BeamNG Modding Tutorial

By BLIJo

In this tutorial I will show you how to make an addon for a vehicle in BeamNG.
I will start with setting up the folder structure, getting meshes and other files from the game and then I will make a simple addon, texture and Jbeam it and pack it so its suitable for the repo.

Step one: What do we need?

To make a mod for beam, you need the following things:

- BeamNG
- Photoshop/Gimp/Paint.net
- Notepad++
- Blender

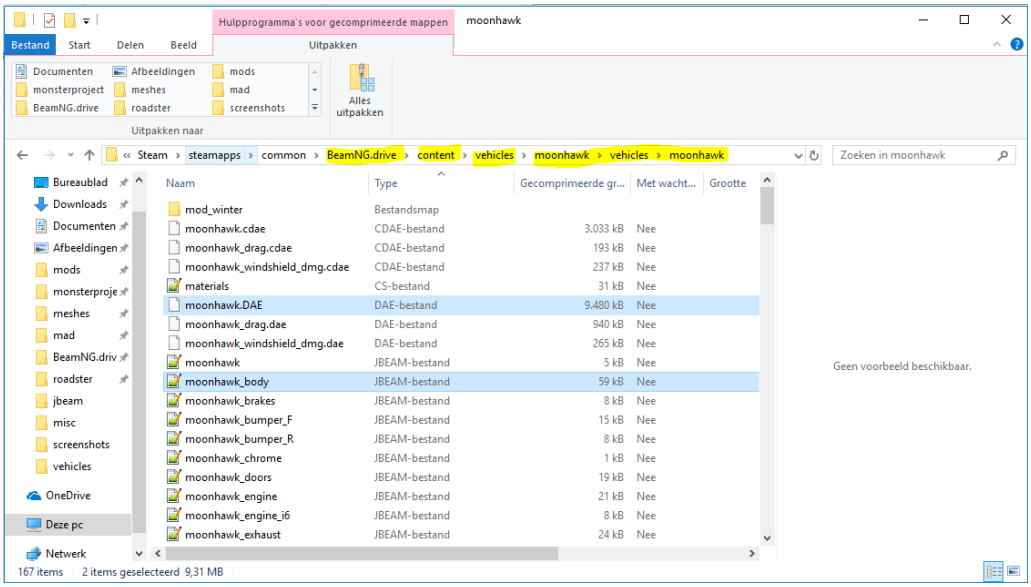
Step two: Getting the vanilla files and making a folder structure.

I want to make a simple addon for the Moonhawk, it will be an ugly box on top of the roof. This makes no sense so that's why 😊.

What do we need for this?

- The mesh of the Moonhawk
- Jbeam file of the body since it will be connected to it.

To get the files, browse to your game directories, open the Moonhawk folder and select these two:

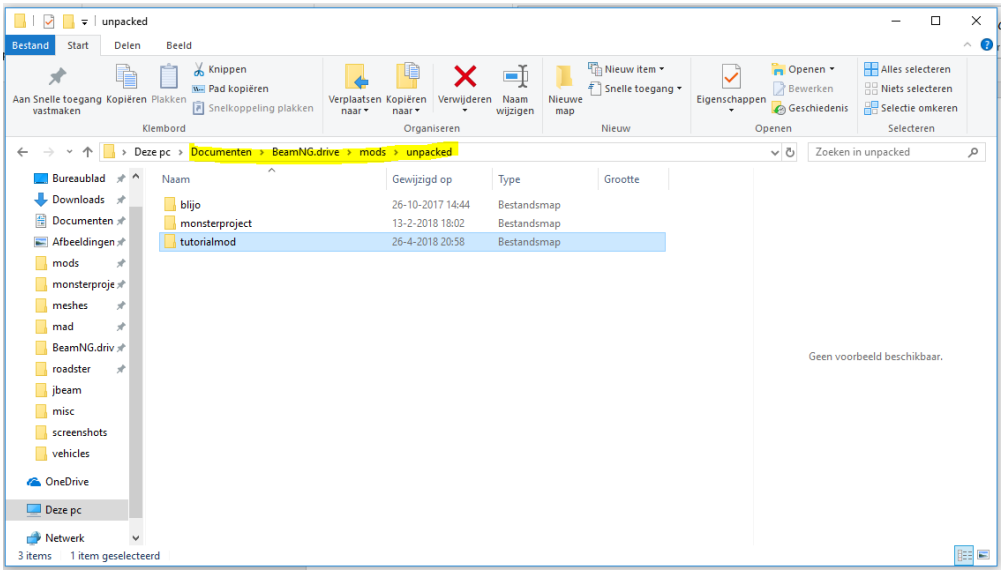


Copy these and paste them somewhere where you can find them, we won't edit them, they are just a reference.

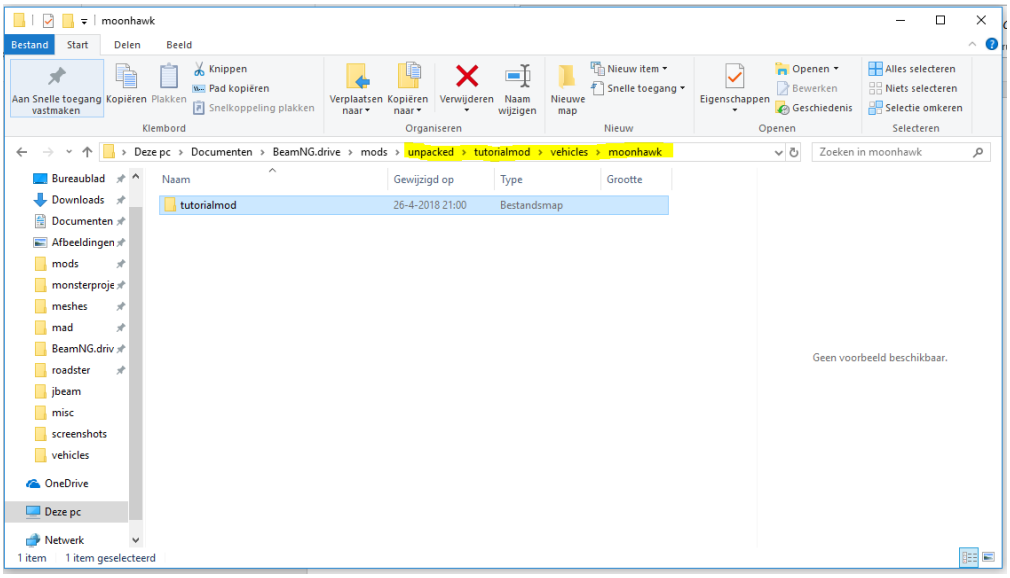
Now let's make the folder structure!

Go to your documents/beamng/mods

When you are in this folder, create a new folder called “unpacked”, in this folder, create a new folder with another name, I will call it “tutorialmod”.



In this folder, you need to make the same subfolders as the game so you will have to add a “vehicles” subfolder in it and a “moonhawk” folder in your vehicles folder. In your moonhawk folder, you should create yet another folder(many folders :D) for this specific mod. I will call it “tutorialmod” again.



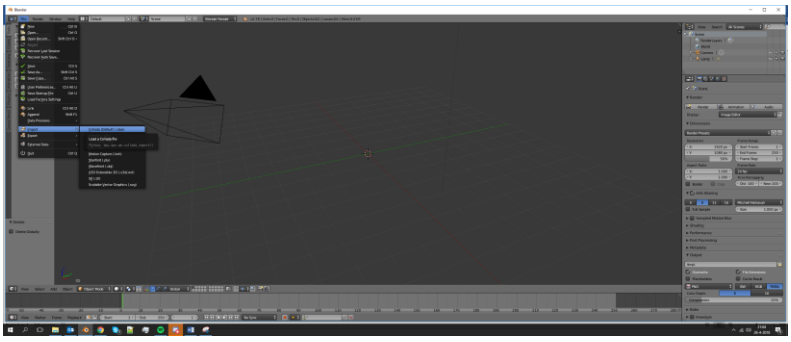
We will work in this folder, all our new files will go in it.

Since I mod BeamNG a lot, I have a special folder for the meshes and Jbeams that I use during the development of new ones.

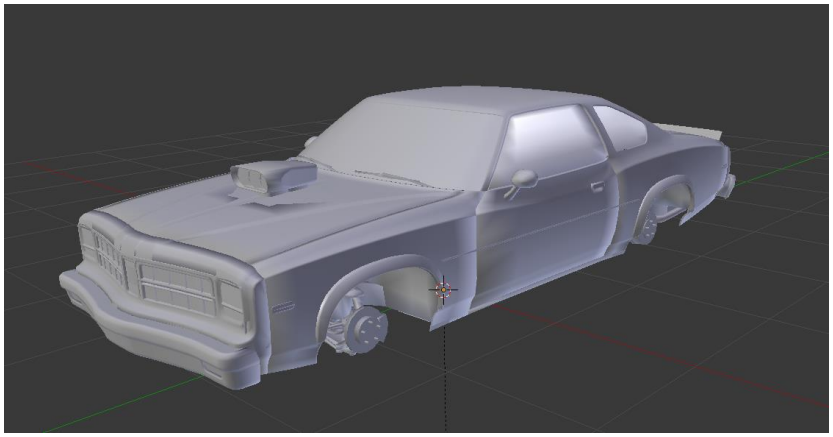
Step three: Making the 3D model!

Now let’s get started!

First, we open Blender. Then you import the moonhawk, you can do that by importing a .dae in the menu on the top left.



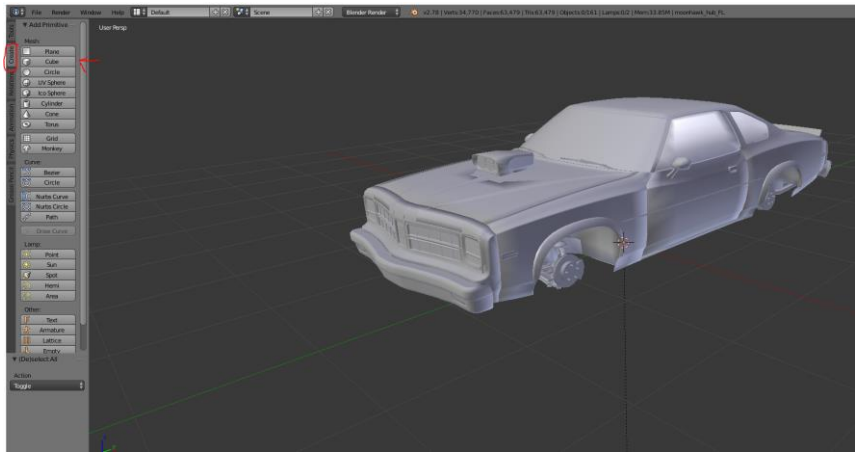
So now I selected the moonhawk and have it in blender.



It looks so ugly since edgesplit isn't enabled, I won't enable it since I don't need the Moonhawk anyways.

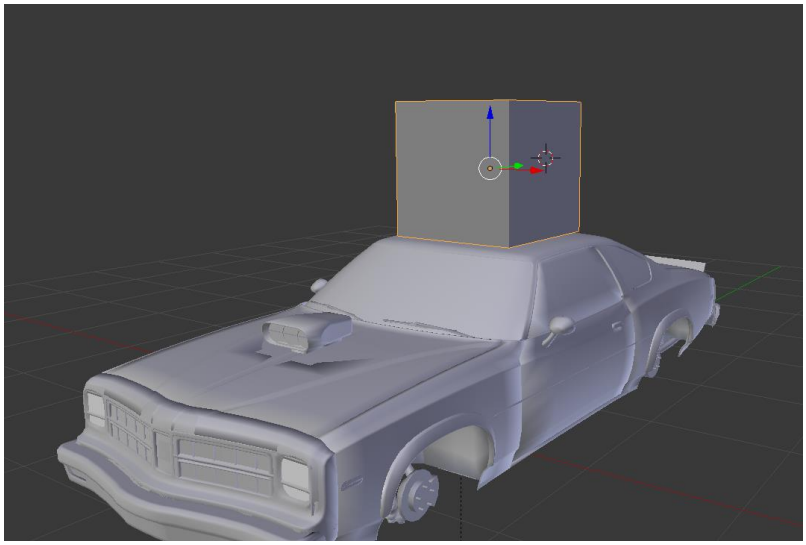
Now let's make our box!

This is easy, you just click create box!



You can select “generate UV’s” when you click on the cube, this will hopefully help us a bit ☺

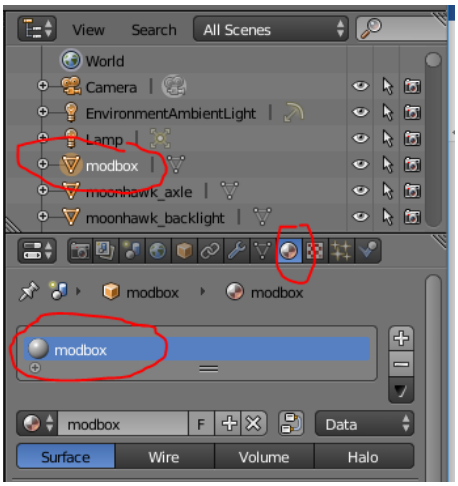
Now we have a box and we will move it to the right location and scale it by pressing “S” and modify it to our likes.



Isn’t it ugly?

Now we press CTRL+A and check all the boxes to apply rotation, scale and location.
You will see that the centerpoint will move to the middle of our working area.

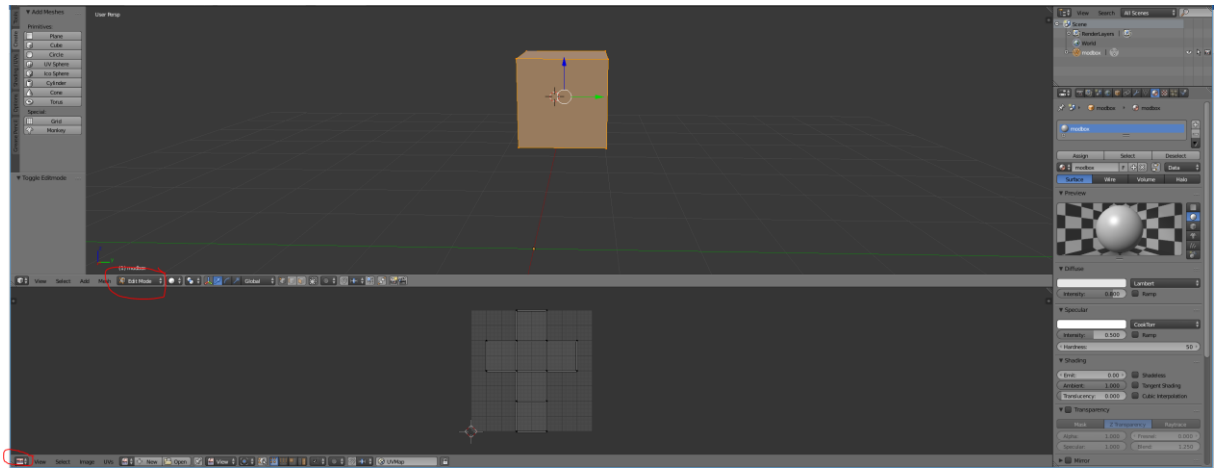
After this, we will rename it and give it a name for the texture:
For the material, press “new” and give it a name, I will call it modbox:



Now remove all moonhawk stuff from your blender file and save it(If you haven’t saved it already)

Step four: UV-unmapping

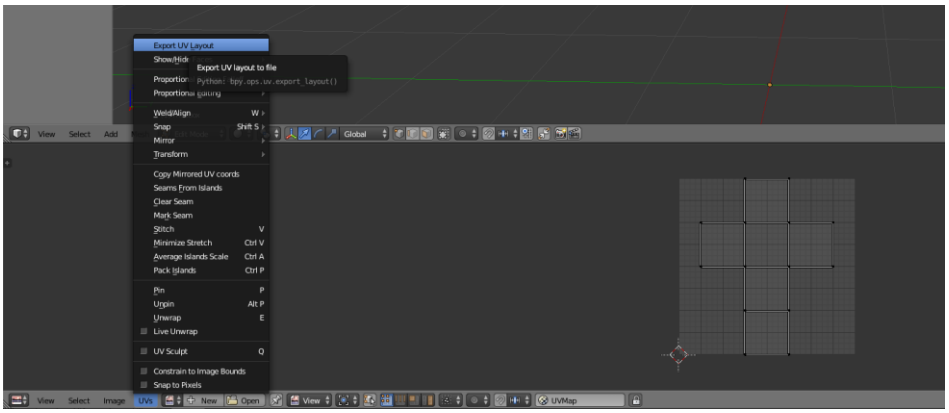
I really hate to do this but in order to get a nice mod, you need to do this.
We select the box and go to edit mode.
We also open a new window/part in blender and open the UV/image editor:



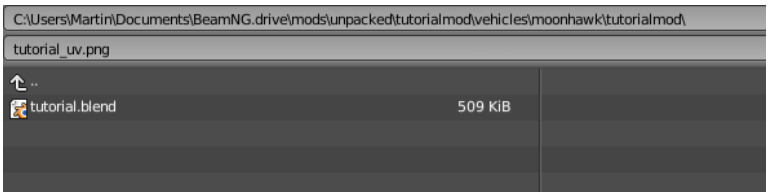
As you can see, selecting “generate UV’s” generated some nice UV’s, these are very useful but we need to know which side is which, this doesn’t really matter, we will find out when putting it ingame.

What we do now is export the UV-layout and export the mesh:

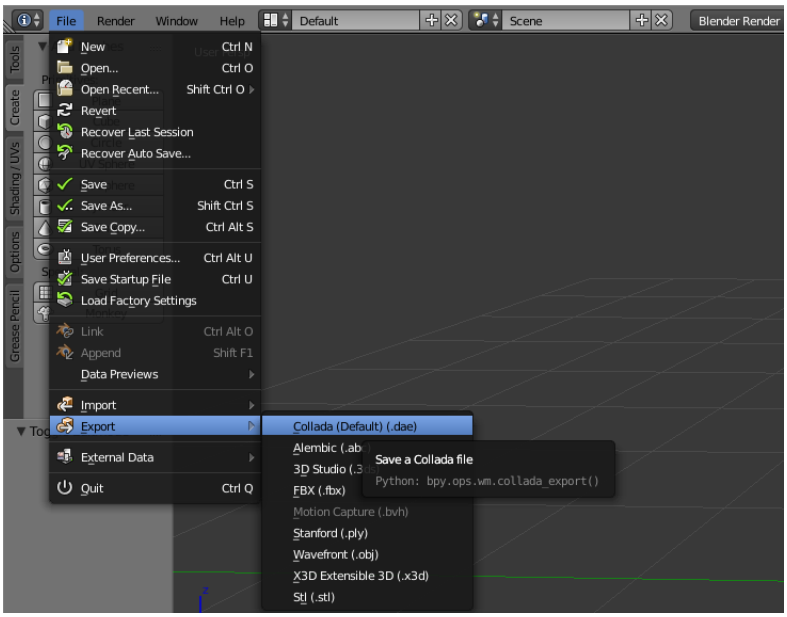
The UV-layout:



We will save the file in our modding folder.



Then we do the same with the mesh, go to the menu and select export Collada(DAE):



We save it to the same folder again.

Then your model is ready!

Step Five: Jbeam

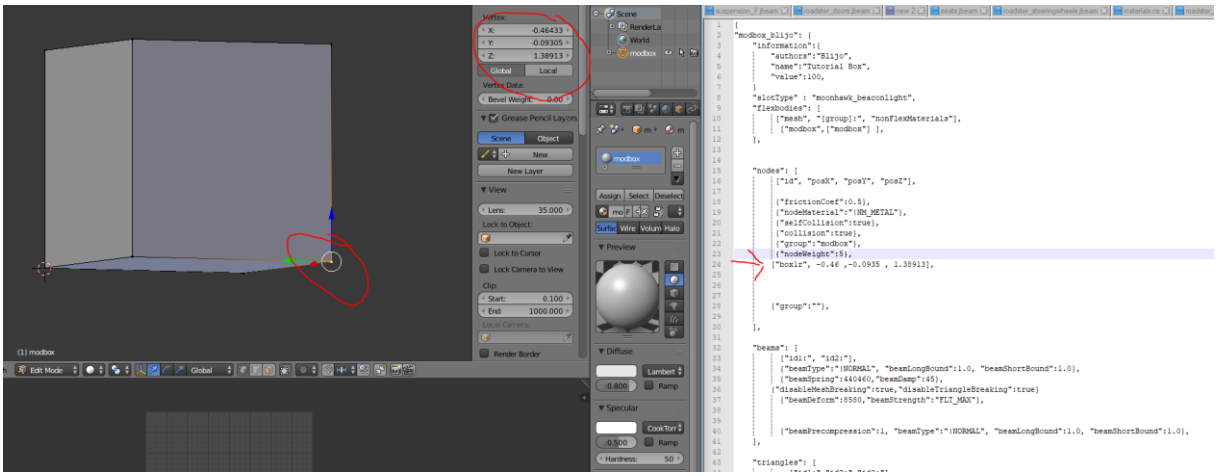
In order to get ANYTHING ingame, you need to make a jbeam, since you are probably new to Jbeam, I will start with the basics:

Open a new file in notepad++ and make it look like this:

```
{                   This is the opening bracket of the jbeam
"": {               Between these two ", you define the partname of the jbeam.
    "information":{
        "authors":"Blijo",
        "name":"",  between these two, you define the screenname of the part(what you see in the selector)
        "value":100,
    }
    "slotType" : "", The slottype is the slot of the jbeam, every part needs a slot, not all parts have them
    "flexbodies": [
        ["mesh", "[group]:", "nonFlexMaterials"],
                    In the flexbody section, you link the jbeam to the mesh: ["meshname",[ "nodegroup"]]
    ],


    "nodes": [              This is the node section, nodes are the weights of the part.
        ["id", "posX", "posY", "posZ"],

        {"frictionCoef":0.5},
        {"nodeMaterial":"|NM_METAL"},
        {"selfCollision":true},
        {"collision":true},
        {"group":""},       The nodegroup links to the nodegroup in the mesh, because of this, not all meshes
        {"nodeWeight":1},   react the same in a collision, your rear bumper isnt linked to the wheels :D
                            The nodeweight is the weight of the node. This is always a bit of a guess if you
                            don't know how much something weighs. I will use 5 KG now.


       {"group":""},

    ],

    "beams": [
        ["id1:", "id2:"],
        {"beamType":"|NORMAL", "beamLongBound":1.0, "beamShortBound":1.0},
        {"beamSpring":440460,"beamDamp":45},
       {"disableMeshBreaking":true,"disableTriangleBreaking":true}
        {"beamDeform":8580,"beamStrength":"FLT_MAX"},
        The beams define the behaviour of the nodes, for the in depth explanation of the properties,
        check the wiki.
        {"beamPrecompression":1, "beamType":"|NORMAL", "beamLongBound":1.0, "beamShortBound":1.0},
    ],

    "triangles": [
          ["id1:","id2:","id3:"],
       {"dragCoef":5}

        Triangles are the aero and the collision technique in beamng


          {"triangleType":"NORMALTYPE"},
    ],
}
}
```

Now let’s make some nodes.

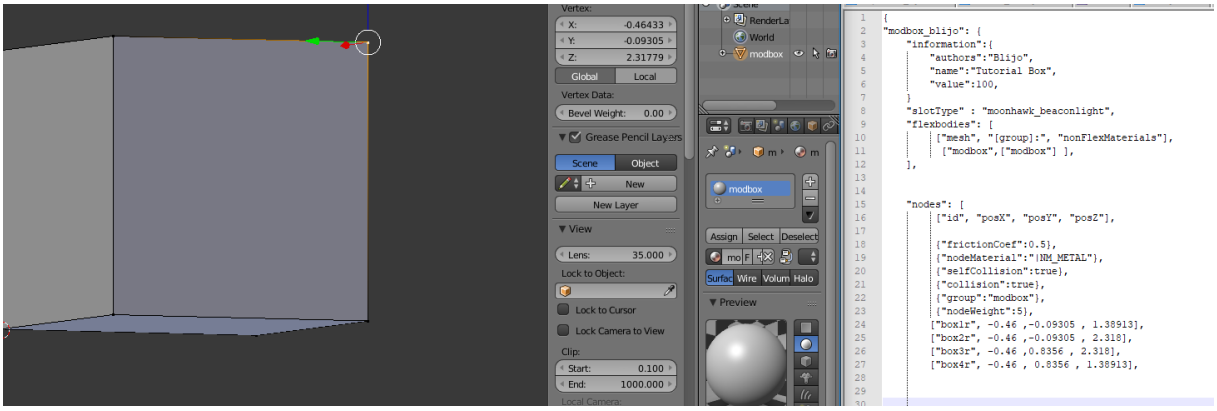
The box has 8 corners and I can use the coordinates from blender:
Each corner will be a node.



The left nodes will be the same as the right nodes but with a positive x-coordinate so we will do these later.

As you can see in the screen, I also added the right slot types, the mesh and the nodegroups. The slottype is from the moonhawk's beacon.

Now, I added all four right nodes.

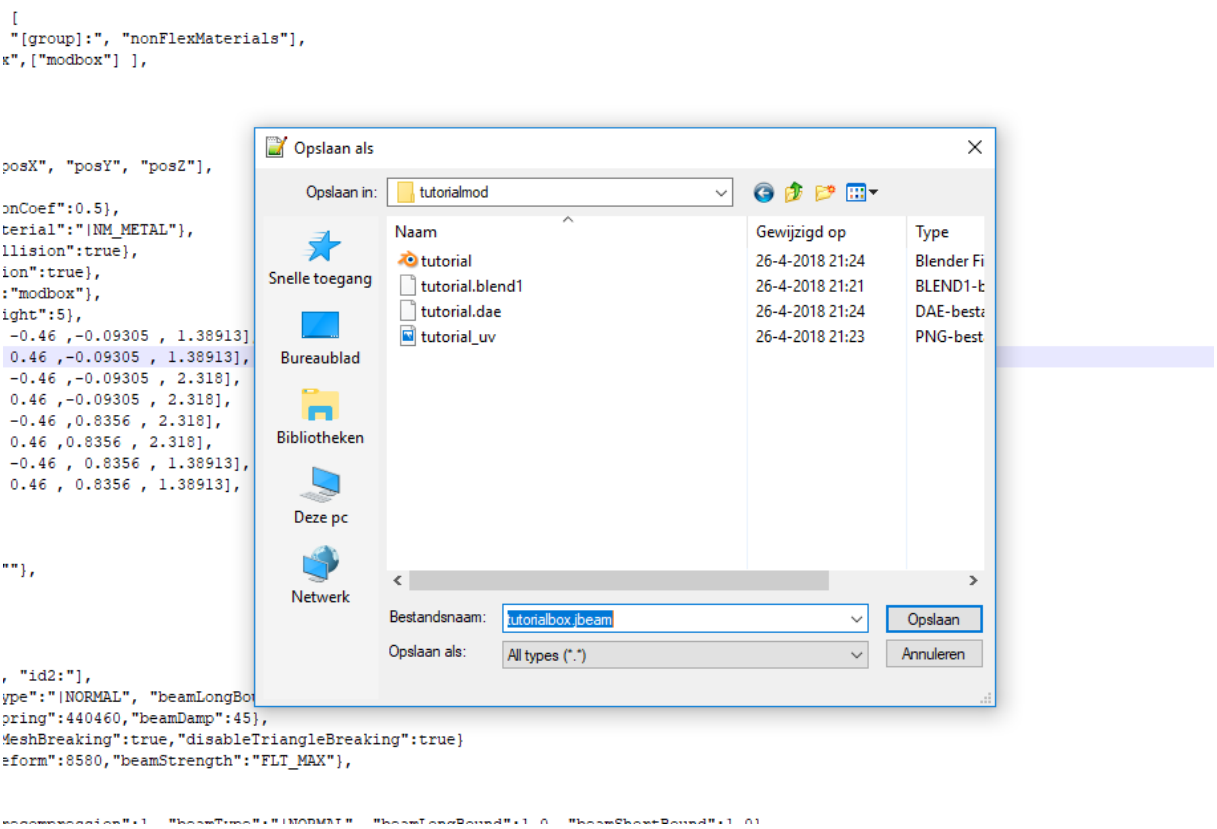


Now I copy the nodes and make left nodes:

```
"nodes": [
    ["id", "posX", "posY", "posZ"],

    {"frictionCoef":0.5},
    {"nodeMaterial":"|NM_METAL"},
    {"selfCollision":true},
    {"collision":true},
    {"group":"modbox"},
    {"nodeWeight":5},
   ["box1r", -0.46 ,-0.09305 , 1.38913],
   ["box1l", 0.46 ,-0.09305 , 1.38913],
   ["box2r", -0.46 ,-0.09305 , 2.318],
   ["box2l", 0.46 ,-0.09305 , 2.318],
   ["box3r", -0.46 ,0.8356 , 2.318],
   ["box3l", 0.46 ,0.8356 , 2.318],
   ["box4r", -0.46 , 0.8356 , 1.38913],
   ["box4l", 0.46 , 0.8356 , 1.38913],
```

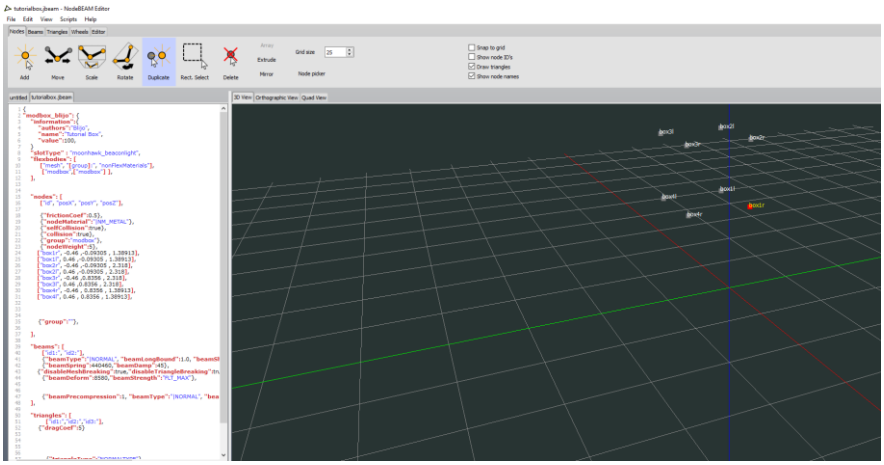
Easy right?

Now save the file as a .jbeam



For the beams and triangles, I will use BNE, you can find it under “utils” in your game files

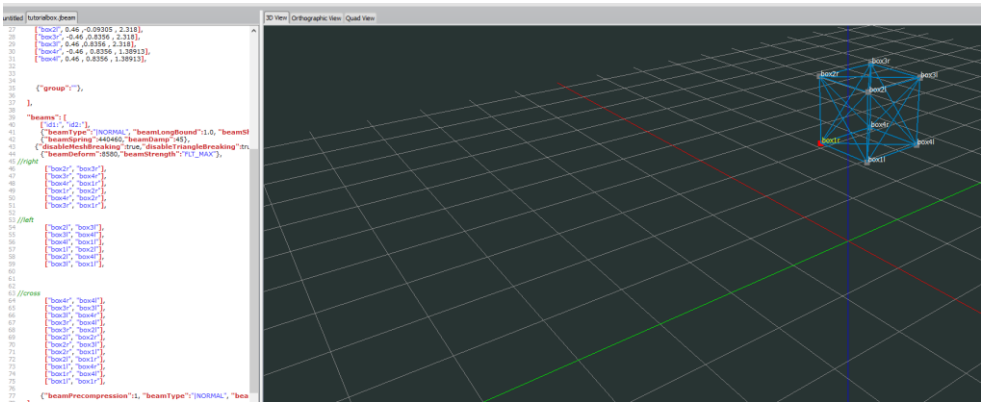
Now, start the program and import the jbeam.



Now we can add beams, go and click on the line below
“{"beamDeform":8580,"beamStrength":"FLT_MAX"},”

And go to the beams section on the top menu and click add single.

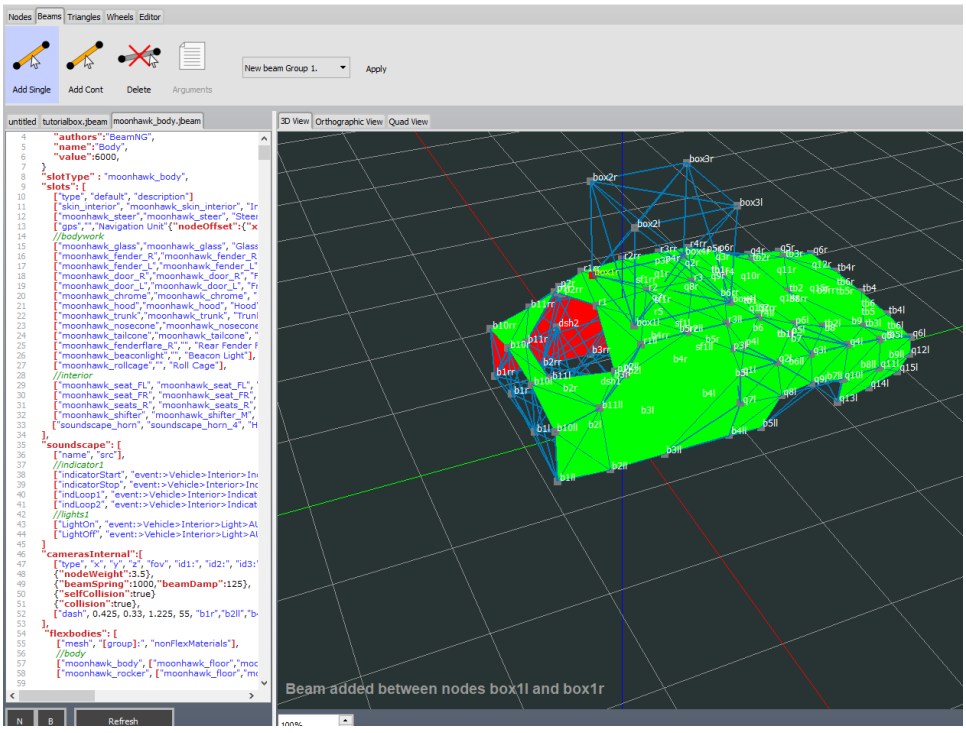
Then make triangles between the nodes with the beams. Then it will look like this, please add comments:



Don’t forget to save!

Now let’s connect it to the body of the moonhawk, I will do this with two breakgroups, more info about them is on the wiki.

For this we open the moonhawk jbeam in BNE.



The code for a breakgroup is:

```
{"breakGroup":"boxmount_R"},
{"breakGroupType":0},

{"breakGroup":""},

{"breakGroup":"boxmount_L"},
{"breakGroupType":0},
```

Breakgrouptype:0 means that when one beam in the breakgroup collapses, the other ones break as well, this is useful for hinges and mounts.

Now save it in notepad and reload the jbeam in BNE.

Now we will add some beams between the roof of the moonhawk and the box, I suggest 3-4 beams per lower node of the box+one per higher node.

For node 1r:

```
{"breakGroupType":0},

   ["box1r", "r1rr"],
   ["box1r", "r2rr"],
   ["box1r", "r1"],
   ["box1r", "r2"],
```

Now do the rest:

```
{"beamDeform":1500, "beams
{"breakGroup":"boxmount_R"}

{"breakGroupType":0},

   ["box1r", "r1rr"],
   ["box1r", "r2rr"],
   ["box1r", "r1"],
   ["box1r", "r2"],

   ["r4rr", "box4r"],
   ["r3rr", "box4r"],
   ["r4", "box4r"],
   ["r3", "box4r"],

//higher nodes
   ["box3r", "r4rr"],
   ["box2r", "r1rr"],
```

And the left one is the same but with l and ll instead of r and rr… Also make sure that the left beams are under boxmount_L!

```
{"beamDeform":1500,"beamStrength":200
{"breakGroup":"boxmount_R"},

{"breakGroupType":0},

   ["box1r", "r1rr"],
   ["box1r", "r2rr"],
   ["box1r", "r1"],
   ["box1r", "r2"],

   ["r4rr", "box4r"],
   ["r3rr", "box4r"],
   ["r4", "box4r"],
   ["r3", "box4r"],

//higher nodes
   ["box3r", "r4rr"],
   ["box2r", "r1rr"],


{"breakGroup":""},


{"breakGroup":"boxmount_L"},
   ["box1l", "r1ll"],
   ["box1l", "r2ll"],
   ["box1l", "r1"],
   ["box1l", "r2"],

   ["r4ll", "box4l"],
   ["r3ll", "box4l"],
   ["r4", "box4l"],
   ["r3", "box4l"],

//higher nodes
   ["box3ll", "r4ll"],
   ["box2l", "r1ll"],
{"breakGroupType":0},
```

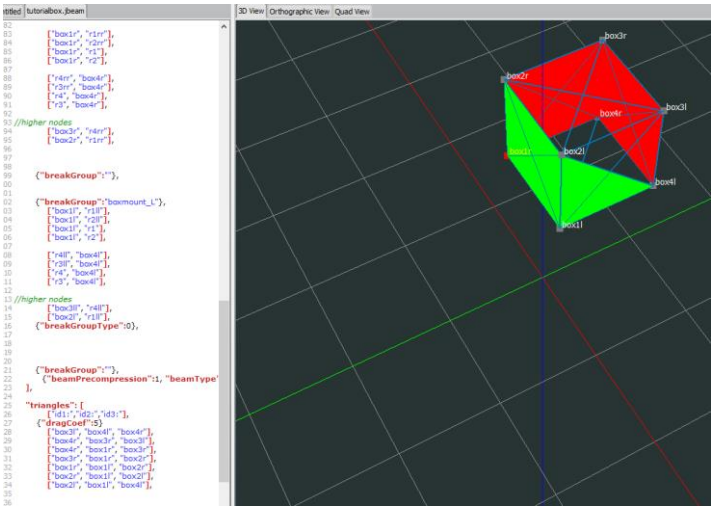
If you did that, save again 😊

Then its time for triangles.

You can close the moonhawk Jbeam.

Then you to the triangles section of the jbeam and click on the line below dragcoef. Then go to triangles and click add triangles, hope you saved since BNE likes to crash at this point.

Now add triangles between the nodes while selecting the nodes counter clockwise.



Do this for the whole box. And save and exit BNE since we are done with the Jbeam!

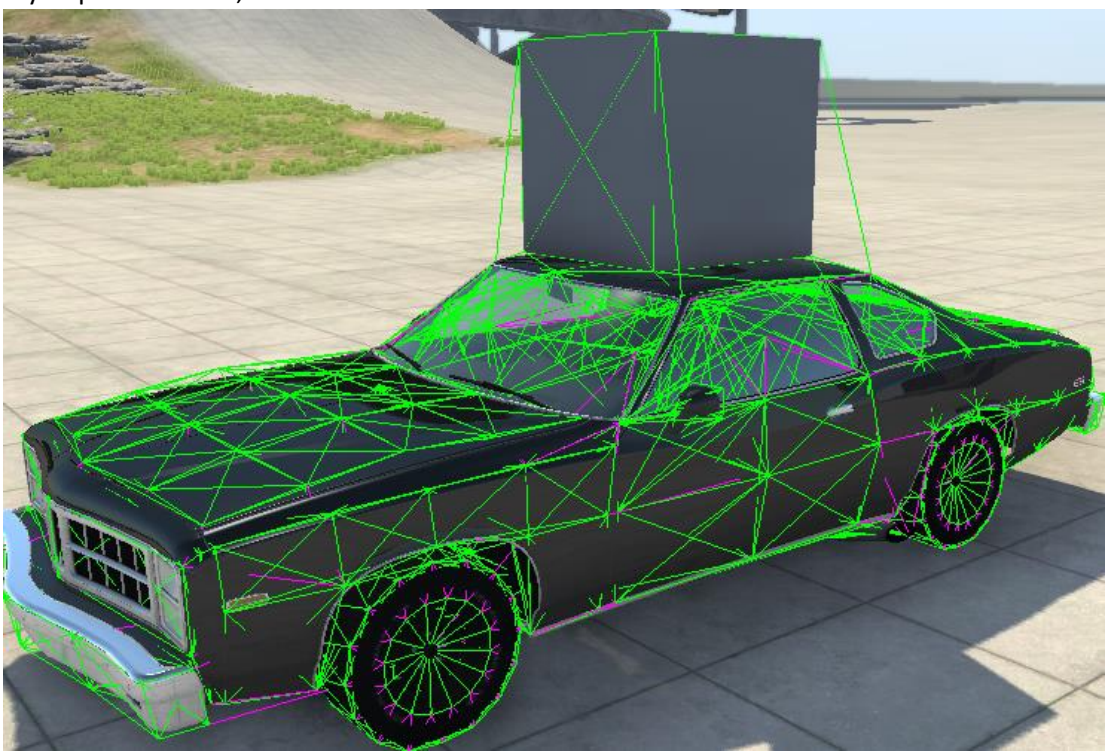
Step Six: first ingame test

Now load beam, select the moonhawk and look for the part under the beacon slot!

Splendid!

Now, if it doesn't behave as you want it to, edit the beam values. For me, the left beams broke so I made them less stiff and stronger by decreasing the beamspring and increasing the beamstrenght and beamdeform.

If you press CTRL+B, it should look like this:



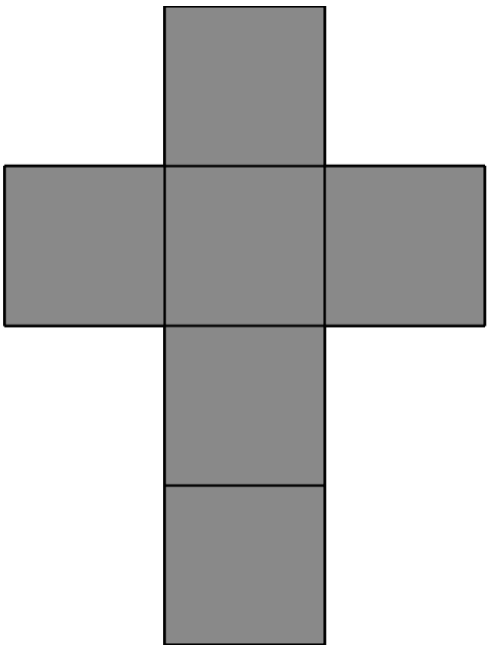
Now its time for the next step, textures.

Step Seven: Textures and last check.

This is easy, you draw some shit on the UV's and call it a day right?

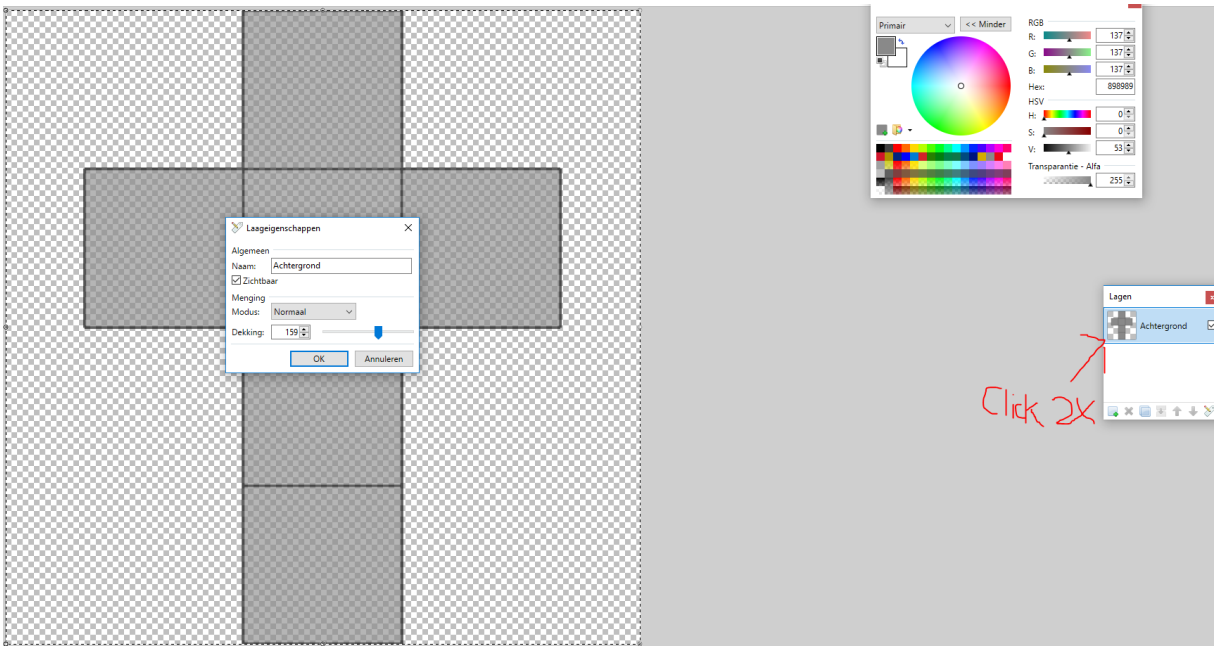
You can do that but you can also make a diffuse, specular and a normal map to make it look better.

Let's open the UV template in paint.net and draw some shit :D



This will be my specular map, tutorialbox_s.dds, the diffuse map will be a more transculent version of this and the normal map will be made with an online tool since Im lazy.

For the diffusemap:



And save it as tutorialbox_d.dds

For the normalmap, save the specular as tutorialbox_n.png and go to http://www.smart-page.net/smartnormal/
and load and save the image.

Now we have our textures but not ingame, how do we do this?

I always copy it from the official vehicles. We make a file called materials.cs in notepad.
In that file you need the following lines:

```
singleton Material(modbox) Name of material in blender
{
    mapTo = "modbox";

   normalMap[0] = "vehicles/moonhawk/tutorialmod/tutorialbox_n.png";
    diffuseMap[0] = "vehicles/moonhawk/tutorialmod/tutorialbox_d.dds";
    specularMap[0] = "vehicles/moonhawk/tutorialmod/tutorialbox_s.dds";
    diffuseColor[0] = "1 1 1 1";
        specularColor[0] = "1 1 1 2";
    specularPower[0] = "128";        Location and filename
    useAnisotropic[0] = "1";
    castShadows = "1";
    translucent = "0";
    translucentBlendOp = "None";
    alphaTest = "0";
    alphaRef = "0";

    dynamicCubemap = true;
    instanceDiffuse[2] = true;
    materialTag0 = "beamng"; materialTag1 = "vehicle";
};
```

Now relaunch beam:

Perfect. Don’t forget to check the console (~key)

If you did everything right, you should not have errors regarding this mod.

Now its time to pack it and share it!

Step Eight: Packing it.

This is pretty easy, go to your work folder and delete the UV and the .blend files. Then go back to unpacked/tutorialmod and select the vehicles folder. Right click on it and zip it. Then rename it to something unique like tutorialbox_blijo.zip.

Now you are ready to share it!