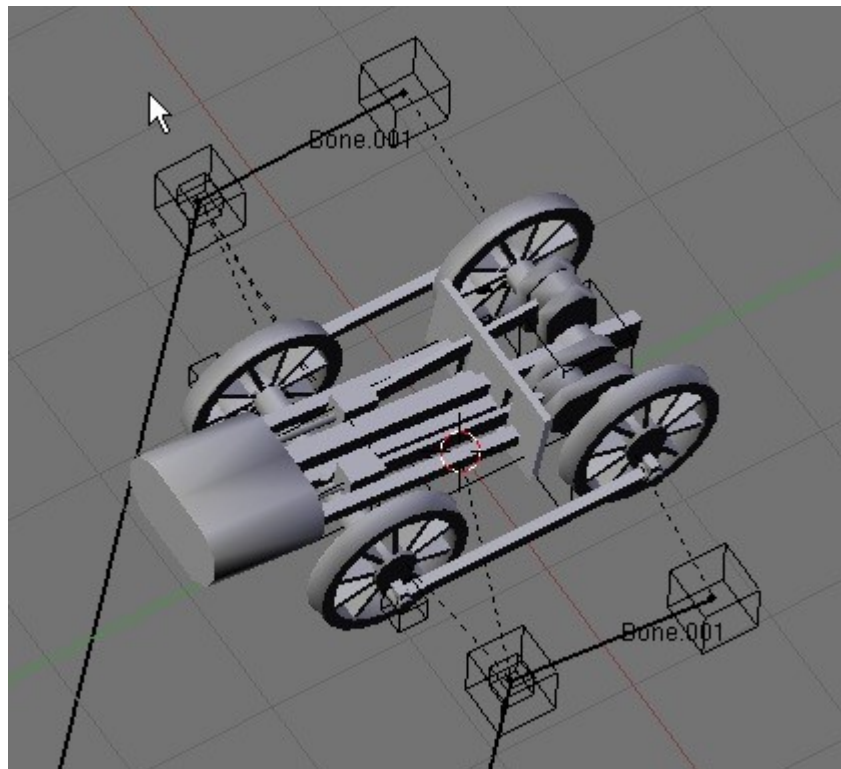


A Blender Tutorial

Building a Simple Loco Part 3 Animation and Exporting © Paul Hobbs 2009



Contents

Simple Animation.....	3
Creating Dummies.....	3
Linking The Bogie.....	5
Unlinking Objects.....	5
Animating the Wheels.....	6
Rotation Constraints	6
The IPO Curve Editor Window.....	7
Exporting the Mesh.....	8
Animation Using Bones (Armatures).....	9
Animating the Right Hand Connecting Rod and Piston.....	16
And Finally.....	17

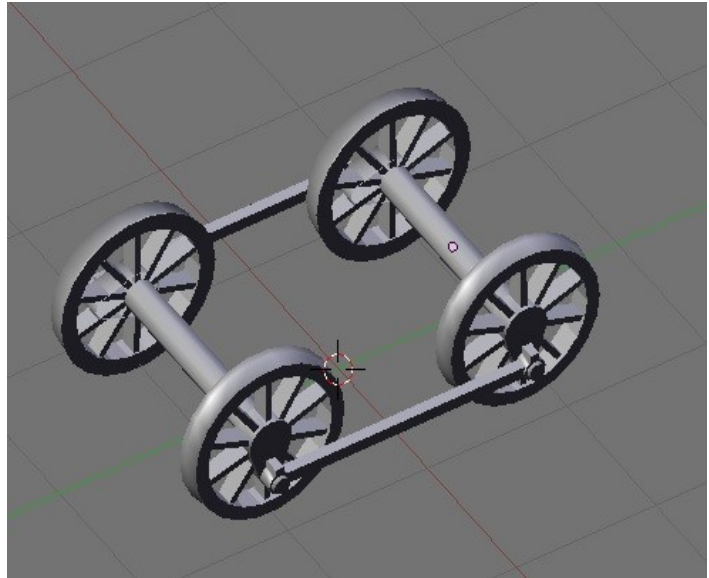
Simple Animation

The simplest vehicle to animate would be a goods wagon, a passenger coach or a diesel loco bogie as these don't have coupling rods.

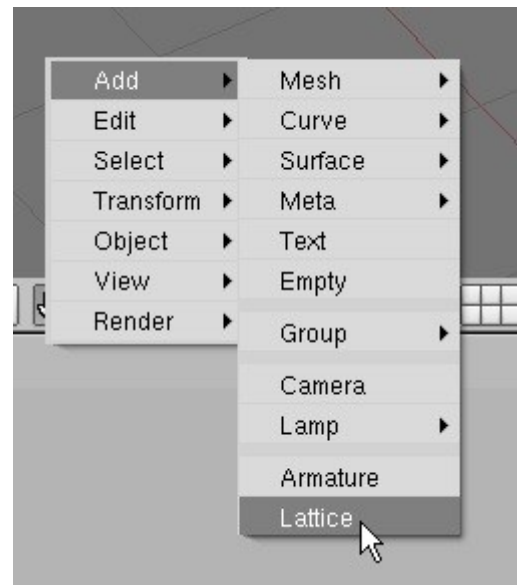
Creating Dummies

We'll start by animating the 'bogie' for the LNER Y7 0-4-0. Here it is, split up into four separate meshes: w1 and w2 (wheel 1 and 2) and rod_l and rod_r (coupling rod left and right).

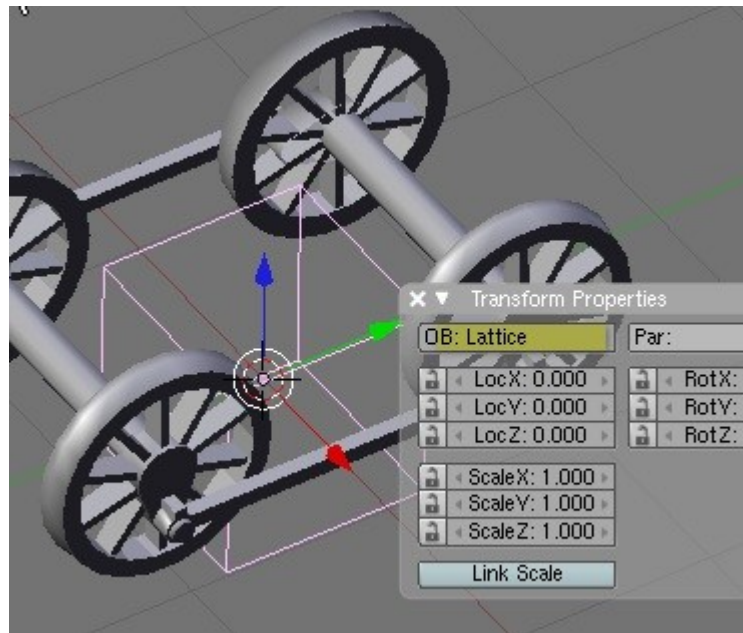
The animation in Blender is very similar to animating with dummies in GMax and Max, so if you've done a few bogies with them you shouldn't have any trouble. As in GMax everything has to be linked to a base dummy called (usually) b.r.main, so we'll start by creating that.



Blender doesn't actually have dummy objects so a 'Lattice' object is used instead. Make sure that the 3D cursor is positioned at 0,0,0 by pressing SHIFT-C and then C and create a Lattice by pressing SPACE-> Add → Lattice.



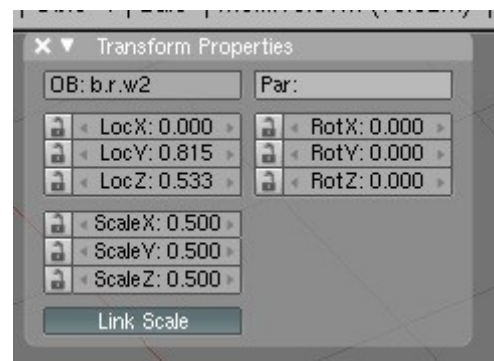
Rename the Lattice object to 'b.r.main' by pressing N and typing the new name in the field highlighted here. **The 'b.r.' part of the name is vital! Failure to use this convention will result in an exporter error!**



Centre the 3D cursor on each of the wheelsets and add two more Lattices called 'b.r.w1' and 'b.r.w2'. The naming convention after the 'b.r.' part is unimportant, but it makes things easier if the dummies have the same names as the meshes.

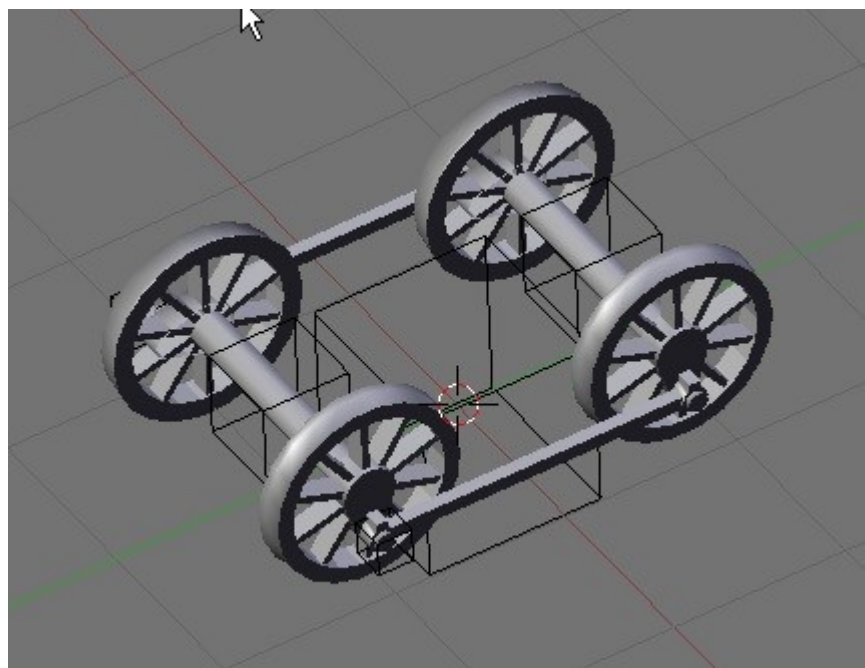
I've scaled the Lattices down to 50% of the original size to keep things tidier. Note that the 'Link Scale' button is selected so that X, Y and Z are locked to the same scale. This means you only have to change one of the fields to uniformly scale an object.

After doing this hit CTRL-A and choose 'Scale and Rotation to ObData' to perform the equivalent of a 'Reset Xform' on the Lattice. This may not be necessary but it's better to avoid nasty surprises later.



Now create two more dummies 'b.r.rod_l' and 'b.r.rod_r' on the centres of the front crankpins. These I scale to 20% of the original size.

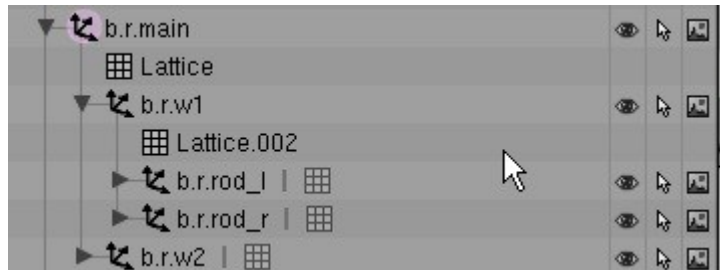
Here is the bogie with all of the dummies created. It's not necessary to create these objects in 'Top' view thankfully, as is the case in GMax and Max. Obviously if you have no connecting rods you don't need the Lattices on the front crankpins.



Linking The Bogie

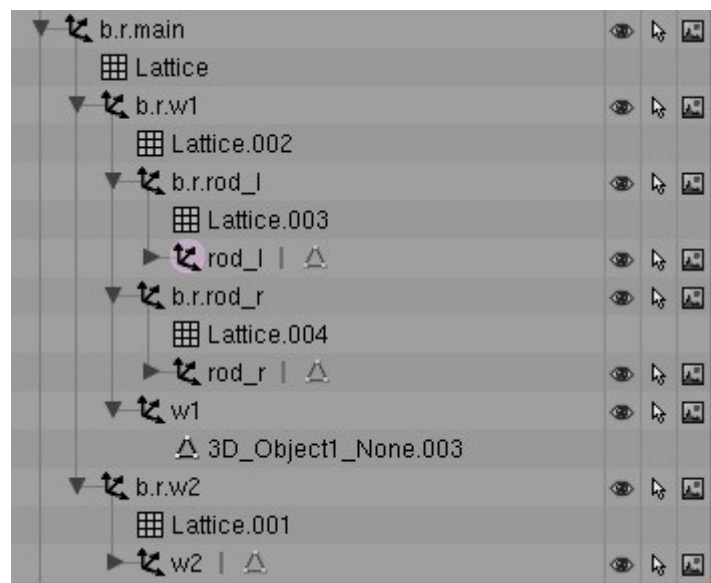
This is very easy in Blender, just select the things you want linked making sure that the last object you select is the parent object, then press CTRL-P and choose 'Normal Parent'. You have to link things in the correct manner though, which is b.r.rod_l and b.r.rod_r to b.r.w1 and b.r.w1 and br.w2 to b.r.main.

In the 'Outliner' window you can see the linking, and also in the 3D view by default dotted lines are drawn to show the linking.



All well and good, but we still have to link the meshes to the appropriate Lattices. This is pretty easy as the meshes and dummies have the same names.

Here is the Outliner after linking the meshes to the Lattices:

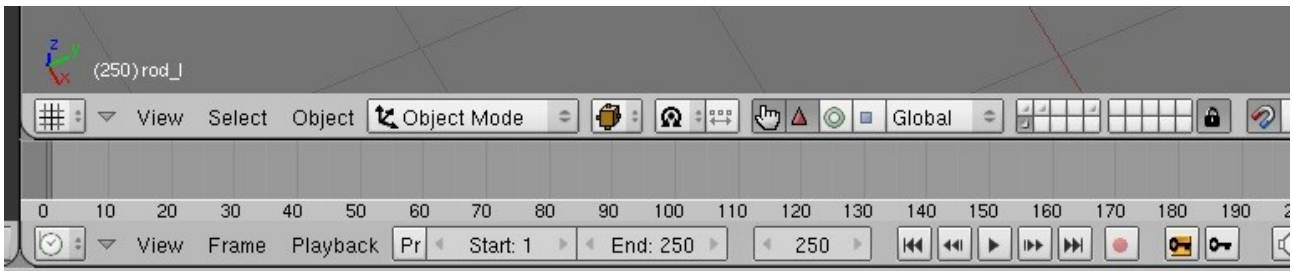


Unlinking Objects

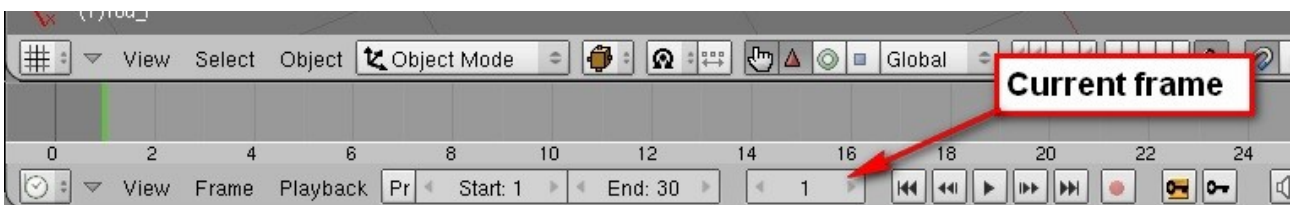
Sooner or later you'll link things together wrongly, but all you need to do is select the object and hit ALT-P with the mouse cursor over the 3D window.

Animating the Wheels

This is also very similar to the process in GMax, animation keys have to be defined for the wheel Lattices. Split the 3D window horizontally and make the new window type 'Timeline'.



Change the 'End:' field to 30, as we want an animation over 30 frames. Choose 'View All' from the View menu to display just frames 1 to 30.



Select the Lattice 'b.r.w1' – **never, ever, animate the mesh object directly, always animate the Lattice!**

Press I to define a key and choose 'Rot' from the menu that appears. In the case of frame 1 we don't need to press I again as we are not moving or rotating anything (this is the starting position/rotation). The green line at the frame 1 position of the Timeline will turn yellow to denote a key has been defined.

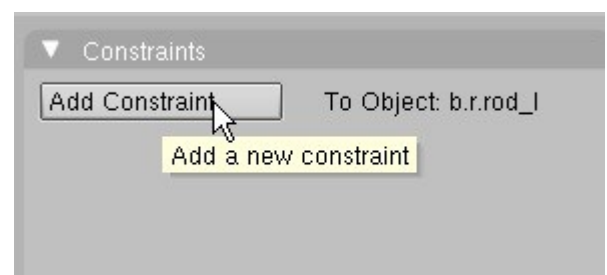
Frame 30 also needs a key, the angle should be 360° so move to frame 30, press I, rotate the Lattice 'b.r.w2' and then press I again.

Repeat for however many wheel sets your bogie has.

If you now play the animation (ALT-A or the 'Play' icon in the Timeline window, ESC to stop it) you'll notice two problems. Firstly, the coupling rods are rotating with the wheels and are not staying horizontal. This is easy to fix by adding a rotation constraint.

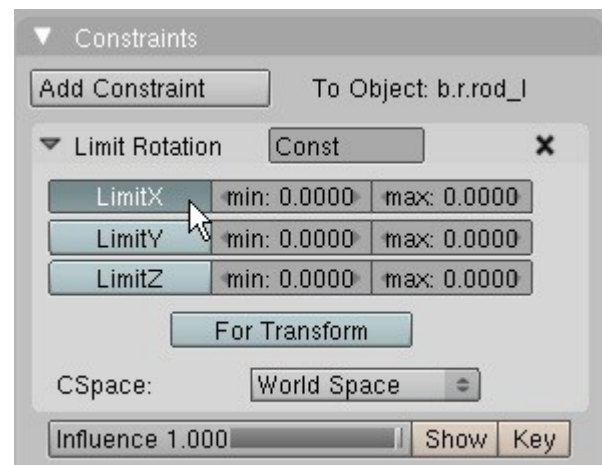
Rotation Constraints

To keep the coupling rods horizontal we need to add a constraint. With the Lattice 'b.r.rod_l' selected press F7 and choose 'Add Constraint'.



From the menu choose 'Limit Rotation' and then select the 'LimitX' button.

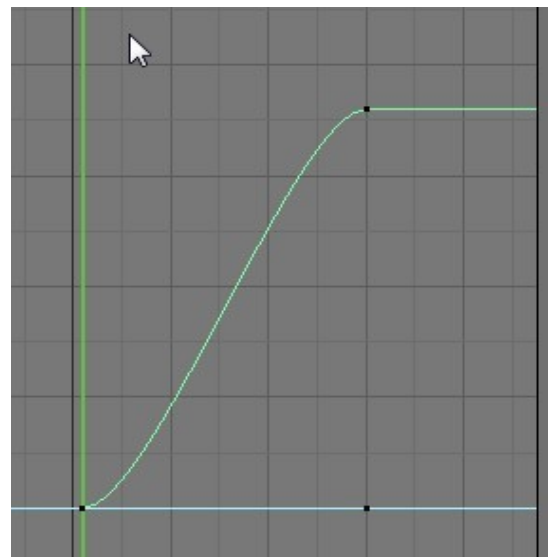
That's it, the coupling rods now stay horizontal. Repeat for the 'b.r.rod_r' Lattice.



The IPO Curve Editor Window

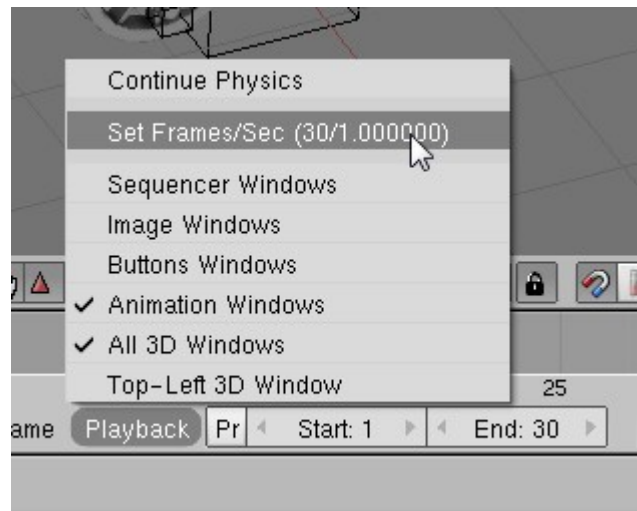
The second problem you will have noticed is that the animation is not smooth. Create another window of type 'Ipo Curve Editor'. Choose 'View All' from the View menu of the curve editor window to see the whole curve for frames 1 to 30.

The curve looks like this:



The uneven rotation is caused by the fact that the curve is not linear, but this is easy to fix by editing the points on the curve. Right click on the curve to select it and then press T. Choose 'Linear' from the 'Ipo Type' menu – the graph will change to be a straight line, i.e. a linear transition. When you play the animation again you may notice a slight pause at the frame 1 position, but the exported animation runs smoothly.

Before exporting the bogie we just need to check that the animation speed is correct, so open the 'Playback' menu and choose 'Set Frames/Sec'.



Change the number of frames to 30 (default is 25).



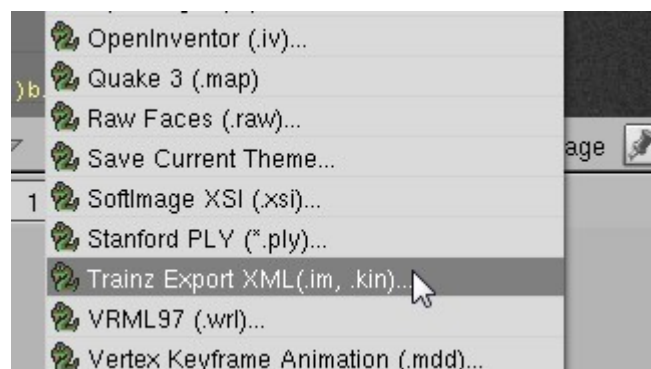
Exporting the Mesh

To export a mesh from Blender it is necessary to install two extra files in the Blender 'scripts' directory found in <your Blender directory>/blender.

Name	Änderungsdatum	Typ	Größe
mesh_solidify.pyc	12.12.2008 11:31	PYC-Datei	7 KB
TrainzMeshImporter.exe	28.10.2008 15:39	Anwendung	284 KB
trainz_export_xml.py	24.10.2008 17:23	PY-Datei	55 KB

'trainz_export_xml.py' can be found [here](#) and 'TrainzMeshImporter.exe' [here](#). The last link is to a page in the TrainzDev Wiki, it will be necessary to register first to gain access to it.

You are looking for the link 'Media:TrainzImporter.rar'. Once you have the files simply copy them into Blender's 'scripts' directory and restart Blender if it was already running. If you did it properly the option to export *.im and *.kin files will appear in the File -> Export menu.

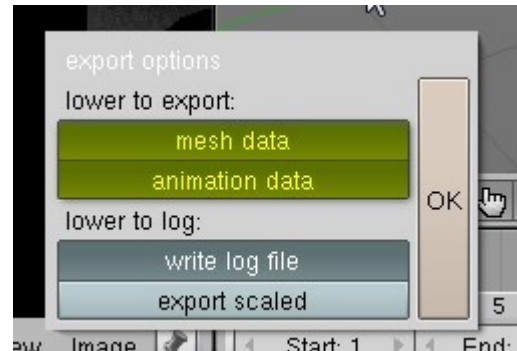


Before exporting you need to ensure that the model has been mapped, has materials applied and has any attachment points needed. The Trainz Exporter has pretty good error messages in case any of these conditions are not met.

The Trainz Exporter always works with what you can see in the 3D window, anything that is hidden won't be exported. In the case of this bogie we need to have the Lattices and the meshes visible and nothing else.

Choosing File → Export → Trainz Export will start the export process, opening the following Dialog Box:

For the bogie we need to select 'mesh data' and 'animation data', for the body just 'mesh data'. 'export scaled' applies a scaling factor to the exported mesh, by default this is set up for where you have created a mesh assuming 1 Blender unit = 1 foot instead of 1 metre.



Click 'OK' to start the export. The mesh will be exported by default in the same directory as the *.blend file and with the same name, but with an *.xml extension. Actually quite a few files are created, including the *.kin and *.im files as can be seen here. The *.xml file is a necessary intermediate step in the creation of an *.im file.

LNER_Y7_f.kin	22.02.2009 13:59
LNER_Y7_f.im	22.02.2009 13:59
LNER_Y7_f.log	22.02.2009 13:59
LNER_Y7_f.xml	22.02.2009 13:59

The *.kin file needs renaming to anim.kin for an animated bogie, the name of the *.im file needs to match whatever is in the config of the bogie.

The model including the animated bogie is available for study [here](#).

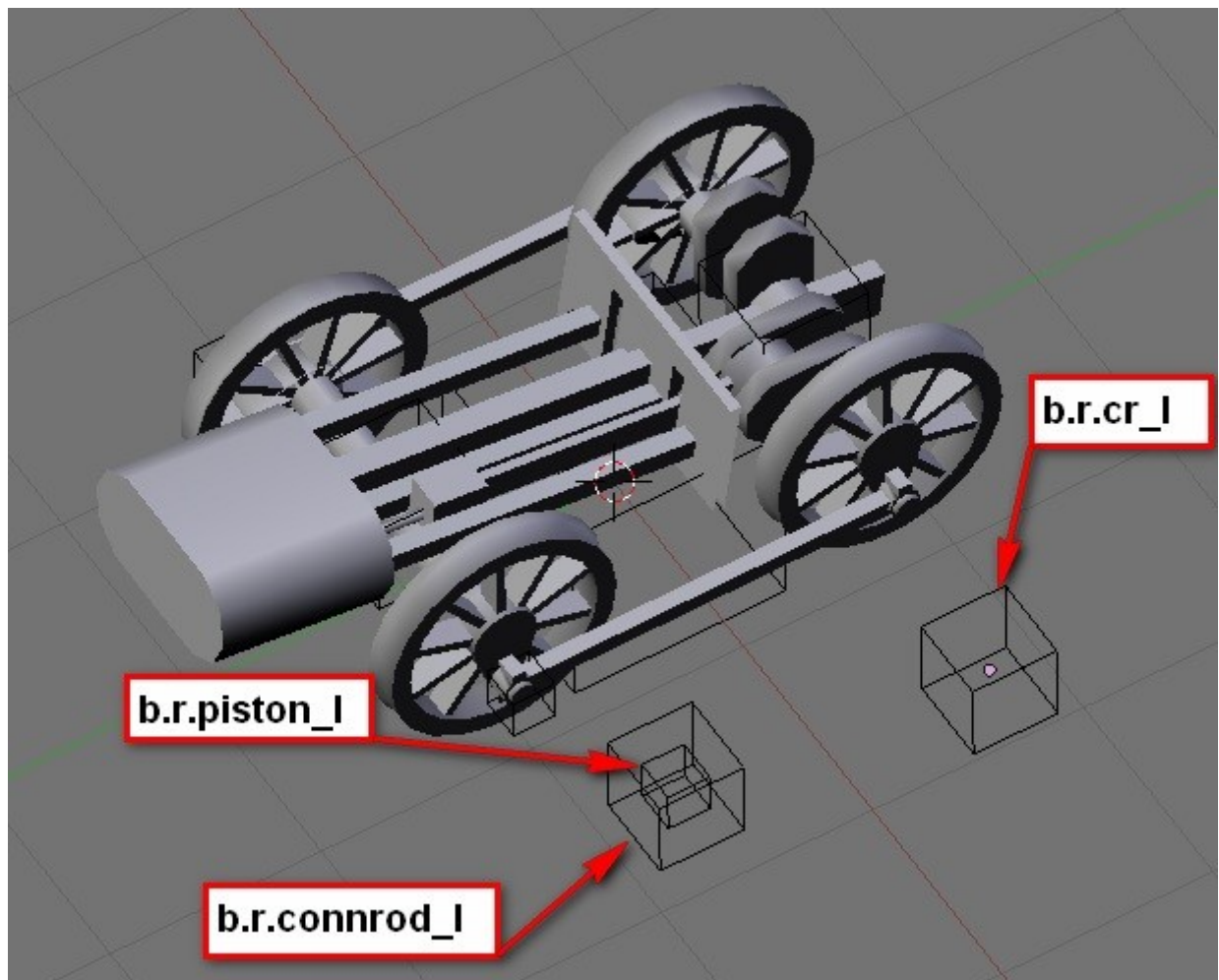
Animation Using Bones (Armatures)

For the situation with a simple bogie with just wheels and coupling rods (or no rods at all) as described above animation using Lattices is pretty straightforward. However, when connecting rods and pistons or outside valve gear is involved things get complicated very quickly when Lattices are used - the rotation and position of each of the Lattices needs to be calculated and keyframes generated for practically every frame.

Luckily another possibility exists which is to use bones, or as Blender calls them 'Armatures'. The Blender exporter in it's current form unfortunately doesn't handle Armatures, but we can use a simple trick to bind Lattices to the Armatures and then export those instead. This is a little more complicated than the procedure in GMax and Max but still far better than calculating the positions of the Lattices manually.

As an example we'll animate the inside motion of the Y7, outside connecting rods and pistons would be identical.

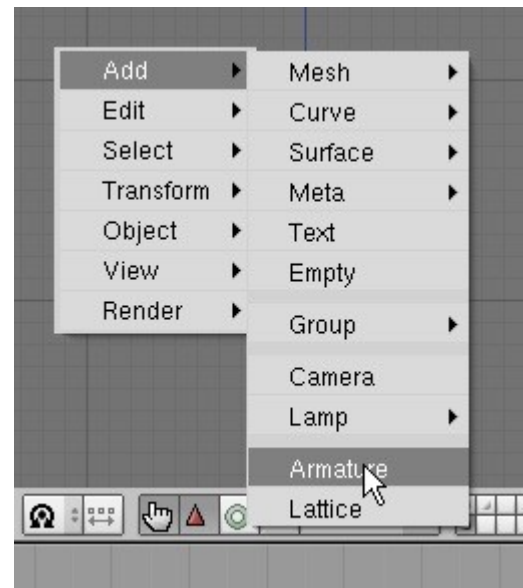
We'll start by creating the Lattices 'b.r.cr_l' (connecting rod big end left), b.r.piston_l' (connecting rod little end left) and 'b.r.connrod_l' (same position as 'b.r.piston_l').



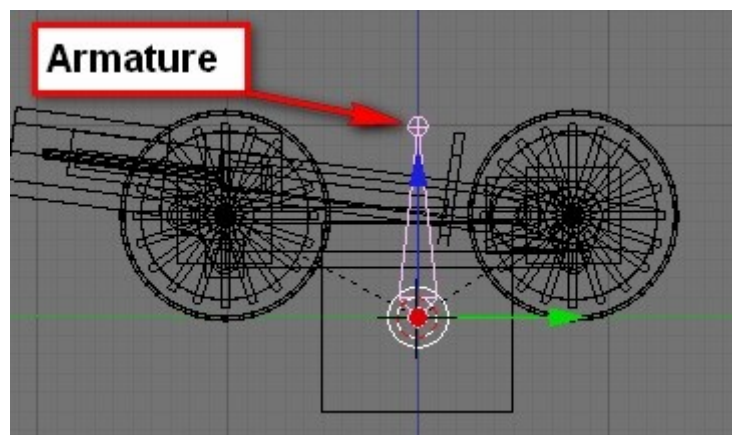
The inside motion meshes were taken from my detail parts library available [here](#). As the Y7 is so small the inside motion needed modifying to fit the available space, and also rotating by -7.5° in X to clear the front axle. I deleted the right hand connecting rod and piston as once the left hand one has been animated we can rotate it 90° and mirror it over to get the right hand rod and piston in the correct position.

Also note that the new Lattices shown here are positioned at X=2 units to make it easier to see what is going on once the Armatures have been added.

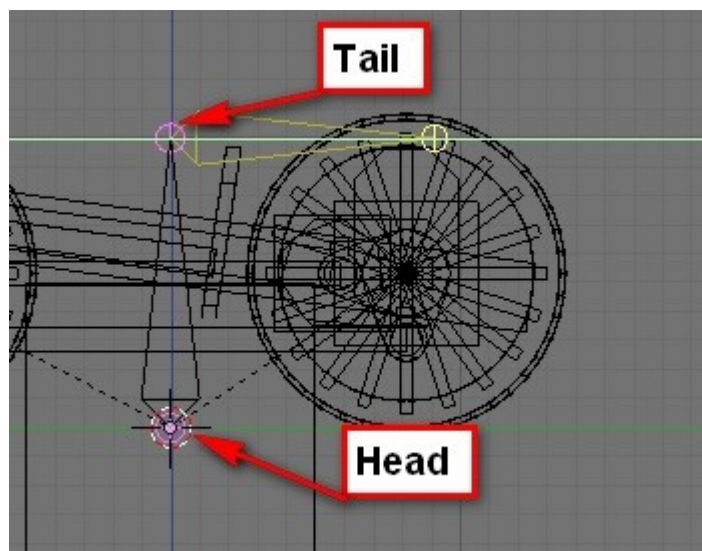
Add an Armature by pressing SPACE and then choosing Add → Armature with the mouse cursor over the 3D window.



The Armature is positioned at the 3D cursor position.



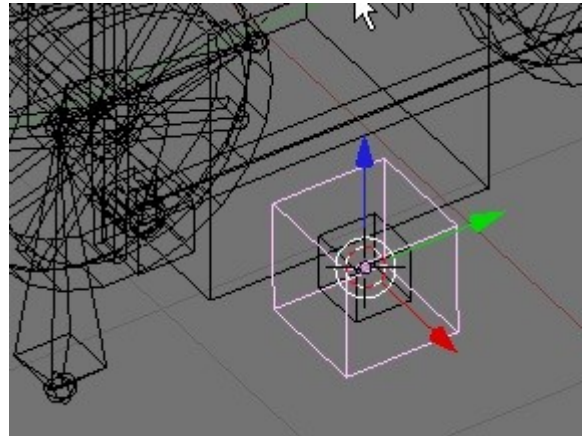
With the Armature selected, press TAB to enter 'Edit' mode. Select the 'Tail' end of the Armature and extrude another Armature in the Y direction.



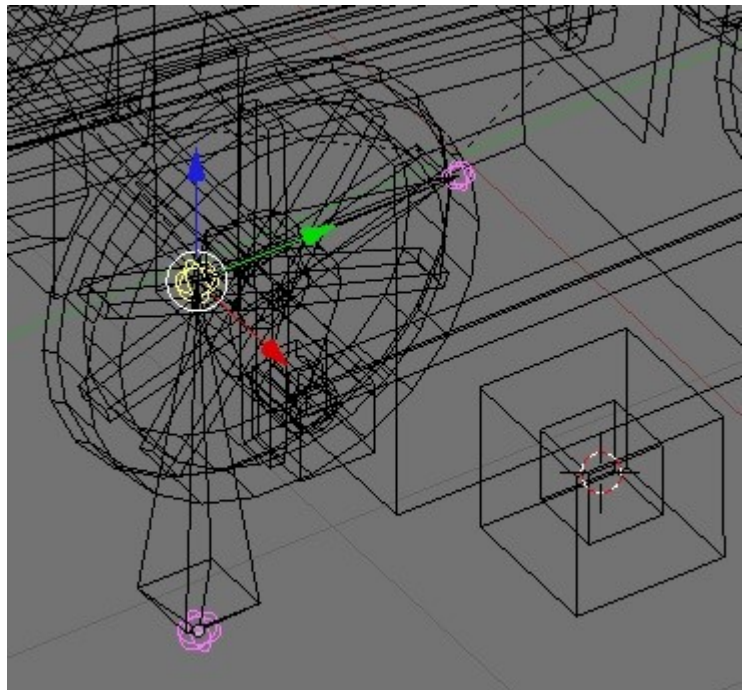
In 'Object' mode the whole Armature system can be moved around, in 'Edit' mode the end points or joints can be moved. The Armature stays connected when the head or tail points are moved. To get the Armature in the correct position we need to move the 3D cursor to the centre of the Lattices and then select a joint to move. Using SHIFT-S and 'Selection →

Cursor' the joints can be positioned.

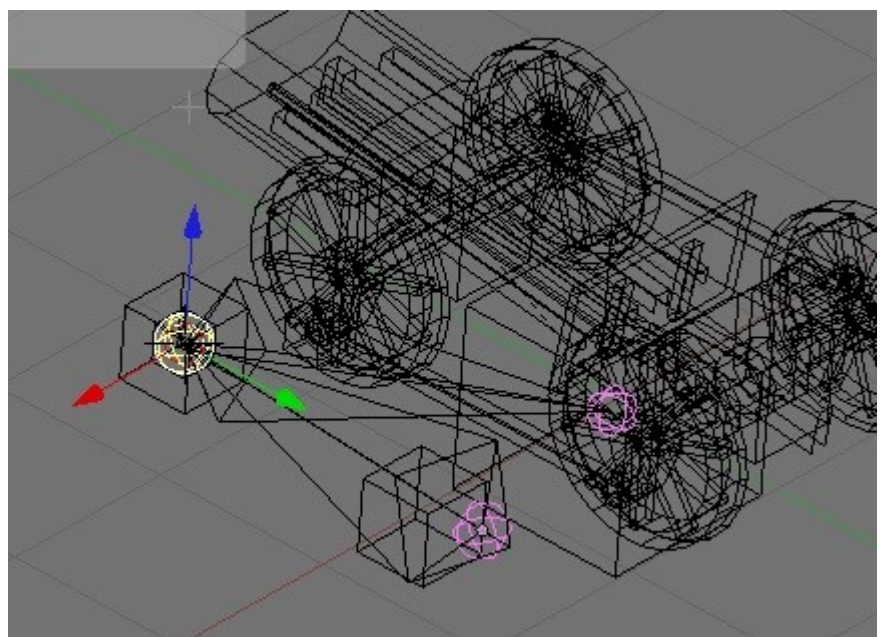
Begin by selecting 'b.r.connrod_l' and then SHIFT-S and choosing 'Cursor → Selection'.



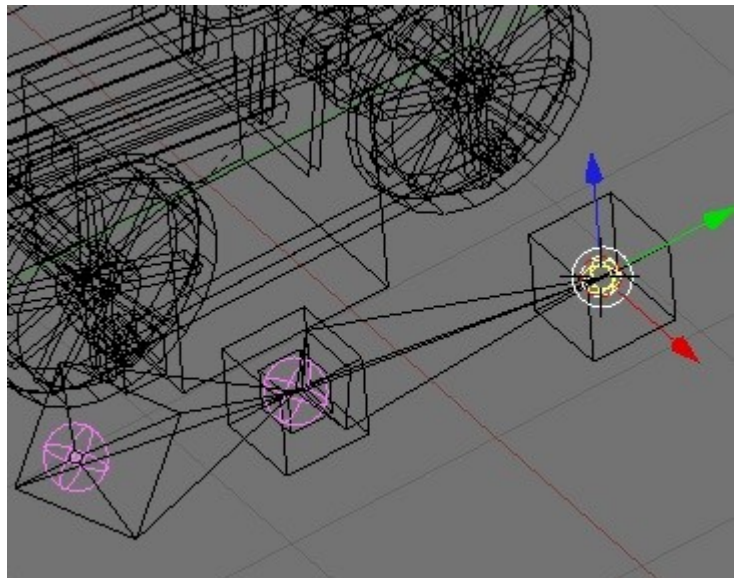
Now select the Armature and enter 'Edit' mode. Select the joint shown and press SHIFT-S and then choose 'Selection → Cursor'.



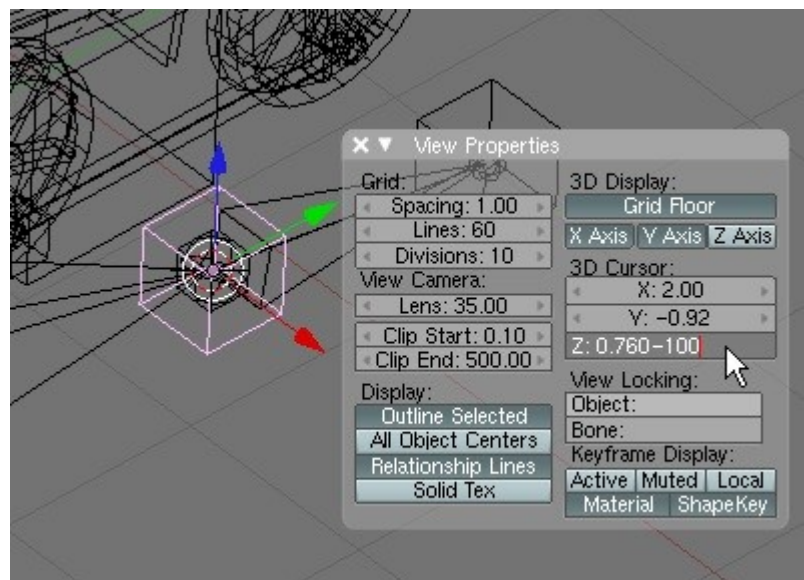
The joint snaps to the 3D cursor.



Repeat for the joint at the connecting rod big end.



Now move the 3D cursor to the centre of 'b.r.piston_l' -100 units **if the cylinders of the loco are horizontal**. In the case of the Y7 they are inclined at 7.5° so we need to do some extra steps.

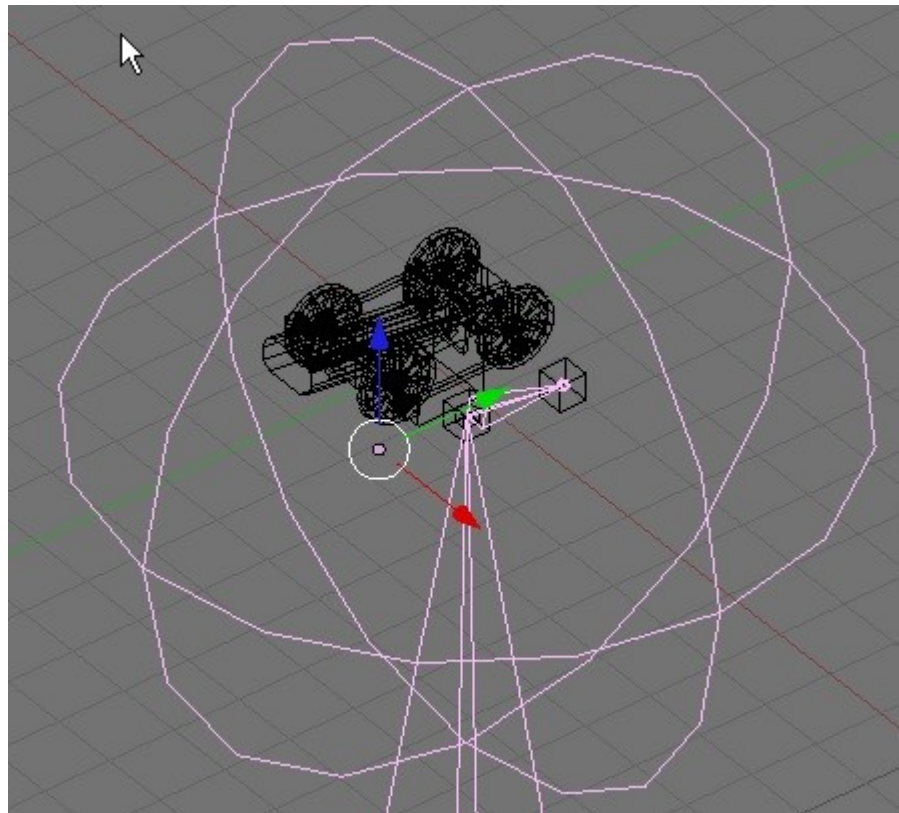


For inclined cylinders make a copy of the 'b.r.connrod_l' Lattice by pressing SHIFT-D. Enter 'Edit' mode and select all the vertices of the new Lattice 'b.r.connrod.001'. Translate the vertices in the -Z direction by 100 units. This will move the Lattice itself but leave its pivot point in the old position. Now switch to 'Object' mode and rotate 'b.r.connrod_l.001' by -7.5°. Select 'b.r.connrod_l.001' and with SHIFT-S move the 3D cursor to the Lattice centre.

The head end of the first bone that we added now needs moving to the 3D cursor.

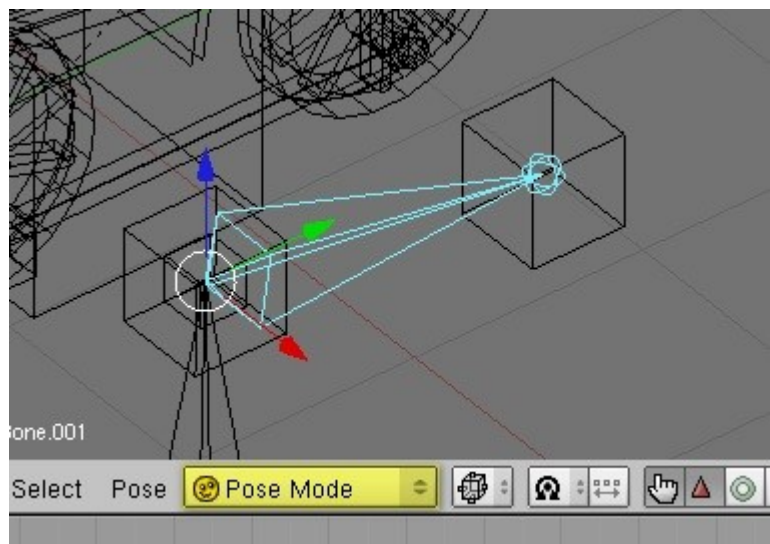
If the cylinders are inclined the Lattice 'b.r.connrod_I.001' can now be deleted, it was only needed to get the 3D cursor in the correct position.

Incidentally the display of the Armatures is not ideal as can be seen here. With the Armature selected you can press F9 to change the display to 'Stick' for example and also display the names of the individual bones which is useful.

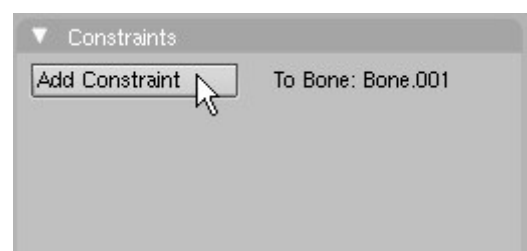


The reason that the first point of the Armature is 100m below ground level and normal to the piston rod is that the 'b.r.piston_I' Lattice will move in a very flat arc which will give the effect of a linear movement of the crosshead, a trick that was suggested by Bill Fock. This makes handling inclined cylinders much easier.

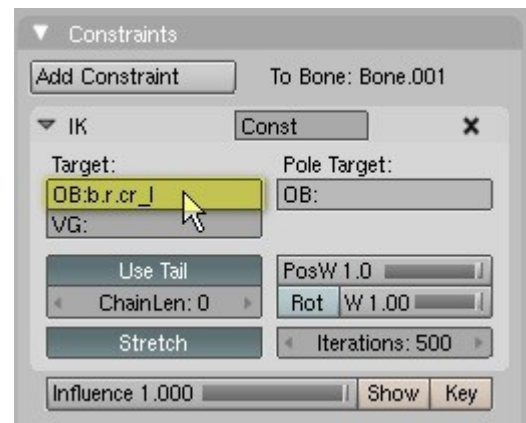
With the Armature selected enter 'Pose' mode and select the bone shown here.



Press F7 and from the menu choose 'IKSolver'.



In the 'Target' field type 'b.r.cr_l'.

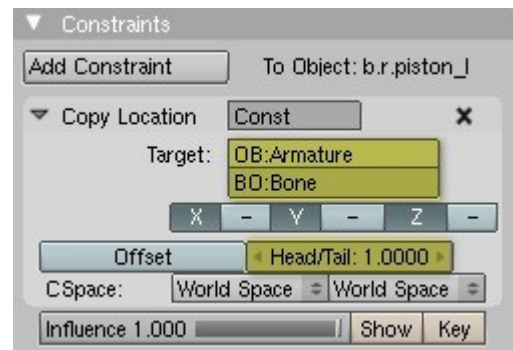


This will lock the bone to the Lattice. If we now link the Lattice 'b.r.cr_l' to 'b.r.w2' we can see the bones in action.

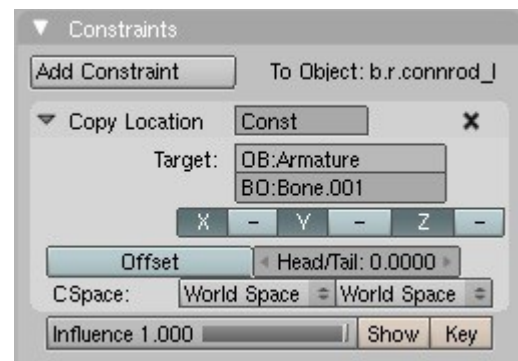
In order to constrain the Lattices 'b.r.piston_l' and 'b.r.connrod_l' to the Armature we have to add a constraint(!) to them. **Remember that the Lattices cannot be linked to the Armature as the exporter can't handle them.**

Select 'b.r.piston_l' and press F7. In the 'Constraints' panel add a 'Copy Location' constraint. In 'Target:OB' enter 'Armature', and in 'BO:' enter 'Bone'. Change the 'Head/Tail:' value to 1 (default is 0).

This will copy the location information from the tail end of the first bone to the Lattice 'b.r.piston_l' which will mean the Lattice will now follow the bone although it is not parented to it.

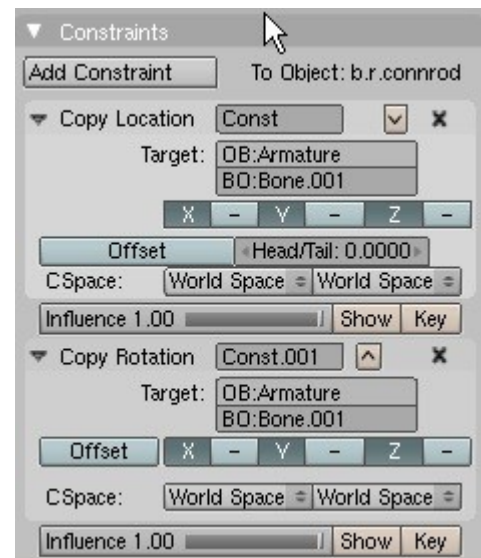


Repeat for 'b.r.connrod_l', except that the 'BO:' field should have 'Bone.001' in it, and the 'Head/Tail:' field 0 (as we want the movement from the head end of the bone in this case).



Note that this will only copy the position of the bone, not the rotation, so we need to add an extra 'Copy Rotation' constraint to the 'b.r.connrod' Lattice as shown here.

If the cylinders are inclined it's necessary to select the 'Offset' button in the 'Copy Rotation' constraint. If this is not selected the Lattice rotates to match the rotation of the bone at frame 1 when it should start out horizontal.



The Lattices 'b.r.connrod_l' and 'b.r.piston_l' must be linked to 'b.r.main' for exporting, and the objects 'piston_l' and 'connrod_l' must be linked to 'b.r.piston_l' and 'b.r.connrod_l' respectively.

Animating the Right Hand Connecting Rod and Piston

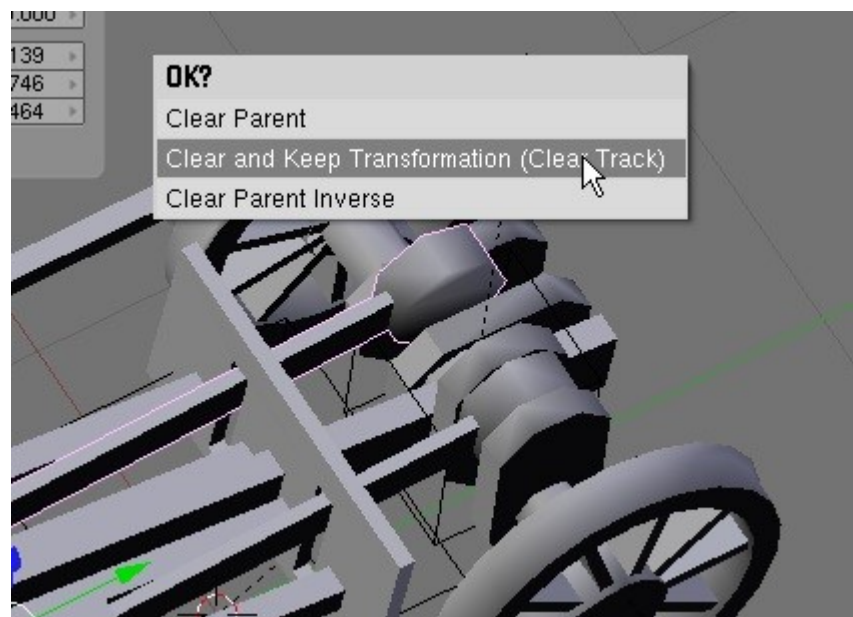
We can make animating the right hand side a bit easier for ourselves by moving the components for the left hand side into the correct position and then mirroring them over for the right side. This is especially useful for complex outside valve gear.

At frame 1 the right hand crank of the crank axle is vertical. Unfortunately as we have animated over 30 frames we can't get the left crank exactly in this position. The easiest thing to do is to simply delete the keyframe at frame 30 of 'b.r.w2', change the animation to run over 32 frames and create a new keyframe at frame 32. The left crank will be vertical at frame 24 now (**don't forget to change the lpo Curve type to 'Linear' though...**), so move to this frame and mirror the objects 'connrod_l' and 'piston_l' over to the right side.

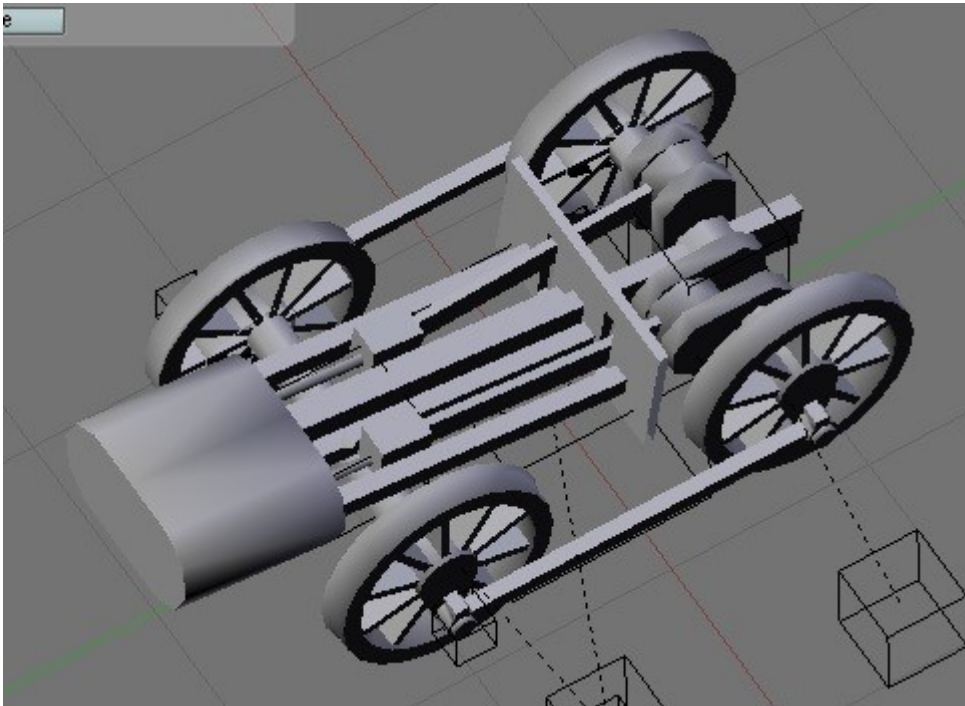
Apply the mirror modifiers to the connecting rod and piston objects and separate the meshes for the right hand side, naming them 'connrod_r' and 'piston_r'.

Now select the two new objects, press ALT-P and choose 'Clear and Keep Transformation (Clear Track)' from the menu.

This will unlink the objects from the left side Lattices but leave the objects in the correct position for the right side.



Move the animation back to frame 1 and reanimate over 30 frames again, not forgetting to reset the Ipo Curve type back to 'Linear'.



All we have to do now is to animate the right hand side in a similar manner to the left side. The completely animated model can be downloaded [here](#).

And Finally

If anybody has any suggestions for improved techniques or problems following this tutorial let me know by mailing me at paul.hobbs@web.de .