

# 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

4x4 Autodrive System. Version 6.1 - 06/09/2009.

Mechanical setup, released under Creative Commons By 2.0 License. France.

## FEATURES :

- All terrains automated driving system, with automatic obstacles detection, not using physics simulation. The Game Engine is not used in this setup.
- Just import your own model of car, wheel and terrain : the system is ready to play.
- Animating the car requires only to edit the curve (trajectory) in top view, and adjust the speed of the vehicle by an unique Ipo curve.
- Driving character provided, automatically animated. You can easily edit the shape and materials of the mesh to create you own character that will still be animated if you don't break the Vertex Groups.
- Automatic tracks creation after each wheel. Generates animated textures for tracks in all ground materials : sand, mud, soil, snow, grass...

**WARNING :** *Due to several bugs in the particle system in Blender 2.49a and 2.49b, the Tracks Creator doesn't allow to render the animated texture based on particles of Object type. So, please, use Blender 2.48a. If you use hair particles to create tracks in grass, 2.48a has also a bug : Hair particles driven by an animated texture made of a sequence of images are not refreshed at frame change. So it is necessary to use a small script written by Atom, to refresh the texture on the ground mesh supporting hair (grass) particles. The script is provided in the blend file.*

**\*\*\* PLEASE READ CAREFULLY THIS MANUAL \*\*\***

## SPECIAL THANKS AND CREDITS :

The speed rotation of each wheel is calculated separately, according to its displacement in the 3D world. This operation is done by a nice Python script kindly written by Jérôme Mahieux (France), known as "littleneo" on BlenderArtists.org, released under Copyleft license. "littleneo" is the author of the famous script "Open City Engine", generating amazing procedural cities. You can find this script on his Website : <http://jerome.le.chat.free.fr>

I also want to thank "Atom", an other member of the same forum, who wrote a very small but very useful script allowing to get rid of an annoying bug in the particle system, while waiting to a real bugfix in a future Blender release.

## LAYERS :

- Layer 1 : Main rig : Armature, Sensors, Empties.

## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

**WARNING :** The bones of the armature are on TWO bones layers. Only the bones which are useful for the use of the system are shown by default (bones on layer 1). The bones layer 2, supports bones essential for the rig to work, but that are not necessary in simple use situation. **Make sure that ALL BONES are visible (Bones layer 1 and 2 enabled) when editing the armature** to adjust the length of the vehicle according to the process described below.

- Layer 2 : Curve, Moving Reference-Meshes, Empties.
- Layer 3 : Basic wheels for testing purpose.
- Layer 4 : Ground mesh, used as reference by the 4 sensors, the curve and the square planes supporting Empty-Front, Empty-Rear, Empty-Steering-Target, Empty-Head-Target.
- Layer 5 : Template meshes for bones Custom Shapes.
- Layer 6 : Car Body Model 1.
- Layer 7 : Car Top (Roof).
- Layer 8 : Car Wheels.
- Layer 9 : Suspension Arms.
- Layer 10 : Blade springs (not used yet).
- Layer 11 : Grey Background Screen, Particles Emitters, and Orthogonal Camera.
- Layer 12 : Particles Objects : small mesh used as Tire Print Sample.
- Layer 15 : Driver-Kit-SAM. Automated driver and Gear/Clutch Remote Control Box.

**WARNING :** This KIT must stay on Layer 15 : As some objects are set to UNSELECTABLE, moving the kit to an other Layer will not move the whole kit and will break it. These unselectable objects are mostly parts of the Remote Control Gear/Clutch Box.

The object "Driver-Kit-Parent" has to be parented to the "suspension" bone. The two Empties Empty-Steering and "Empty-Head" have to be parented to the Driver-Kit-Parent when you import them into an other blend file. To use them you will have to parent them to "Empty-Steering-target" and "Empty-Head-Target".

If you want to import this Kit alone in an other file, DO NOT FORGET to unparent these Empties and parent them to the "Driver-Kit-Parent" object before importing the Kit.

- Layer 16 : Car Body Model 2.
  - Layer 19 : Cameras.
-

## **4x4 AUTODRIVE SYSTEM** for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

- Layer 20 : Car Body Template and Arrow mesh.

**NOTE :** All objects used in the car rig are set to NOT RENDERABLE.

### **CAUTION ! SCRIPT ACTIVATION :**

Please type Alt+P from FRAME ONE in the text window, to activate the wheelrotation.py script. This script drives the rotation speed of the wheels. As it is activated at Frame 1, you will have to launch the animation from Frame 1 at least 1 time. otherwise, your wheels will not rotate. Please read the script itself for more informations about wheels naming and setup.

### **PLAYING THE ANIMATION :**

Alt+A to launch the animation.

### **SUGGESTIONS :**

- 1 - First, check the Setup with Layers 1,2,3,5 Enabled.
- 2 - Check the Setup with Layers 4,6,7,8,9 Enabled.
- 3 - Check the Setup with Layers 4,6,8,9,15 Enabled, and play with the Gear/Clutch remote control : look at the driver's hands and feet. It is an animation tool, not a realtime tool.
- 4 - Check the Setup with Layers 1,2,3,11 Enabled : Type Alt+A from Frame 1 to see the Tracks Creator in action. The four Particles Emitters will drop tires prints after each wheel, while the tracks are baked in cache.

### **PRINCIPLE OF THE SETUP :**

This is the 6.1 release, using armature and Empties.

This rig doesn't contain interdependencies, so it is more responsive than previous versions.

The Car body is parented to the blue Custom bone named "Suspension".

Bones are placed on 2 bone layers. You can disable the visibility of the bone layer 2 for normal use. Only the Bones used as parents for the wheels will be visible, as well as the Axles and the Suspension bone. These bones have custom shapes made from meshes used as templates and put on Layer 5.

The principle of the ground level and obstacles detection is based on four "Altitude Sensors".

## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

These sensors are grouped in two meshes named "Sensors-Front" and "Sensors-Rear". "Sensors-Front" follows the "Empty-Front" through a Copy Location constraint and tracks the "Empty Rear". "Sensors-Rear" is parented to "Sensors-Front".

These sensors are affected by ShrinkWrap modifiers : This means that the vertices of the square meshes are projected onto the surface of the Ground mesh, along the Z axis.

The Vertex at the center of each square mesh is used as a sensor to detect the level of the terrain. An Empty is Vertex Parented to the central vertex of each square mesh.

These 4 empties are used to locate and align the bones Axle-Front-DN and Axle-Rear-DN, in the 3D world, using Copy Location and Track To constraints.

As the geometry of the car chassis can't be kept by using a single rig based upon these two axles, the informations given by this basic rig are reported onto copies of these axles, placed at the level of the wheels centers. Axle-Rear-UP is locked on the Z axis of the vehicle and copies the Y rotation of Axle-Rear-DN.

Empty-Rear-Ref-UP is parented to Axle-Front-UP and gives a precise location to Axle-Rear-UP.

The choice of this "Double Stage" rig allows to get rid of geometrical deformations of the "chassis", but the drawback is a very slight loss of accuracy in the contact between the wheels and the Ground. However, it is noticeable only in extreme slopes, or when passing over a very big hole, and can be easily corrected manually if necessary. There are at least 3 methods to do this correction. We will see that later.

The "Sensors-Front" Object is driven on the curve by the "Empty Front", through a Copy Location constraint, and is pointing to Empty-Rear, Through a Track To Constraint.

The empties "Empty-Front", "Empty-Rear", and "Empty-Steering-Target", are Vertex parented to Small square meshes all driven by the same dLoc Ipo, and following the curve thanks to a Curve Modifier. An other square mesh is added, named "Head-Target". This mesh supports an Empty named "Empty-Head-Target", used to drive the behaviour of the Automated Driving Character.

So, all the empties move on the curve with the same speed. This method is much more handy and accurate than using Follow Path constraints.

The dLoc Ipo curve allows to adjust the Speed of the car. The Front wheels are parented to the bones named "Steer-Front.R" and "Steer-Front.L". These bones copies the direction of the "Steer-Pointer" bone. The "Steer-Pointer" Object tracks the Empty-Steering-Target. The influence of the Track To constraint, as well as the location of the Steering-Target square mesh on the curve, can be adjusted, to adjust the Steering angle of the wheels.

The Rear wheels are parented to the bones named "Rim-Rear.R" and "Rim-Rear.L".

## **4x4 AUTODRIVE SYSTEM for Blender v2.48a**

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

The behaviour of the rear wheels on their trajectory can be adjusted by moving the Square mesh supporting the Empty-Rear along the curve, in the chosen direction, accordingly to the Ipo (dLocX or dLocY). Playing with the distance between the square mesh "Rear" and the center of the rear axle, will give more or less "Trail Effect" on the rear wheels. No distance means that the rear wheels will pass exactly in the tracks of the front wheels, and is not very realistic in narrow curves.

This said, reducing this distance on a small part of the trajectory is one of the methods to correct the loss of altitude accuracy in extreme torsions of the chassis in high slopes. IT IS NOT THE BEST METHOD though.

### **IMPORTING YOUR OWN TERRAIN MESH :**

**IMPORTANT :** Do not forget that several important objects are using the Ground Mesh, through ShrinkWrap Modifiers.

So, the Ground Object name is important. To avoid troubles when importing the setup, give to your own terrain an other name, and import the Ground mesh.

Then, select your terrain and the Ground mesh, and with the Ground mesh selected, type CTRL+L and choose Mesh Data.

The Ground mesh will keep its name, but will get all its properties from your own terrain mesh. So, the new Ground mesh will be identical to your own terrain.

After that, you can delete your own original terrain mesh. To be safe, always keep a copy of it in a file.

### **IMPORTING THE SETUP INTO A PERSONNAL PROJECT :**

The easiest and safest way of using this setup is by importing your scene content into this file.

However, in many cases, you will probably prefer to do the opposite. So here is the process :

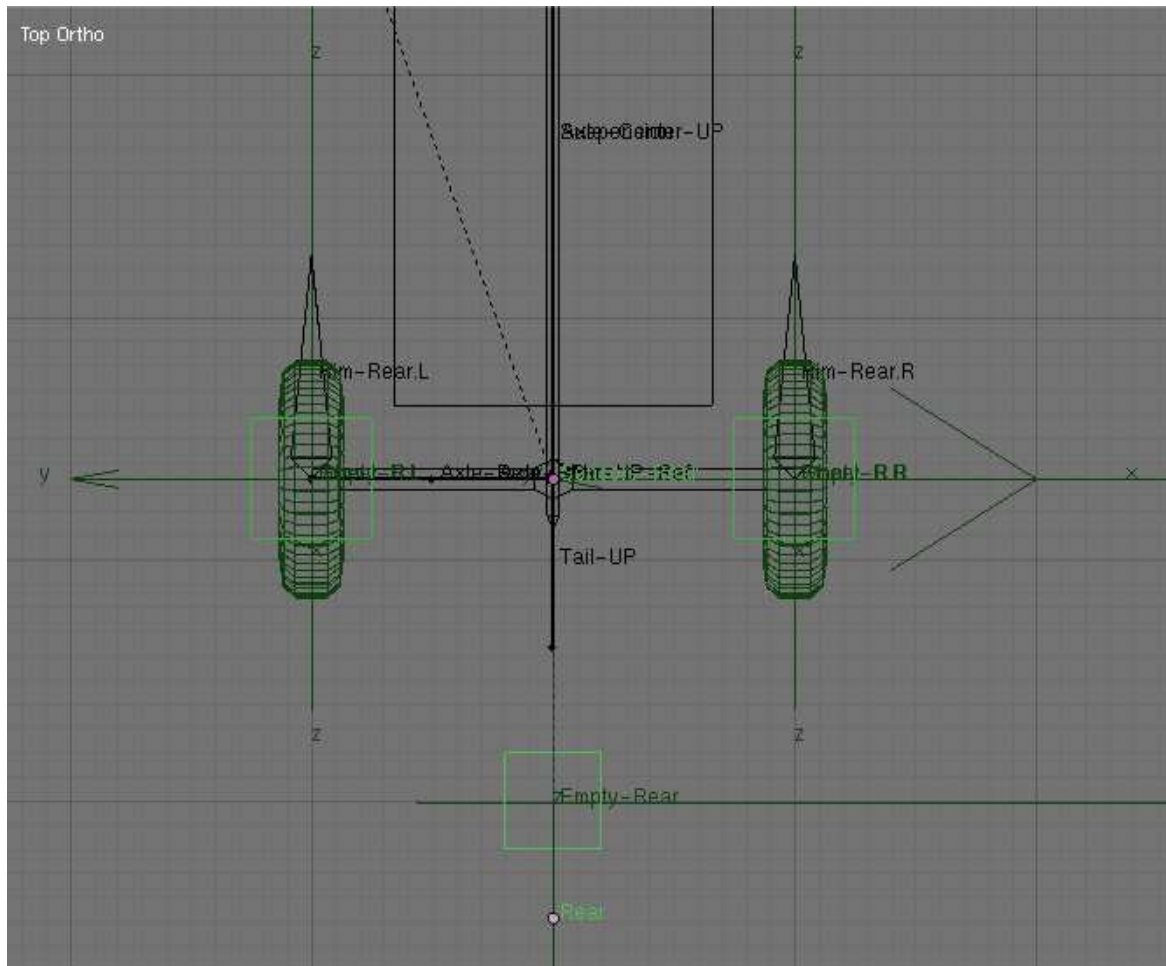
- 1 - Append the Groups named "Car-Setup" and "Rear".
- 2 - Select an object included in the Car-Setup Group.
- 3 - Select all the objects in the same group.
- 4 - Scale all the selected objects Up or Down, to match the WIDTH of your own Car Body model. I mean the distance between the Right and Left wheels.
- 5 - When it is done, you have to adjust the LENGTH of the rig, to match the distance between the Front and Rear wheels.

## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

a - Select all the objects included in the "Rear" group. these objects are the meshes named "Sensors-Rear", and "Rear".







## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

c - Select the Armature. In Edit Mode, select all the bones belonging to the rear part of the armature, including the rear tips of the "Axle-Center-UP" and "Suspension" bones.



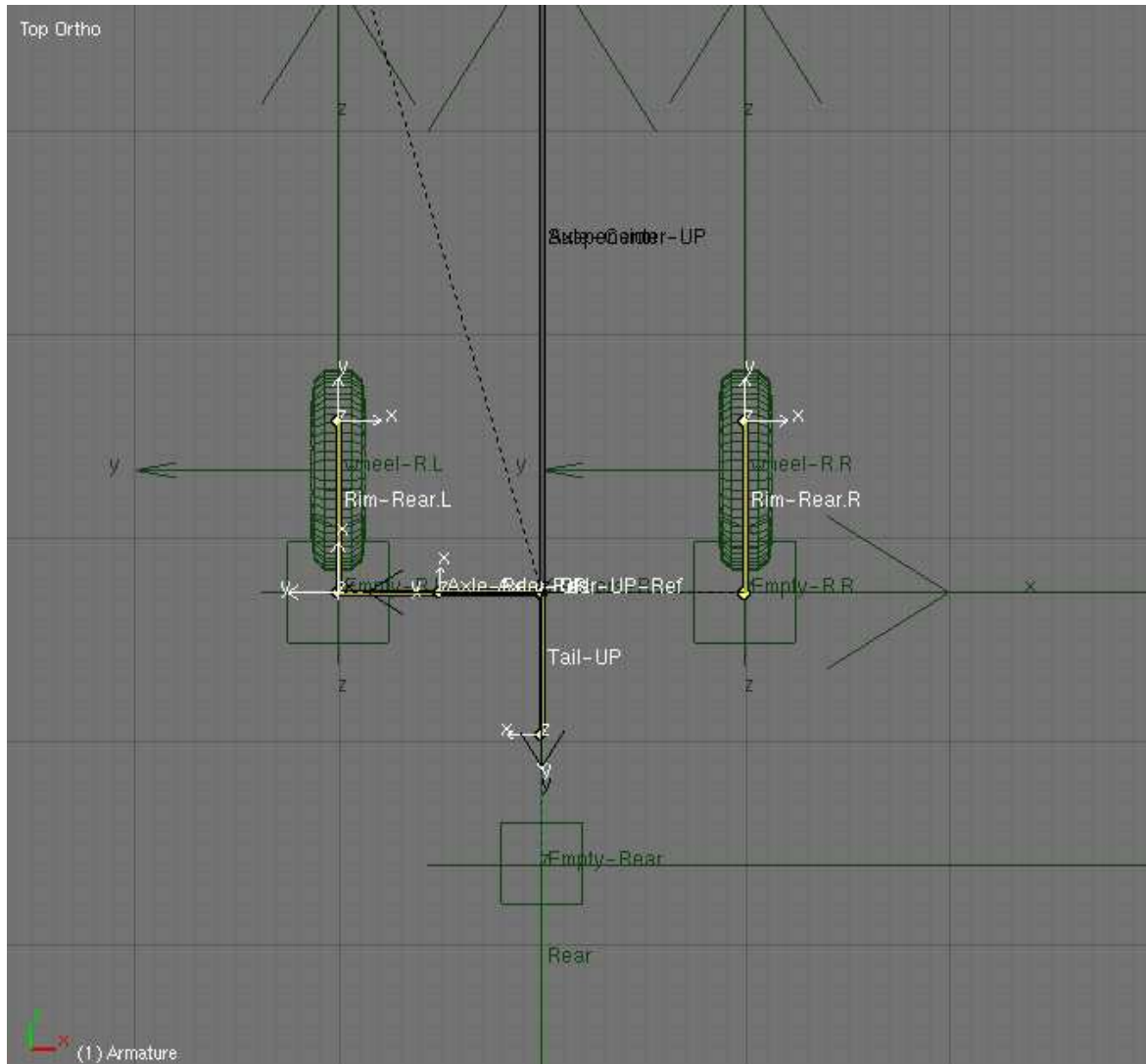


## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

d - Move the bones in the local Y direction, to match the distance between the front and Rear wheels. The displacement value is the same as the value noted in "b".



## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

When you exit the Edit Mode, the rear wheels have followed the bones, and are at the good place along the Y axis, but they are not at the good altitude in Z direction. Now we will fix this :

e - In Pose Mode, select the bone named "Axle-Rear-DN".

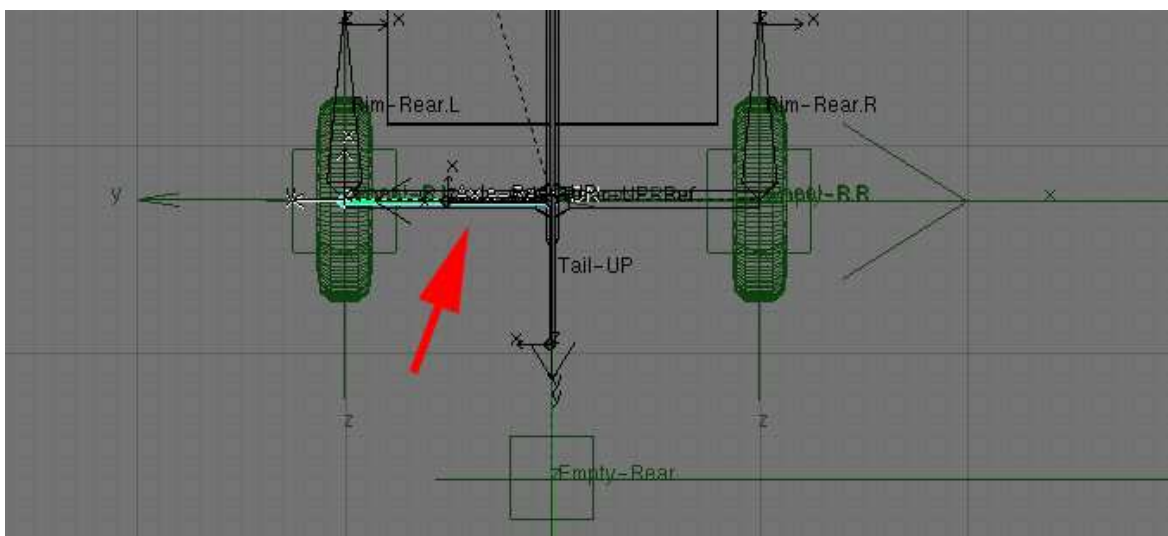
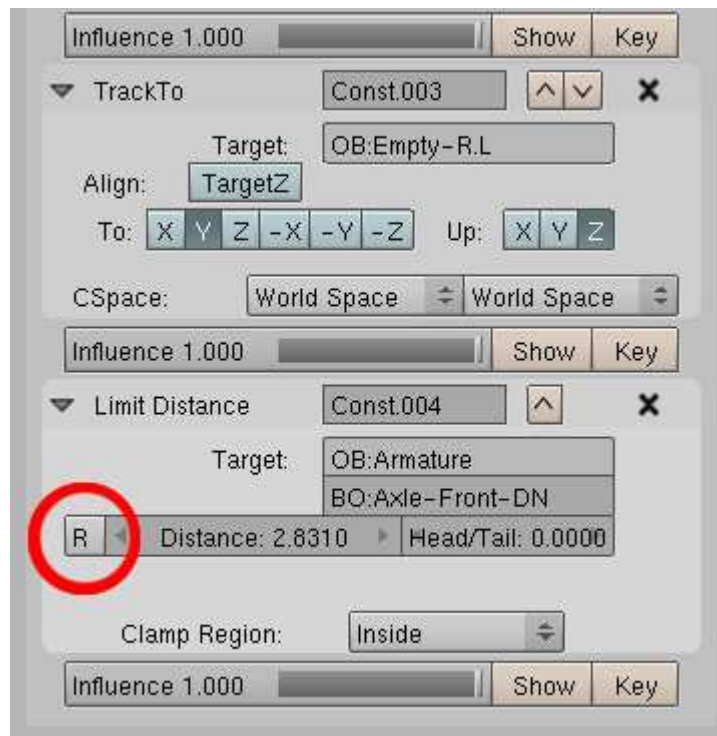


## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

f - In the constraints panel, look at the Limit Distance constraint : Click on the R button to reset the Distance limit. Now, the "Axle-Rear-DN" bone is at the good location, and the rear wheels should be at the good altitude.

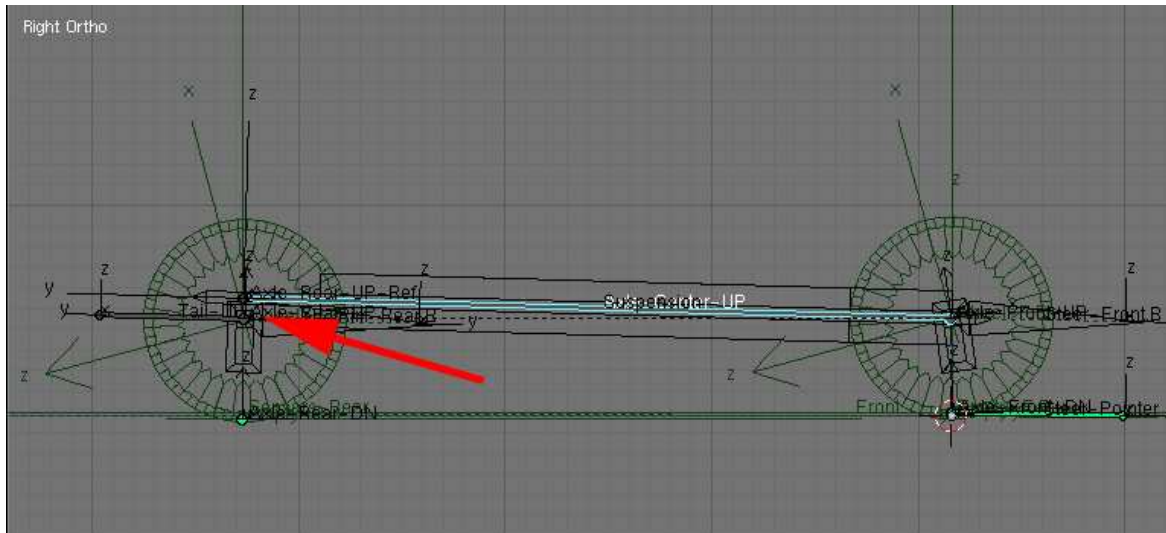


## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

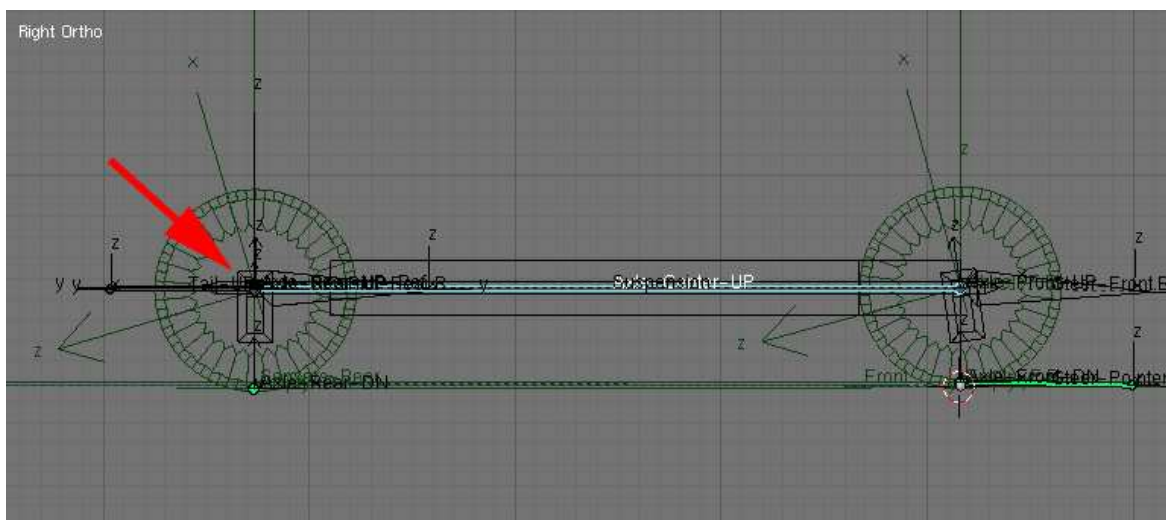
An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

g - Still in Pose Mode, look at the armature in side view : You may notice that the bone named "Axle-Center-UP" is no more horizontal.



Select it and rotate it on its X local axis to make it pointing at the "Axle-Rear-UP" bone. This time it should be horizontal, and the "Suspension" bone as well.



**NOTE :** if the rear wheels are not exactly at the same altitude as the front wheels, it can be corrected by selecting the Empties "Empty-R.L", and "Empty-R.R", and moving them slightly in the Z direction.

8 - Parent your Car Body model to the bone named "Suspension". This bone has a custom shape. It is the blue board with an arrow shape.

## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

### ADDING YOUR OWN WHEELS :

It is better to add your wheels AFTER resizing the setup.

- 1 - Select the bone named "Steer-Front.R", and type SHIFT+S. Choose Cursor -> Selection.
- 2 - Select your wheel model, and type SHIFT+S. Choose Selection -> Cursor.
- 3 - Parent your wheel to the bone named "Steer-Front.R".
- 4 - Repeat the same operation for the 3 other wheels, parenting them to bones "Steer-Front.L", Rim-Rear.R and Rim-Rear.L.
- 5 - **IMPORTANT** : Make sure that the rotation axis of your wheels is the Y axis. This condition is necessary for the script to work. The Y axis must also be pointing to the left, otherwise you wheels would turn backward. The dimension of the wheels on Z axis is used as Diameter by the wheelrotation.py script to calculate the rotation speed when the wheel is moving in the 3D space.

### ALTITUDE CORRECTION :

There are 3 (at least) methods to compensate altitude errors. As it acts directly on the wheels, method 1 is the most efficient.

- 1 - The perfect contact between the wheels and the Ground level can be adjusted thanks to the 4 Empties parented to the sensors. These Empties Empty-F.R, Empty-F.L, Empty-R.R, Empty-R.L, are in a Group named "Sensor-Adjust".

You can adjust the altitude of the four wheels in one time, by selecting the four empties, and moving them along the Global Z axis when the car rig is on a Flat Horizontal Ground.

If you have to adjust the wheels level on a bumpy terrain, in extreme stress conditions, you can use the local Z axis instead of the Global Z axis of the Empties. In this case, you may have to adjust each wheel independently.

**IMPORTANT** : If you do such correction, it is important to insert at least 3 Loc keyframes : One before and one after the range of frames where the correction is needed. These keyframe are necessary to isolate the correction area from the rest of the trajectory, to avoid propagating the correction in areas where they are not needed, and where the effect could be the opposite of the expected one.

Then, go to the frame in between, in the location needing the maximum of correction. Move the concerned Empties in ZZ (local) direction, to adjust the altitude of the wheel, and Insert a third Loc Keyframe.

Use the same method for each wheel needing a correction.

## **4x4 AUTODRIVE SYSTEM for Blender v2.48a**

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

In most cases, the Bezier interpolation of the Ipo is sufficient to give the appropriate correction to all the frames in the correction area. If not, just insert some more keyframes or adjust the handles of the Ipo Bezier curves.

2 - In most cases, wheels altitude issues are due to a shift of the Sensors meshes, especially in important transversal slopes and when the chassis (armature) is submitted to an extreme torsion.

A transversal shift of the "Sensors-Rear" mesh can be compensated by hand by moving the sensors mesh in its local direction by G+XX. Like in the previous method, it is necessary to insert a keyframe before and after the area where the trouble is visible, and to apply the shift correction in the middle of this frame range.

3 - You can also correct an altitude issue by constraining the rear wheels to pass in the same tracks as the front wheels. As it is not realistic in narrow curves (when the altitude issues mainly occurs), it is NOT RECOMMENDED. To do that, you have to make the Ipo of the "Rear" mesh a single user, and move this mesh on the local Y axis. The same precautions as previously must be taken : insert a Loc YY Keyframe before and after the correction area, and apply the correction by other keyframes in between.

One more time : use this method with CAUTION, and if you can AVOID it. It is the worse of the 3 methods.

### **EDITING THE TRAJECTORY :**

In TOP view, just edit the curve at will. Remember that very narrow curves will give less accuracy on the contact between the wheels and the ground.

The speed of the car and its starting point on the trajectory can be adjusted by playing with the dLocY Ipo curve named "Speed on Trajectory". This Ipo is shared by the small square meshes following the curve.

The default dLocY Ipo curve given in this example is set to linear, but can be set to bezier to get smooth speed variations.

### **CHANGING THE INITIAL DIRECTION OF THE CURVE :**

In this blend file, the initial direction of the curve use as guide for the trajectory is the Y axis. You might want to change that :

1 - Select the curve and type SHIFT+S. Choose Cursor -> Selection.

2 - Select all the objects belonging to the Car-Setup Group.

3 - Rotate the whole group around the Curve center point, using the 3D Cursor as rotation point.



## **4x4 AUTODRIVE SYSTEM for Blender v2.48a**

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

4 - Select one of the square meshes following the curve, and change the "Speed on Trajectory" Ipo curve from dLocY to dLocX.

5 - If you notice that the curve no longer passes by the center of the rig :

a - Select the square mesh named "Rear" (the one supporting the Empty-Rear).

b - Type SHIFT+S and choose Cursor -> Selection.

c - Select the Curve and bring the Curve origin on the 3D Cursor : SHIFT+S and Selection -> Cursor.

6 - If you have rescaled the scene, or at least the setup in the scene, you will have to edit the Ipo curve to adjust the speed to the new world.

### **ADDING OBSTACLES ON THE TRAJECTORY :**

There are at least 3 methods to add obstacles on the terrain :

1 - Making an obstacle in the same mesh as the terrain mesh. These obstacles can be extruded from the ground, or can be separated from it. In the second case, the ShrinkWrap modifiers will have some difficulties to detect the exact level of the obstacle because several layers of mesh faces will give an average altitude detection. So, if it is possible, it is better to delete the vertices of the ground below the obstacle.

2 - The ShrinkWrap modifiers can use two objects as references, so a second object can be used to create obstacles.

3 - It is also possible to use an ordinary object as apparent obstacle. In this case, you will have to create an invisible obstacle (material with alpha) in an object detected by the ShrinkWrap modifier. This solution helps the wheels to climb smoothly over the obstacle. It is very useful when the obstacle has sharp edges.

### **PRINCIPLE AND USE OF THE TRACKS CREATOR :**

The Tracks Creator is a feature allowing to create tires prints (tracks) after the wheels of your animated vehicle.

The setup can be used to create animated textures of the tires tracks. Usually, these textures are made of sequences of images.

Everything needed to create the tires tracks is on Layers 11 and 12.

Layer 11 : Grey Background Screen, Particles Emitters, and Orthogonal Camera.

Layer 12 : Particles Objects : small mesh used as Tire Print Sample.



## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

NOTE : You will notice that there is an offset between the Print Sample and the Particles Emitters. This seems to be a bug, and I have seen a report on the Bugtracker page about that. So, if it is fixed in Blender 2.5 or in any future release, a small correction could be necessary in this setup.

The JPG format is well suited for this use. The texture that you will render for your animated tires tracks will have to be detailed enough, but the compression factor is not an issue, because the tracks don't need to be very accurate, and you may even need to blur them. The image of the texture will be rendered with a high resolution. 2048x1536 is enough in most cases if the Length/Width ratio of your terrain mesh is 4/3.

As the image will have to be UV mapped on the terrain mesh, and also used by a Displacement modifier if you want the tracks to be in real relief, the X and Y dimensions of the rendered texture will have to perfectly match the size of the terrain.

So, we will use a camera set to Orthogonal, to capture the images of the animated tracks. This camera will be put right over the center of the terrain mesh (See Layer 11), and will be adjusted to get the entire surface of the terrain mesh exactly inside the bounds of the camera field.

When you will have rendered the sequence of images in the wanted resolution, you will have to load the first image of the rendered sequence in the UV/Image Editor. Then, you will select all faces of the terrain mesh and type "U" with the cursor in the 3D window.

By selecting "Project from view (Bounds)", you will apply UV coordinates to the terrain mesh. After that you will have to add the animated texture to the Terrain material, and set the texture coordinates to UV. Then, you will be able to use the animated texture with various parameters of your choice, in the Map To panel, as usually.

The animated texture will also be usable for a real displacement, by adding a Displace Modifier to the terrain mesh. You will use UV coordinates in the Modifier as well.

Using a Displace Modifier allows very realistic tracks, but requires a highly subdivided terrain mesh. So it is very consuming in RAM memory, and the animation can be slowed down. We'll see below that a turnaround can be used to avoid crashes due to high memory needs.

WARNING : If you have added some grass on your terrain by using hair particles, you will certainly want to control the length of the grass in the tires tracks to simulate the grass crushed by the car wheels. At this point, you will probably encounter some troubles, because of a bug in the particles system, not allowing to refresh the texture at frame change. Don't worry : This setup includes a very small but efficient Python script written by "Atom", solving this problem while waiting for a real bug fix in a future Blender release. I have named his script "GrassParticlesDebug.py" in this setup. To use it, the object on which the animated texture is applied must be selected. Obviously, this object will be the Terrain mesh (Ground). This script Enter and Exit Edit Mode for the selected object, forcing the Particle System to refresh and so to use the new image in the sequence.

## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

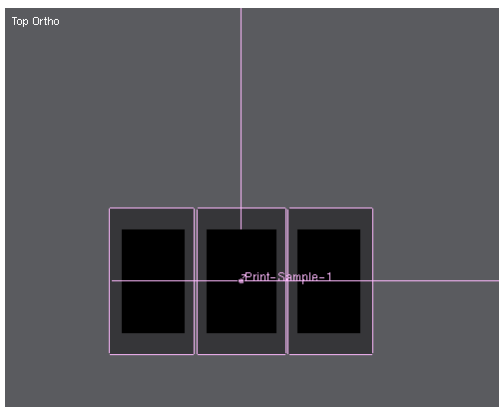
---

Atom's script to fix the particles refresh bug :

```
import Blender
from Blender import Window,Scene

print "->Frame Change"
scn = Scene.GetCurrent()
obj= scn.getActiveObject()
Window.EditMode(0)
Window.EditMode(1)
Window.EditMode(0)
```

Under the terrain mesh, we will put a large plane with a material set to a medium grey color (R=127; G=128; B=127). RGB values of 128 can also be used. This Grey color is neutral when used with a NOR map parameter in the Map To panel. This allows to add bumps and holes with the texture, by using darker or lighter value in the greyscale, between black and white. This large grey plane will be used as a "Background Grey Screen".



The animated texture of the tires tracks will be made by rendering tracks leaved by four particles emitters parented to the rear axle of the armature (the "Axle-Rear-UP" bone), and the two bones named "Steer-Front.R" and "Steer-Front.L".

These particles emitters are the simplest meshes ever built : they are made of ONE SINGLE VERTEX. Enable the axis and name in the 3D view, otherwise you will have difficulties to see and to select them. As the emitters have no faces, you will set the Emit From parameter to Verts

(VERTEX).

The particles emitters will be put at the ground level, and at the X and Y location of the wheels. Remark that the front emitters are parented to the bones supporting the wheels. Experiments have shown that in some animations better results have been achieved with the tracks emitters only following the movement of the axles, and not rotating with the Steering. So, it is up to you. If you are not satisfied by one method, feel free to experiment the other way !

The particles type of the particles systems will be set to objects. The particles will be small meshes with materials in greyscale color. These small meshes will be used as samples, and will represent a small part of the tires surface.

The provided Print Sample is made to match the relief of the provided wheel model. You will be able to edit the little mesh or to create other prints sample to match your own tires design.

The number of particles for each emitter will be set to the same number as the frames number in the animation. This way, at each frame, a single print sample will be dropped by each particle emitter, just below the tires.

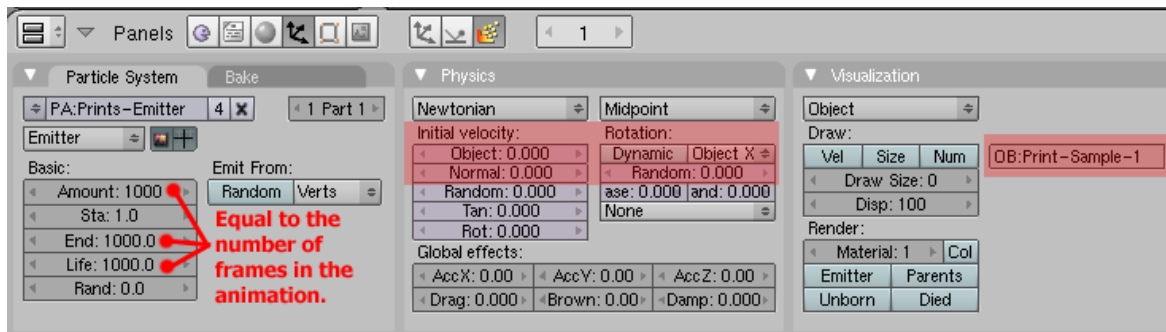
## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

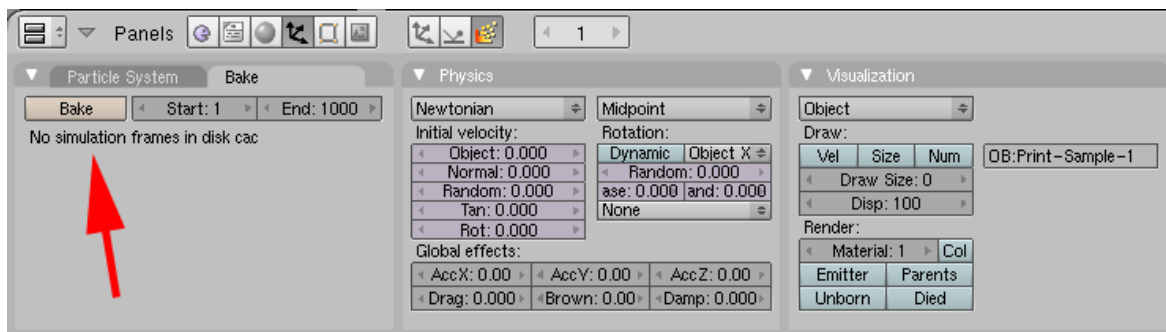
---

As we don't want the tracks to vanish during the scene shot, the life of the particles must be set to at least the duration of the animation. The baking range of the particles system must also match the duration of the animation.

All the emitters will share the same Particles System, but if you create various print samples, you will have to select the sample object FOR EACH EMITTER, because objects used as particles are linked to the emitter itself, not the Particle System.



To create the tracks, just select one of the four Particles Emitter and click on the "Bake" button in the Bake panel of the Particle System. Make sure that the Baking range matches the animation range.



During the process of creating the animated texture, the terrain mesh is not necessary. When baking the Particles, the terrain can even be deleted. Deleting the terrain mesh doesn't affect the animation, because for the ShrinkWrap modifier, the terrain is still kept in memory.

The positive thing is that deleting the terrain will free some memory somewhere (except for the ShrinkWrap Modifier), and will allow the baking process of particle to avoid crashing. It is very useful when the terrain is large and highly subdivided, as it is generally the case when using a Displace Modifier.

**WARNING : DO NOT SAVE THE FILE WITH THE TERRAIN DELETED.**

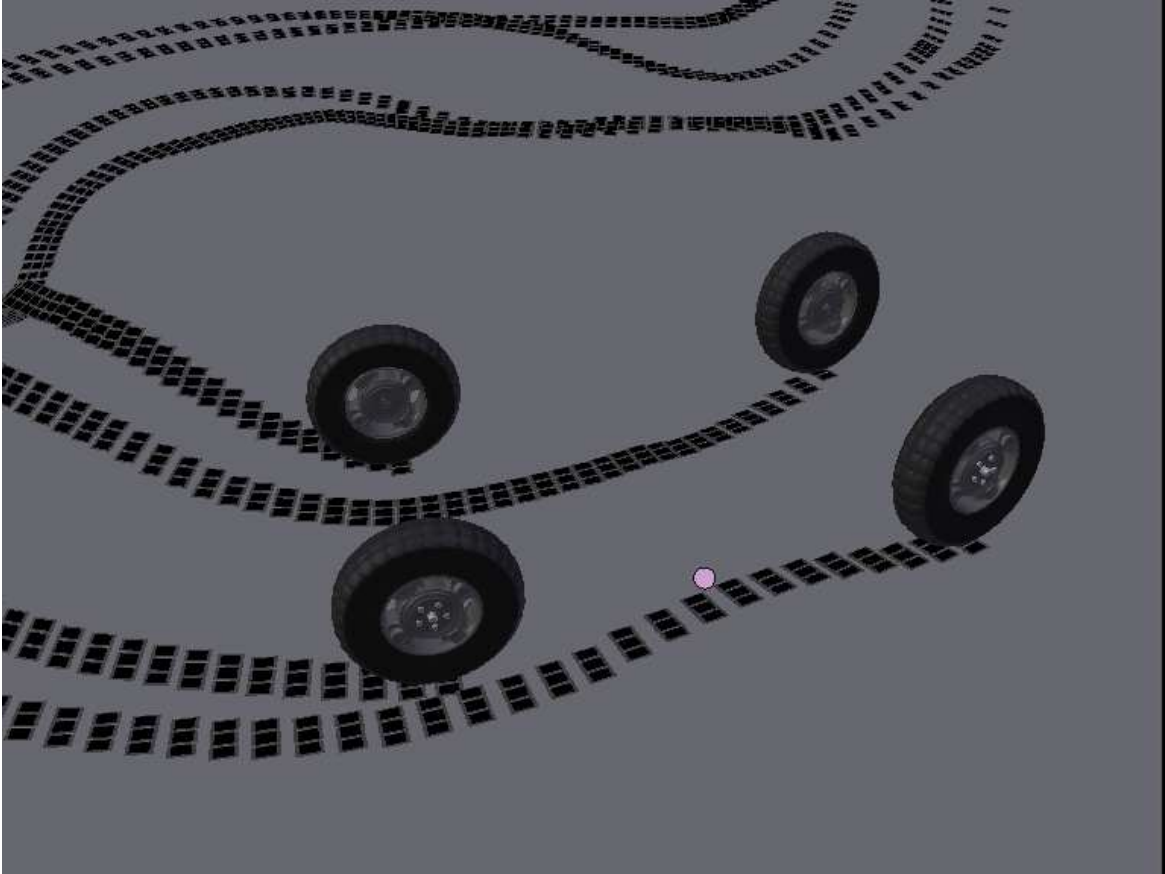
It is not necessary to save the blend file after baking the particles, because the particles are saved in cache, anyway. Close the blend file and open it again, to recover the terrain mesh.

---

## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---



When the particles are baked, you can render the animated texture. At this point, you must hide the terrain mesh. It mustn't be rendered, and will be on a layer that will not be selected when rendering. The goal is to render with the orthogonal camera only the prints samples over the neutral grey background.

On the image, you can see some gaps between the prints samples. It is mainly because of the perspective. As the texture is rendered in top view and with a camera set to Orthogonal, these gaps are filled in the animated texture. However, some little gaps can occur sometimes. If this happens and the gaps can be reduced by enlarging a bit the print sample mesh in edit mode.

No lamps are necessary to render the animated texture, because the materials of the grey background screen and the materials of the prints samples will be set to Shadeless.

If you want to get smoother tires tracks, you can enable the "Do Composite" button, and adjust the values of the blur filter, in the Compositing panel (Screen 7), before rendering the sequence of images.

**IMPORTANT :** When you will apply your animated texture to a Displacement Modifier, you will be able to see the terrain mesh deformed in the 3D window. As the " Altitude Sensors " are following the ground level, the real deformation of the terrain mesh is detected by the ShrinkWrap Modifiers. This has an influence on the Z location of the wheels, and is



## 4x4 AUTODRIVE SYSTEM for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

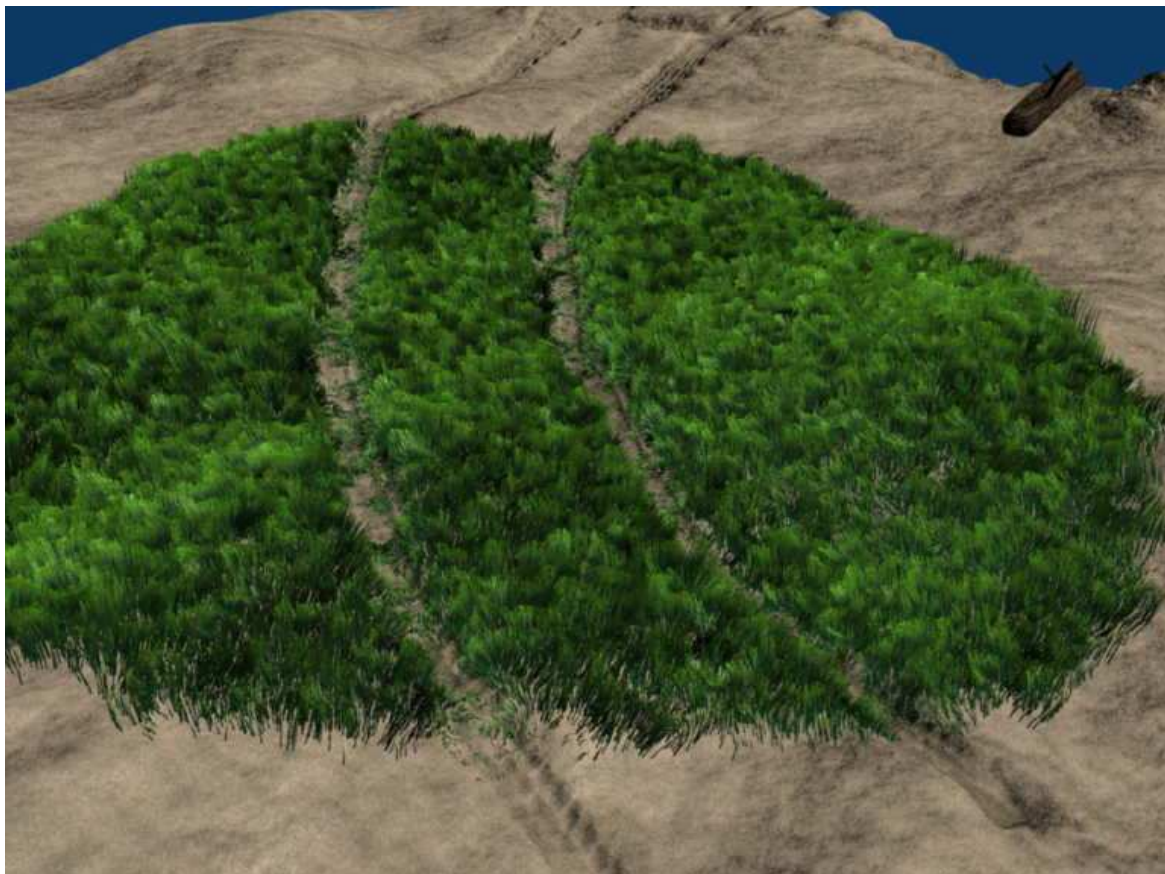
---

mostly visible on the front wheels. You can add a frame delay of + 1 or - 1 frame on the image sequence of the tracks texture. Doing that, you move the first print sample after, before or right under the center point of the front wheels. The visible effect is the adjustment of the Z position of the front wheels in the depth of the tracks.

You will notice that the real relief of the displaced ground mesh introduces some vibrations in the wheels. These vibrations can be adjusted by tweaking the contrast of the animated texture, and can add realism to the animation.

As I told you at the beginning of this manual, due to several bugs in the particle system in Blender 2.49a and 2.49b, the Tracks Creator doesn't allow to render the animated texture based on particles of Object type. It sometimes work in simple files but doesn't seem to work with the file of this setup. So, please, use Blender 2.48a.

### **EXAMPLE OF WHAT CAN BE DONE WITH BOTH DISPLACEMENT MODIFIER APPLIED TO THE GROUND MESH AND GRASS MADE OF HAIR PARTICLES :**



### **LAST WORDS :**

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

## **4x4 AUTODRIVE SYSTEM** for Blender v2.48a

An automated setup allowing your car model to run on all kinds of terrains, with automated ground level and obstacles detection.

---

If you use this setup, the provided models or the script in your animations, please don't forget us in the credits.

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

Best regards,

Philippe.