

# The Snowman of Doom

A Beginner's Guide to 3D Modeling

for Neverwinter Nights

by TheBarbarian

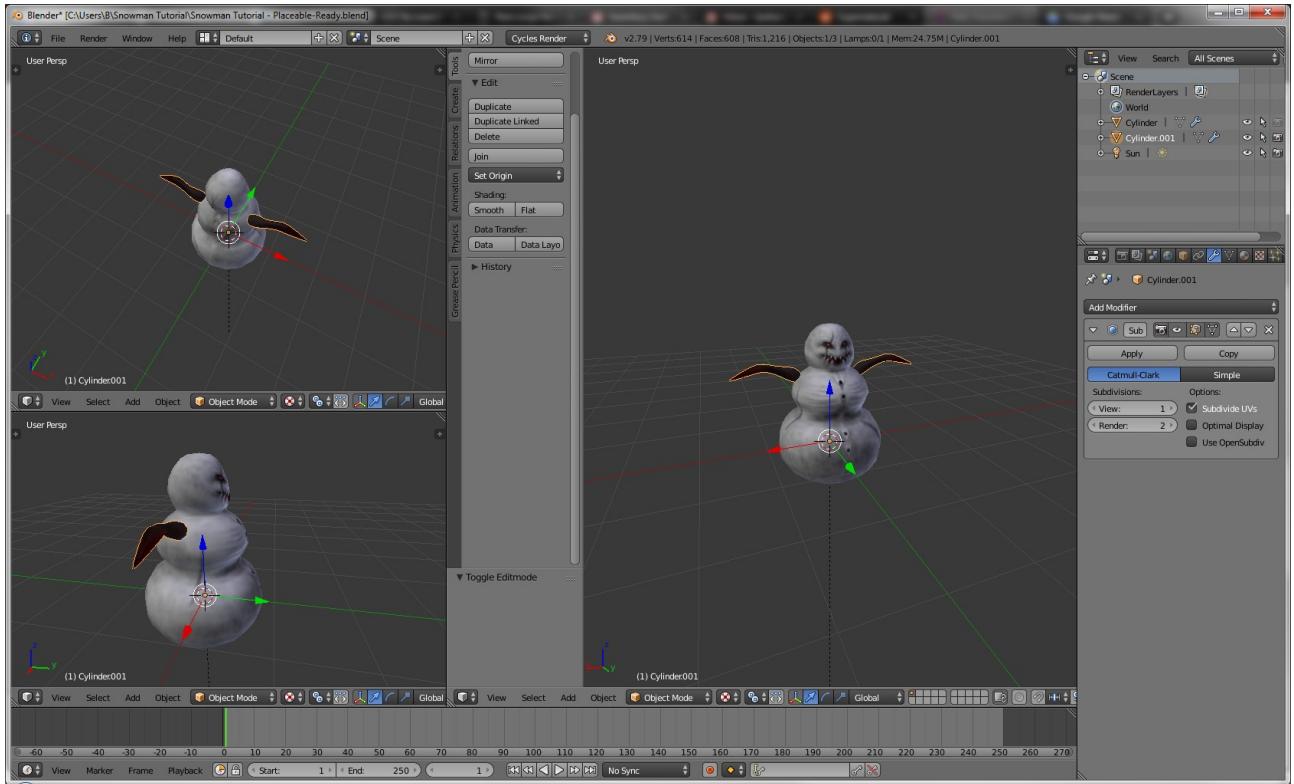
## Part 1.3 - Exporting as a Placeable

### Table of Contents

- Intro Page / Resuming.....	2
- Creating the AuroraBase.....	3
- Parenting to the AuroraBase.....	4
- Adding the Nodes (Walkmesh & Dummy Helper).....	6
- Shaping the Walkmesh.....	8
- Positioning the Use / Impact Nodes.....	13
- Setting up the Material.....	14
- Check the Image Formats.....	18
- Check the Individual Object Settings.....	19
- Check the Positioning and Size of the Mesh.....	21
- Apply Transforms.....	25
- Export!.....	27

## - Intro Page / Resuming

We resume the quest for custom asset creation. Hooray! Starting point:



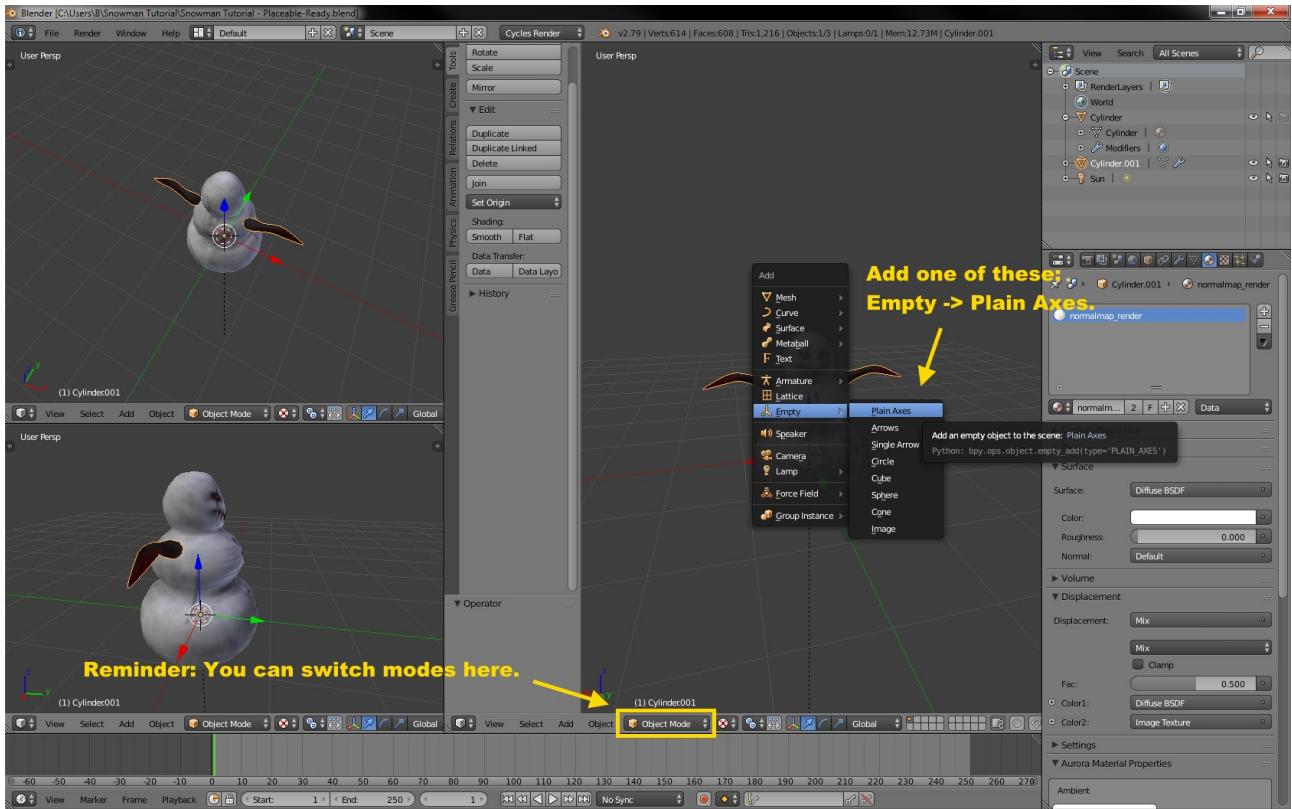
This part of the tutorial is reusable, and doesn't strictly require a hand-crafted model (as we've been covering in Part 1.1 and Part 1.2). You can also import models from free asset sites, in this way. To follow along with the tutorial at this point, you need to have:

- An UV-mapped mesh
- A diffuse texture for that mesh
- A normalmap for that mesh
- A specular map for that mesh

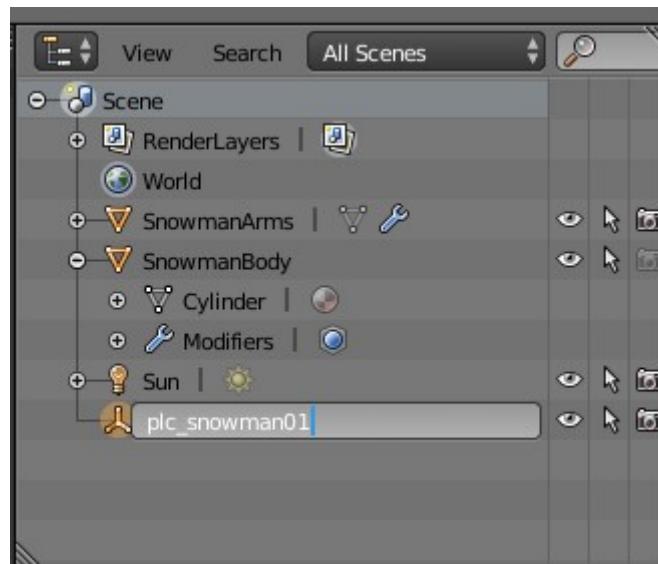
Information on the side: Unlike the Enhanced Edition version, 1.69 Neverwinter Nights and earlier do not support normal- and specular maps. To set up a 1.69-compatible placeable, you only need an UV-mapped mesh and a diffuse texture.

## - Creating the AuroraBase

First things first, we need our AuroraBase - the root of our to-be-exported model. Let's add an empty Plain Axes object.



Switch to Object Mode, then hit SHIFT+A to open the Add menu.

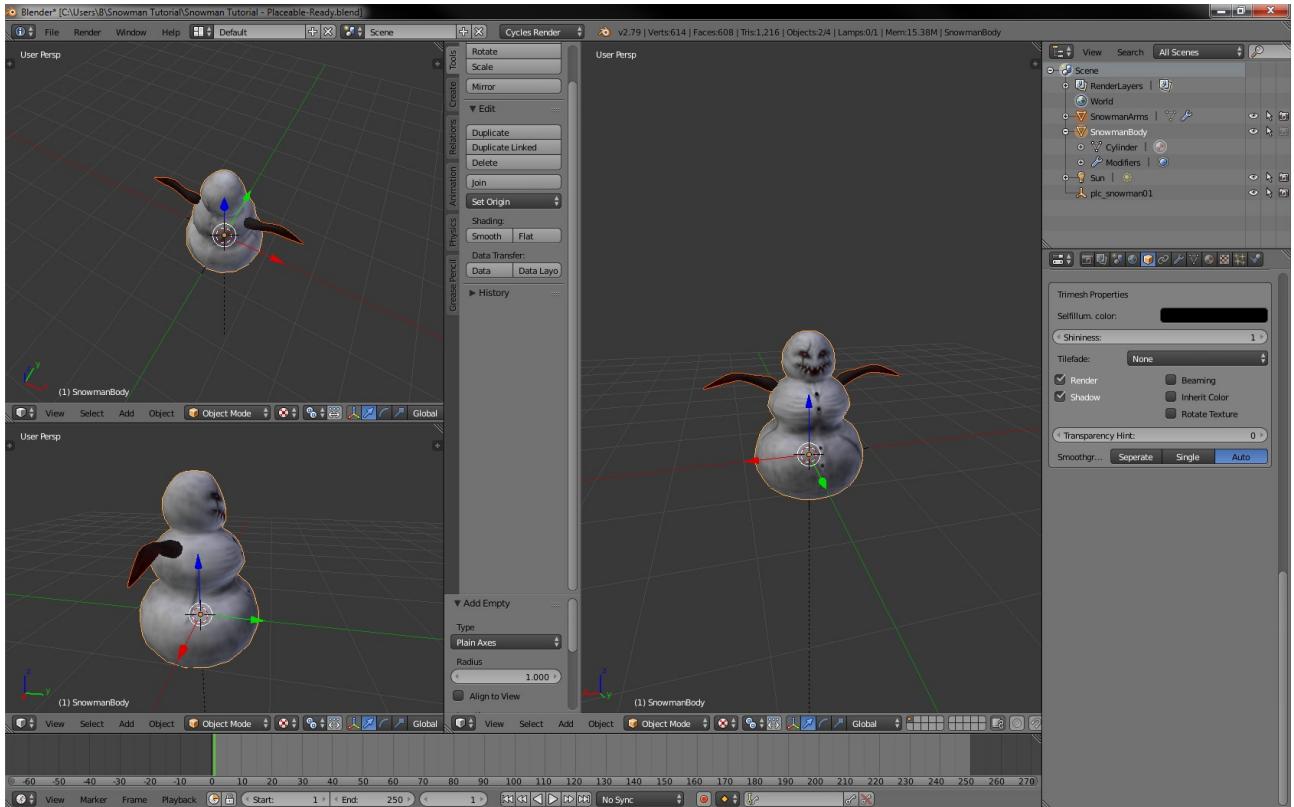


Now, rename the plain empty. The name of the AuroraBase (our plain axes empty) will also be the name of the exported model.

## - Parenting to the AuroraBase

Next up, we need to link everything that is supposed to get exported along with the AuroraBase, as part of this model, to the AuroraBase.

Select the objects that are meant to be part of the final model ingame. In my case, as seen in the picture below, these are SnowmanArms and SnowmanBody, the two objects that contain the mesh parts for the body and the arms.

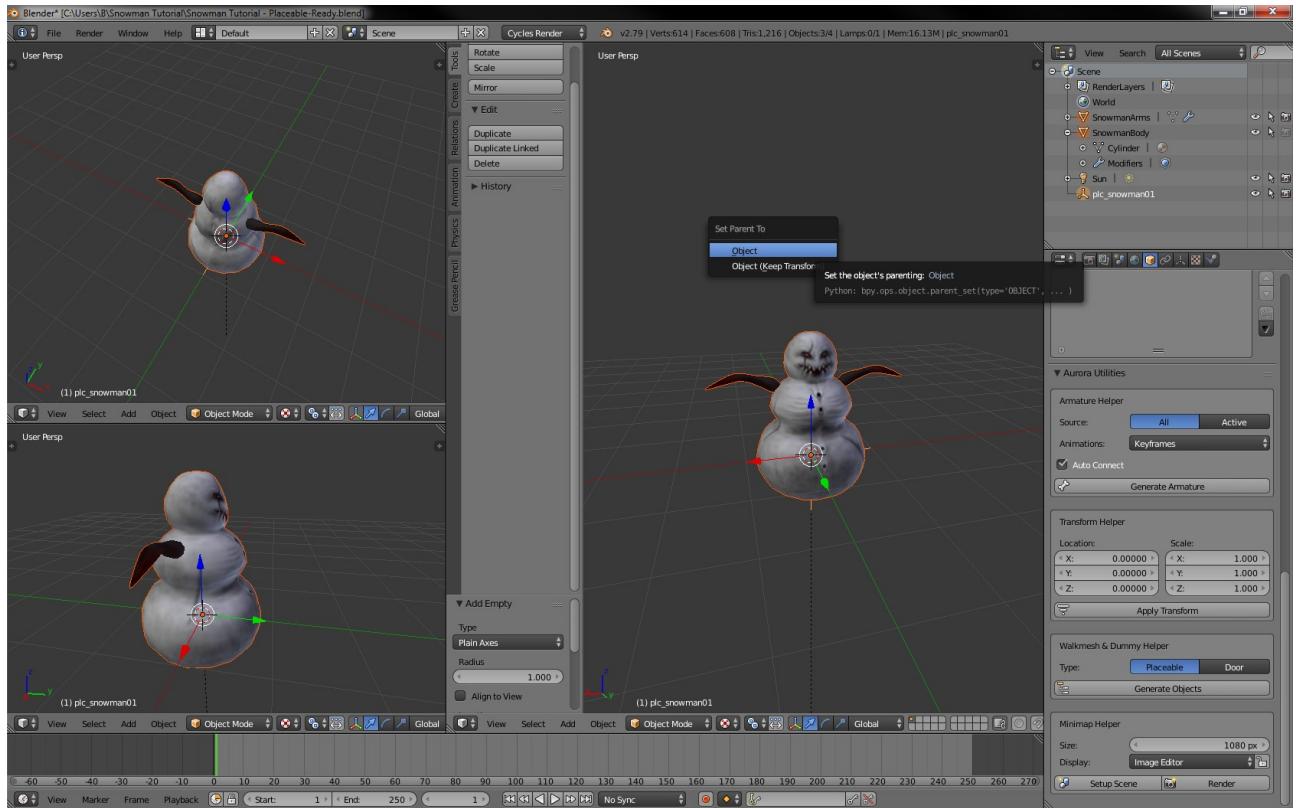


First **click** one of them, then **hold SHIFT** and **click** on the other to add the next object to the selection.

Last of all, select the AuroraBase (in this case, plc\_snowman01).

(and then, turn to the next page, because the images for the next step didn't fit onto this one anymore so I put them on the next page and now you have to turn it at a somewhat awkward location and that's just something we're all going to have to get through. together. as a family!)

... and now, with the objects selected and the AuroraBase being the last object to be selected, hit **CTRL+P**.



Either option should work in this case, just click one of 'em.

So, we just set the AuroraBase to be the *parent* of the other selected objects (SnowmanArms and SnowmanBody). They're linked to it, now, and will be exported along with the AuroraBase.

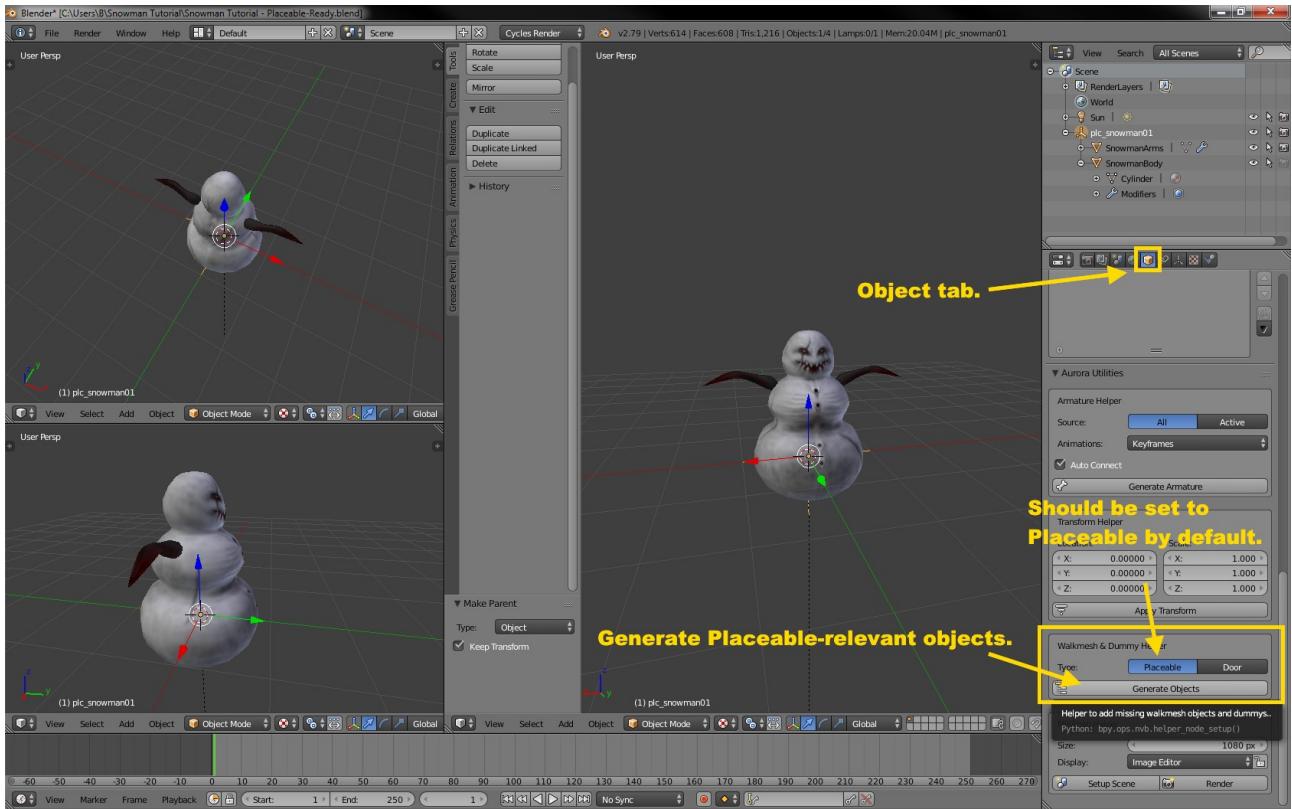


Watch out for this; if an object in the scene is not a child of the AuroraBase (or a child of a child of a child...), then it won't be exported into the MDL file, and therefore won't show up in the game.

Sidenote: You can also drag and drop objects onto one another in the scene list to do this.

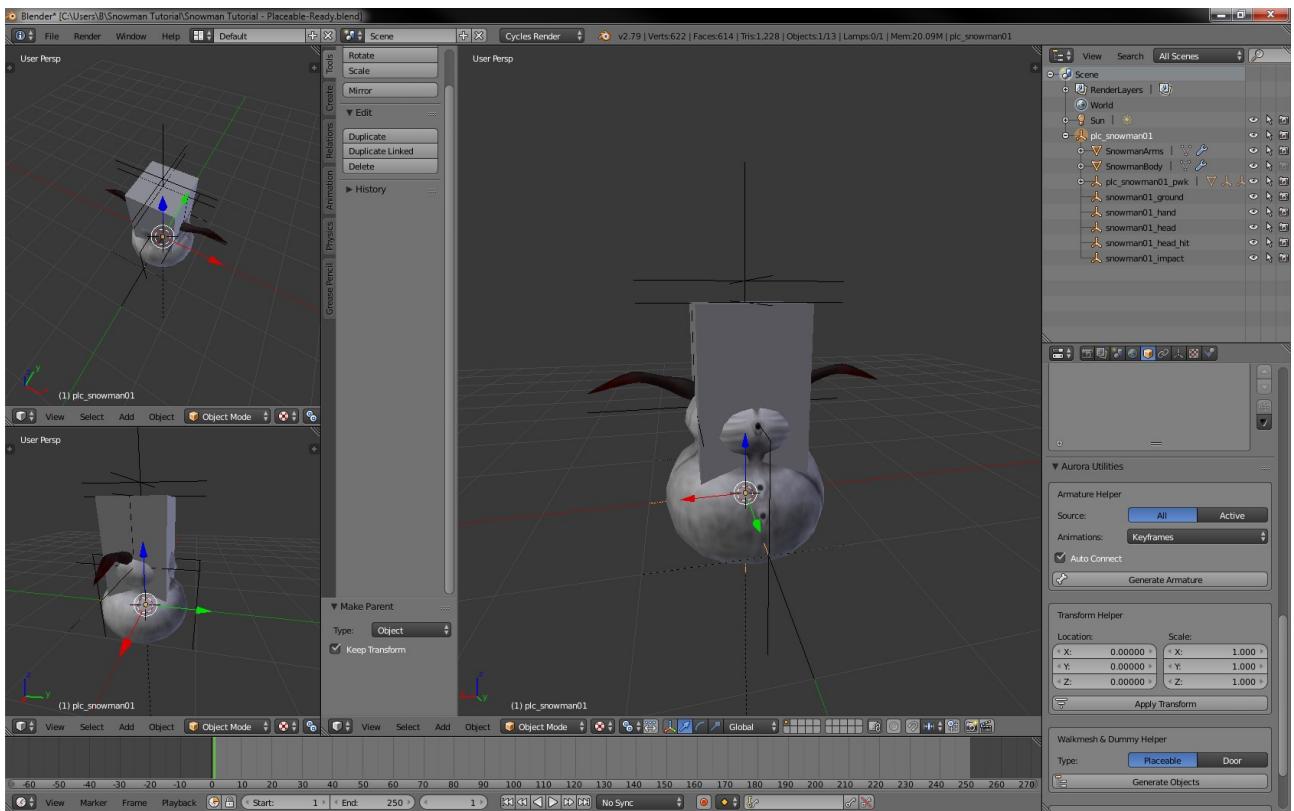
## - Adding the Nodes (Walkmesh & Dummy Helper)

OK, now, Symmetric set up a lovely helper utility in Neverblender that we'll be making use of now. Switch to the Object tab, and scroll down until you find the Walkmesh & Dummy Helper:



Click the button to generate the helper objects for a placeable.

Also, don't forget to save every once in a while. **CTRL+S / CTRL+SHIFT+S / CTRL+ALT+S**.



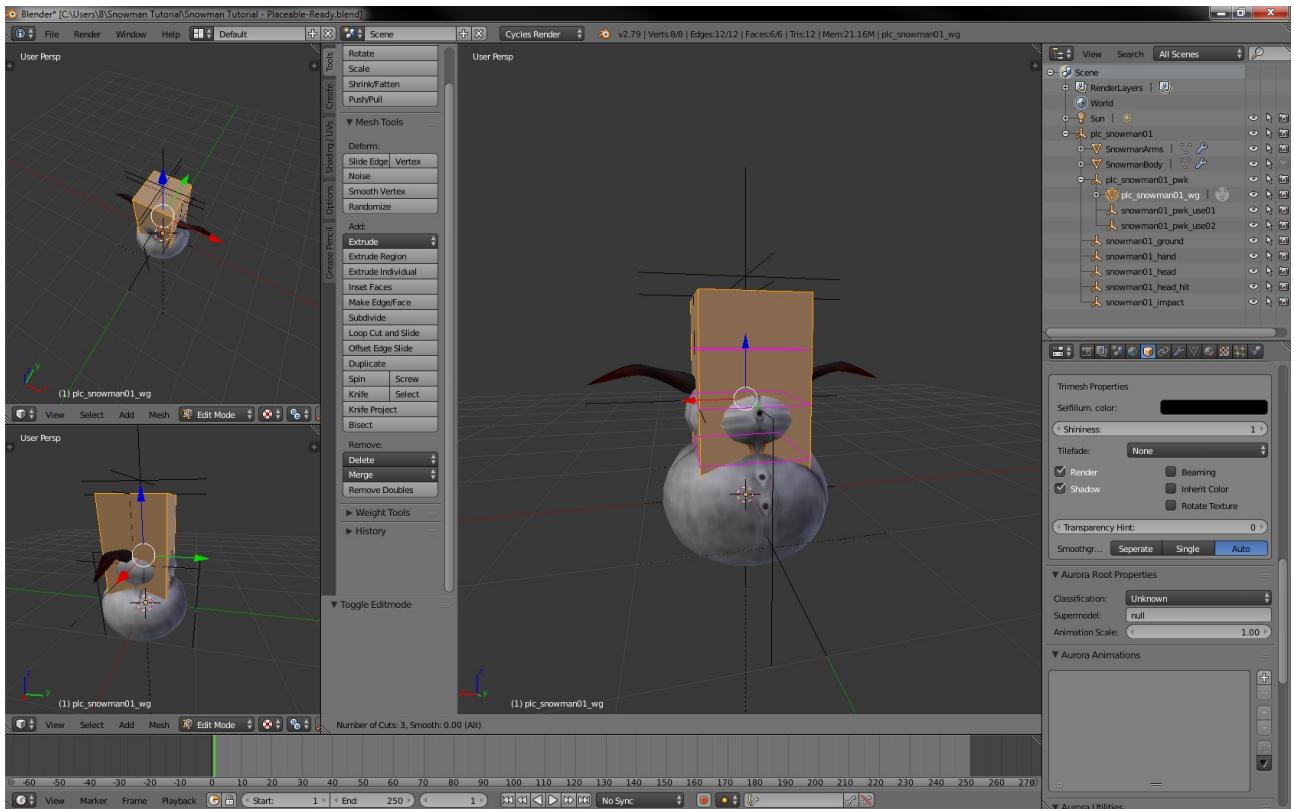
So, whoompfh! What do we have here? It's a whole bunch of new objects. Who are they? What do they do? Why is there a great big blocky box in the middle of the snowman now? We have questions, and we have answers.

The empties are, by and large, use/impact nodes. They designate the root locations at which particular types of VFX will be placed when applied to the placeable, or where players will attempt to move in order to use the placeable.

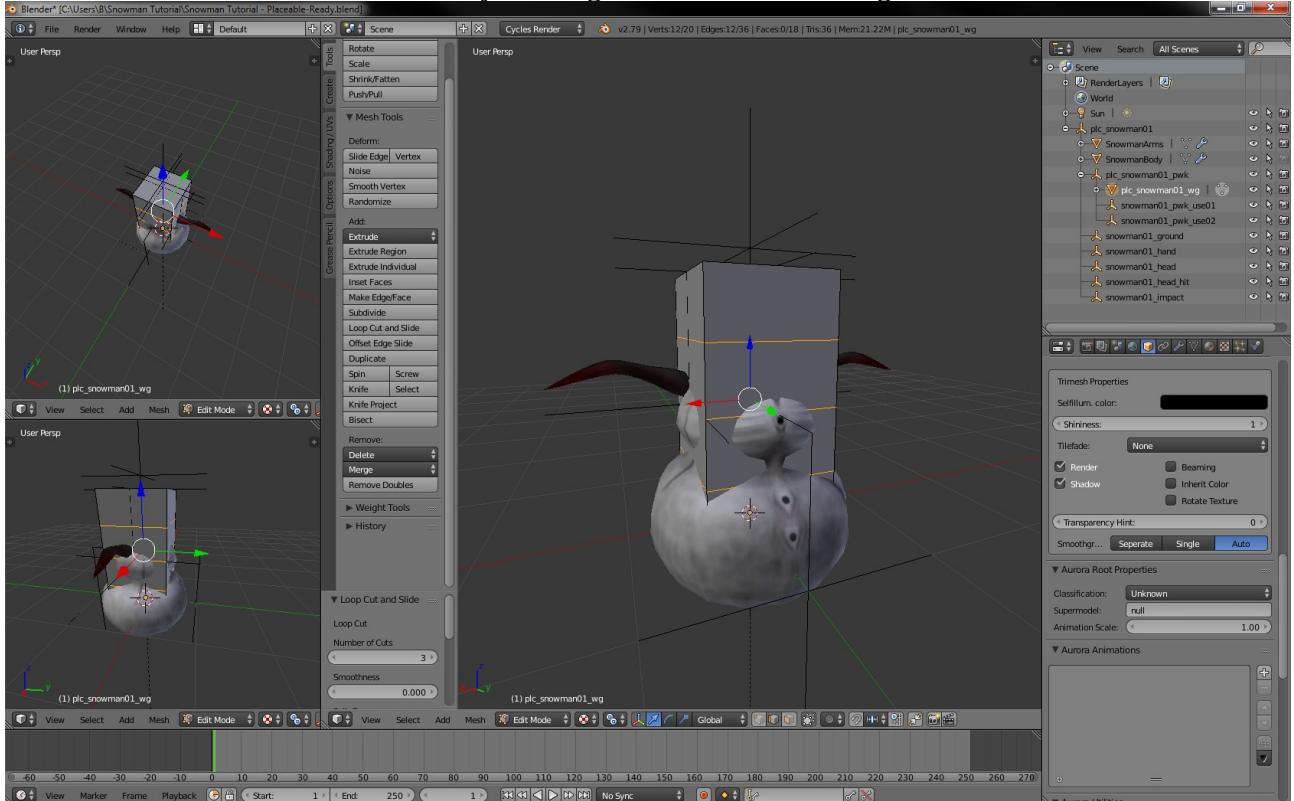
The great big blocky box is for the walkmesh. Think that this box will be invisible ingame, but the area within will be marked non-walkable, so players can't walk through the placeable.

## - Shaping the Walkmesh

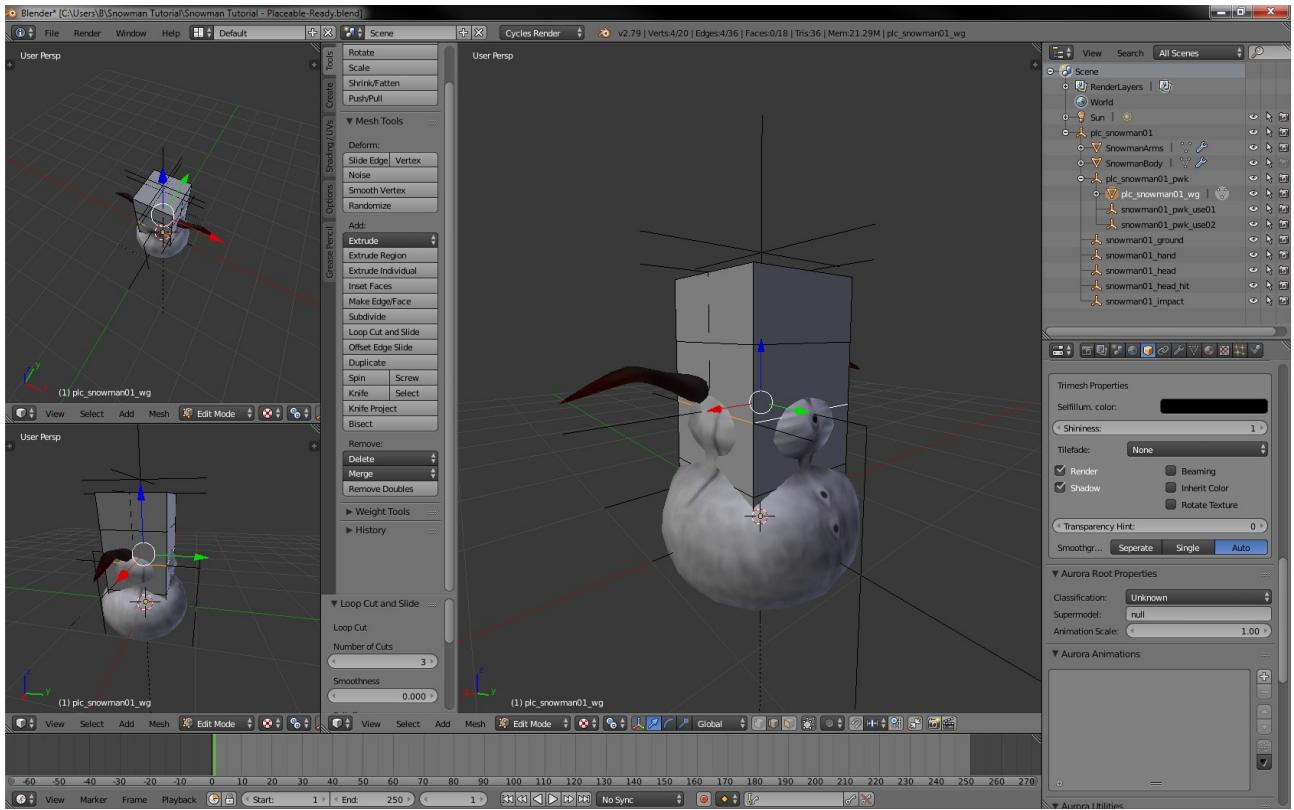
Let's shape that walkmesh so that it fits the snowman. Select the walkmesh box, and switch to Edit Mode:



You can hold **CTRL+R** and point, then click, at parts of the mesh to place Loop Cuts. Roll your mouse wheel to increase or decrease the amount of cuts, and click again to confirm. Then, you can either move the location of the cuts by moving the mouse, or click again to confirm.

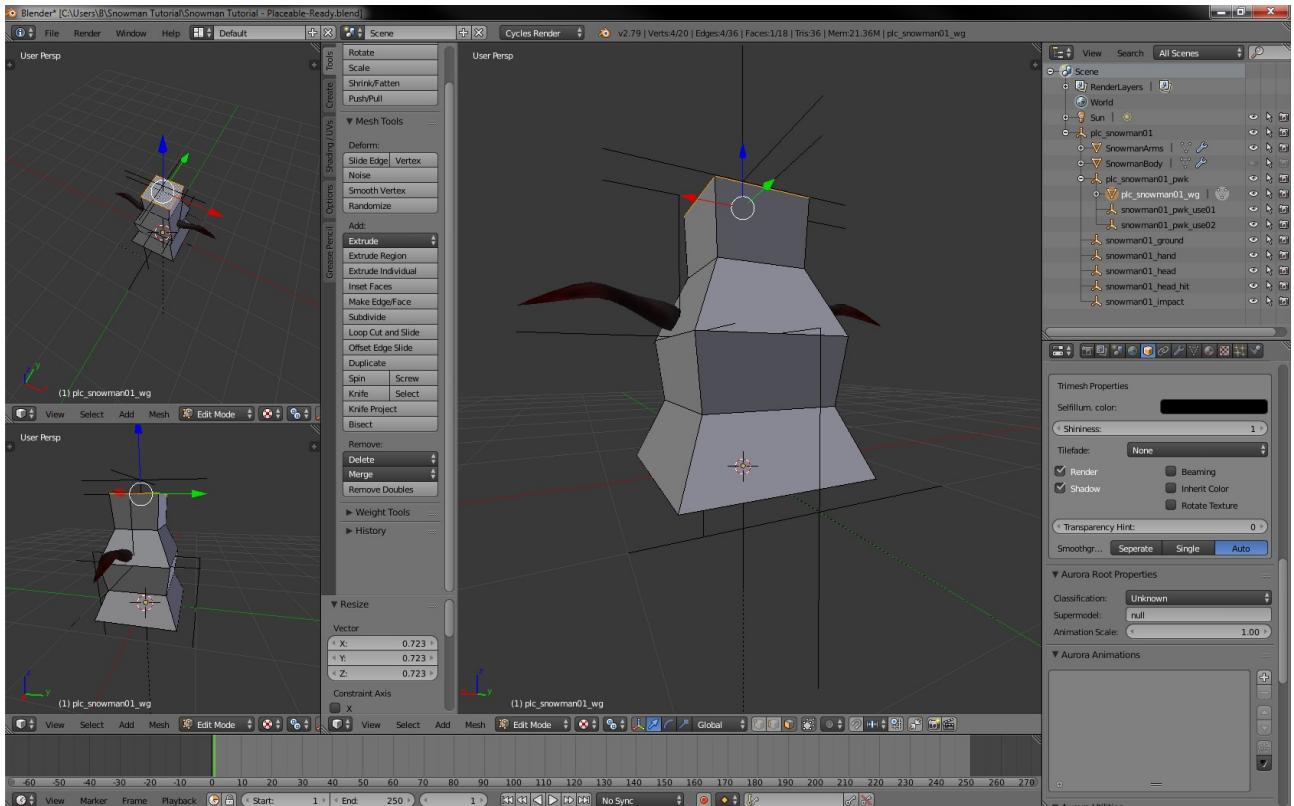


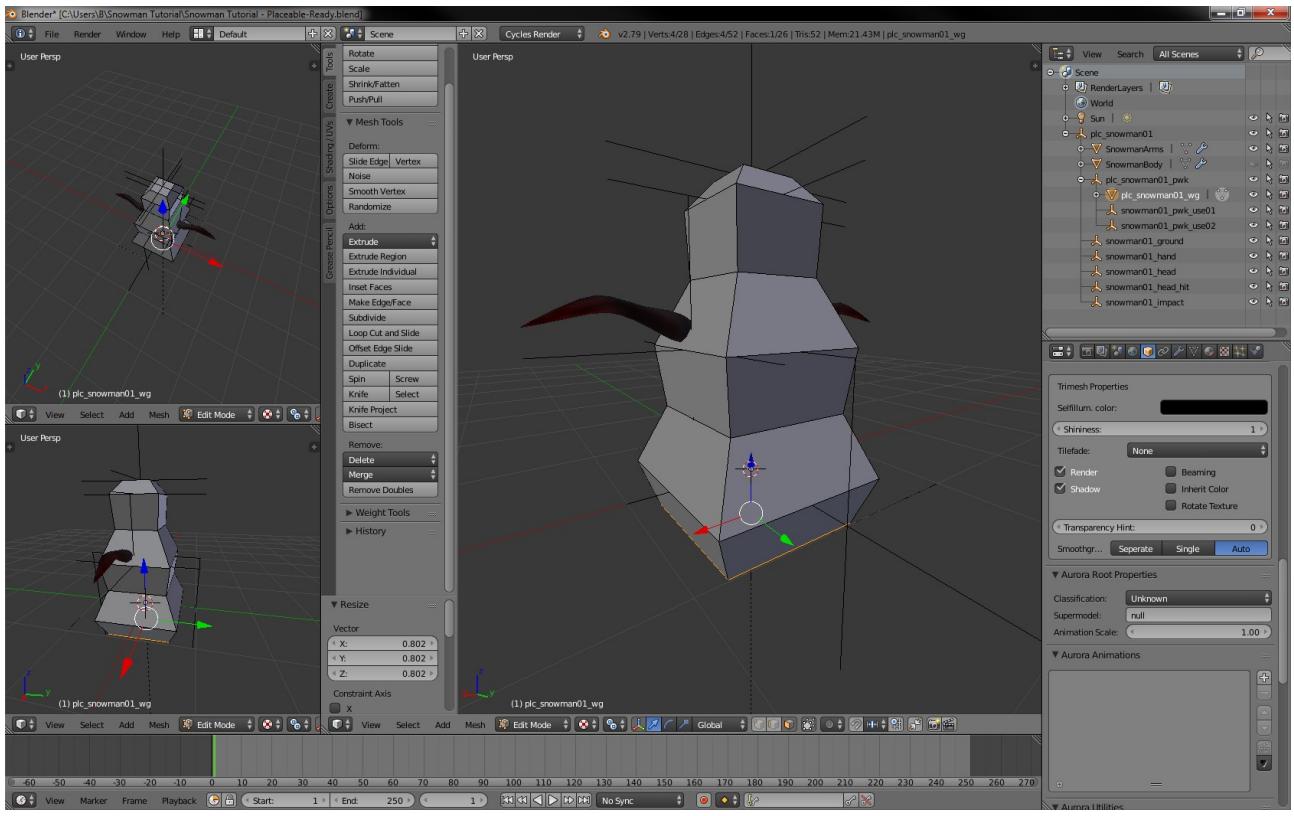
**Hold ALT+SHIFT while clicking an edge** to attempt to select other linked edges.



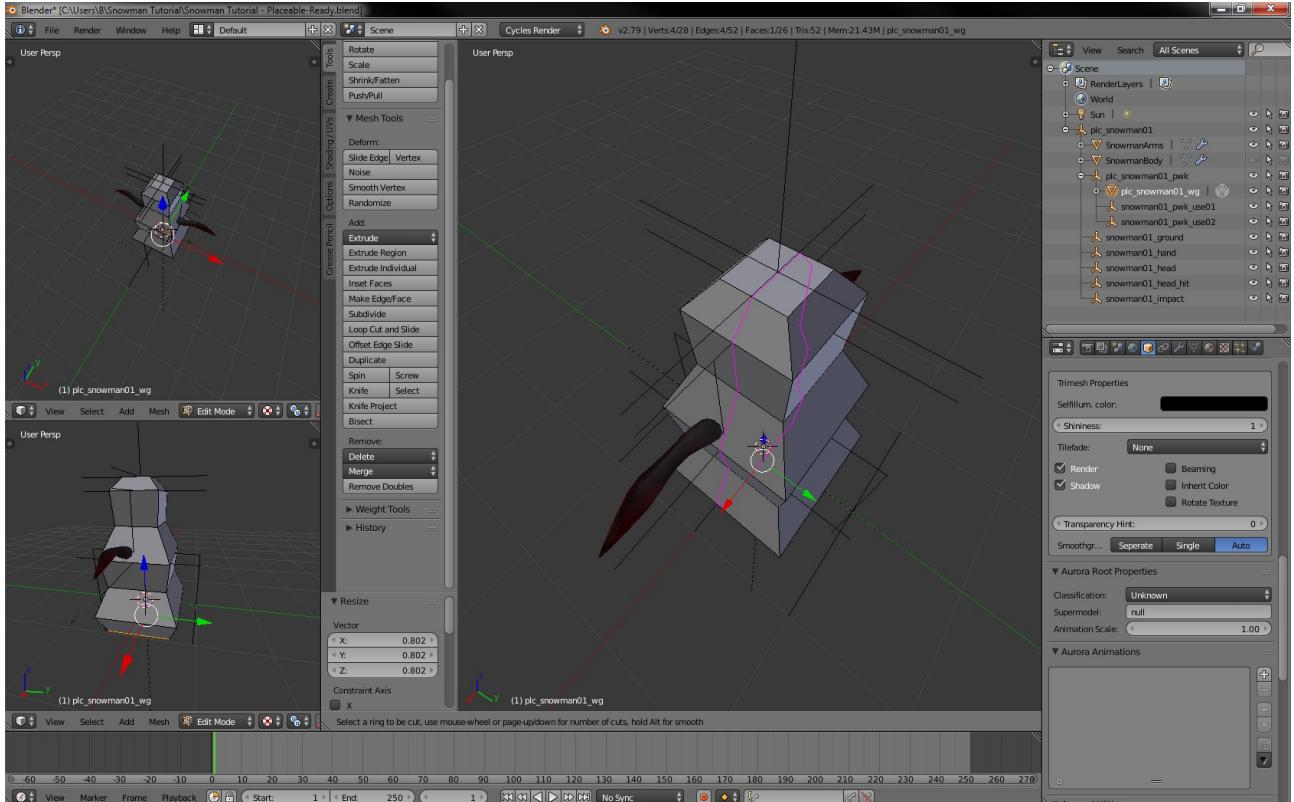
With something selected, hit **S** to scale it up or down.

You can also temporarily switch the snowman body invisible by hitting the eye icon in the objects list.

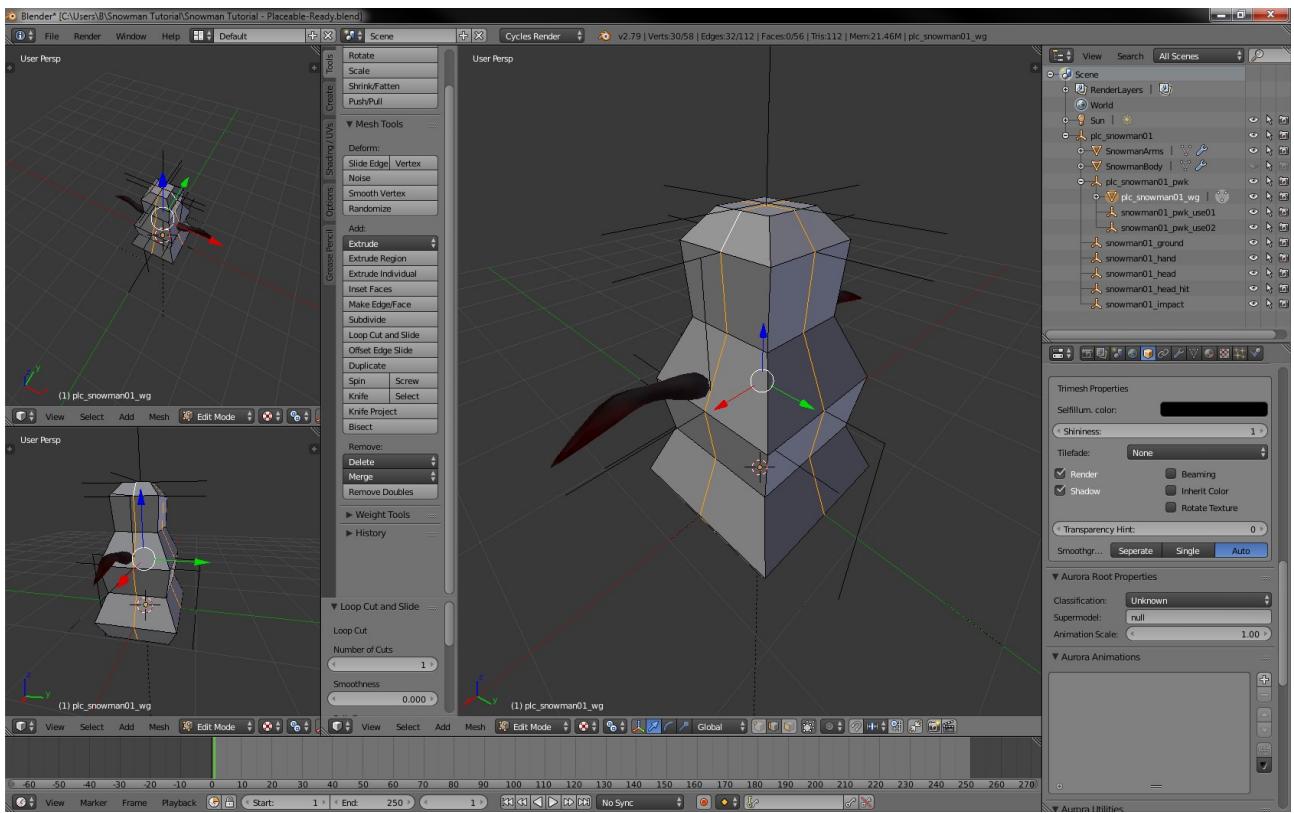




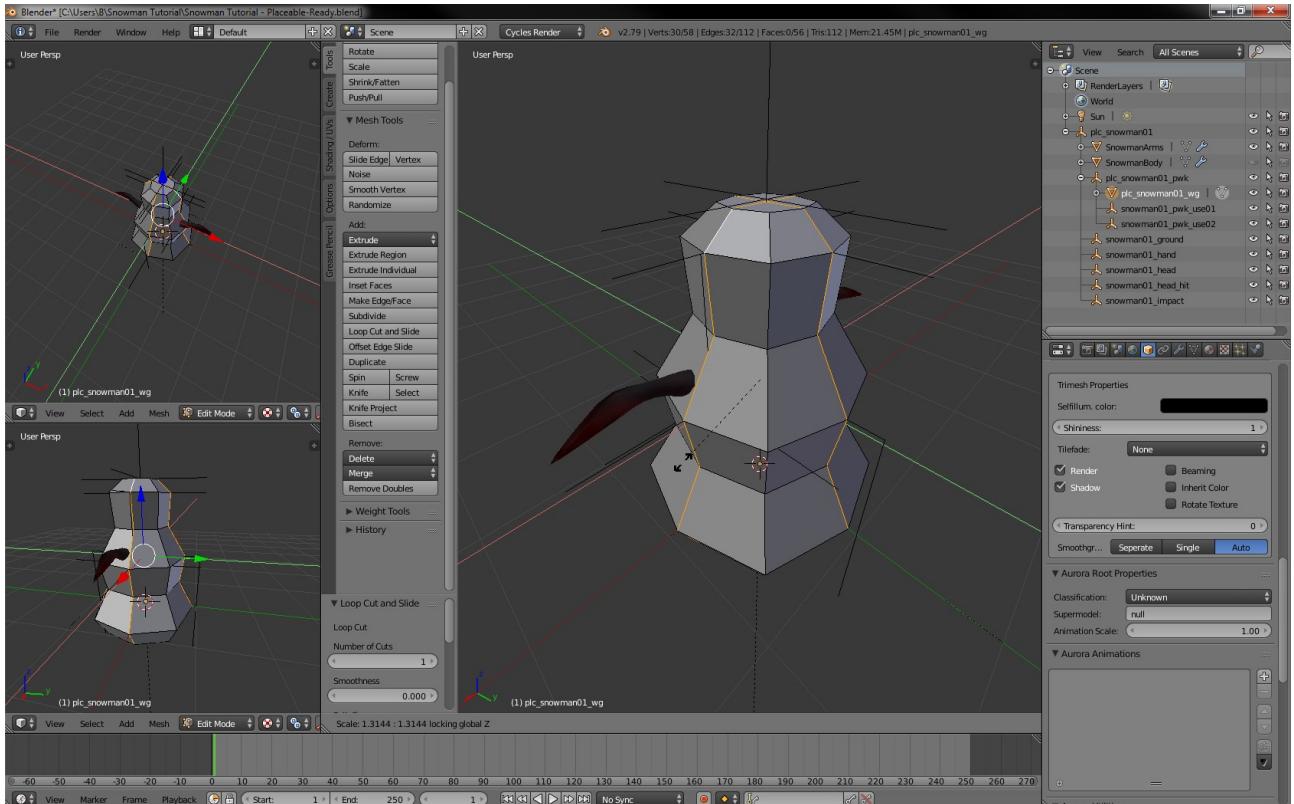
Reminder: Select faces and hit E to extrude them.



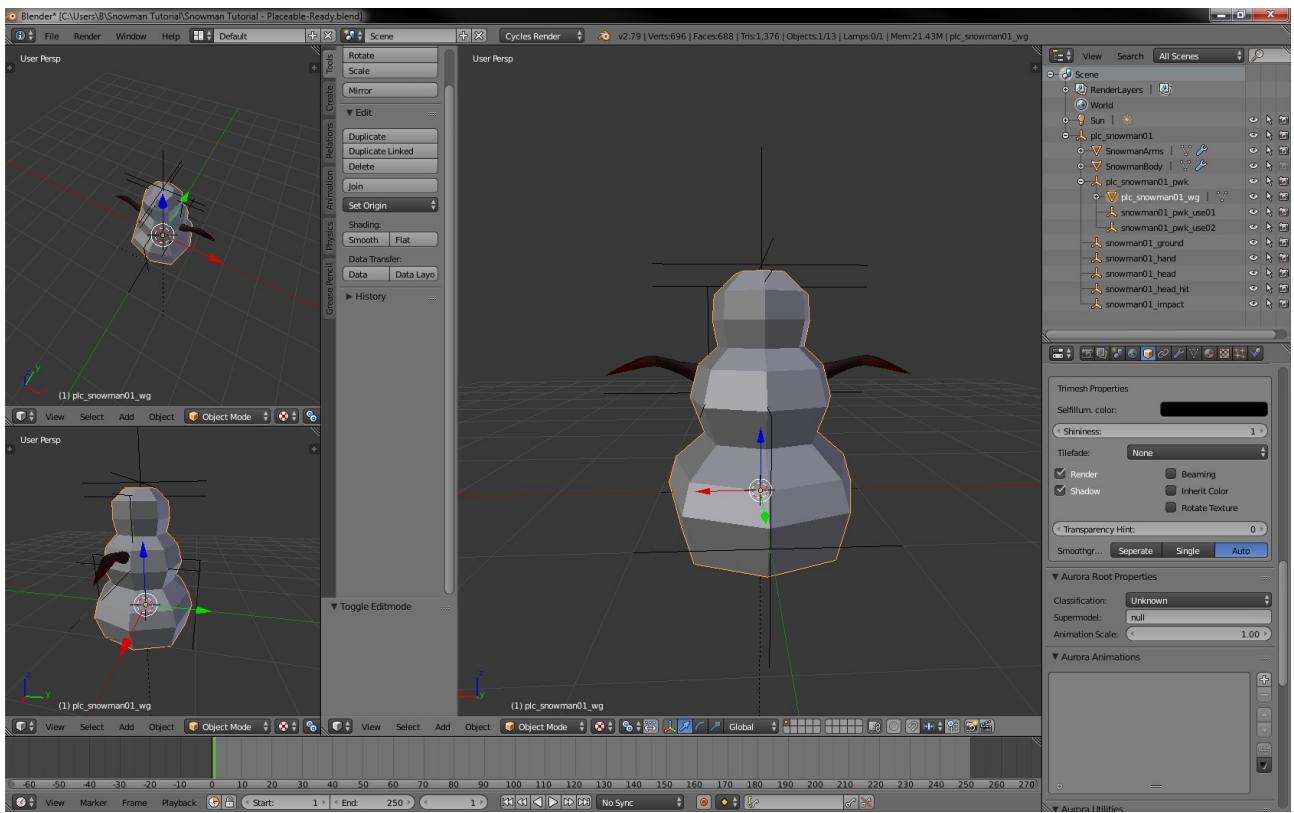
Loop cuts can also be placed vertically; same method as usual, just pointing at a different part of the mesh.



While scaling something, hit **SHIFT+Z** to exclude the Z axis while scaling.



Give the walkmesh roughly the shape of the snowman. Let it be a little bit bigger than the snowman itself, so the player cannot get close enough to the mesh to clip through it.



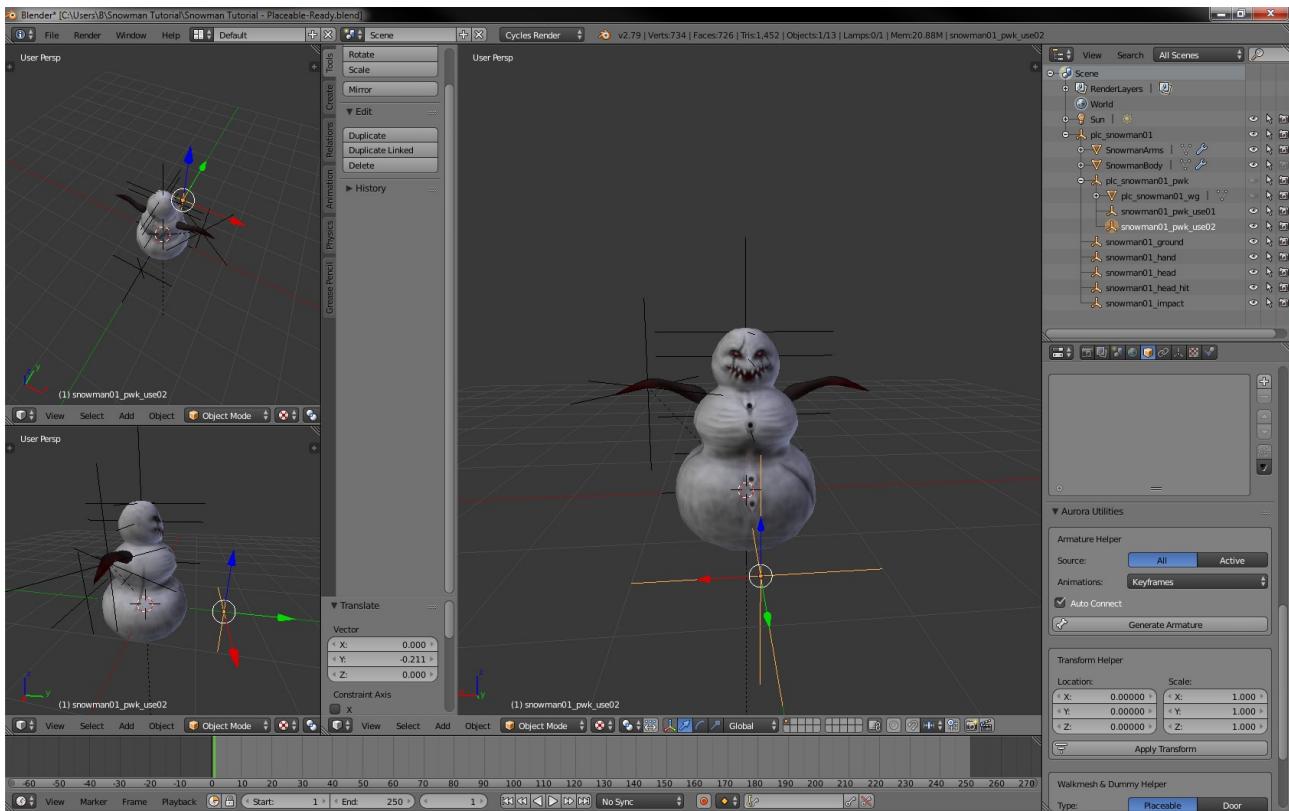
There we go, it's a walkmesh. Doesn't include the arms, but, heck. I don't think it really needs to, in this case.

Try to keep the walkmesh models as low in complexity as you can. Honestly, the one I've set up above is probably a bit overkill, too. A cylinder would've done just fine, here.

(this would be a pretty good time to save!)

## - Positioning the Use / Impact Nodes

OK, now, check and make sure that the Use and Impact nodes for the placeable are in good locations:



Just grab them and reposition them, while you're in **Object Mode**. Hit **G** and then **X**, **Y**, or **Z**, to move the node along a particular axis, until it's whereever you want it to go.

Note the following:

The Use nodes should not be within the area covered by the walkmesh, since the player can't move there.

The Head and Head\_Hit nodes should be in the head of the snowman, or in case of headless placeables, somewhere around the top of the model.

The Impact node should be somewhere in the middle of the model; this is the location that, for example, spell projectiles will be firing at.

The Ground node, *generally speaking*, should stay where it is. Floor-aligned spell VFX may be applied at this location, like the Entangle vines.

If a placeable has hands or arms, then moving the Hand node there can't hurt.

(save point!)

## - Setting up the Material

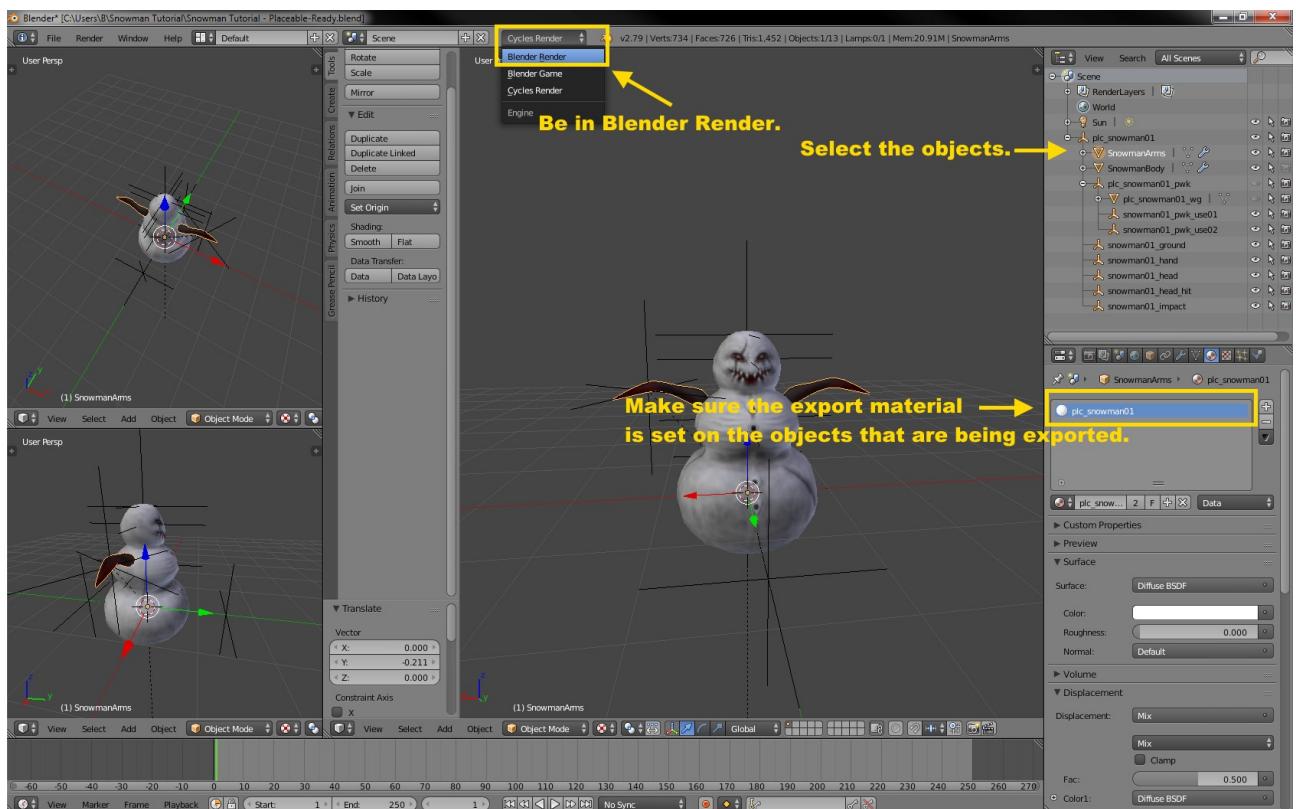
OK, gettin' closer. Let's set up the material for export.

To chuck information at you before we go through the steps one by one, to export right, the material needs the following:

- 1.) All textured mesh parts should be using a material that is intended for export (or, to say, you should check each object one at a time and make sure that they're all using a material you intend them to use).
- 2.) The textures must be set on the material (also: to be on the safe side, I also always name the textures to match their image files, but I'm not sure that's actually necessary).
- 3.) The images must be linked to the correct textures (as seen below, our export material has three textures set on it, and each of those textures has a different image linking to it (\*\_d, \*\_n, \*\_s)).

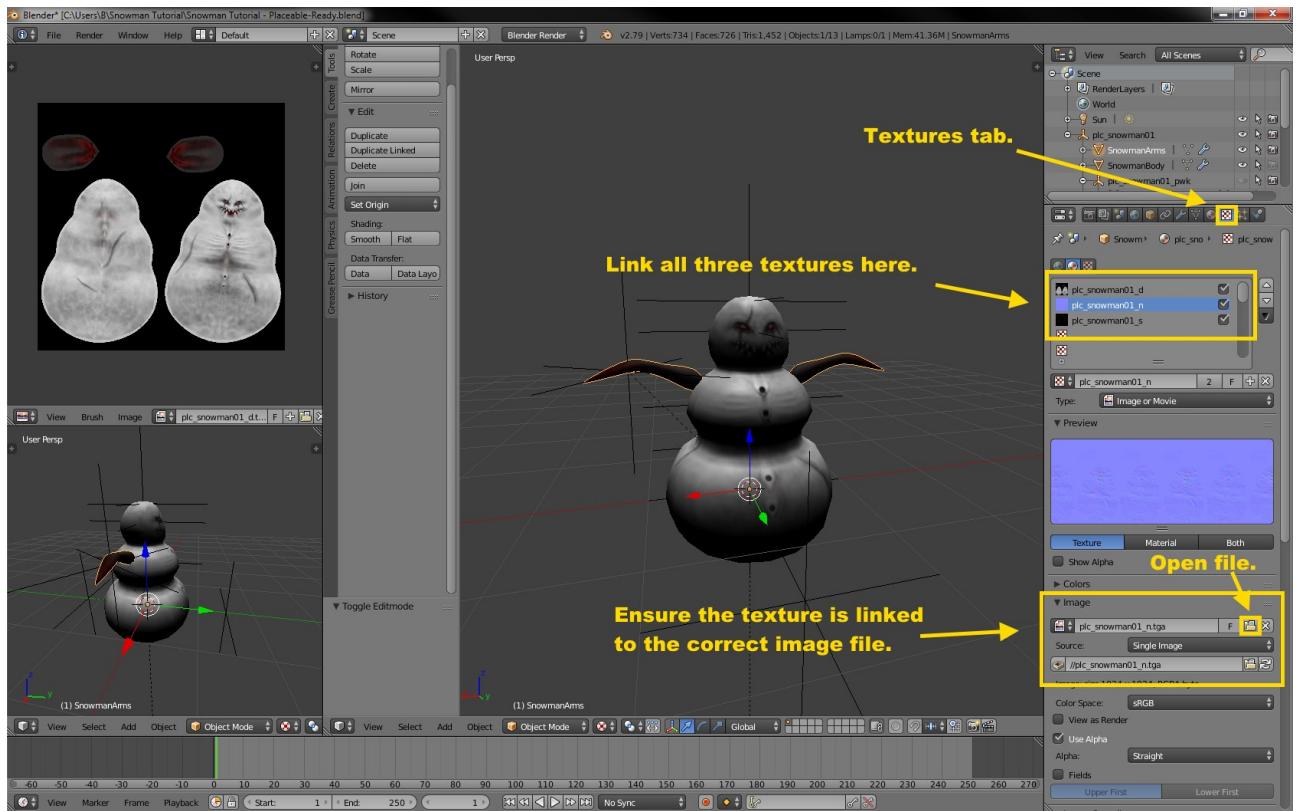
OK, let's go through the process:

Switch to Blender Render, and make sure that all your mesh objects use a material that is intended for export.



In my case, these objects are SnowmanBody and SnowmanArms, both of which need to be using the material `plc_snowman01`, which I'm preparing for export here.

Next up, create and link the textures.



This is a throwback to Part 1.2 of the tutorial; setting up a new material, adding three textures to the material, naming everything clearly, and linking the appropriate images to the textures. I'll be going with a material I'll call `plc_snowman01`.

If you've been following along with the meshcrafting and texturing tutorial, then you've currently got your texture files named `snowman_d`, `snowman_n`, and `snowman_s`.

You can use these names if you're working alone on a practice snowman, but be aware: File names need to be unique once we import them into the game, so if you're making more than one snowman, in which all the textures are different, it'd be very wise to give each and every one an unique number or name - and the same goes for the images that belong to it.

Example:

MODEL NAME	MATERIAL	TEXTURE 1	TEXTURE 2	TEXTURE 3
<code>plc_snowman01</code>	<code>plc_snowman01</code>	<code>plc_snowman01_d</code>	<code>plc_snowman01_n</code>	<code>plc_snowman01_s</code>
<code>plc_snowman02</code>	<code>plc_snowman02</code>	<code>plc_snowman02_d</code>	<code>plc_snowman02_n</code>	<code>plc_snowman02_s</code>

It's maybe a bit pedantic, but this way, the likelihood of anything getting tangled up is extremely low.

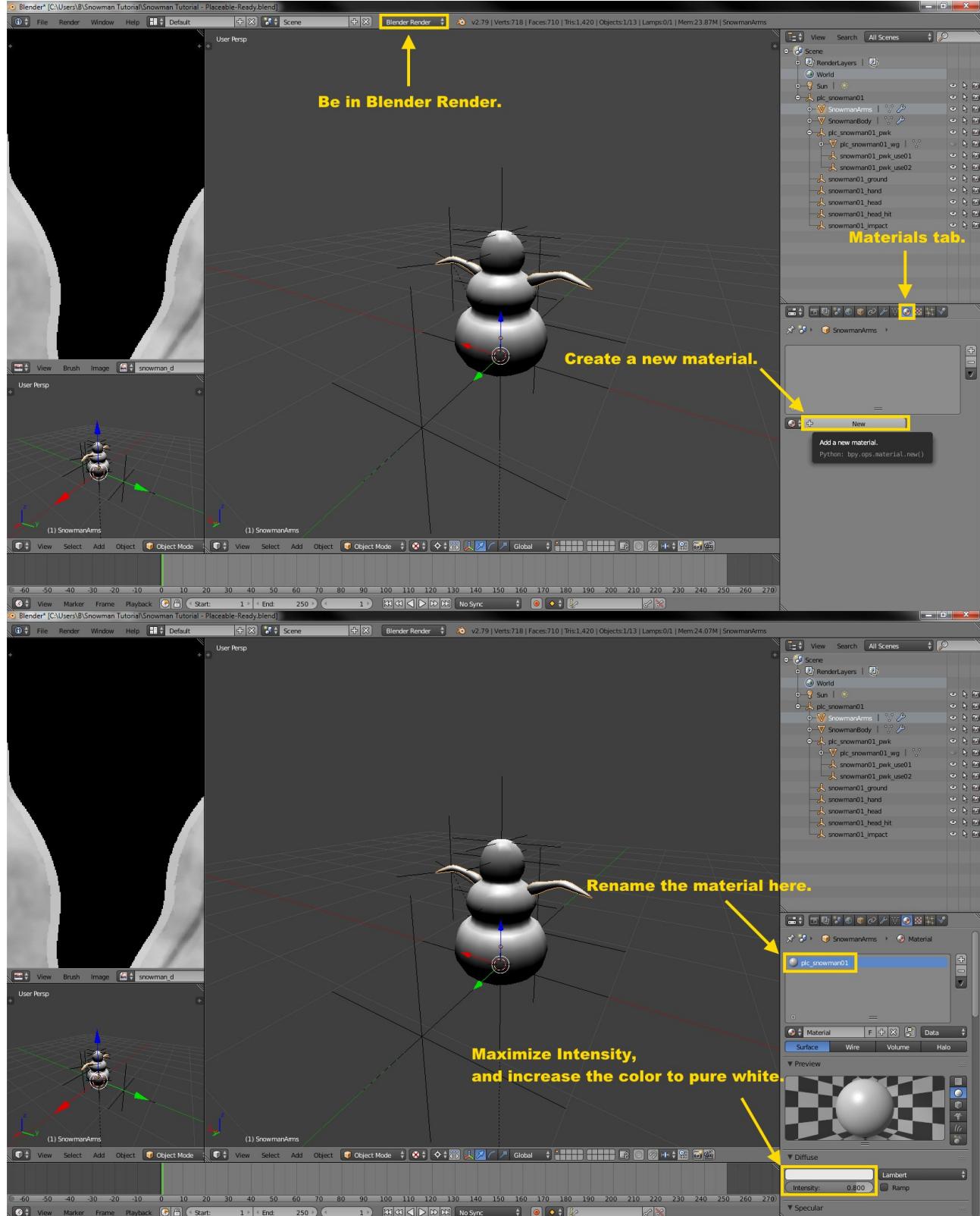
Note that you can reuse the same image file for multiple different models, but if you do this, any changes made to the image will affect *all* models that reference that image. This is extremely useful and done on purpose very often; tilesets, for instance, generally reuse the same images to have identical types of stone, wood, or grass, on all the separate tile models.

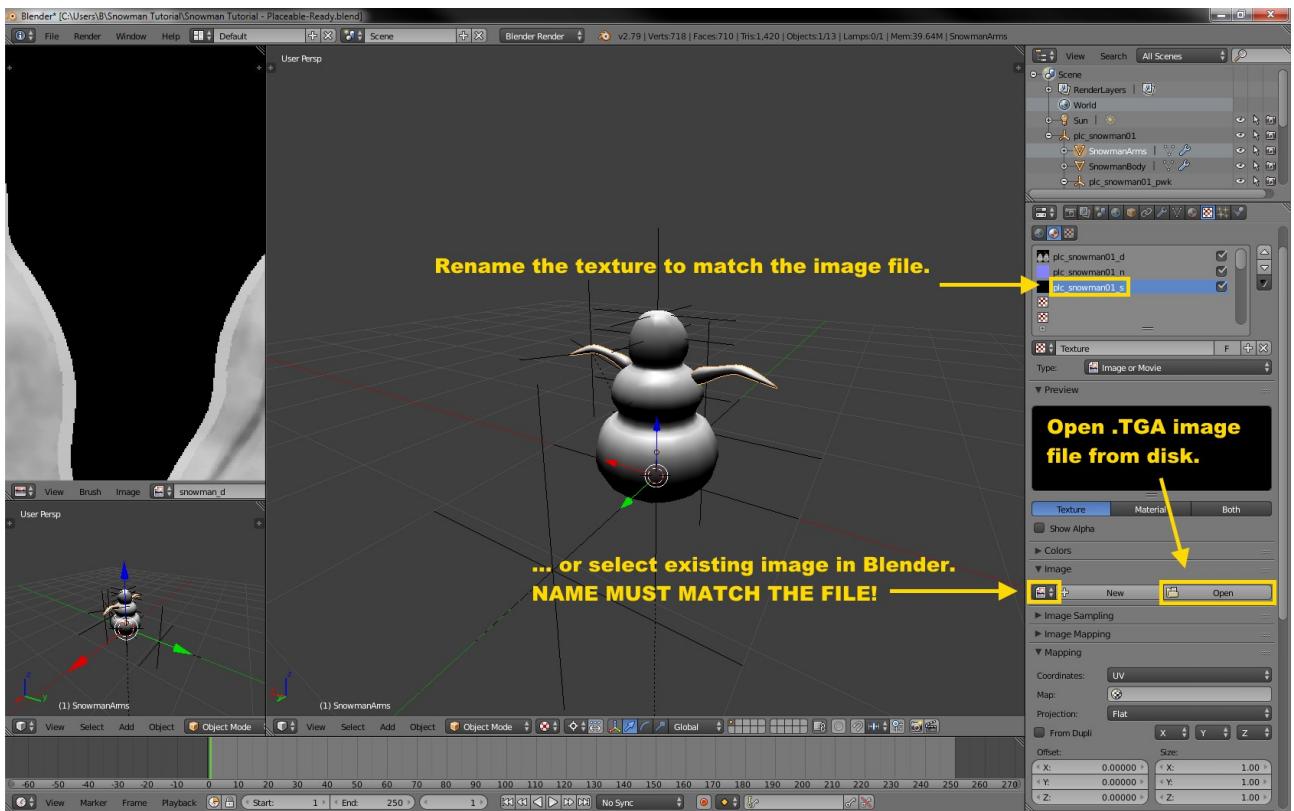
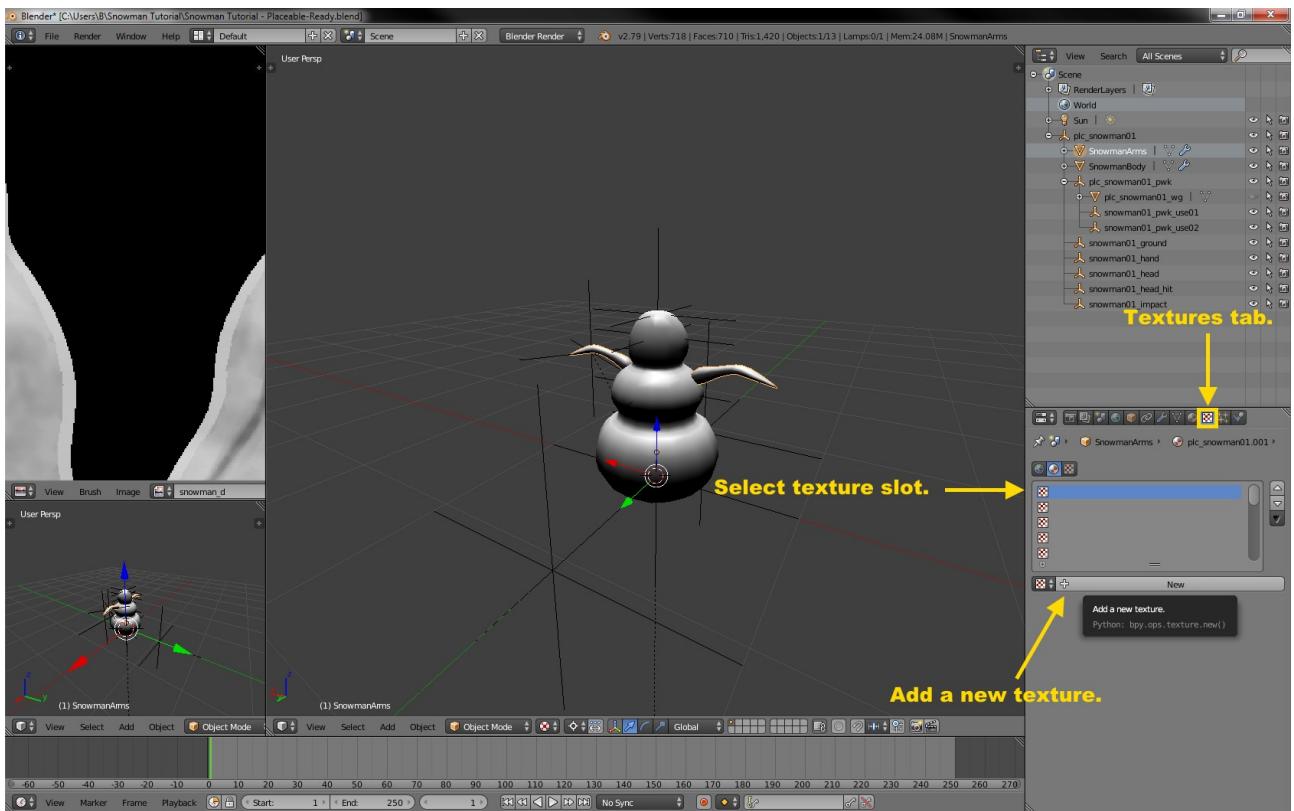
(you should still be saving every once in a while)

The following pages showcase how to create materials and set up images. They are here as reminders, in case you've forgotten, or in case you're a reader who came here specifically to learn how to set up models for export, like ones from third party sites.

So if you've already completed the above steps, skip straight ahead to the Check the Image Formats-section. You're already done here.

Otherwise:





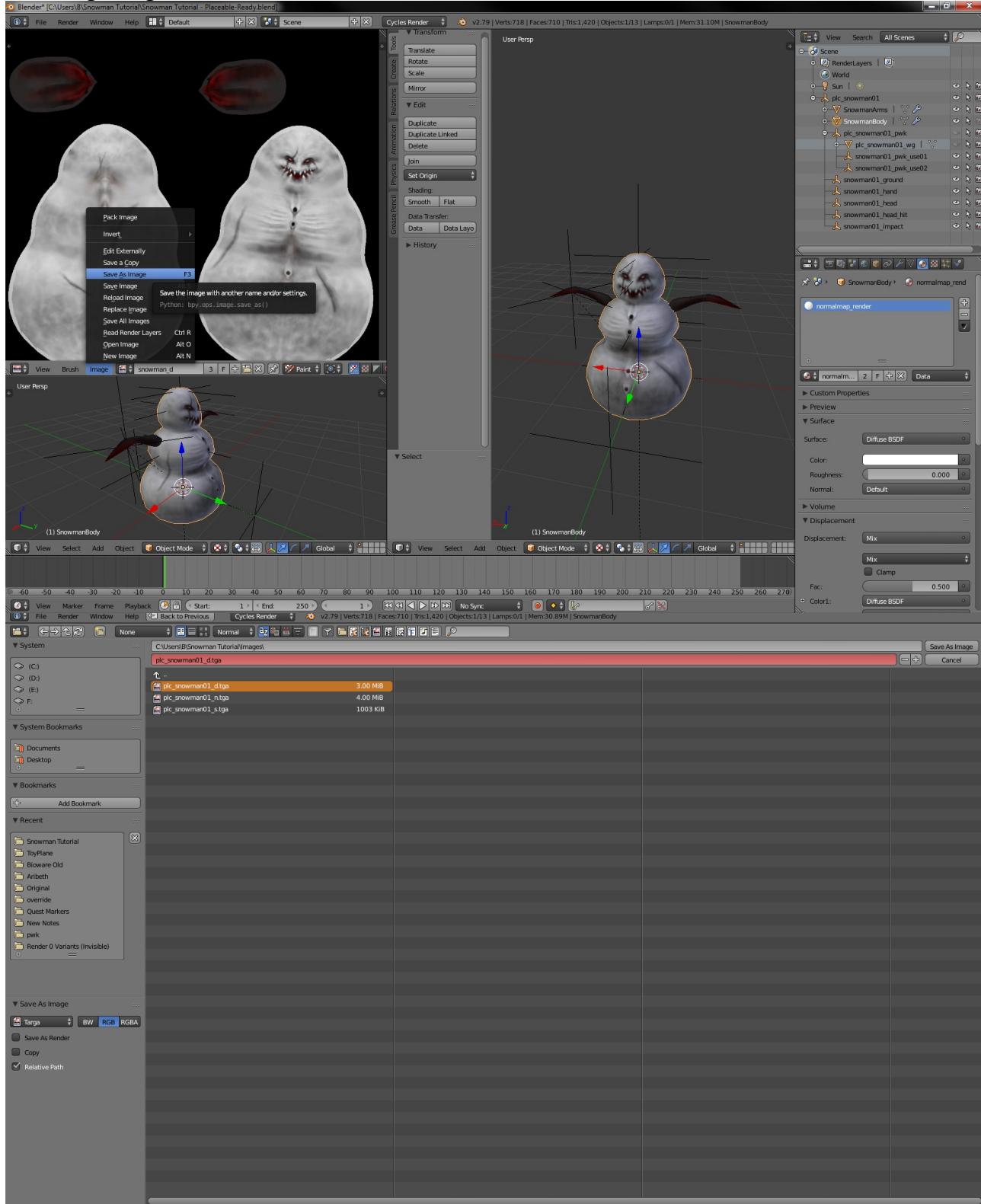
Yep. That's about the way this goes.

Watch out for the image file names. They need to match the actual image files from now on.

## - Check the Image Formats

If your image files are at this point something other than .TGA files, it's time to TGA-ify them.

In the UV/Image Editor, you can click on Image and select Save As, to save a copy of the image, including using a different name or file format.



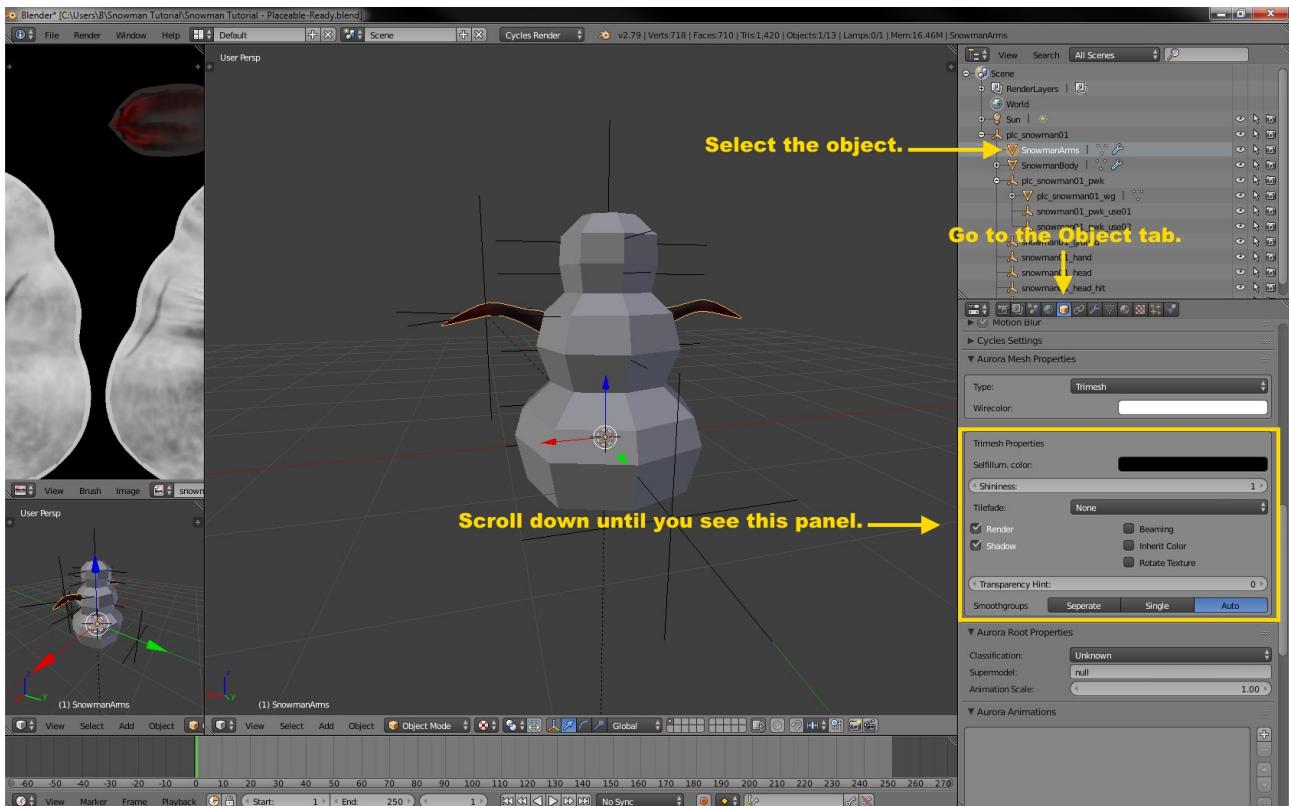
Save the diffuse, normalmap, and specular map, as TGA files. Make sure that the names of these files match the names of the textures/images on the textures!

## - Check the Individual Object Settings

OK, next up, information:

Each object in the model has its own individual settings, among them things like Render (is this object being visibly rendered ingame?), Shadow (does this object cast a shadow ingame?), and Self-Illumination (Is this object individually lit to some degree, independently from the area lighting?).

So before we export, we need to check through the objects that could be getting rendered ingame and tick settings active or inactive. In the case of the snowman I'm prepping for export here, these are SnowmanArms, SnowmanBody, and plc\_snowman01\_wg (the walkmesh box).



In the case of this snowman, I'm going to put the following settings onto the following objects:

### SnowmanArms

- Render 1
- Shadow 1

(The arms are visible ingame, and cast a shadow)

### SnowmanBody

- Render 1
- Shadow 1

(The body is visible ingame, and casts a shadow)

### plc\_snowman01\_wg

- Render 0
- Shadow 0

(The walkmesh box is not visible ingame, and does not cast a shadow)

On shadowcasting:

Note that it's frequently desirable to turn off Shadow on small additional decor objects. In the case of the more complex snowman, for instance, I deactivated Shadow on the scarf, as it makes next to no difference ingame.

It's also frequently desirable to deactivate Shadow on the original mesh object and set up a separate shadowbox object, when you've got an object that is a complex shape constructed out of significantly many more polys than the shadow would need to have. I've seen shadows cast by complex meshes crash the game. So, as a ground rule, when crafting complex meshes, consider setting up a separate, simpler shape and settings

- Render 1
- Shadow 0

... on the original mesh, but

- Render 0
- Shadow 1

... on the simple one that will be casting the shadow.

That aside, Self-Illumination Color lets you do things like GLOWING RAINBOW GNOME EYES OF MADNESS!



(his name is Cackles and he is nice)

I think for the scope of this tutorial, these three will do. Check out the Neverblender documentation for more information on the various other settings, if you're interested:

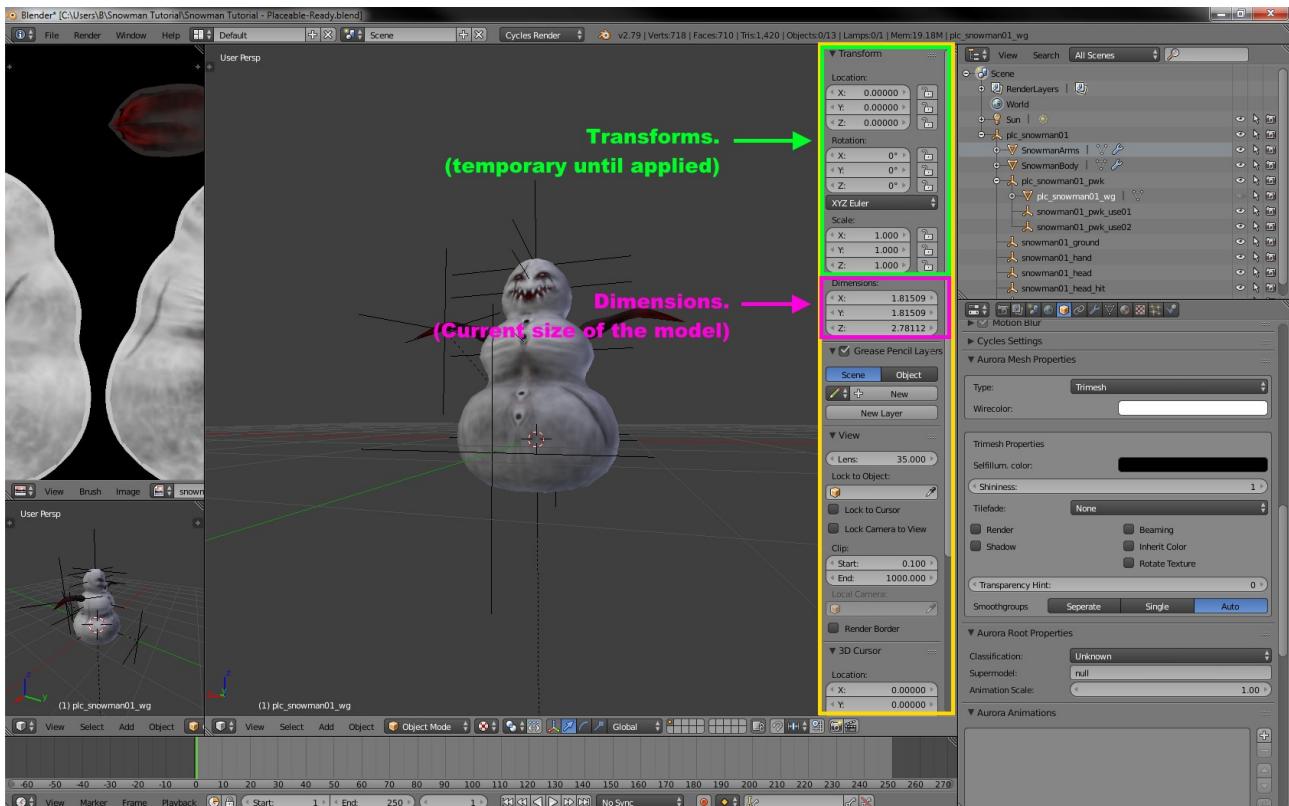
<https://nwn.wiki/display/NWN1/NeverBlender#NeverBlender-Export>

Moving on. PS: Saving frequently is still cool.

## - Check the Positioning and Size of the Mesh

When you're working in 3d space, it's often difficult to tell by eye what the size of a model is in relation to others, since they can all be scaled up or down and maintain their proportions. So, time to check.

Hit **N** to open the right-hand menu.



The dimensions one is the one we want, right here, right now.

In NWN, areas are built out of tiles, like a big gameboard made up out of squares:



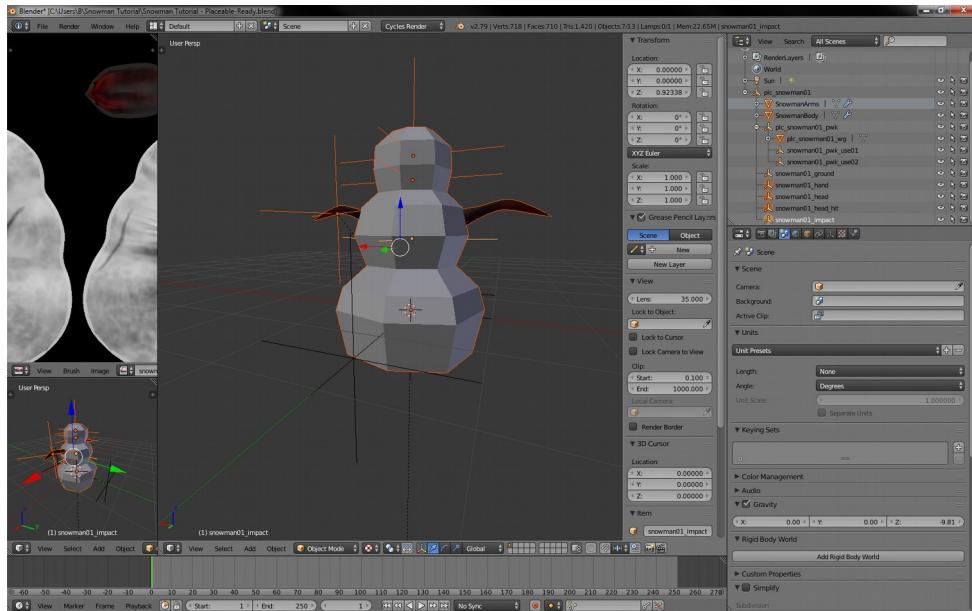
Each of these tiles translates to 10.0 x 10.0 in Blender's default units. If a placeable model has X and Y dimensions of 10.0 and 10.0, then it will be exactly one tile in size ingame.

Far as this tutorial goes, what I need to do before I can export is move the snowman up a little. My snowman would be sunken into the ground, if I export him like this.



See the position of the origin point? The visible base of the snowman is significantly below that point.

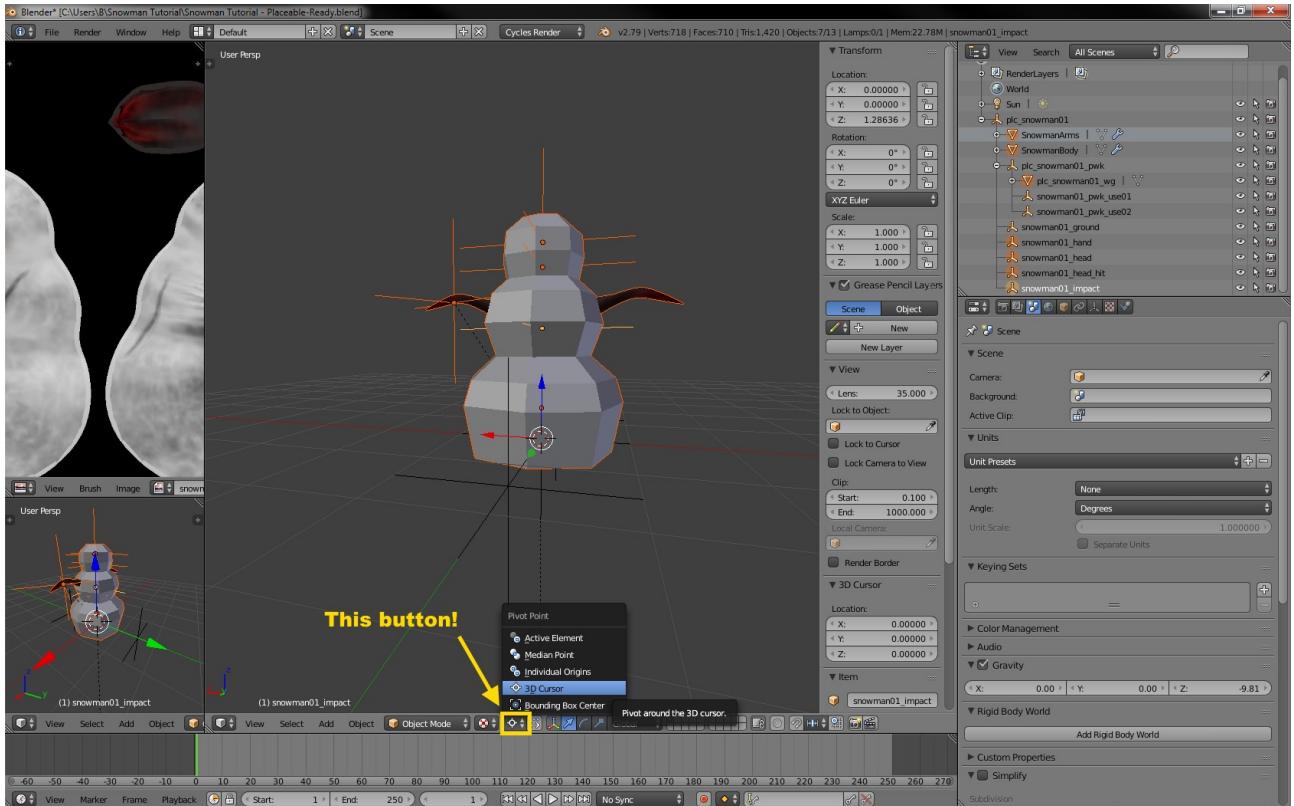
Now, because I've already positioned the impact nodes and already created the walkmesh (educational! value!), I can't just grab the body and the arms and move them. I need to select all the relevant nodes (and the walkmesh, too!) and move them all at the same time.



Note that you can use **SHIFT+G** to select groups of objects, such as selecting the AuroraBase and then hitting **SHIFT+G**, then **C**, to select all children of the AuroraBase at once.

I'm also going to scale the snowman, while I'm at it. Specifically, I'll scale it down. It's a bit large at the moment.

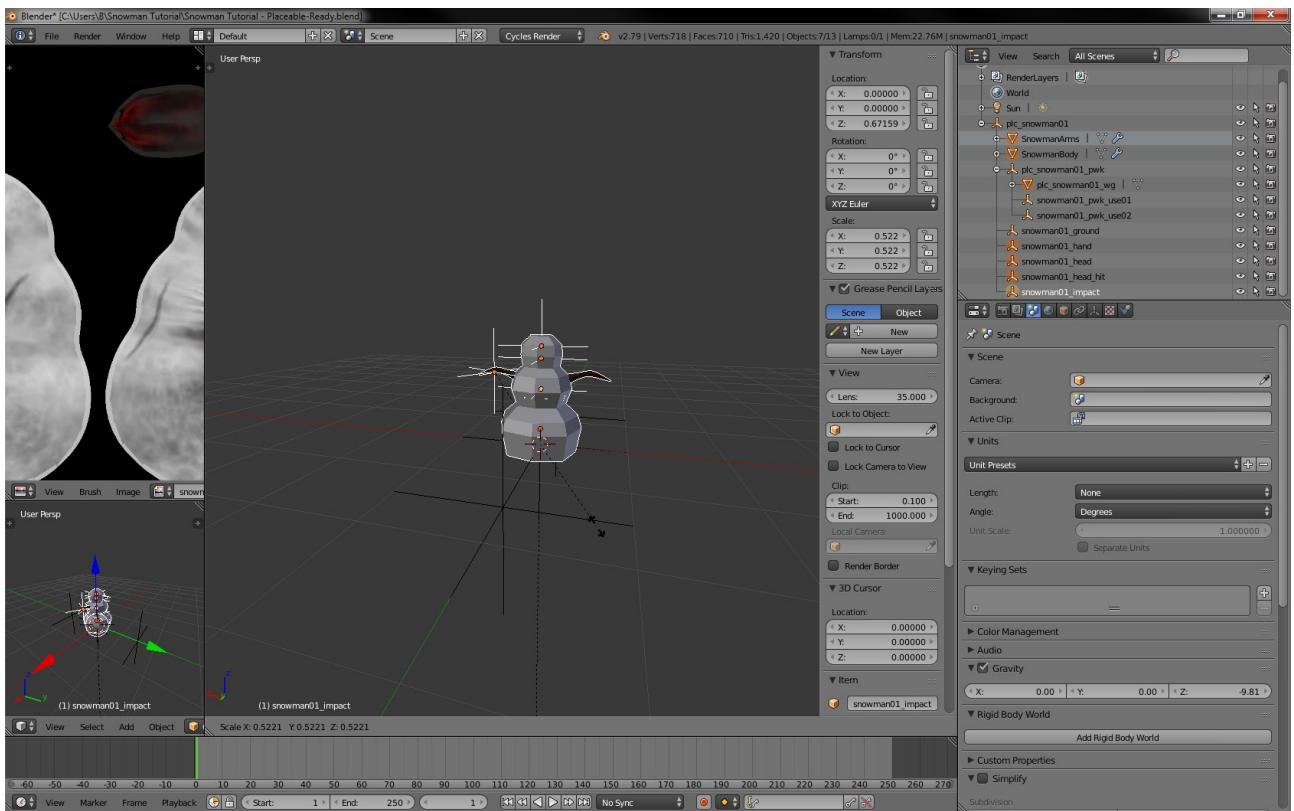
New trick, and possibly about the first time we're using the 3d cursor for anything in this tutorial: Blender also lets you specify the center location when you're rotating/scaling the selection, and one of the locations you can specify for this is the 3d cursor.



Hit **SHIFT+S** -> Cursor to Center to place your 3d cursor in the center of the scene, if you've misplaced it at any point.

By doing this, we can put the 3D cursor anywhere we want (including in the middle of the scene, like here) and then scale/rotate around that point.

Be in Object Mode when you scale your snowman, and reminder: Hit **S** to scale.



Tiny snowman sez hi!

Now, we have something going on here that can potentially really wonk up the export - **unapplied transforms!** The horror!

(ps you should be saving pretty much all the time)

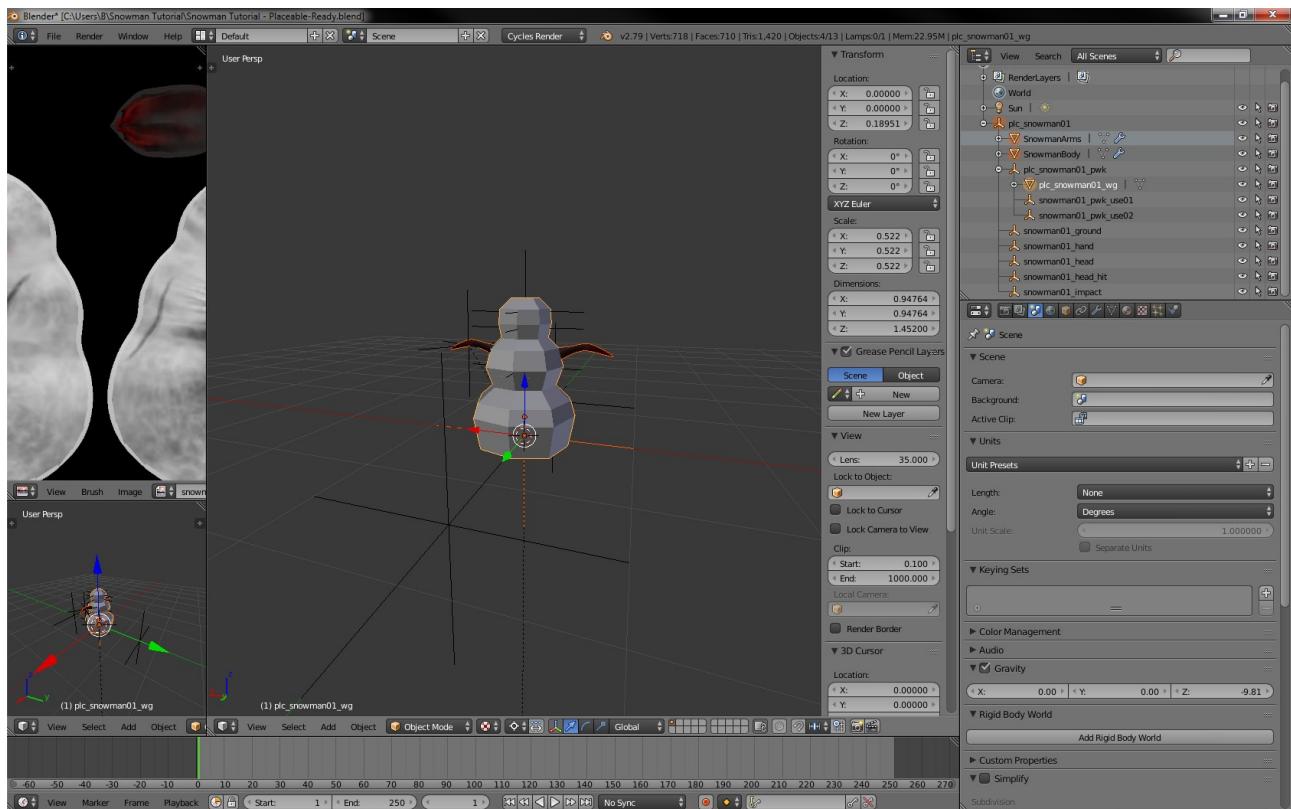
## - Apply Transforms

OK, info time:

Remember how Modifiers were temporary; something affecting the model underneath the modifier, but preserving the original state (allowing the modifier to be deactivated or moved up and down in the stack) until they were *applied* to the object that they were affecting?

This works for transforms (increase or decrease size, move position elsewhere, rotate along any axis), too. So at the moment, if we've been scaling or repositioning objects, there are a bunch of unapplied transforms here.

First, we want to apply **all** the transforms on the visibly rendered mesh objects, on the AuroraBase itself, and on the walkmesh box. Select those four. In my case, these are plc\_snowman01, plc\_snowman01\_wg, SnowmanArms, and SnowmanBody.

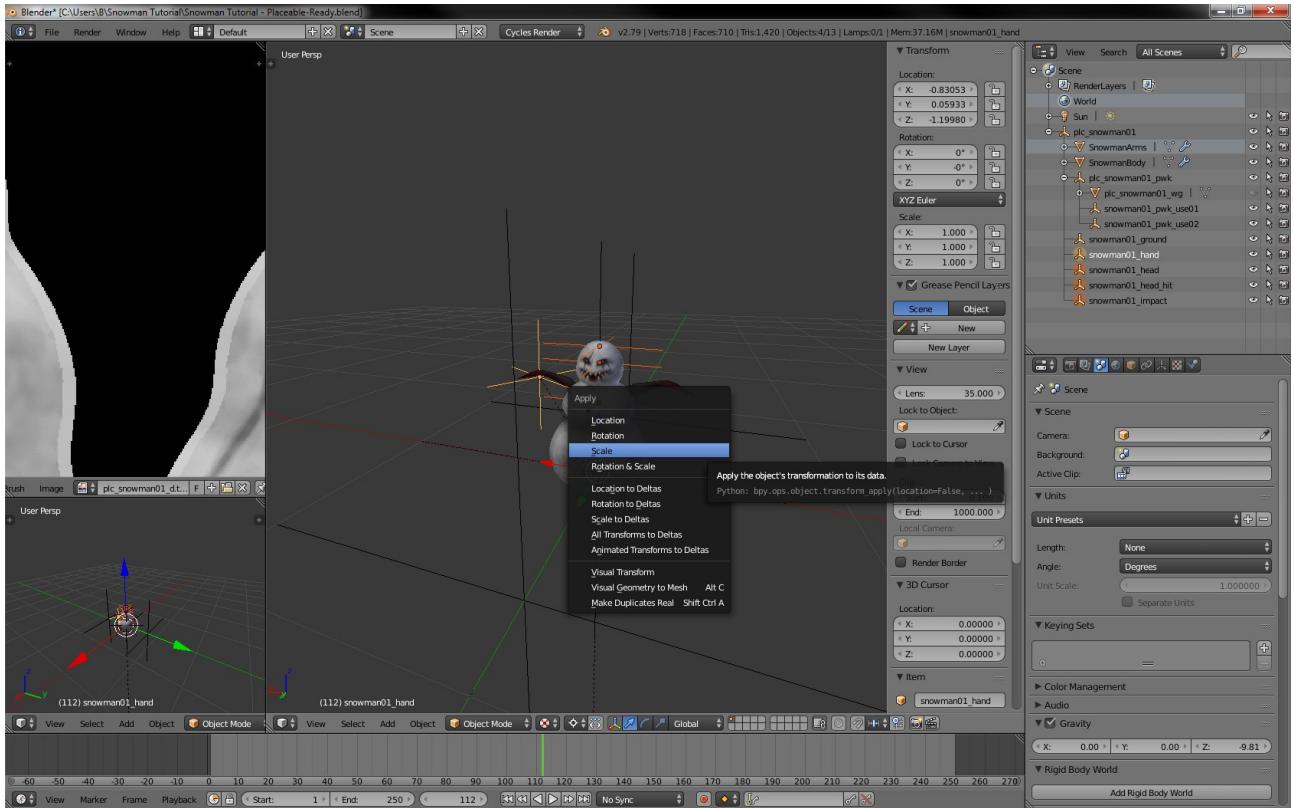


Now, hit **CTRL+A**, and choose **Location**. Repeat, and choose **Rotation & Scale**.

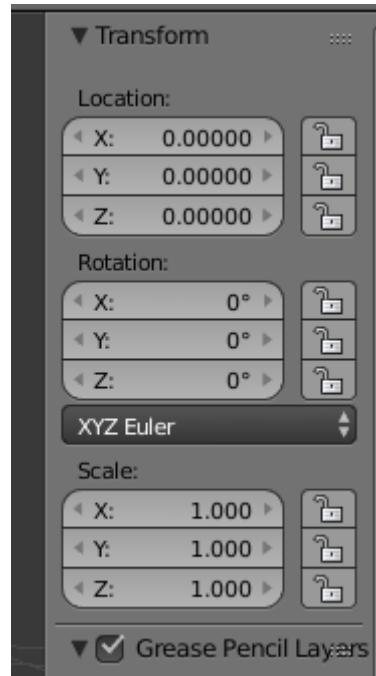
Next, we want to select the nodes we've previously scaled along with the snowman. In my case, these are snowman01\_hand, snowman01\_head, snowman01\_head\_hit, and snowman01\_impact.

If you've followed along with the tutorial precisely, then you'll notice that they currently have Location, Rotation, and Scale transforms on them. For these guys, we just want to apply the Scale transforms. If we apply the Location transforms, their origin points should jump back to the middle of the scene, resetting the position of the entire node.

So, with the aforementioned nodes selected, once again, **CTRL+A**, but this time, choose **Scale**.



After doing this, give all the objects a last checkover to be thorough. Look at this thing up here:



The empty node objects are allowed to have Location and Rotation transforms set on them. We preferably do not want **any** location, rotation, or scale transforms, to be set on the visible mesh objects, on the AuroraBase, or on the walkmesh box. Same goes for any potentially involved invisible shadowbox objects.

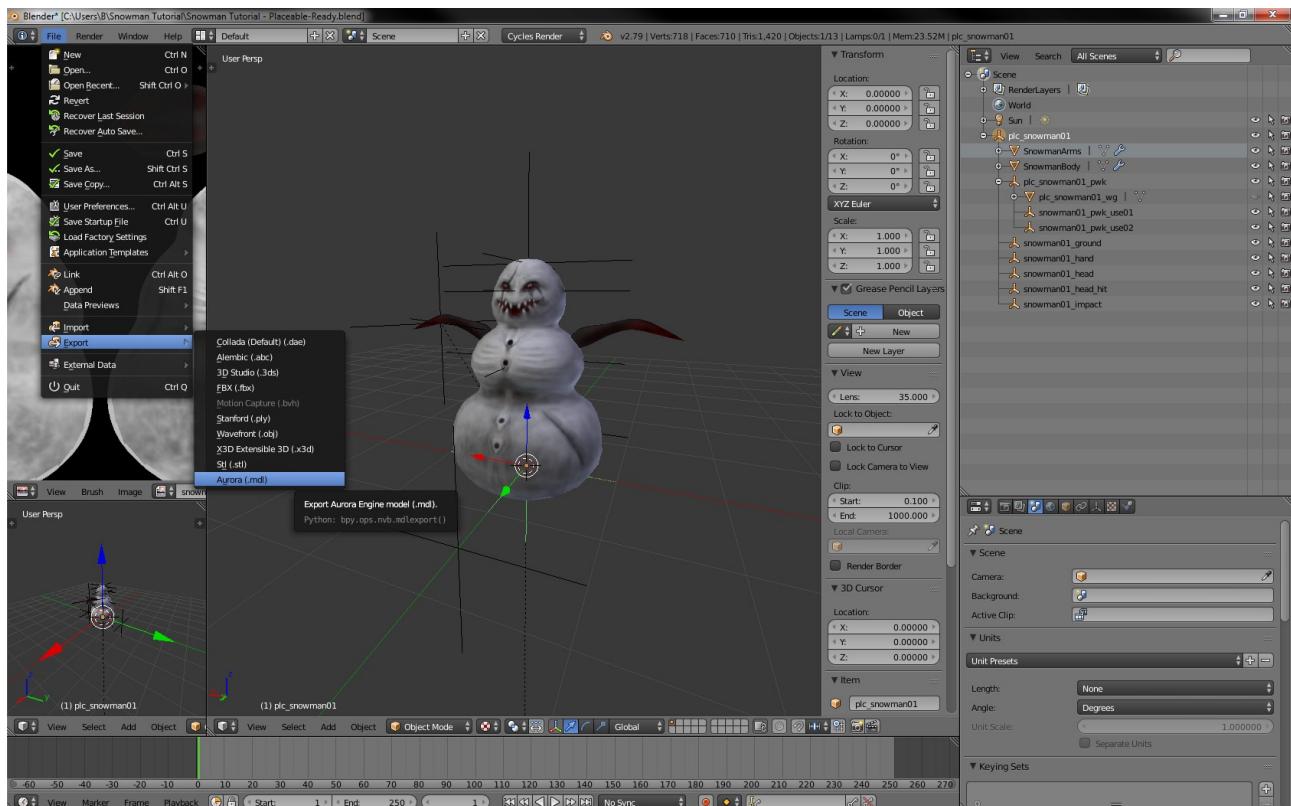
Also, this is a good time to save! 'cuz you know what? We're done.

## - Export!

If everything went off without a hitch, which I rather hope it does because it will make me look terribly foolish as a hobbyist teacher if it didn't (**fact:** if you did something wrong, it's actually *my* fault for not providing clearer instructions!), we should now have a model set up ready for exporting. :-D THE TIME OF GETTING THE MODEL INTO THE GAME IS AT HAND!

Our last step for Part 1.3:

Select the AuroraBase, and then hit File -> Export -> Aurora (.mdl).



Save the resulting .mdl file **using the exact same name that the AuroraBase has**. In my case, this is **plc\_snowman01**.

On the off chance that you want to practice setting up models for export sometime, I'm going to chuck the following links at you:

<https://nasa3d.arc.nasa.gov/models>  
[https://www.turbosquid.com/Search/3D-Models/free?  
exclude\\_branded=1&exclude\\_editoriallicense=1](https://www.turbosquid.com/Search/3D-Models/free?exclude_branded=1&exclude_editoriallicense=1)

When using models from free asset sites, always look for the usage permission details put forward by the author, and do try to be respectful and appreciative when you interact with them. :-) There's usually a real person behind the free stuff on the internet, and they're generally not getting paid for the time and effort they've put into this stuff, given that it's available for free.

So, yeah! Guess what? We're done here. Time to go get this thing into the game!

Next up: PART I, SECTION 4 - ADDING A PLACEABLE APPEARANCE TO NWN