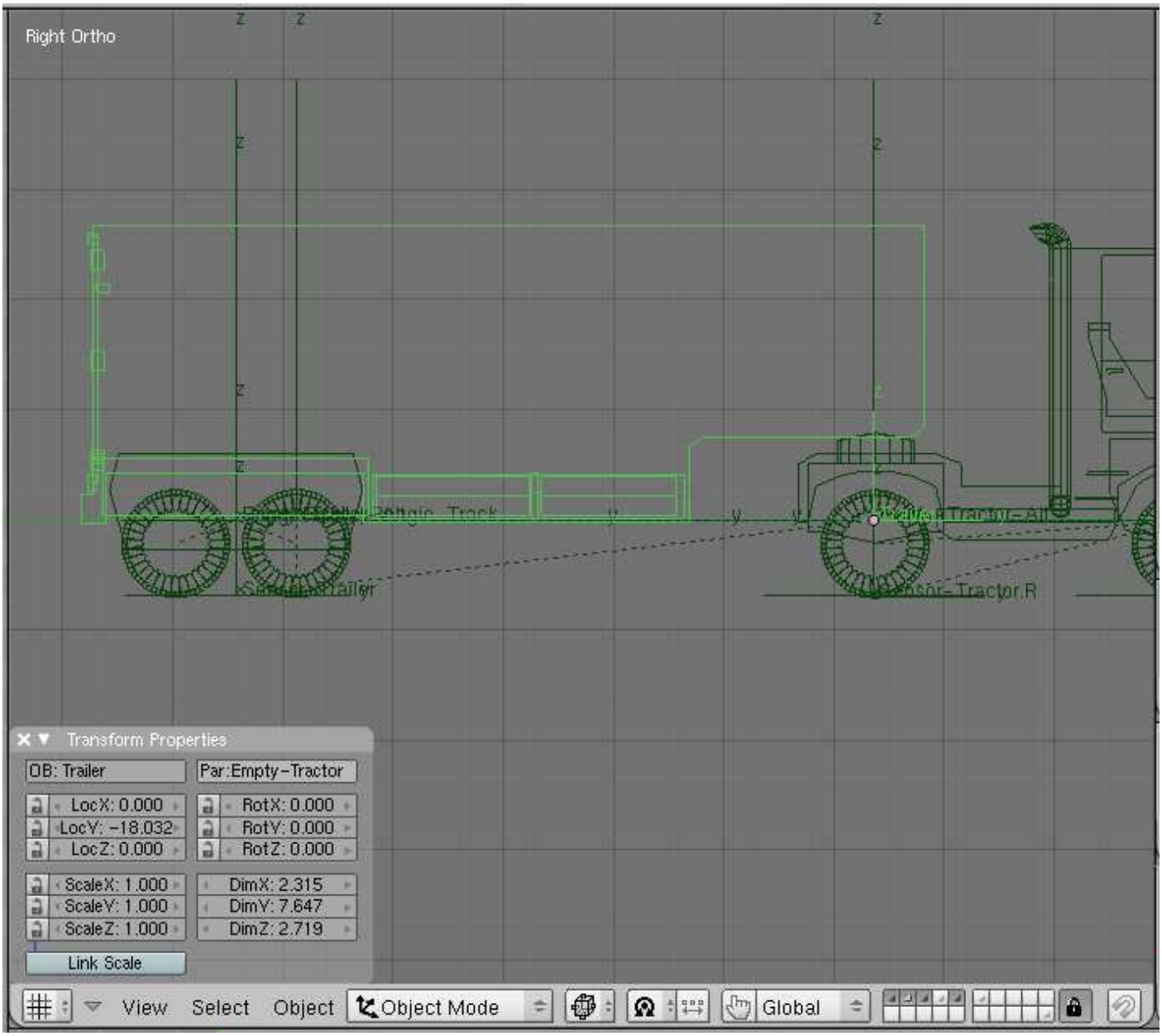


#102



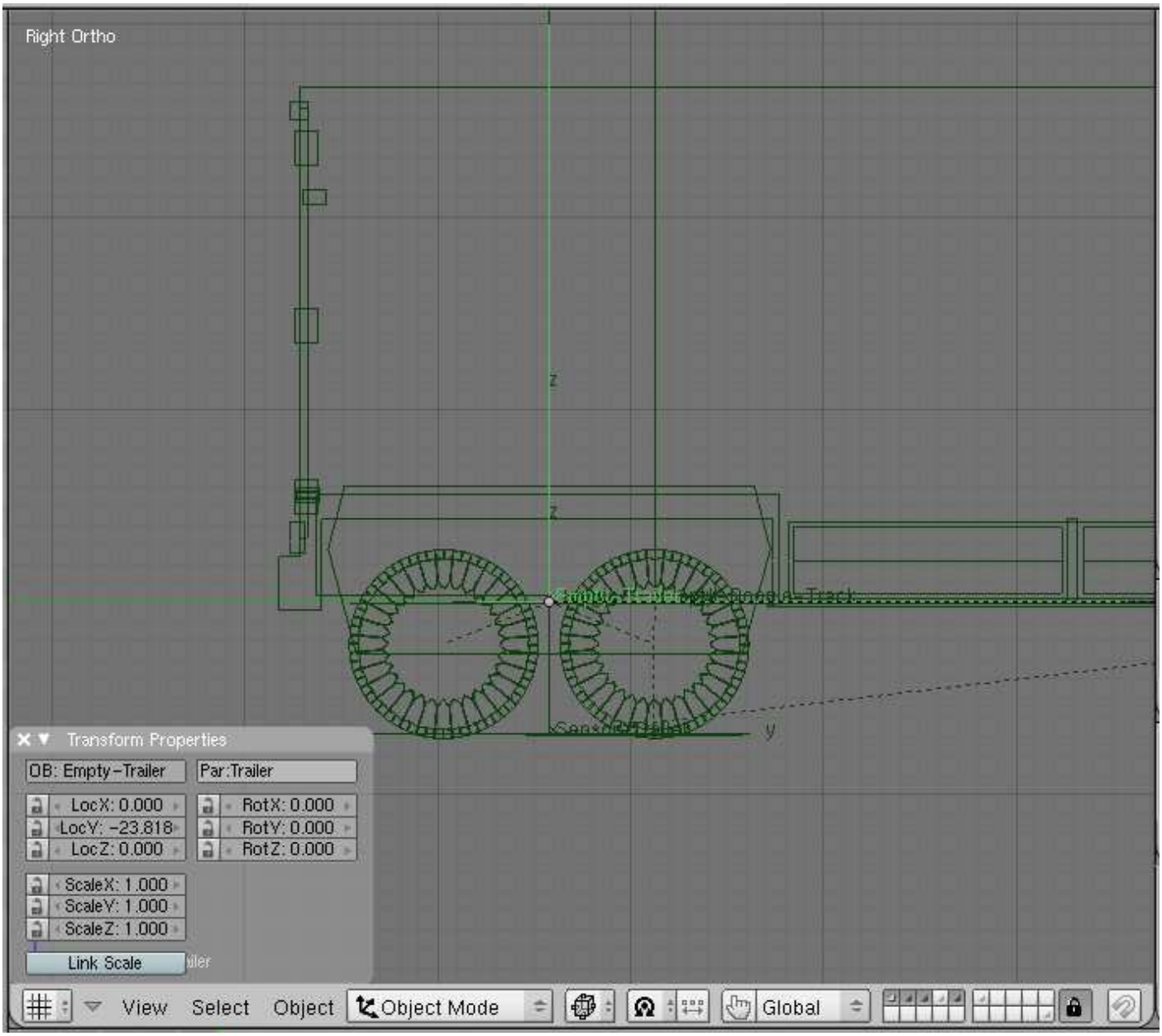
#102C

- The trailer uses exactly the same method. The trailer's mesh is build with its center point set at the location of its rotation point. I mean the point where it is attached to the rear of the tractor : the same location as the Empty-Tractor. The trailer is tracking an Empty named "Empty-Trailer", parented to the trailer, and set in the middle of the rear wheels, just above their axle, at the same altitude as the Center point of the Trailer Mesh.
This way, the "self tracking" keeps the horizontality of the trailer. The whole system can be seen like a car tracted by a car. It is no more complicated.

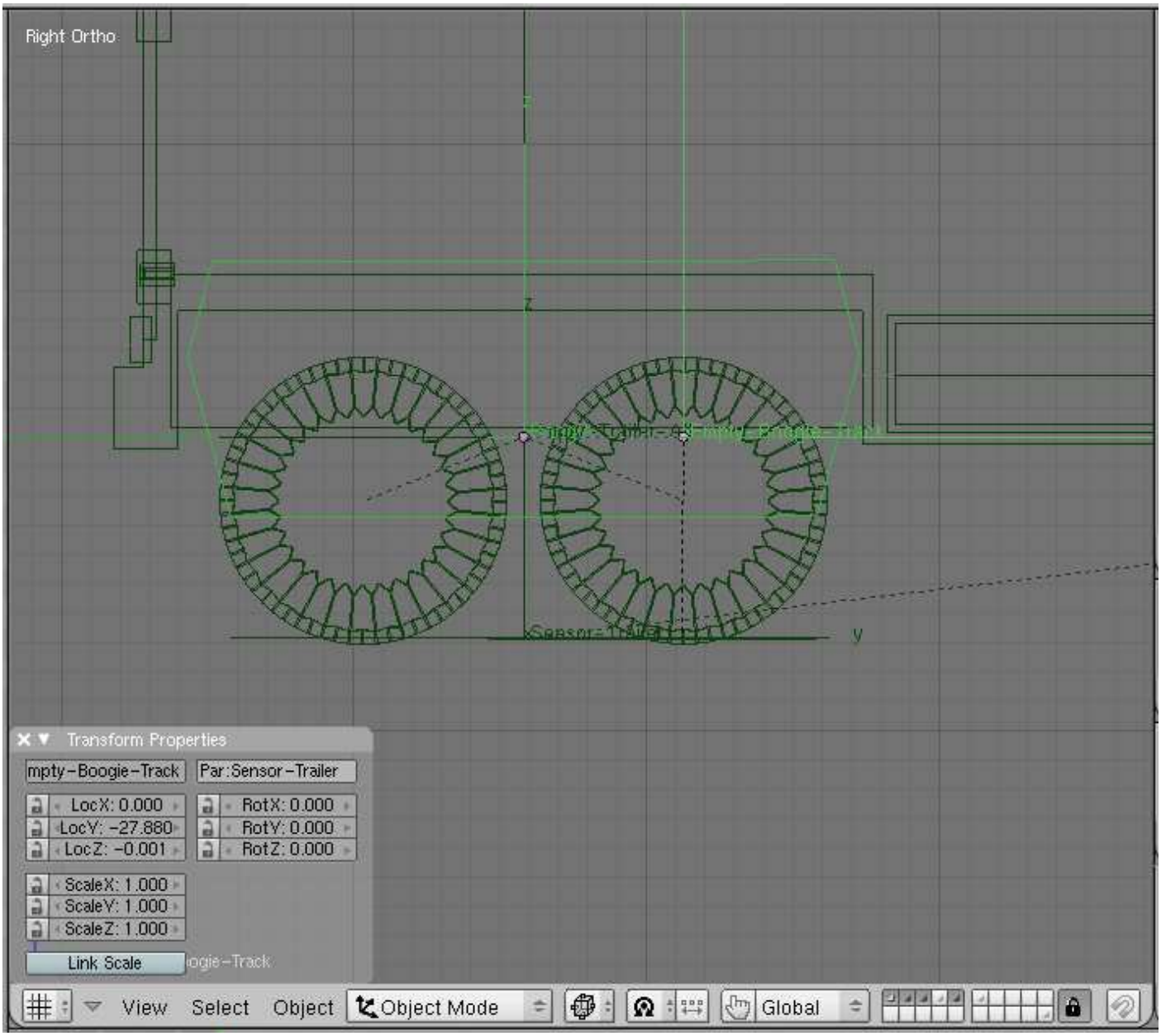


#103

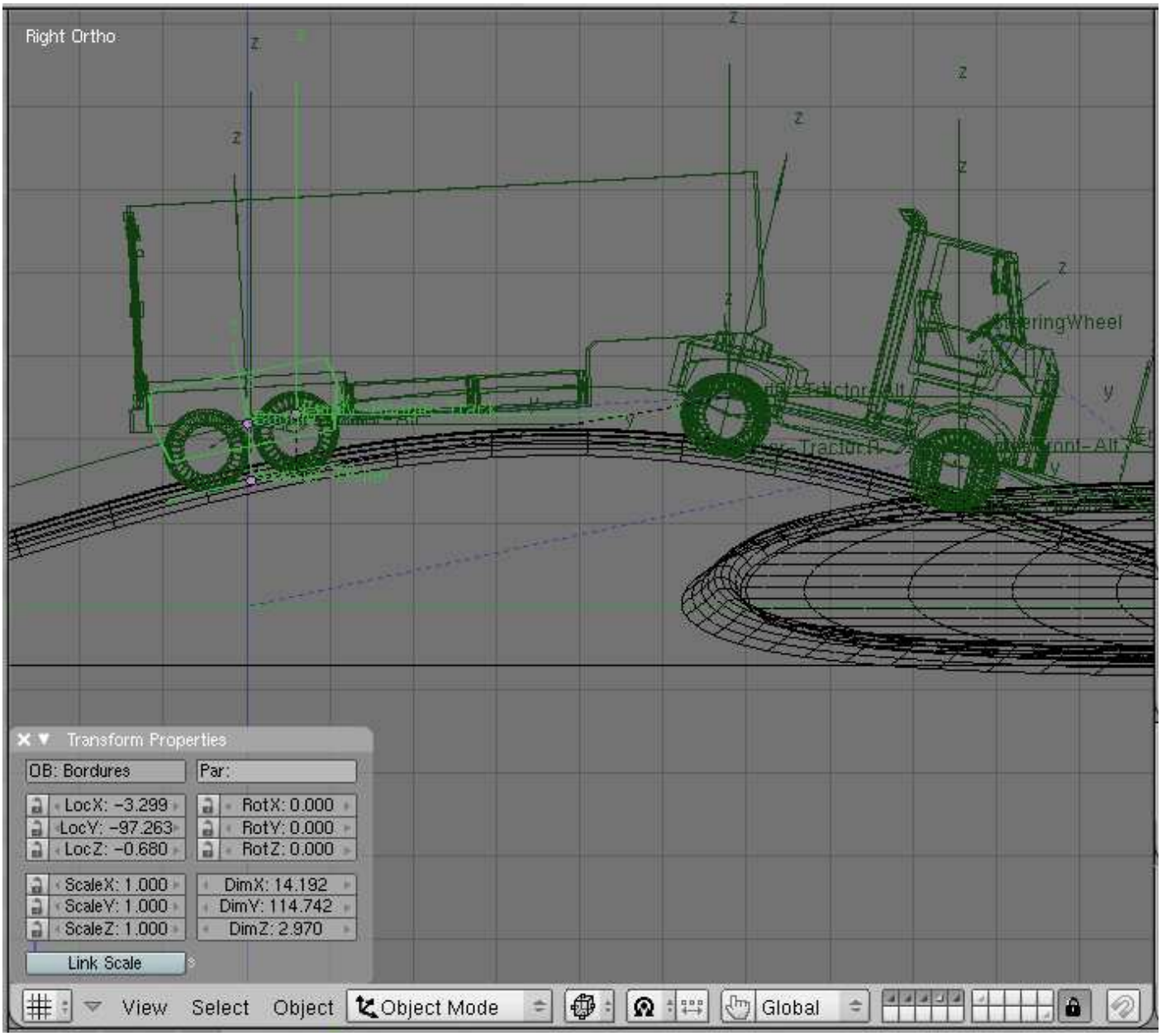
- The wheels of the trailer are parented to a system of boogies, parented to the trailer and tracking an Empty Vertex Parented to a vertex on a sensor mesh. This Track To constraint allows the wheels to follow the slope of the road.



#104


Right Ortho
Transform Properties
mpty-Boogie-Track
Par:Sensor-Trailer
LocX: 0.000
LocY: -27.880
LocZ: -0.001
RotX: 0.000
RotY: 0.000
RotZ: 0.000
ScaleX: 1.000
ScaleY: 1.000
ScaleZ: 1.000
Link Scale
View
Select
Object
Object Mode
Global

#105



#106

- Secondary animation is easy, and can be done by duplicating the meshes of the tractor and the trailer.

- Make the original meshes not renderable and parent their copy to them.

- You can simplify the original meshes to avoid wasting computer ressources.

- By rotating the copies (renderable) around their X and Y Local axis, you can add some realism to the animations in the curves or when using the brakes.

LAST WORDS :

I hope that this tutorial is understandable and that you will find this setup useful. Feel free to send me questions or comments at 3d@3d-synthesis.com or on Blenderartists.org. English is not my native language, so if you find typo, grammatical or spelling errors, please tell me as well...

I would be very happy to see what you did with this system, so don't be shy and please send me a link to your works.

Happy Blending !