Beginner's Guide to 3D Art Assets

Buildings

Introduction

This guide is a step-by-step tutorial that will walk you through how to create 3D art and add it to your ModBuddy project. It focuses on the 3D art for buildings in districts, including buildings in the city center district. It assumes that you already have a ModBuddy project started, that you have basic knowledge how to navigate ModBuddy, and that you have basic knowledge of your operating system. The operating system I use is Windows 7, so be aware that other operating systems may have different file paths and may have compatibility issues with some of the programs listed. There are multiple different ways to create 3D art, and this guide will walk you through just one of those ways. At the end of this guide you will have a 3D Model of a building that you created in your mod.

Programs and Files You Will Need

- Sid Meier's Civilization VI Development Tools (SDK)
  - In Steam Client Application hover over LIBRARY tab and select TOOLS. Scroll down list until you find Sid Meier's Civilization VI Development Tools. Right click and select Install Game.
- Sid Meier's Civilization VI Development Assets
  - In Steam Client Application hover over LIBRARY tab and select TOOLS. Scroll down list until you find Sid Meier's Civilization VI Development Assets. Right click and select Install Game.
- CivNexus6 (Author: Deliverator)
  - Go to (https://forums.civfanatics.com/resources/civnexus6.25987/) and click the Download Now button. After it is downloaded right click the zip file and select Extract All. Set destination to C:\Program Files (x86) and click Extract.
  - Note: I had an issue with this starting. This is how I fixed it:
    - On your desktop, press the Windows Key + R to open the Run window. Type regedit and press OK. Navigate to HKEY_CURRENT_USER\Software\Firaxis\Tools. If you do not have a Tools folder yet, in Firaxis, right click, hover over New, and click Key from menu. Name it Tools. In Tools, right click, hover over New, and click String Value from menu. Name it ToolAssetPath. Right click on ToolAssetPath and select Modify. For the Value Data field enter in C:\Program Files (x86)\CivNexus6_1.2.3 and press OK. The folder you have CivNexus6 in may have a different name because you may have a newer version, so enter whatever your CivNexus6 folder is called instead of CivNexus6_1.2.3.
- Deliverator's Blender Scripts (Author: Deliverator)
  - Go to (https://forums.civfanatics.com/threads/deliverators-blender-scripts.497821/#post-12507334) and click the Download Now button. After it is downloaded right click the zip file and select Extract All. We will add this to Blender later.
- Blender (2.79 or higher)
  - Go to (https://www.blender.org/download/) and click the Download Blender button. After it is downloaded install it.
  - Note: I recommend using a 3 button mouse, you technically don't because you can change it to keyboard buttons.
- Normal to Map Node (Author: David Gayerie)
  - Go to (http://digicreatures.net/articles/cycles-normal-baking.html) and click downloadable file normal_to_map.blend. We will add this to Blender later.
- Gimp (2.8.22 or higher)
  - Go to (https://www.gimp.org/downloads/) and click the Download GIMP directly button. After it is downloaded install it.
- Gimp DDS Plugin (Author: Shawn Kirst)
  - Go to (http://registry.gimp.org/node/70) and click gimp-dds-win32-2.0.2.zip button. After it is downloaded right click the zip file and select Extract All. Set destination to C:\Program Files\GIMP 2\lib\gimp\2.0\plug-ins and click Extract.

Optional Programs and Files That Could Be Helpful

- Notepad ++ (quickly view code with formatting outside of ModBuddy)
  - Go to (https://notepad-plus-plus.org/download) and click the Download button.
- IrfanView (quickly view DDS files outside of Gimp)
  - Go to (http://www.irfanview.com/) and under download the current version 64 bit click IrfanView. After it is downloaded install it.
  - Go to (http://www.irfanview.com/) and under download the current version 64 bit click Plugins/AddOns. After it is downloaded install it.

Section 1 — Import Base Game Asset In To Blender

This section is optional. If you are wanting to create a building from scratch just skip this section, however it would be a lot easier to get your building's positioning right if you imported a base game asset and deleted everything but the base and built your building up from that. This section will walk you through how to take the 3D art of a base game building and import it in to Blender for editing.

1) In Windows Explorer navigate to the following folder:
   - ...\Documents\Blender Projects
2) Create a new folder in the Blender Projects folder and name it CivNexus6.
3) In Windows Explorer navigate to the following folder:
   - C:\Program Files (x86)\Steam\steamapps\common\Sid Meier's Civilization VI SDK Assets\pantry\Geometries
4) Copy the FGX file (File Extension: .fgx) of the base game building you want to use and paste it in to the following folder:
   - ...\Documents\Blender Projects\CivNexus6
   - Note: To figure out what is what, you can preview the asset in Asset Editor in ModBuddy.
     - In ModBuddy click the TOOLS tab and select Launch Asset Editor.
     - In Asset Editor just click File and select Open Entity.
     - Click Asset in the list on the left box, then type a building name in the search bar at the top.
     - Click the asset you want to check from the list on the right box and click OK.
     - Once it loads you will see what the asset looks like in the box to the right called Asset Previewer.
     - Look under the Geometries tab for the file name. It is usually the first one listed. For example, the monument is DIS_CTY_AW_Pieces_DIS_CTY_AW_Monument, so in step 4 you would be looking for the file DIS_CTY_AW_Pieces_DIS_CTY_AW_Monument.fgx to copy and paste in to the folder you created in step 2.
5) Open the CivNexus6 application.
   - C:\Program Files (x86)\CivNexus6 _1.2.3\CivNexus6.exe
   - Note: The folder you have CivNexus6 in may have a different name because you may have a newer version.
6) Click Open, navigate to ...\Documents\Blender Projects\CivNexus6, select the FGX file (File Extension: .fgx), and click OK.
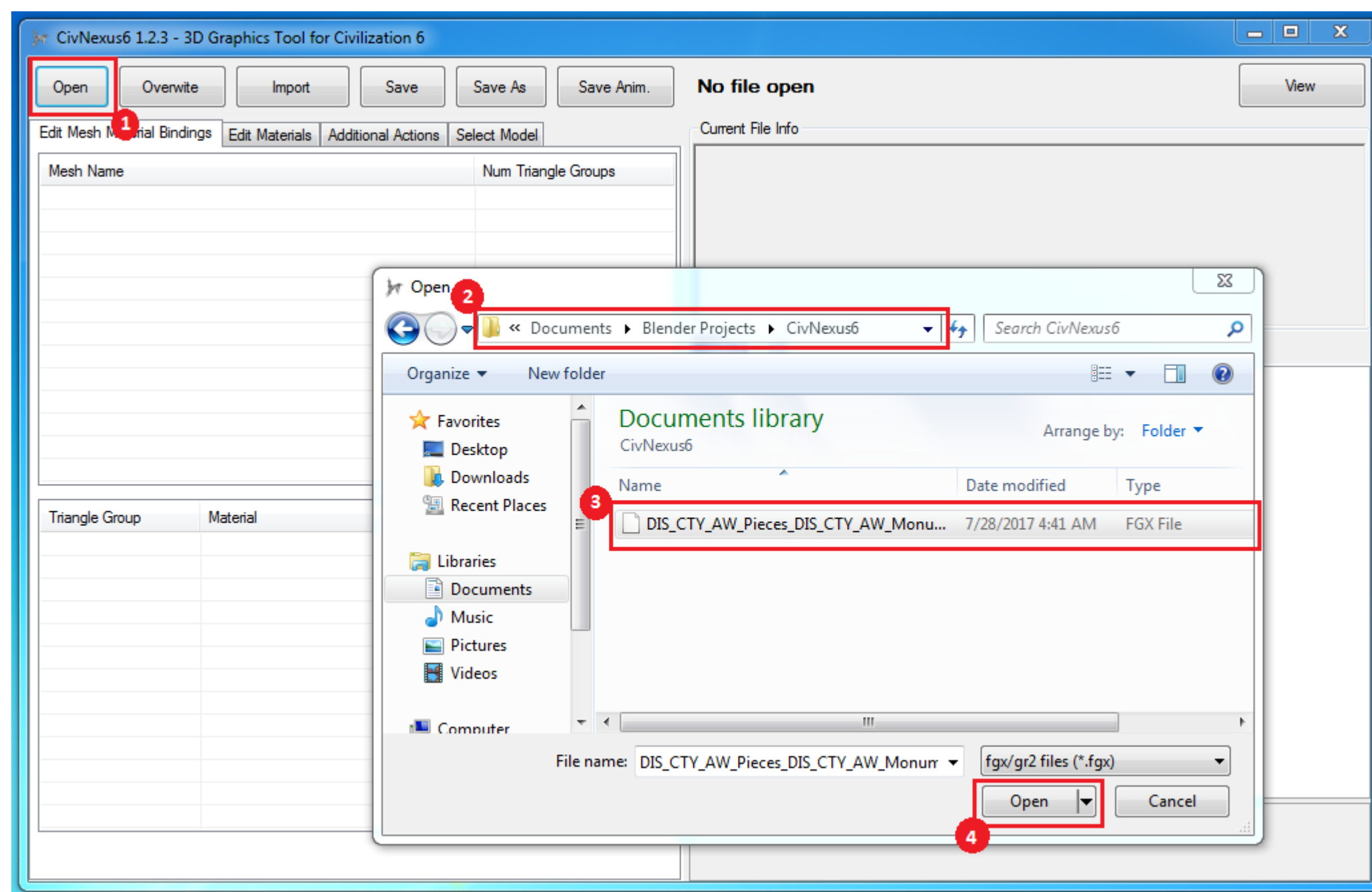

7) Click on the Additional Actions tab, then click the Export Models (All Models) button. The file has now been saved in CivNexus6 format (.cn6). You can exit out of CivNexus6 now.
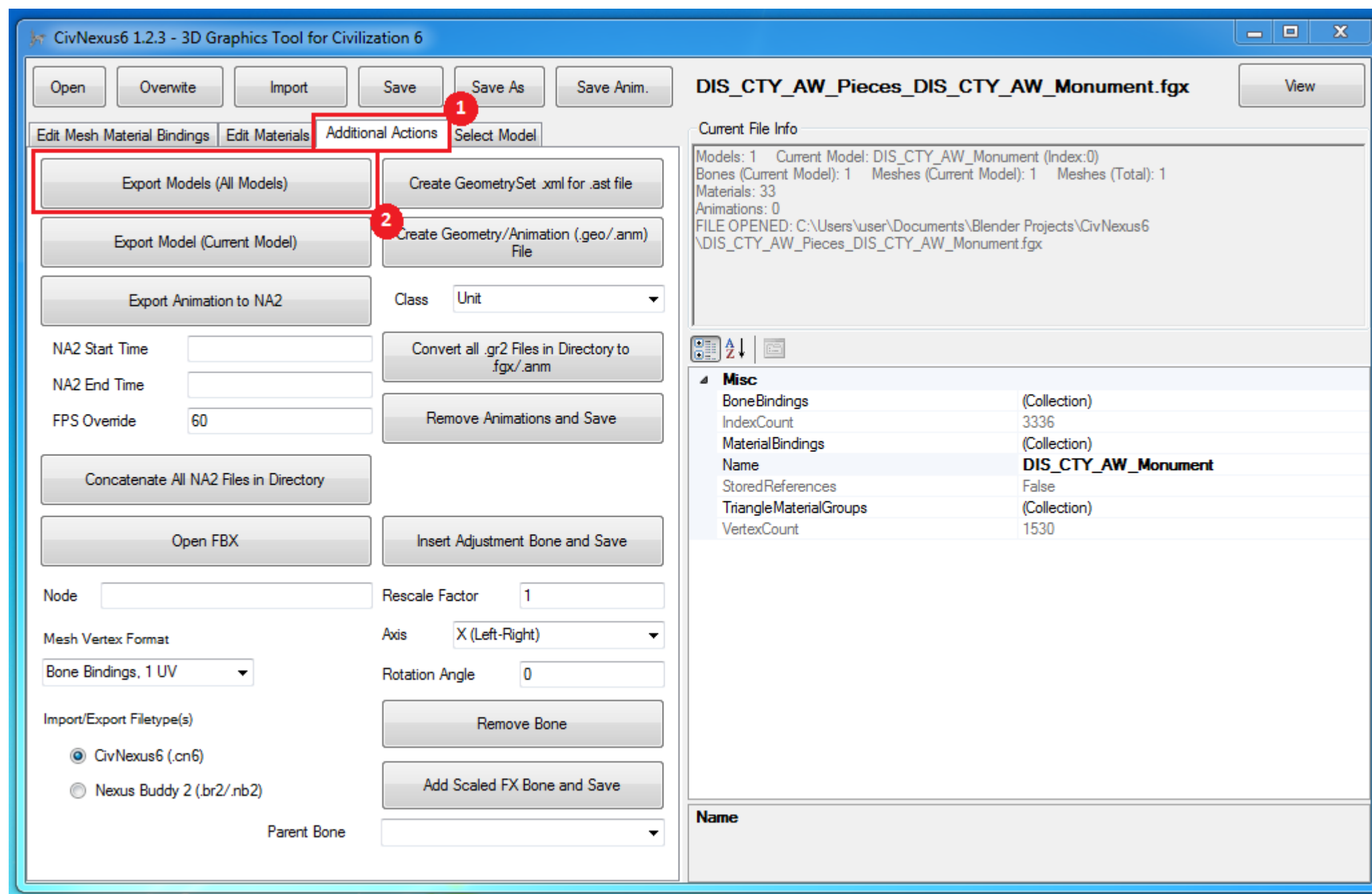
This section will walk you through how to start creating a 3D model, and how to set it up so that it is ready for texturing. Be aware that this is not a Blender tutorial. I recommend searching YouTube for a beginners Blender tutorial before starting this section. However, I will give a brief description on how to start creating a 3D model and list a few useful hot keys for those who prefer jumping right in and figuring things out on their own.

1) Open the Blender application.
   • C:\Program Files\Blender Foundation\Blender\Blender.exe
2) Click File, and select User Preferences, click the Add-ons tab, and click Install Add-on from File.
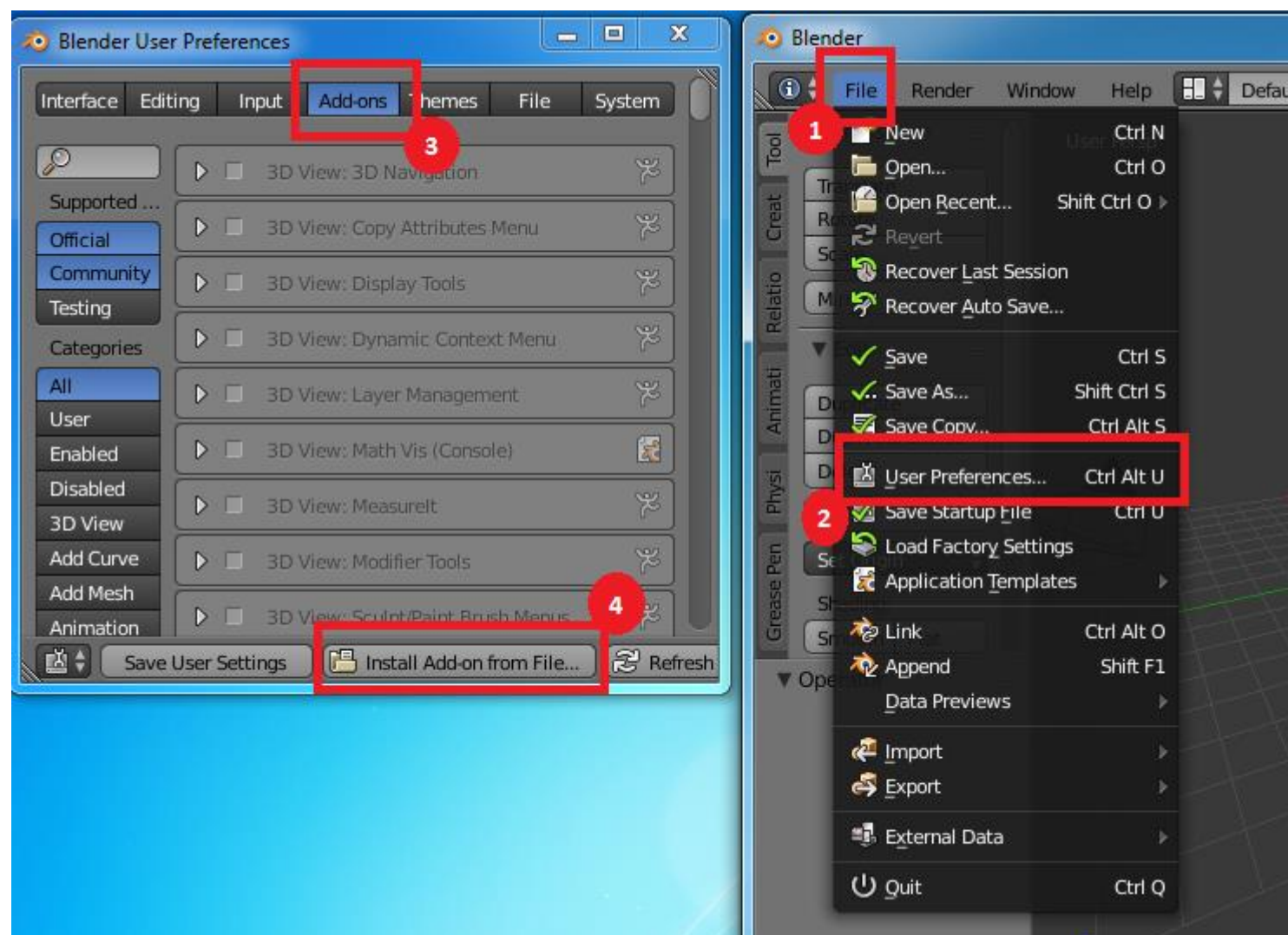
3) Navigate to ...\Downloads\Blender-Scripts-14.4\Blender-Scripts-14.4\Blender-2.7-Addons, or wherever you extracted Deliverator's Blender Scripts to, hold the shift key and click all 4 files in the Blender-2.7-Addons folder, with all 4 files selected, click the Install Add-on from File button. Under Categories, click User, and make sure the CivNexus6 scripts are all checked, then click the Save User Settings button. You are now able to import/export CivNexus6 files in Blender.
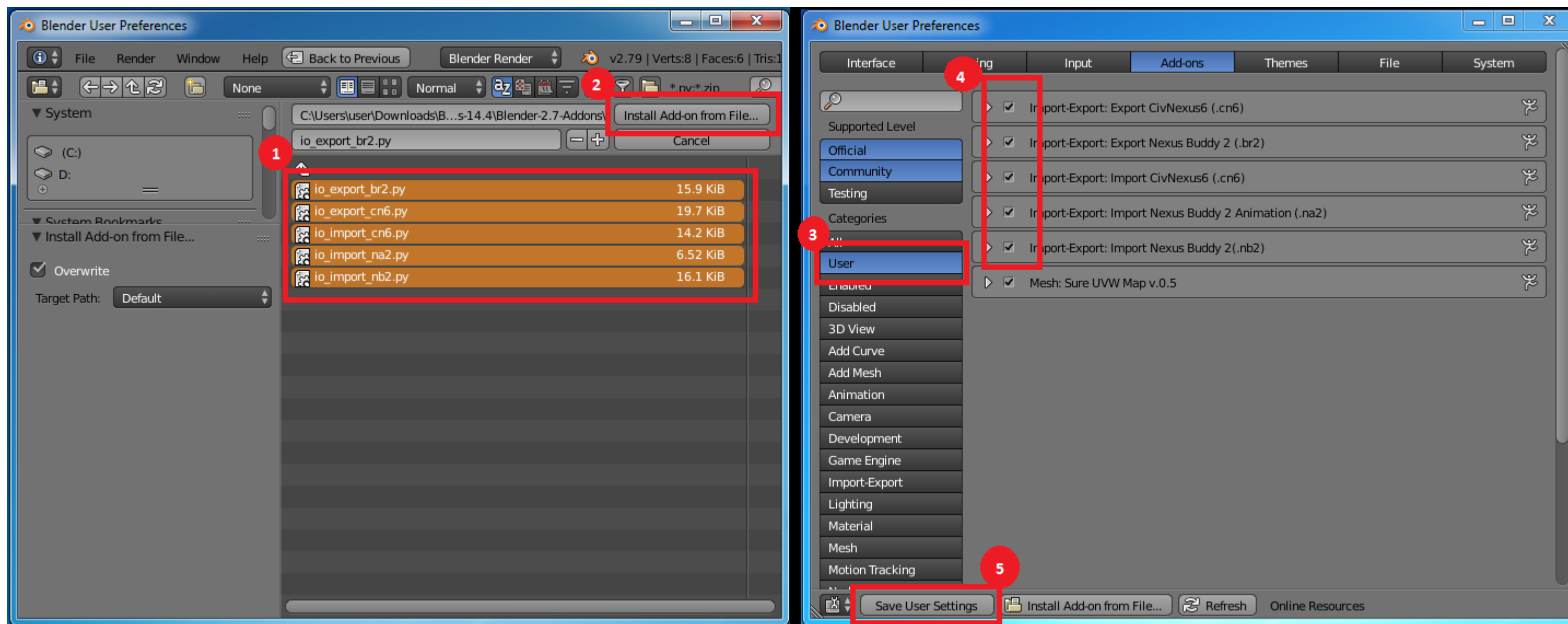
4) In the big center window (3D View) you should see 3 objects. A cube (mesh), a weird triangular thing (camera), and weird circle thing on a stick (lamp). Hold the shift key and right click on the weird triangular thing (camera) and weird circle thing on a stick (lamp). All 3 of the objects should be selected now because the cube is already selected by default. You can tell if an object is selected because it will be outlined in yellow or orange. If the cube is not outlined in orange then just hold shift and right click on it to include it in your selection. Press the delete key, and left click OK to confirm.
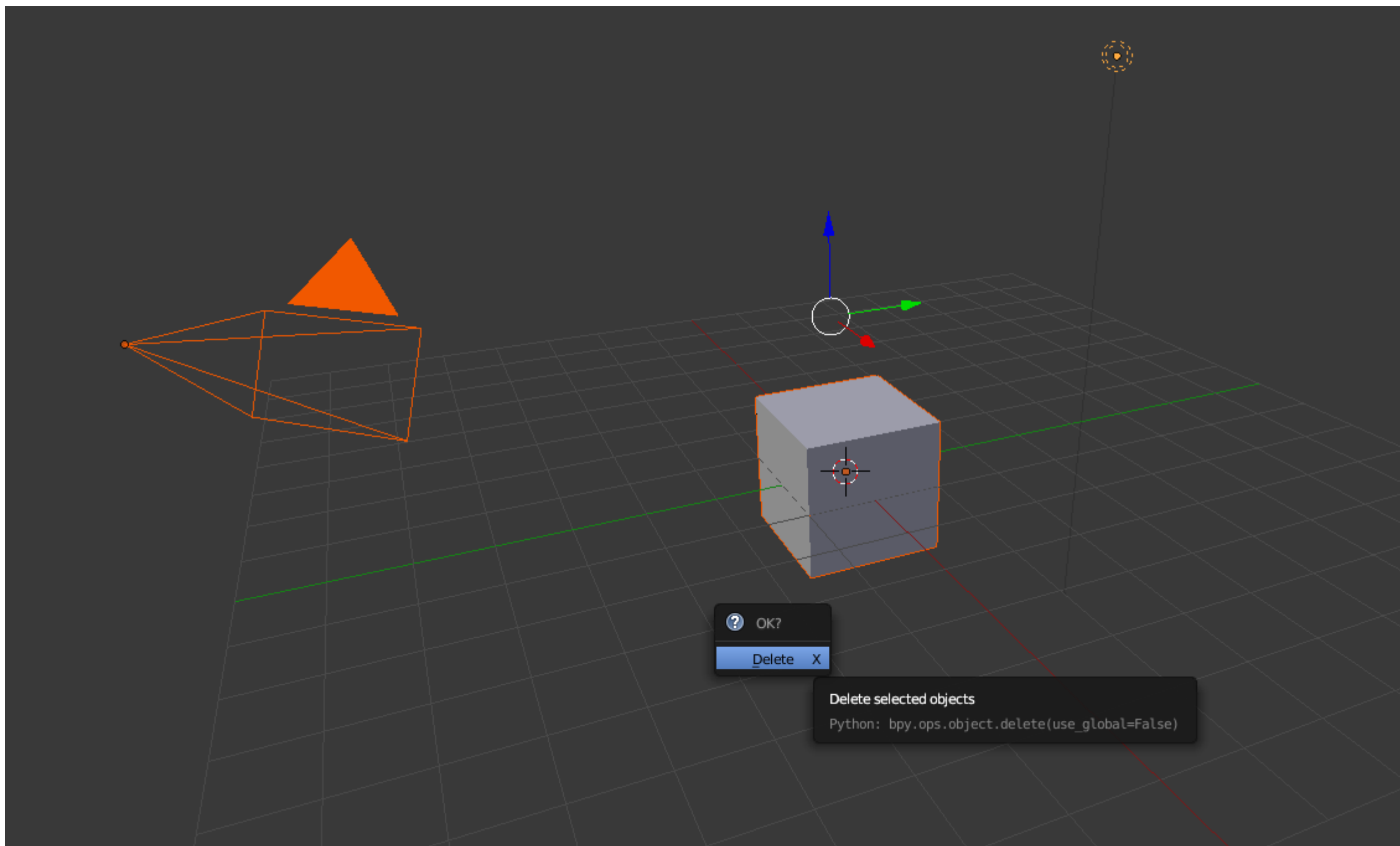

5) Now look at the panels on the right hand side of the screen. You see the horizontal row of small icons of a camera, pictures, a circle with a cylinder behind it, etc. This panel is called the Properties Menu. You want to click on the circle with a cylinder behind it one, it is called Scene. Now look below that and you should see a section called Units. You want to change the Length to Metric, just click on the black button that has None on it and select Metric. Then below that you want to change Unit Scale to 0.01, just click in the box with all the numbers in it once, type 0.01 and press the Enter key on your keyboard. Now move your mouse cursor so that it is in the big center window (3D View) that had those 3 objects you deleted earlier. Press the N key on your keyboard. You will see a new vertical panel pop up to your right, this panel is called the 3D View Properties Menu. Look under the section titled View, and look for Clip. Under Clip you will see Start and End with a numerical value to the right of each of these. You want to change the numerical value for End to 100m, just click once in the box with the numbers in it once, type 100m and press the Enter key on your keyboard.
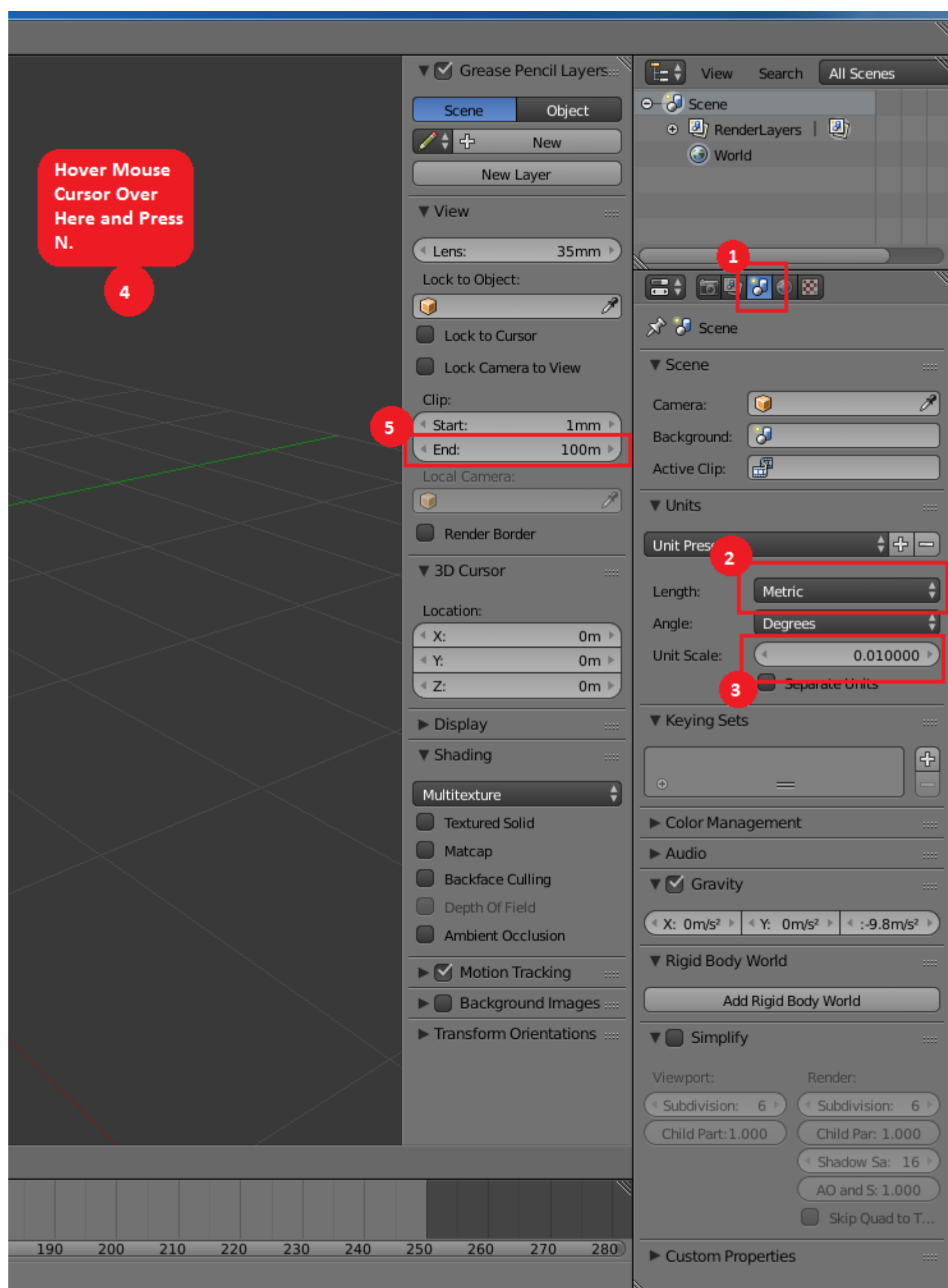
- Look at the very top bar, click the black button that says Blender Render, and select Cycles Render from the drop down menu.
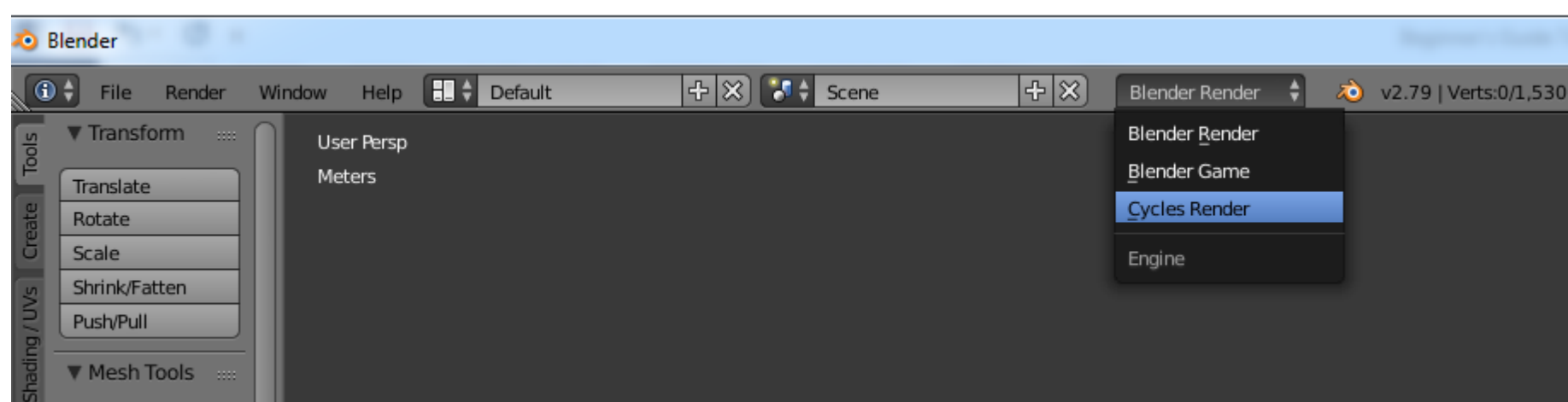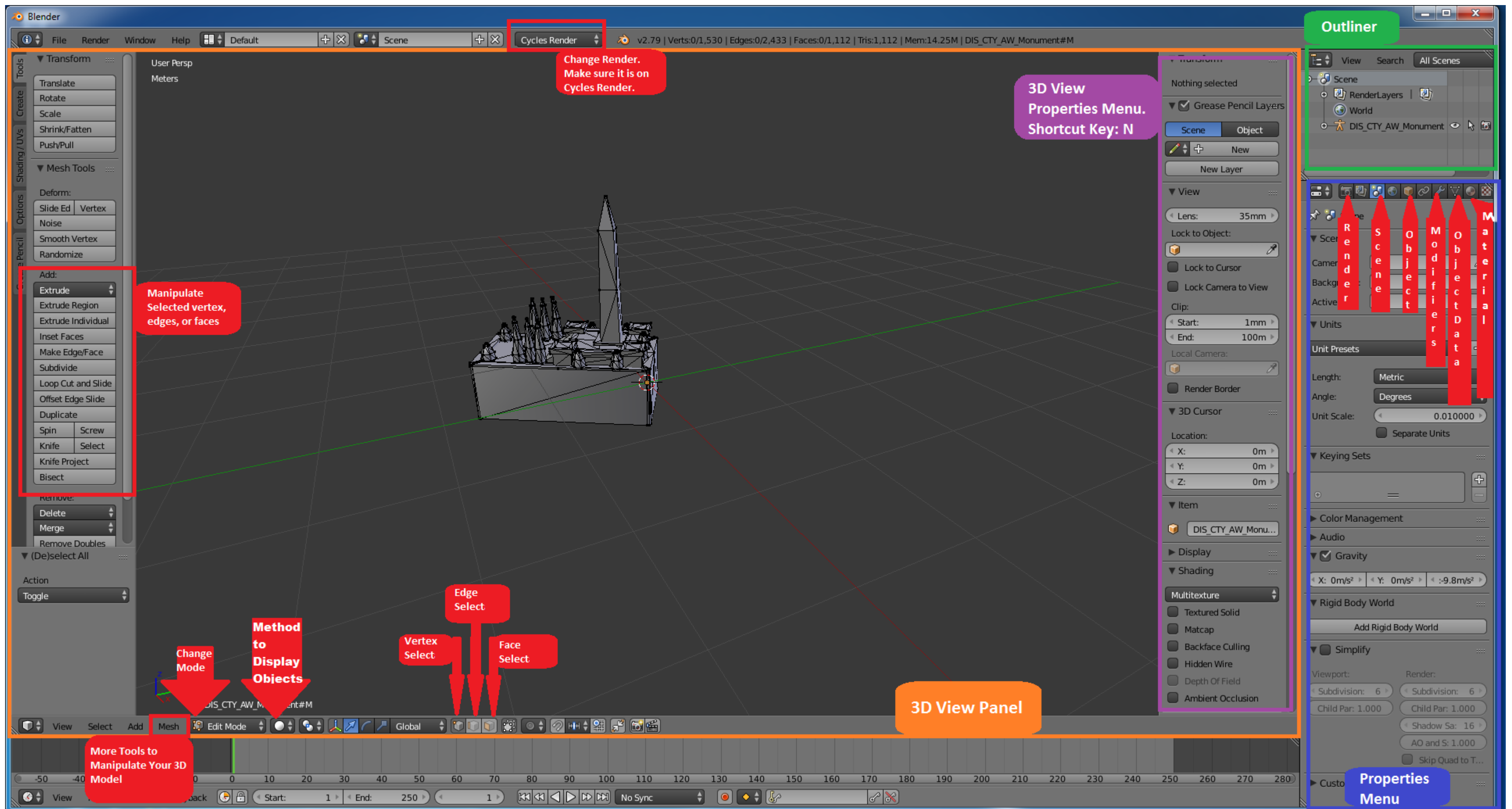

- Note: You are going to want to do what we did in steps 4 and 5 every time you start a new project. And this has to be the first thing you do before you start making stuff or it will cause issues with meshes made in Blender's default units. Setting the units resolves scaling issues when loading your 3D model in to the Civilization 6 game engine. And you only want your 3D model in your file, the camera and lamp is for making pictures and videos in Blender.

6) If you are creating a 3D model (Mesh) from scratch just skip this step. To load a base game 3D model (mesh) in to blender, click File, select Import, and click Civ Nexus6 (.cn6). Navigate to …\Documents\Blender Projects\CivNexus6 and select the CivNexus6 (.cn6) file you created in Section 1, click Import CN6 (.cn6).

- Go ahead and save your project, click File then select Save As. Note: What you name this will be the name of a number of your 3D art asset files so keep this in mind when you are coming up with a name. For example, I named one of my blender projects DIS_CTY_ESO_MagesGuild, DIS for district, CTY for city center, ESO for esosorcdc, and MagesGuild for the building's name.
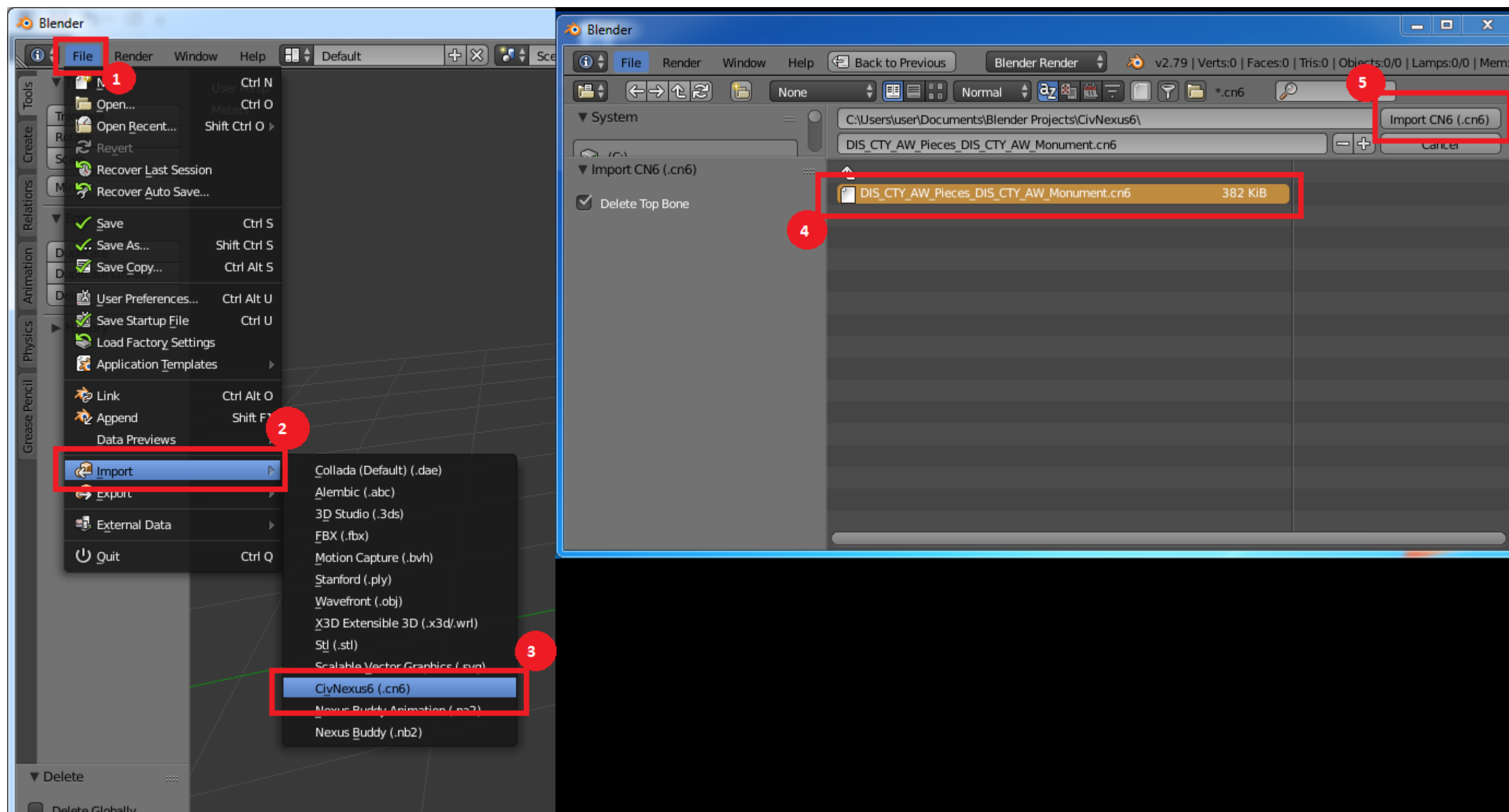
7) If you imported a base game 3D model (Mesh) just skip this step. To create your own 3D model (Mesh) from scratch, look at that menu bar right below the big center window (3D View), click Add hover over Mesh, and select Cube.

- Go ahead and save your project, click File then select Save As. Note: What you name this will be the name of a number of your 3D art asset files so keep this in mind when you are coming up with a name. For example, I named one of my blender projects DIS_CTY_ESO_MagesGuild, DIS for district, CTY for city center, ESO for esosorcdc, and MagesGuild for the building's name.
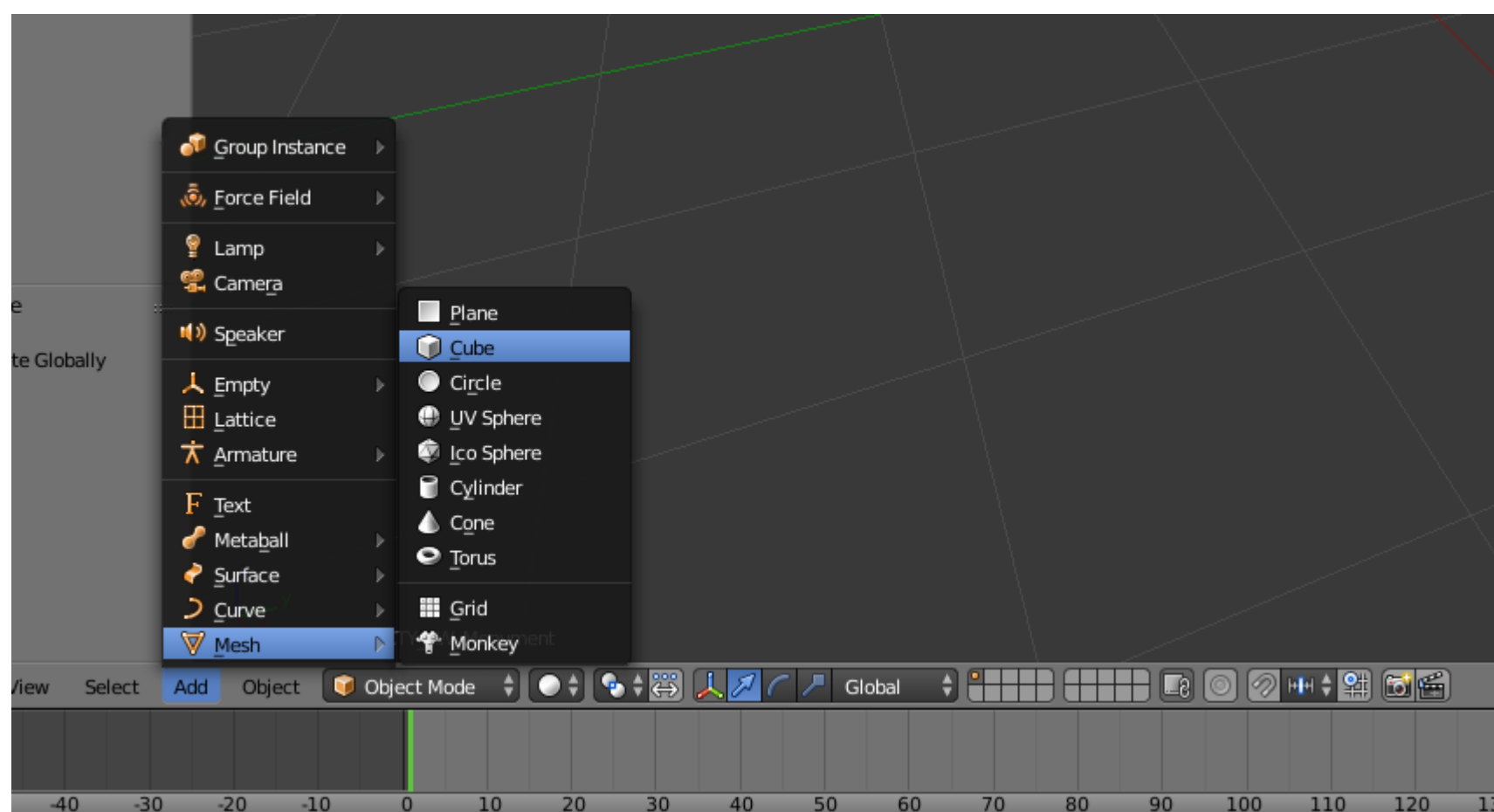

8) Now this is where you create your 3D model (Mesh), or edit the base game 3D model (Mesh) depending on which route you are going. The Mesh is the geometry of a 3D model and this is technically what you will be making in this step. I will now give you a brief description on how to do this to help you get started, but like I said earlier I recommend you watch some YouTube tutorials on this first. Tip: Look at the size of the base game monument building in the image below in relation to the grid below it to get an idea of how big to make your building, since you know how big the monument looks in game.

- To zoom in and out, use your mouse wheel.
- To rotate your view, click and hold the middle mouse button (your mouse wheel) and move your mouse around.
- To pan your view, hold the Shift key then click and hold the middle mouse button (your mouse wheel) and move your mouse around.
- To select and object, right click on it.
- To select multiple object, hold the Shift key and right click on each object.
- To box select, press B key and left click and drag. Holding down shift while doing this deselects.
- To circle select, press C key and hold down the left mouse button and drag. Press Enter or right click to exit circle select mode. Holding down shift while doing this deselects. Use mouse wheel to increase circle size.
- To deselect everything, press the A key while something is selected.
- To select everything, press the A key while nothing is selected.
- To undo action, hold ctrl and press the Z key.
- To actually start manipulating your 3D model (Mesh), select the object (right click on it), look at that menu bar right below the big center window (3D View), click the black button that says Object mode, and select Edit Mode.
- To move objects or selected parts, press the G key and drag mouse. Press Enter to complete, right click to cancel. This is called Grab. You can also type numeric values.
- To create a new mesh out of a selection, press the E key and drag mouse. Press Enter to complete, right click to cancel. This is called Extrude.
- To lock a move or extrude on an axis, after pressing G or E, press X, Y, or Z key to lock to that axis (shift + Z to lock on X & Y).
- To divide a selected face in to multiple equal faces, hold Ctrl and press the E key, then click Subdivide from drop down menu.
- To cut a face, press the K key, left click on one end of the face and then left click on the other end, then press enter.
- To rotate a selection, press the R key and drag mouse. Press Enter to complete, right click to cancel.
- To scale a selection, press the S key and drag mouse. Press Enter to complete, right click to cancel.
- The Outliner is another way to select objects in Object Mode. It shows you all the objects you have in your scene and if objects are parented to each other.
- Changing the view of your 3D model (Mesh) is very useful. The Wireframe view lets you see through your mesh to select hard to reach vertices. Solid is the default.
- And that is some of the basics of Blender. You would use a combination of all the actions listed above to create your building. From now on I will assume you at least know all the actions and terms I talked about in all the steps above.
- Below is an image showing you some basic parts in Blender that you need to know.

Outliner

3D View Properties Menu. Shortcut Key: N

Change Render. Make sure it is on Cycles Render.

Manipulate Selected vertex, edges, or faces

Change Mode

Method to Display Objects

Vertex Select

Edge Select

Face Select

3D View Panel

More Tools to Manipulate Your 3D Model

Properties Menu

Render | Scene | Object | Modifiers | ObjectData | Material

- Note: You will be making a hollow building, do not make rooms on the inside or anything, and do not leave holes for the windows or anything, it needs to be one complete object minus the bottom, you do not need to put faces under it, and do not make it to detailed, the less vertices you need to achieve what you are going for the better.
- Note: The position of your 3D Model in Blender will be the position of it in game, with position x = 0 and y = 0 being the center of a tile and z = 0 being ground level of a tile. Load base game buildings in to Blender to get an idea of position since you know where base game buildings show up in game. You can also look at the image from section 6 step 5 to get an idea of where to position your 3D Model in Blender.

9) Now that you have your 3D model (Mesh), you need to Triangulate it. In other words, it has to be made up of only triangles. In Object Mode select your 3D model (Mesh), then in the Properties Menu click the Modifiers button. Click the black button that says Add Modifier, under the Generate column select Triangulate, and then click the Apply button. You won't see any difference in Object Mode, but once you go in to Edit Mode you will now see that your mesh has been triangulated.
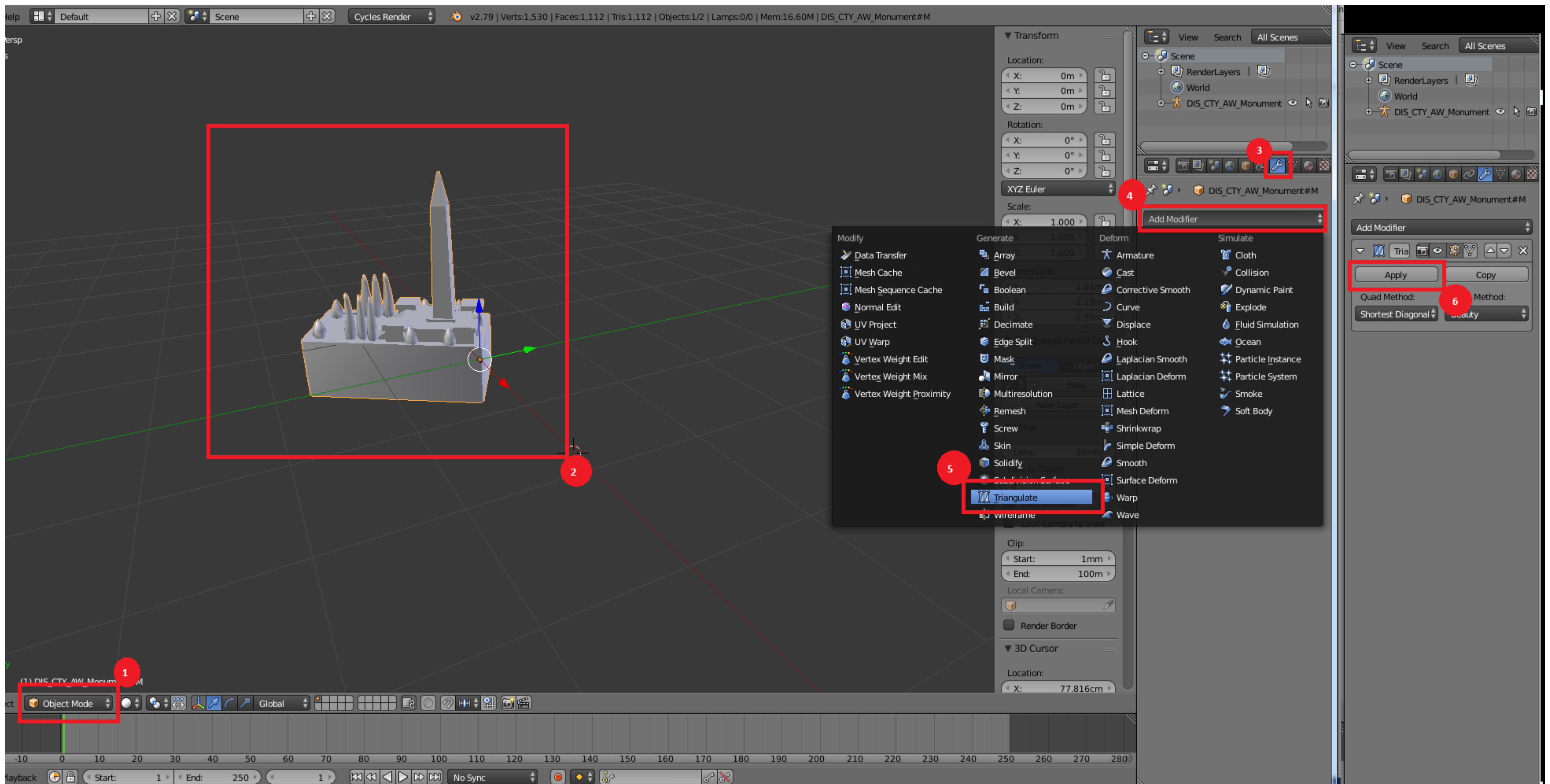
10) In Edit Mode with Edge Select active, visualize if you were going to lay your 3D model (Mesh) out on a flat 2D plane, what edges would you need to cut so that it could lay down completely flat. Now go around your mesh and select all the edges that would need to be cut, then hold Ctrl and press the E key and click Mark Seam. Mark seams in a complete edge loop to separate parts of your 3D model (Mesh) if needed.
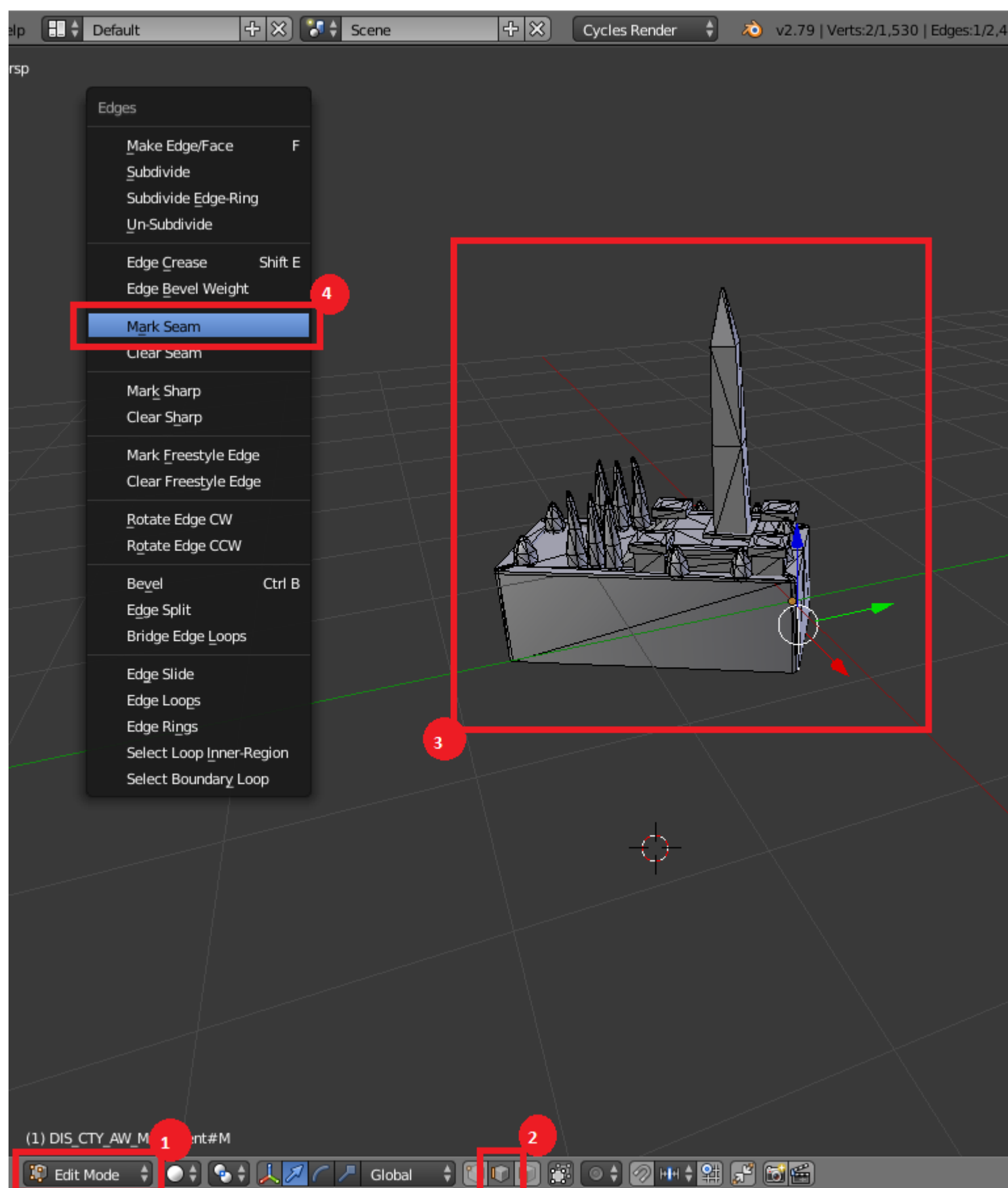

- Note: This step is technically not required. It allows you to tell Blender how to unwrap your mesh, and fixes UV islands that have overlapping parts, as well as island proportion issues so that you get a nice clean unwrap.

11) If you are using an imported base game 3D model (Mesh) you can skip this step since you already have a bone. In Object Mode, click on Add, then hover over Armature, then select Single Bone.
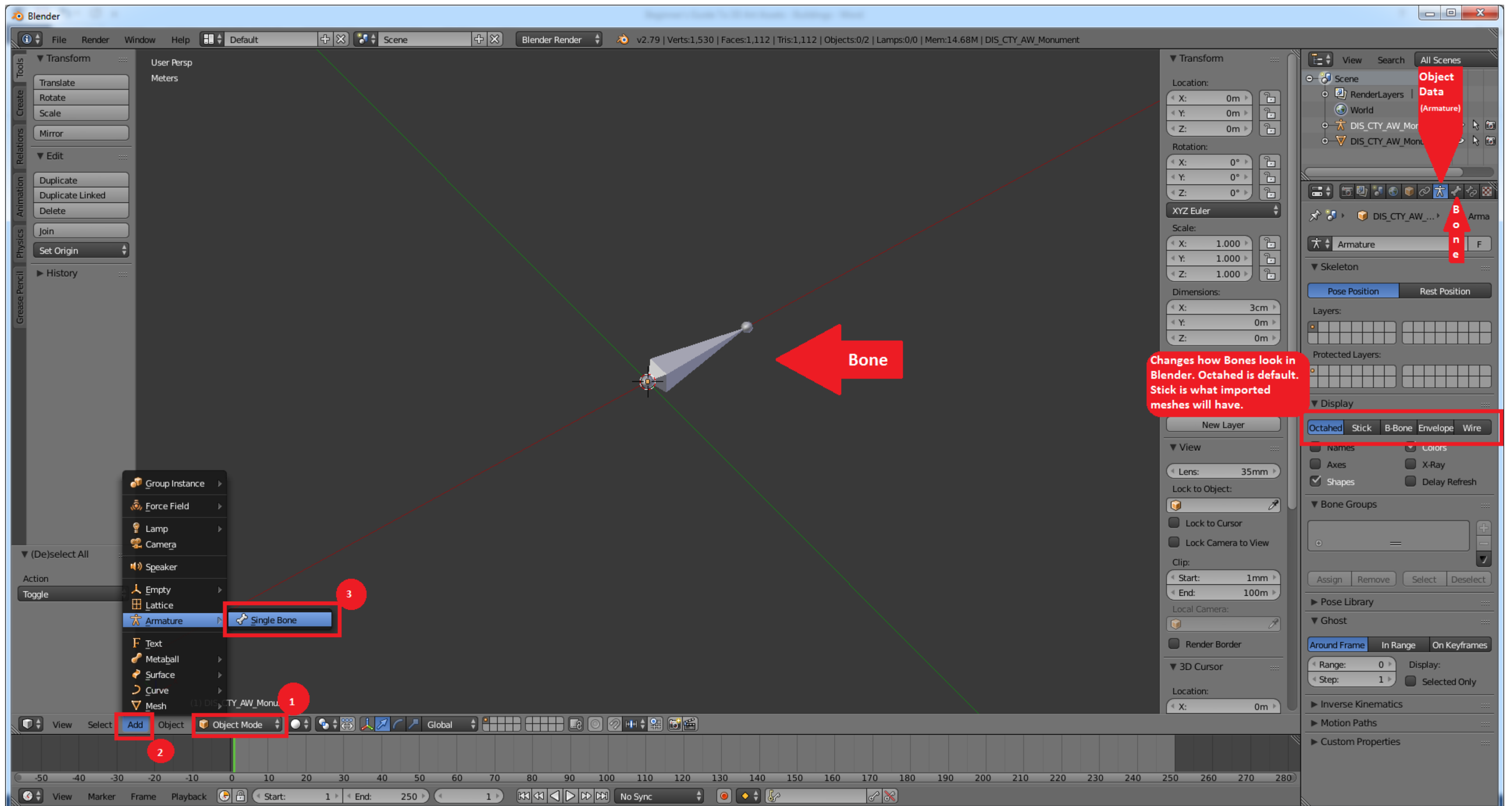

- Note: You can select bones and go in to Edit Mode to move, scale, and rotate it just like you do with your mesh. If you want to place your Bone inside your mesh you can, but it does not matter where the Bone is for buildings. Tip: Changing the Method to Display Objects to Wireframe makes it easier to place a Bone inside of a mesh.

12) If you created a 3D model (Mesh) from scratch just skip this step. In Object Mode, select both your 3D model (Mesh) and the Bone, so your 3D model (Mesh) will be outlined in orange and the Bone in yellow. Hold down Alt and press the P key, then select Clear Parent. It may be hard to find the bone, as you can see in the picture below it shows up as a little stick. Left clicking the Bone in the Outliner Panel will select it and make it easier to find in your 3D View Panel.
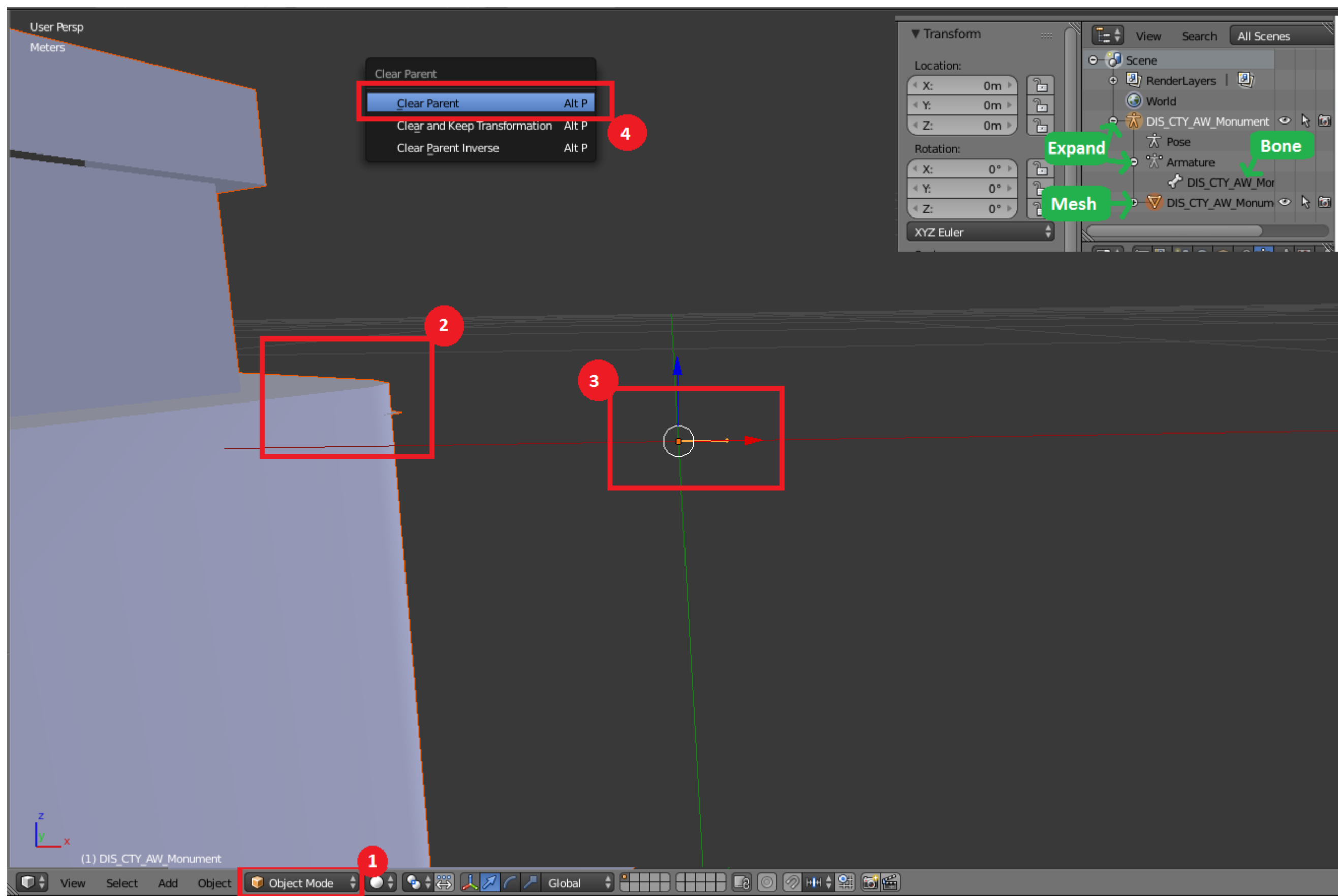
- Note: The reason why we clear the parenting is to resolve an issue with exporting to CivNexus6 format. You will re-parent your 3D Model (Mesh) to the Bone in the next step.

13) In Object Mode, select your 3D Model (Mesh) first, then select your bone second, so now both objects are selected with your 3D model (Mesh) outlined in orange and your Bone in yellow (very important). Now hold Ctrl and press the P key, then select With Automatic Weights under Armature Deform. You will notice in your Outliner that your 3D Model (Mesh) is now included under the Armature tree. You will also notice if you move the Bone in Object Mode, your 3D Model (Mesh) will move with it. Tip: Moving your Bone in Edit Mode does not move your 3D Model (Mesh) with it. In the Outliner panel double left click the text of your Armature object, this is the one at the very top of the Armature tree, type in Root and press the Enter key, the Armature object is now renamed to Root. We rename the Armature object because later we will use this name in ModBuddy and it will make identifying the Armature object easier.
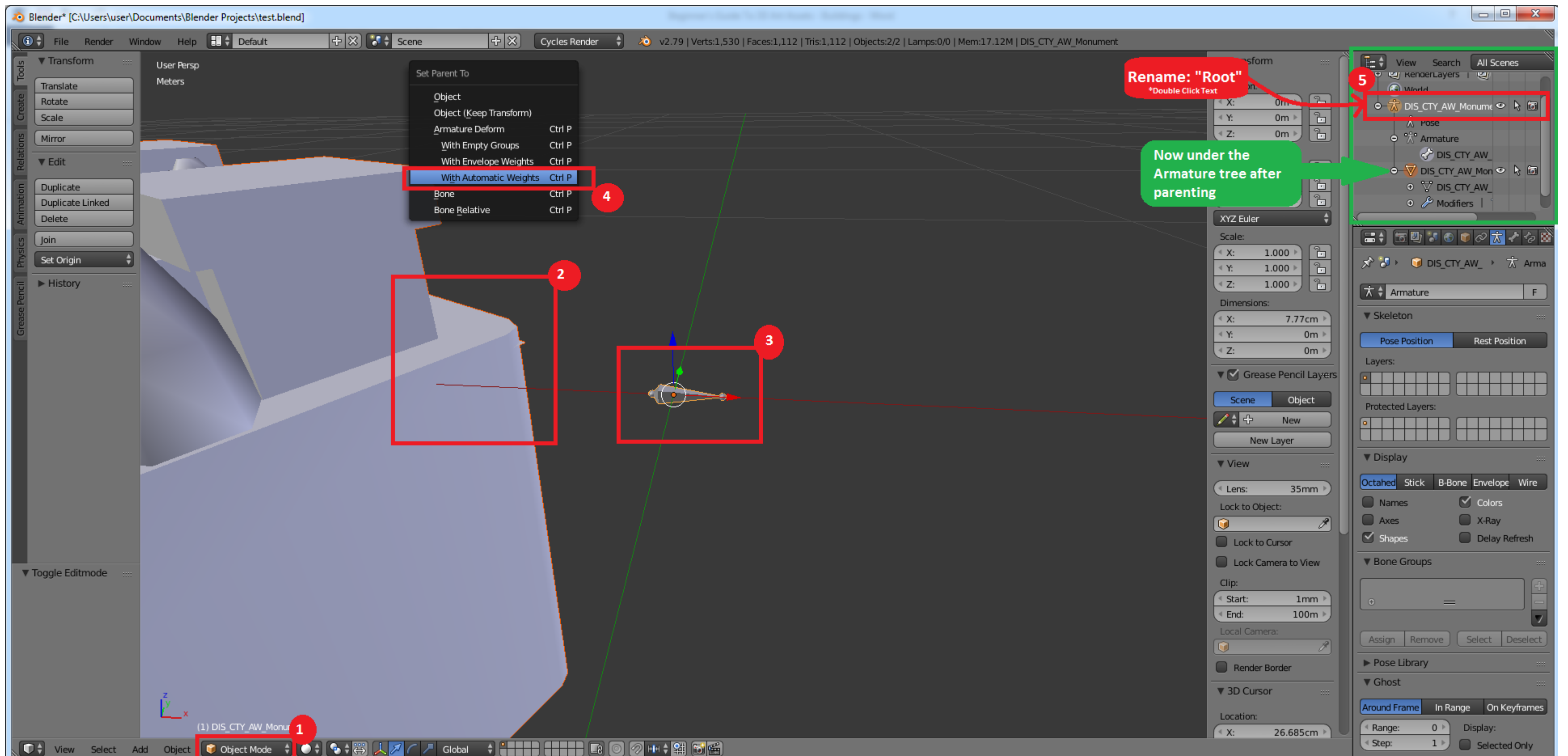

- If you did not put your bone inside of your 3D Model (Mesh) you will see a warning message on the top menu bar, ignore it.
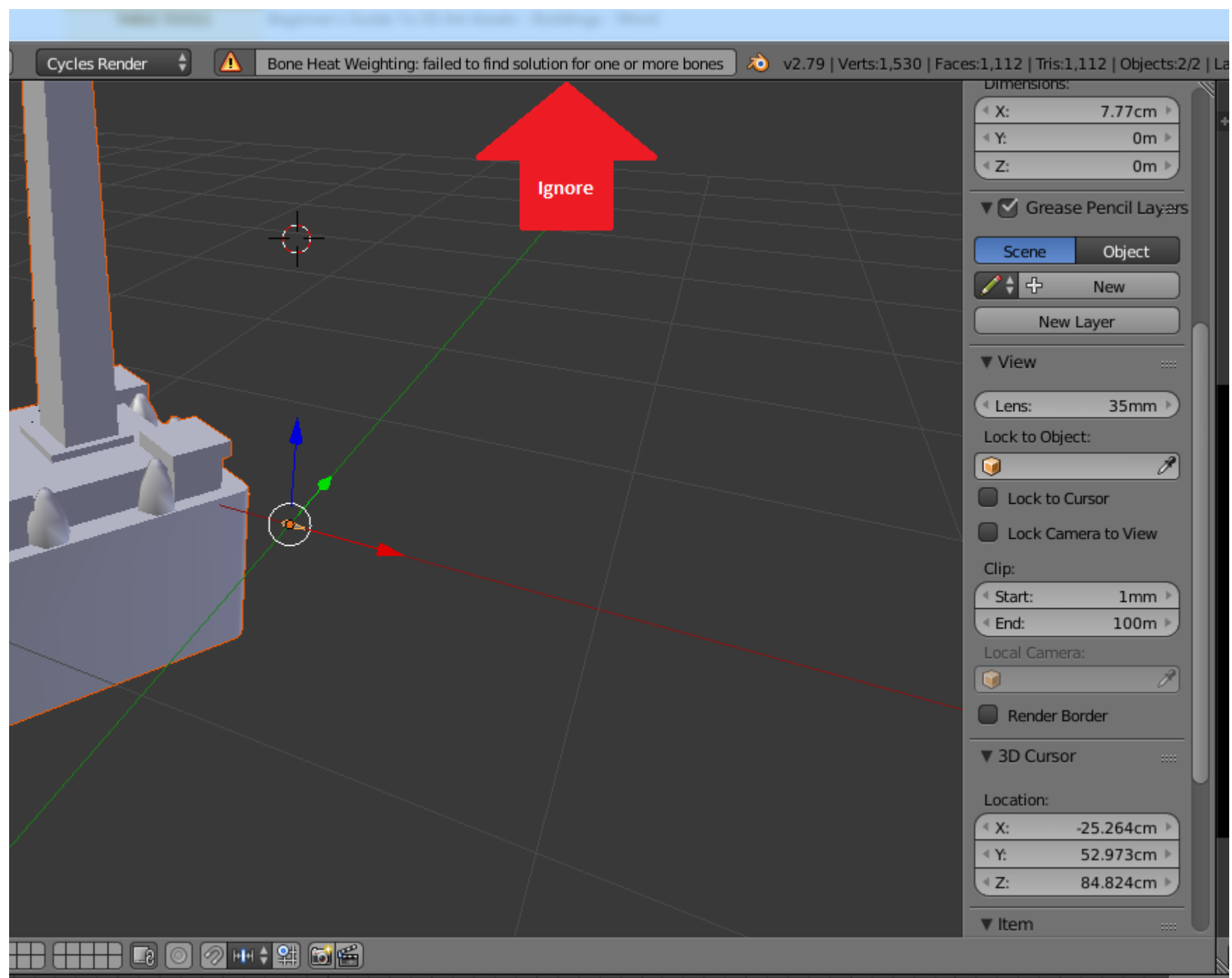
- Note: Put very simply, a Bone is what moves a 3D Model (Mesh), and an animation is what tells the Bones to move. Every object needs a Bone whether it moves or not, so even though we are making a building it still needs at least one bone.

14) In Edit Mode for your 3D Model (Mesh). Press A key to deselect your 3D Model (Mesh), then press the A key again to select your entire 3D Model (Mesh). This ensures that everything is selected. Under the Properties Menu click Object Data. Under Vertex groups, you will see either a vertex group called Bone (if you created a new 3D Model (Mesh)), or a vertex group called VERTEX_KEYS and another one with the name of the base game building you imported (If you imported a base game 3D Model (Mesh)). Make sure the Bone vertex group, or the vertex group with the name of the base game building you imported, is selected (will be highlighted in blue), and click the Assign button. We do this to make sure every vertex is assigned to this vertex group. If you have unassigned vertices you will have issues with exporting from Blender or Importing to CivNexus6.

Created own 3D Model (Mesh)

Imported Base Game 3D Model (Mesh)
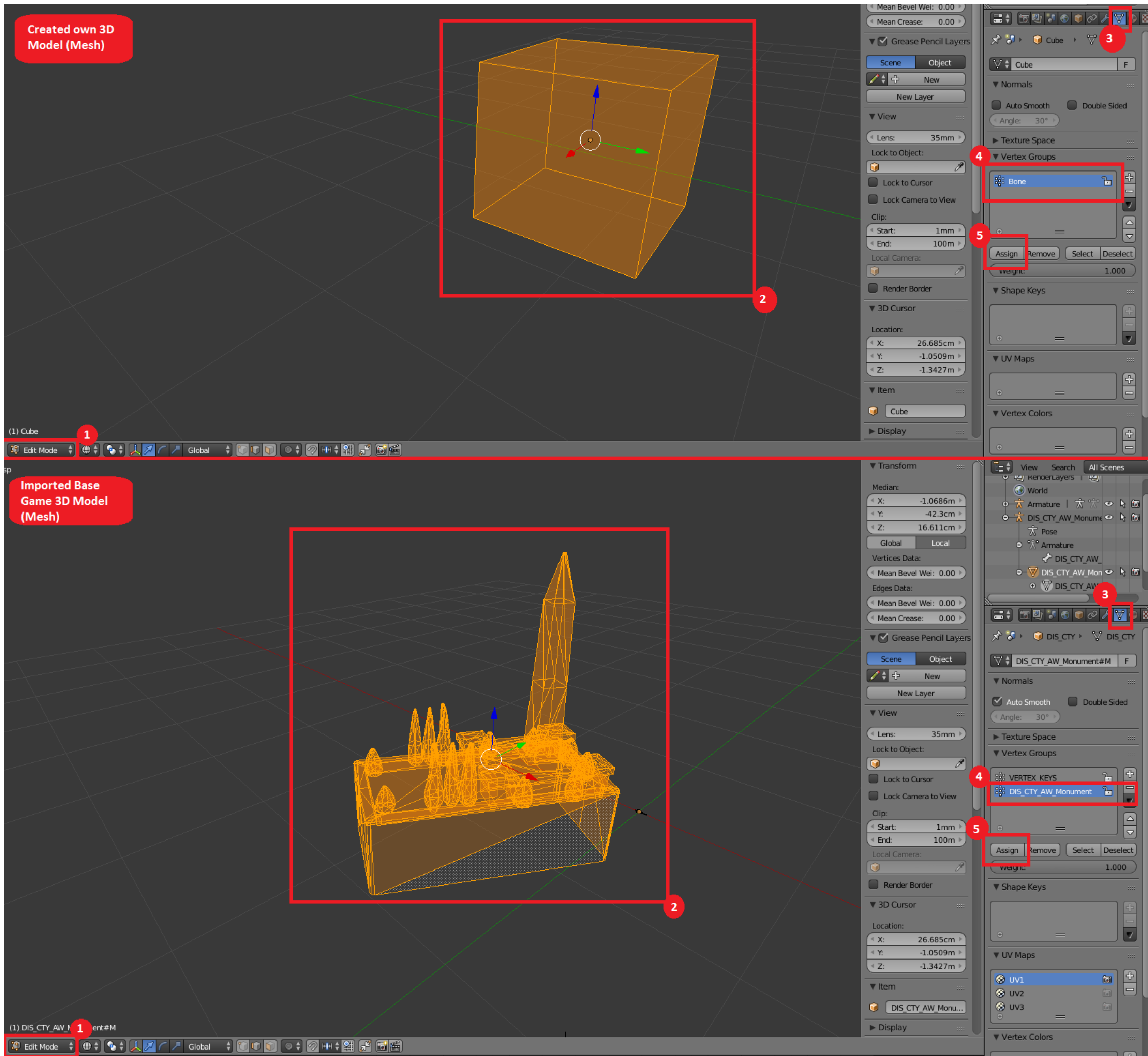
15) In Object Mode select your 3D Model (Mesh). Under the Properties Menu click Material. You will see a box that contains an entry or a number of entries of a grey circle followed by the text. This is called your Material. Click a Material to select (it will be highlighted in blue when selected), and press the button to the right with the minus sign. This will delete it. If you have multiple Materials, delete the rest of them. Now the box should be empty.
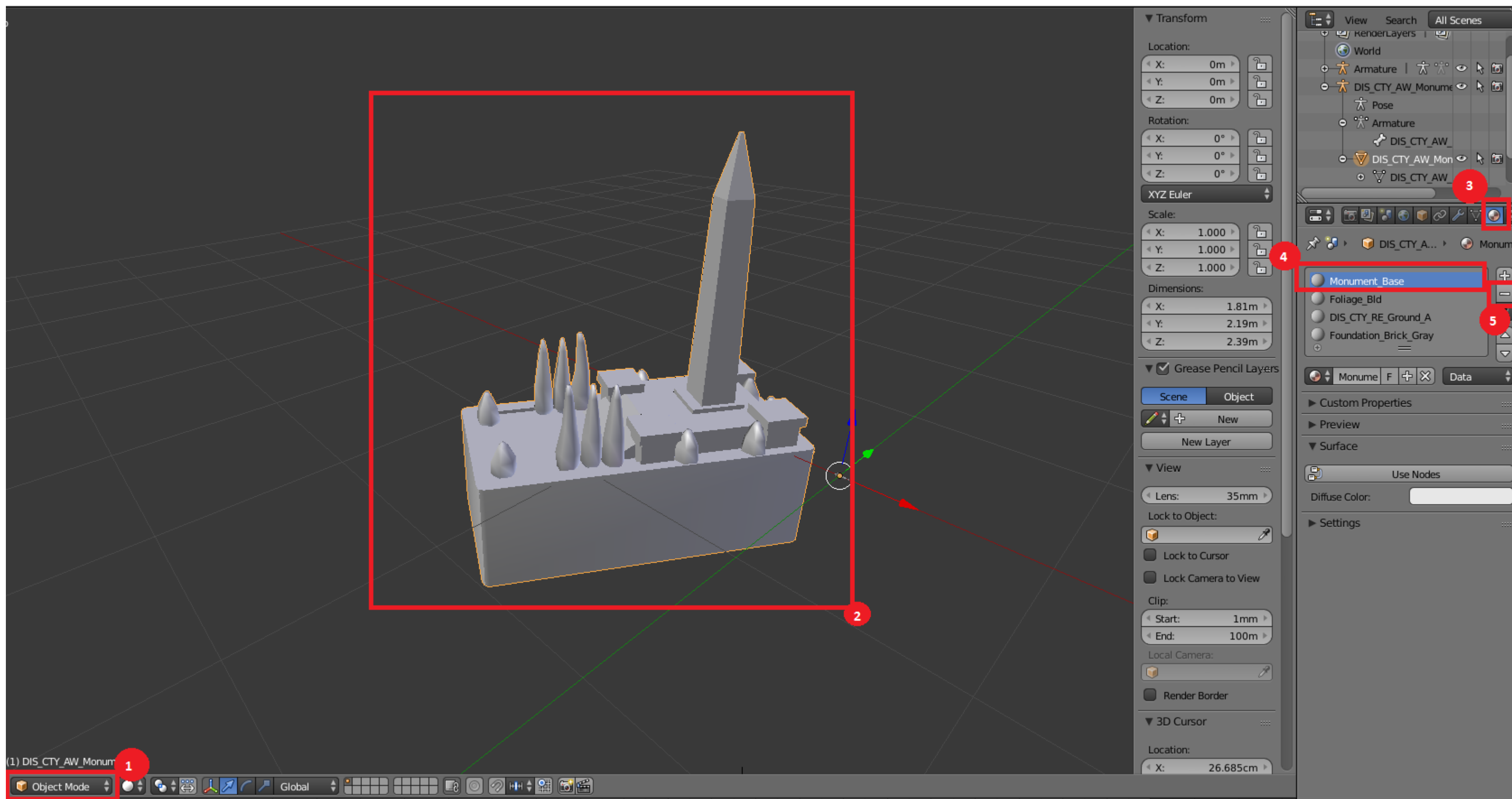
16) In Edit Mode for your 3D Model (Mesh), click Face Select, then select your entire 3D Model (Mesh). Under the Properties Menu click Material (if not already brought up). Click the button with the plus sign. This adds a new material slot and you will see a new button appear below that says New. Now make sure the new material slot is selected and click the New button. This adds a new material to the new material slot. Now with the new material selected, click the Assign button. Because your entire 3D Model (Mesh) was selected, every face of your 3D Model (Mesh) is now assigned to this Material. You can name your material by double left clicking on it, typing in a name, and pressing enter. It does not matter what you name it right now.
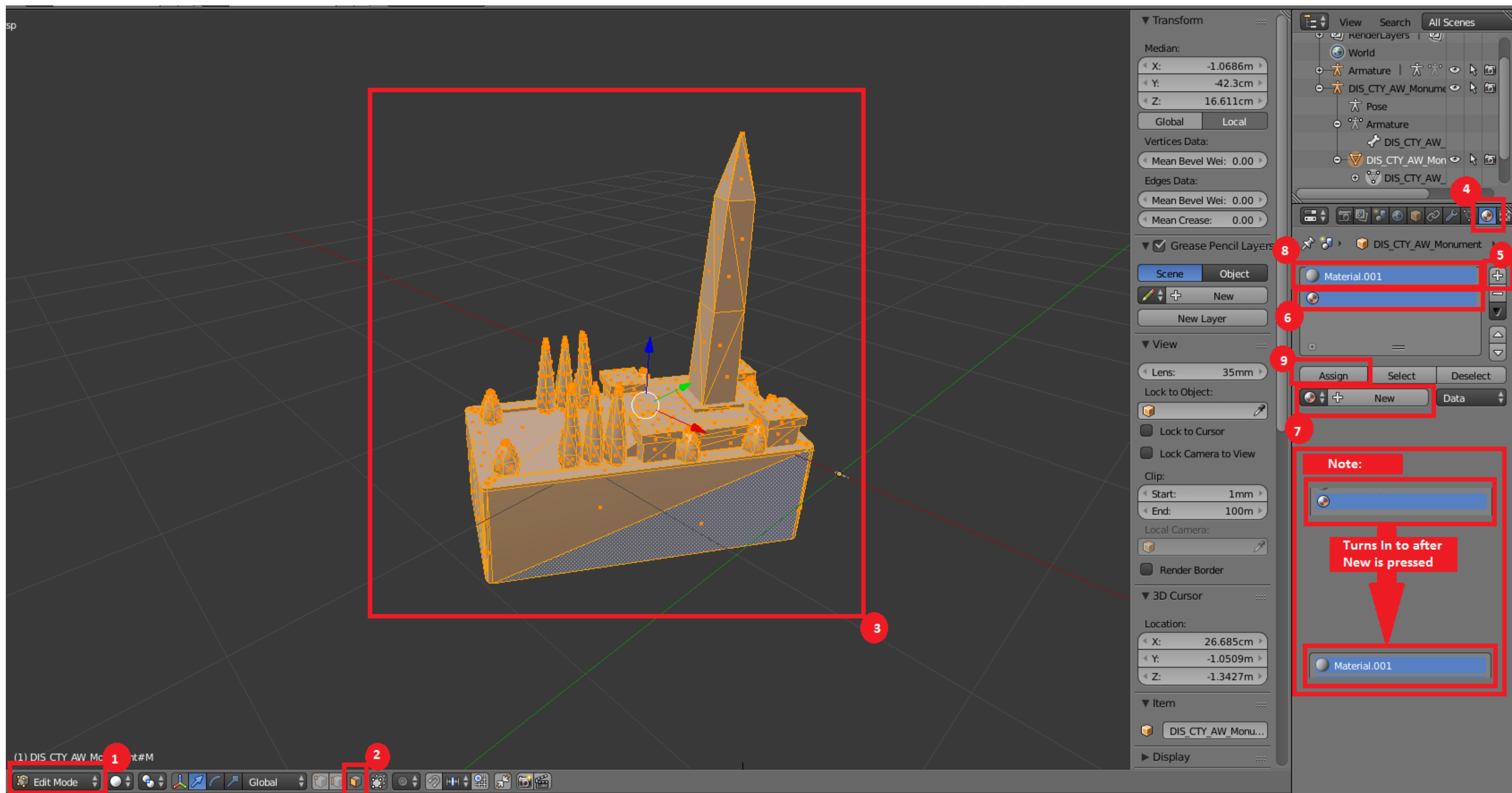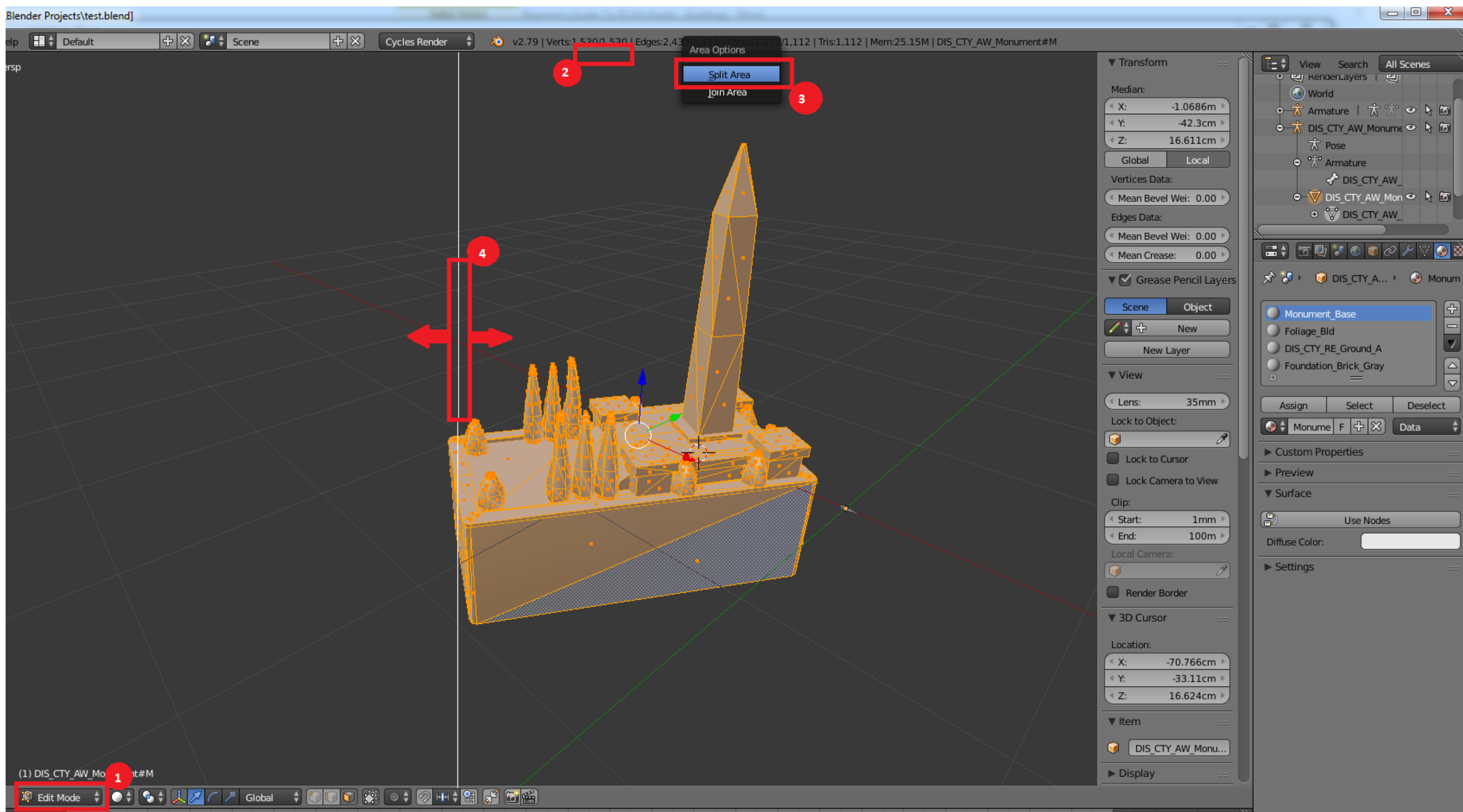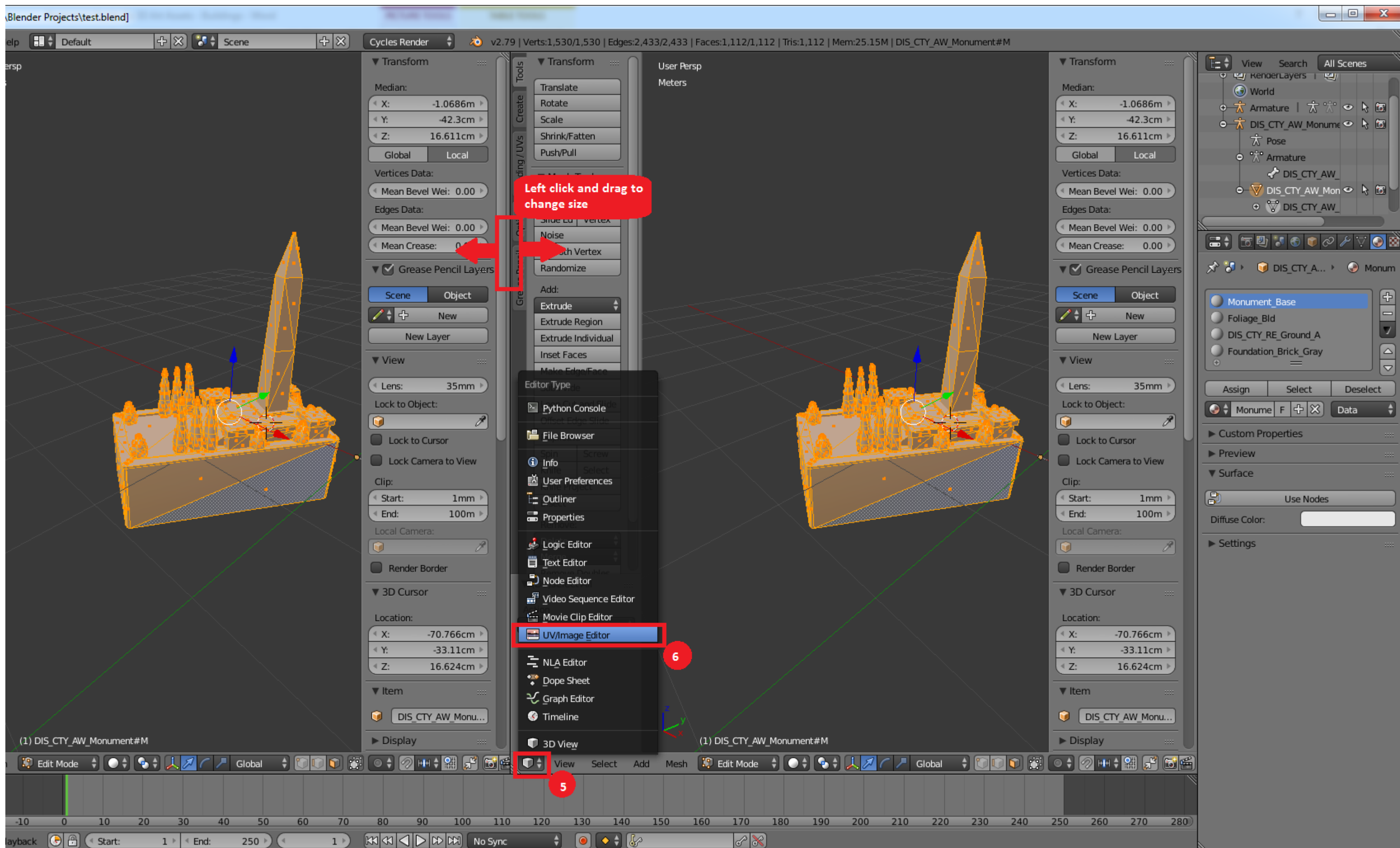

17) In Edit Mode for your 3D Model (Mesh), click Face Select. Now that you have one material created and assigned, you will want to do this with every different type of material that your 3D Model (Mesh) will have. The first material you created and assigned to your whole 3D Model (Mesh) usually will be for the material that consists of the harder to select faces so you won't have to go around and select all of these. For example, your building's trim around edges and window frames. As you assign parts to other materials, it removes it from the first material you created. This is why we just assigned the whole 3D Model (Mesh) to the first Material we created. So now go around your 3D Model (Mesh) and select faces that are all one type of material. For example, all the faces that make up the roof would be one. Do the same thing you did in step 16 and create a new material in the Properties Menu under Material, then with your new material selected, click Assign. This will assign all of the faces you currently have selected to this material. Repeat these steps for every material you want to have in your 3D Model (Mesh). For example, a material for the trim, for the roof, for the walls, for the windows, for the doors, etc.
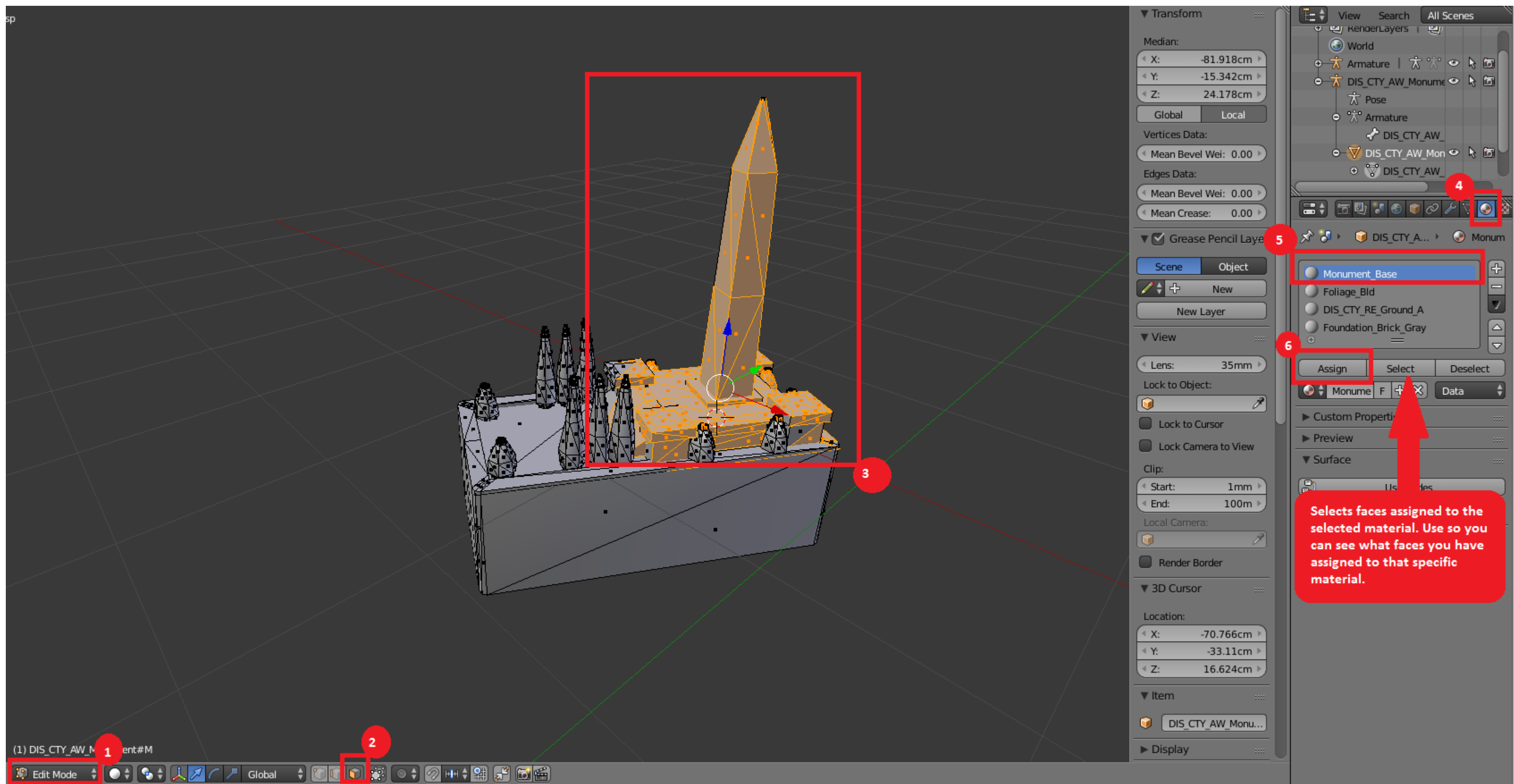
Selects faces assigned to the selected material. Use so you can see what faces you have assigned to that specific material.

- Note: In the Properties Menu under Material, with a material entry selected, use the Select button to select all faces assigned to that material. This is helpful if you want to check what faces are assigned to what material.

18) In Edit Mode for your 3D Model (Mesh), hover your mouse at the top of your 3D View Panel between where it ends and the top menu bar begins, you will see it turn into an up/down double arrow, right click and select Split Area. You will see a white line pop up running down the 3D View Panel, moving your mouse moves this line, move the line to about the center of the 3D View Panel and left click. This will split the 3D View panel in half with the 3D View panel on both sides. Now on the right side one look to the bottom left of the panel and you will see a little button with a white cube in it, click this button and select UV/Image Editor from the list. This will turn the right panel in to the UV/Image Editor panel. So you should see the 3D View panel on the left and the UV/Image Editor panel on the right now. To adjust the size between these two panels, hover your mouse between them two, you will see it turn into a left/right double arrow, left click and hold, drag to desired size, then release the left mouse button.

19) In Edit Mode for your 3D Model (Mesh), in the 3D View panel, select your entire 3D Model (Mesh), press the U key and select Unwrap. This creates a UV map of your 3D Model (Mesh), which is your 3D model (Mesh) laid out on a flat 2D plane. This action is called UV Unwrapping. In the UV/Image Editor Panel, on the bottom menu bar, click the Keep UV and edit mode mesh selection in sync button, since you should still have your entire 3D Model (Mesh) selected in the 3D View panel the entire UV map will now become selected as well. With your mouse cursor in the UV/Image Editor panel, hold Ctrl and press the P key. This action is called Pack Islands, it reorganizes your UV Map and puts spaces between islands, which is what you want when baking. An island, in regards to your UV map, is a connected mesh group. If you imported a base game 3D Model (Mesh) you will see a UV map of the original building already in the UV/Image Editor Panel, just ignore this, every time you do a new unwrap it replaces the old one.
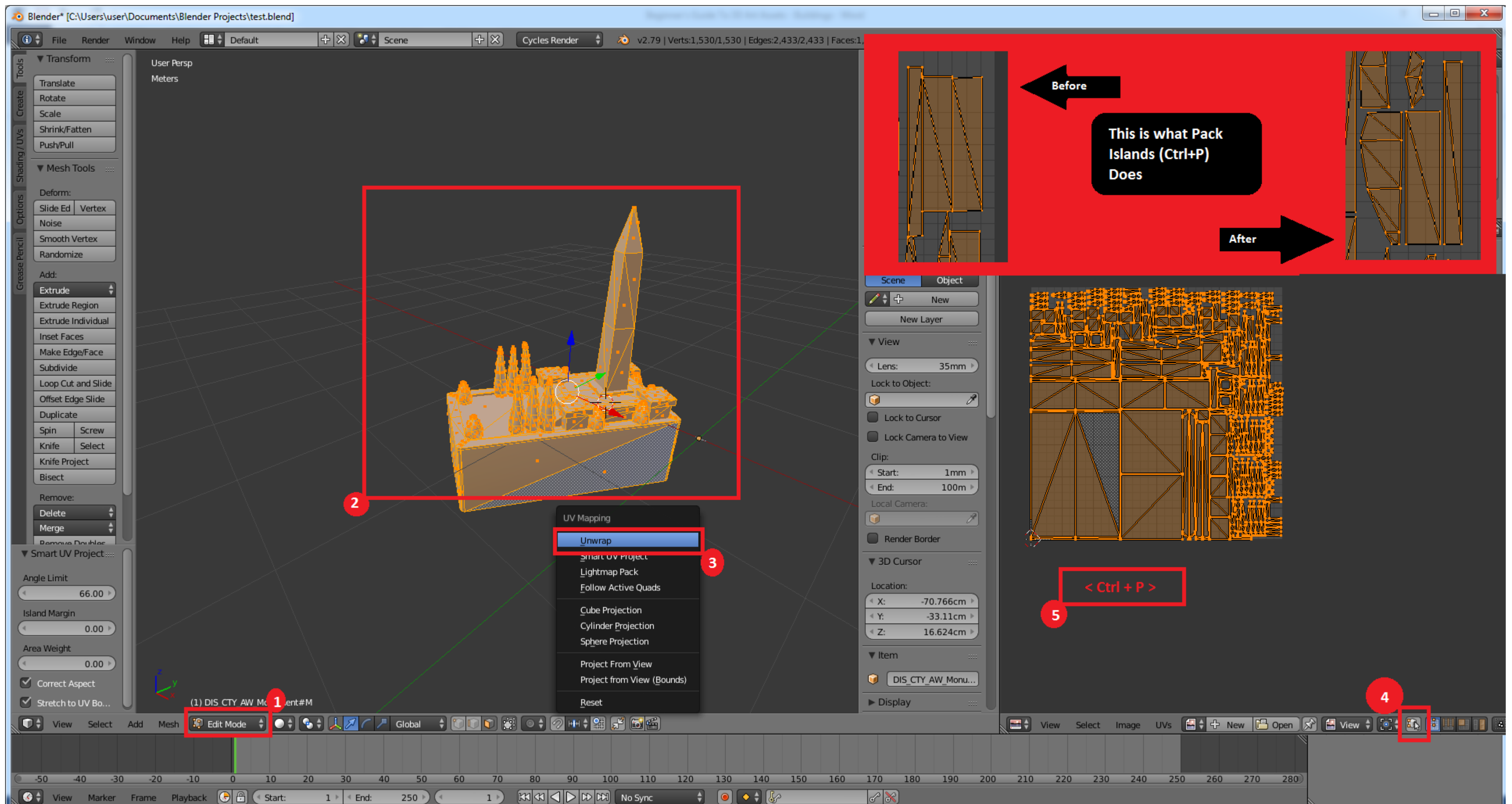
- Note: This is what you were visualizing back in step 10, and where you marked your seams is where Blender cut to lay out the 3D Model (Mesh). If you did not mark seams you should choose Smart UV Project instead of Unwrap. Though as I said before you might get overlapping connected parts and you do not want that or textures will bleed on each other during baking, so you would need to move vertices around in your UV map to fix the overlapping parts. You move stuff around the same way as you did with your 3D Model (Mesh).

20) One last thing I would like to mention again is the position of your 3D Model (Mesh). I noted it in step 8, but I would like to reiterate in case you missed it. How your 3D Model (Mesh) is positioned in Blender will be reflected in game, with position x = 0 and y = 0 in Blender being in the center of a tile in game and z = 0 being ground level. You can look at the image from section 6 step 5 to get an idea of where to position your 3D Model in Blender. So for example, if your building is going to be where the market is, then you will need to position your 3D Model (Mesh) in Blender so that when you load it up in game it will be were the market is. As I mentioned in the introduction for section 1, the easiest way to do this would be to load a base game building, then delete everything except the base and build off of that, then the size and position will be exactly how you want it to be. If you didn't build off of a base game 3D Model (Mesh), then when you load your building in to the ModBuddy AssetEditor and see that your building is not where you want it, you will need to go back in to Blender, reposition it in Edit Mode, then redo the exporting part of section 3 and all of section 4.

21) And that concludes Section 2 of this guide. You created a 3D Model (Mesh) of a building and it is now ready for texturing.

This section will walk you through how to texture your 3D Model (Mesh) in Blender using PBR maps, then how to bake them on to your UV map using Cycles Baking, and finally exporting your project. If you rather use a different method of texturing then you can skip to step 17 of this section after you are done. If you are curious about other ways to texture, I will mention that one of those is to save your UV map as an image, load it in to a photo editing software like Gimp, and texture it in there. However, in this guide we will be going over PBR maps and Cycles Baking only.

*Note: Cycles Baking your textures in Blender is not required what so ever. Cycles Baking is a difficult process. You can simply save your UV map as an image and paint on it in a photo editing software like Gimp, then just skip down to step 17 in this section and resume. In the UV/Image Editor panel, click UVs, then click Export UV Layout. Type what you want to name the file, and click the Export UV Layout button. You will now have your UV map as an image in PNG format. Open the file in an image editing program of your choice and paint. In section 2 step 19 we clicked the Keep UV and edit mode mesh selection in sync button, so when you select a face on your UV map in the UV/Image Editor panel, it will also become selected in the 3D View panel. Use this to identify which faces on your UV map corresponds to which faces on your 3D Model in Blender, so you know what is what while painting your textures. Skip down to step 17 to continue.
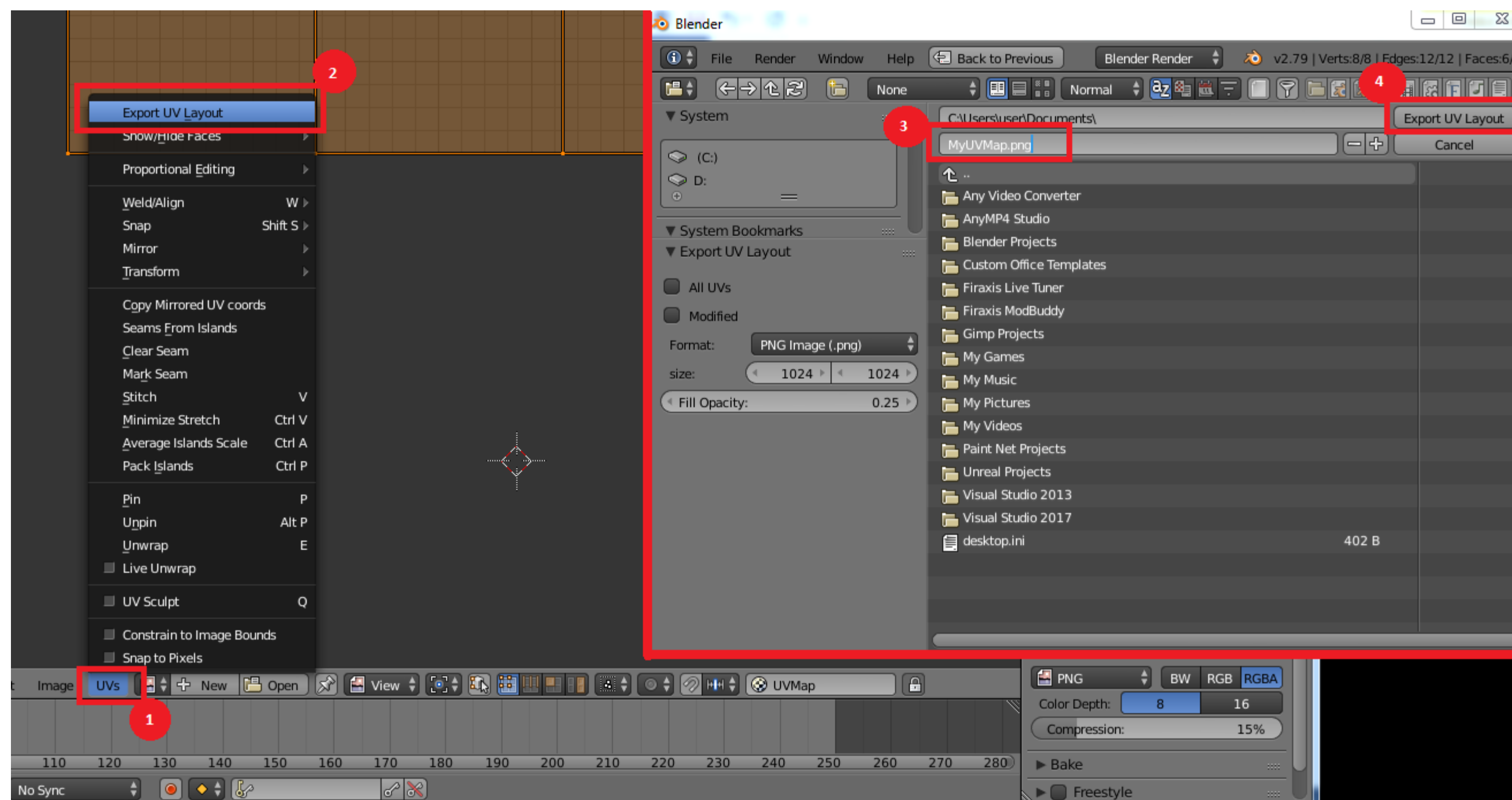
1) The first thing you need to do is get PBR (Physically-Based Rendering) textures for your building. You may also hear it be called Substance Materials, or PBR Substance Materials. A PBR texture of a material will consists of multiple different types of PBR maps that, when combined, make up the texture. There are two ways to go about getting PBR textures, downloading them from the internet or making them yourself in Blender. In this guide you will be downloading them from the internet. There are many different websites where you can download PBR textures for free. You can simply google search free PBR textures and you get a whole list. Be sure to download all of the different PBR map types listed for a specific material. I will go ahead and list two good ones that I have used before.
   - https://www.poliigon.com/
     o Click Free Textures Icon, or click Explore Full Library Button and search free textures.
     o Download 1K image sizes (1024x1024). The whole PBR set will come in one download.
   - https://www.textures.com/
     o Under Categories, click on Substance.
     o Download 1024px image sizes (1024x1024). You will need to download each type of PBR map separately for a material.
     o Note: This site requires you to make an account, it is free, and it has a 15 images per day download limit and a 1024px size limit for free accounts.
   - Below is a list of common PBR maps you will see while searching.

| PBR Map Type	Other Names	Description
Color Map	Albedo/Base Color/Diffuse Map	The base color image of your texture
Ambient Occlusion Map	AO	Tells which areas will be less exposed to light
Normal Map (Bump Map)	Normals	Gives your texture depth and features
Height Map (Bump Map)	Displacement Map	Gives your texture depth and features
Glossiness Map	Gloss Map	Tells which areas will have reflection (inverted Roughness Map)
Roughness Map		Tells which areas will have reflection (inverted Glossiness Map)
Metalness Map	Metallic Map	Tells which areas are metallic
Do not download the following. They are used in a specular workflow and are not needed for the Principled Shader.		
Specular Map		Tells the amount of light reflected by each part of your texture
Reflection Map		Tells which parts of your texture has reflection

   - Note: Make a folder to keep all of your PBR texture downloads in, for example create a folder called PBR Textures in My Documents.
2) Now that you have all of your PBR textures, open the Blender application and open your project, if you don't have it open already. In Object Mode, select your 3D Model (Mesh), and change the Method to Display Objects to Material. Make sure your render type is set to Cycles Render. In the very bottom panel below the 3D View panel, hover your mouse at the top of the Panel between where it ends and the where the 3D View panel begins, you will see it turn into an up/down double arrow, left click and hold, drag up to increase the panel size to take up about half of the screen, then release the left mouse button. Now look to the bottom left of the panel and you will see a little button with a white clock in it, click this button and select Node Editor from the list. Under the Properties Menu click Material. Click your first material to select it, and under the Surface section click the Use Nodes Button.
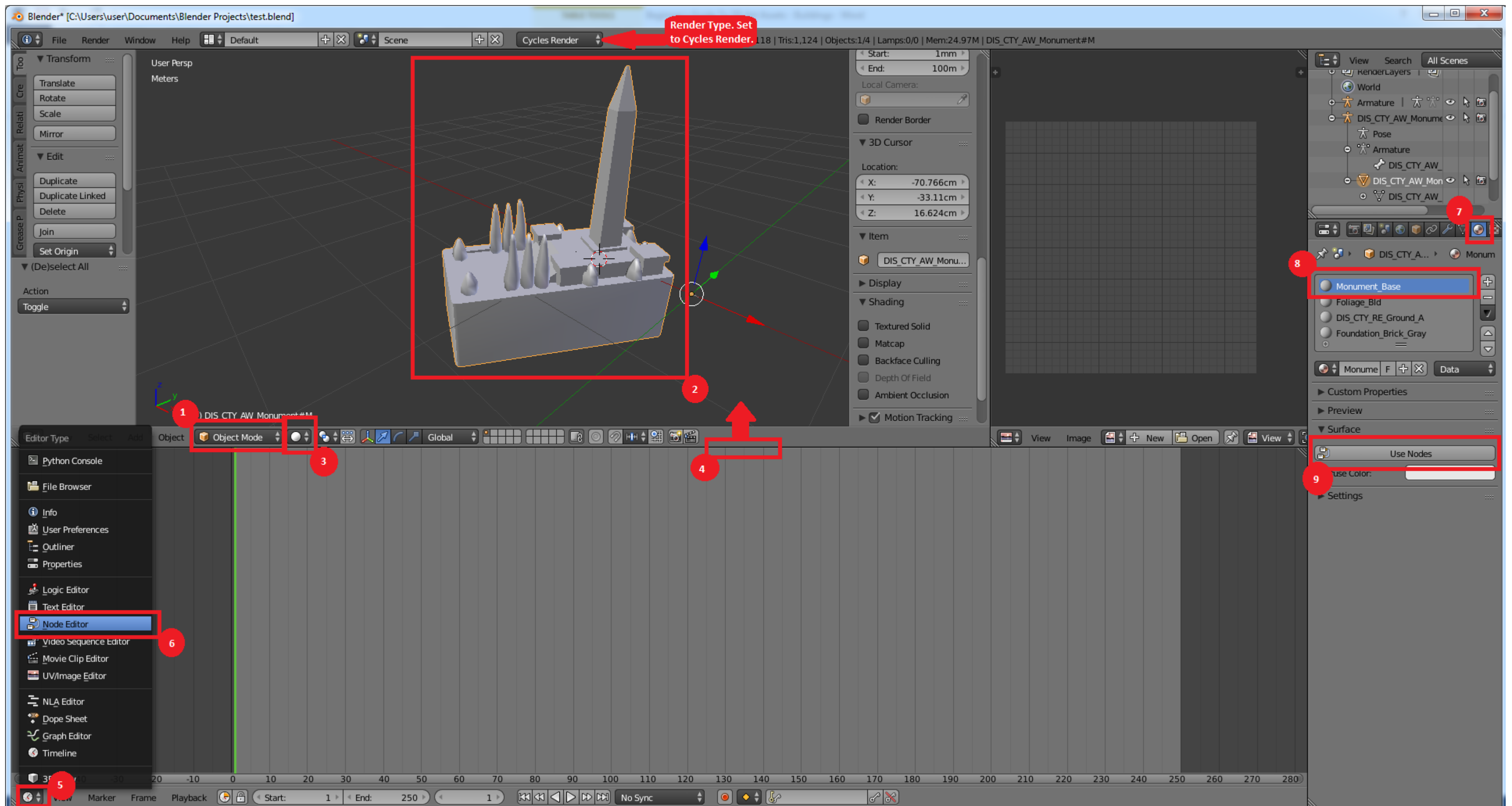
- Note: Position your 3D Model (Mesh) in the 3D View panel so you can see the whole thing, like shown in the image above.

3) In the Node Editor Panel. Tip: Zooming, moving, and selecting are the same as they were in the 3D View panel. The two small window looking things that have a white line between them are called Nodes. That white line shows that the two nodes are connected. Select just the node titled Diffuse BSDF and press delete. Now you should only have one node titled Material Output. On the bottom menu bar, click the Add button, hover your mouse cursor over Shader, and click Principled Shader. You will see a new node pop up in the Node Editor panel labeled Principled BSDF. To move a node just left click and hold, then drag your mouse, and release the left mouse button to place. Now you see the little green dot on the top right side of the Principled BSDF node next to the label BSDF, left click it and hold, then drag your mouse cursor over to the first little green dot at the top left of the Material Output node labeled Surface, you will see a yellow string trailing behind as you move, then it will turn white when you are over the little green dot on the Material Output node, now let go of the left mouse button. You have successfully added a node and connected it to another node. Tip: You can disconnect two nodes by holding down Ctrl, then left click and hold near the center the white line you want to detach, drag your mouse across the white line, and release the left mouse button. You will see your cursor turn in to a knife and trail a small white line as you do this.
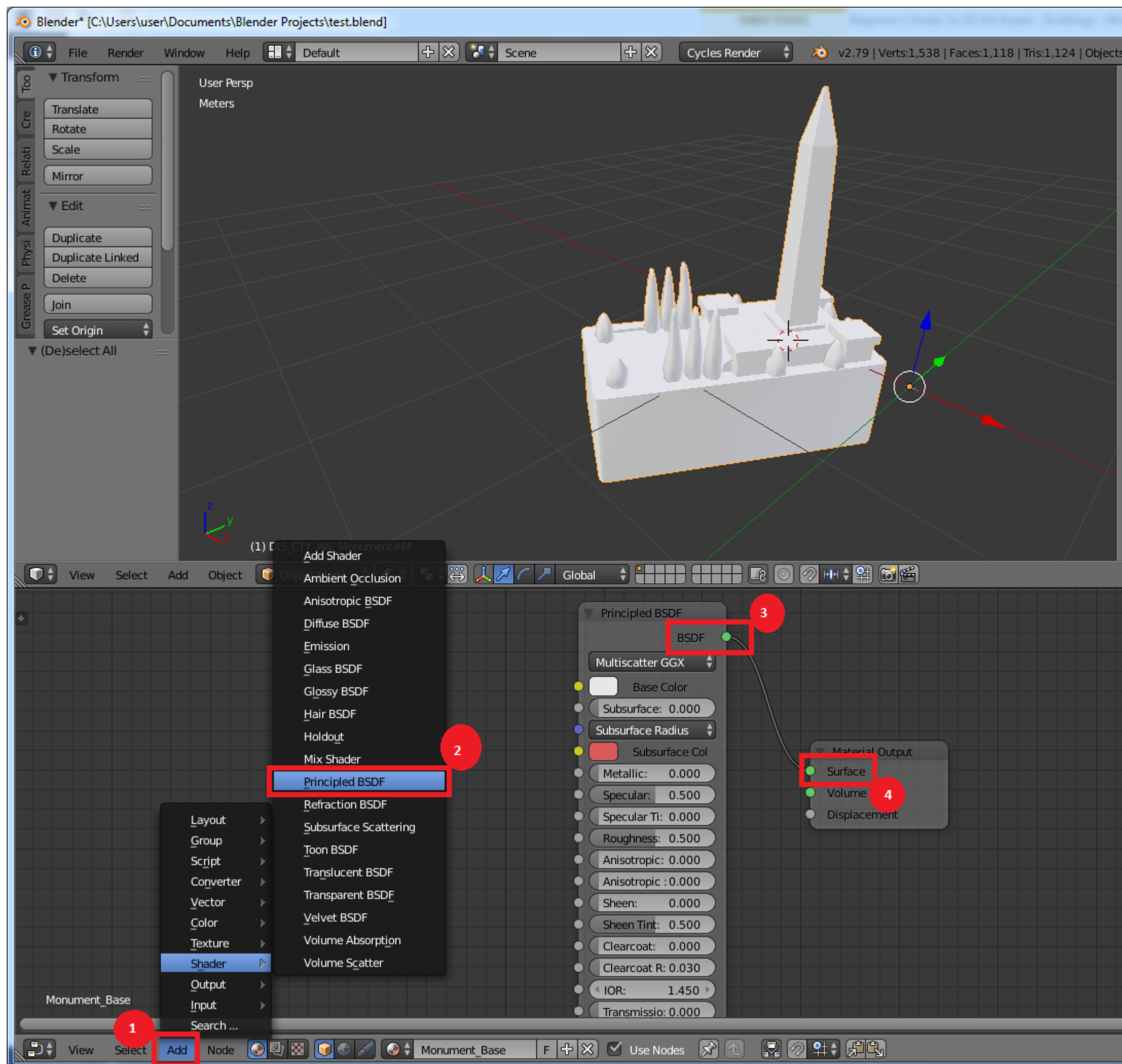
- Note: The Principled Shader is an all-in-one shader based on the Disney principled model also known as the PBR Shader. This shader is what you want to use when using PBR maps to texture your 3D Model (Mesh). It was added to Blender in version 2.79, if you have an older version of Blender this shader will not be there so you will need go download the newest version of Blender.

4) In the Node Editor panel. To add your Color Map PBR texture, click Add, hover your mouse cursor over Texture, and select Image Texture. In the Image Texture node click the third black button from the top titled Flat and select Box (Method to Project), then click the grey button titled Open, navigate to the folder you have your PBR textures in, select the Color Map of the texture you want to use for this material, and click Open Image. In the Node Editor Panel, click Add, hover your mouse cursor over Color, and select MixRGB. In the MixRGB node click the black button titled Mix, and select Multiply. In the Node Editor panel, click Add, hover your mouse cursor over Input, and select Texture Coordinate. In the Node Editor panel, click Add, hover your mouse cursor over Vector, and select Mapping. Now connect the small blue dot labeled Generated in the Texture Coordinate node to the small blue dot labeled Vector (on the left side) in the Mapping node. Connect the small blue dot labeled Vector (on the right side) in the Mapping node to the small blue dot labeled Vector in the Image Texture node. Connect the small yellow dot labeled Color in the Image Texture node to the small yellow dot labeled Color1 in the MixRGB node (Multiply). Connect the small yellow dot labeled Color in the MixRGB node (Multiply) to the small yellow dot labeled Base Color in the Principled BSDF node. If your Display Method is set to Material, You will see that the texture has appeared on your 3D Model (Mesh) in the 3D View panel. Tip: You can change the scale of your texture in the Mapping node by changing the numerical values for X, Y, and Z under the Scale section. You can use negative numbers to make it smaller. Space your nodes out as shown in the image so you can have room to add the other nodes.
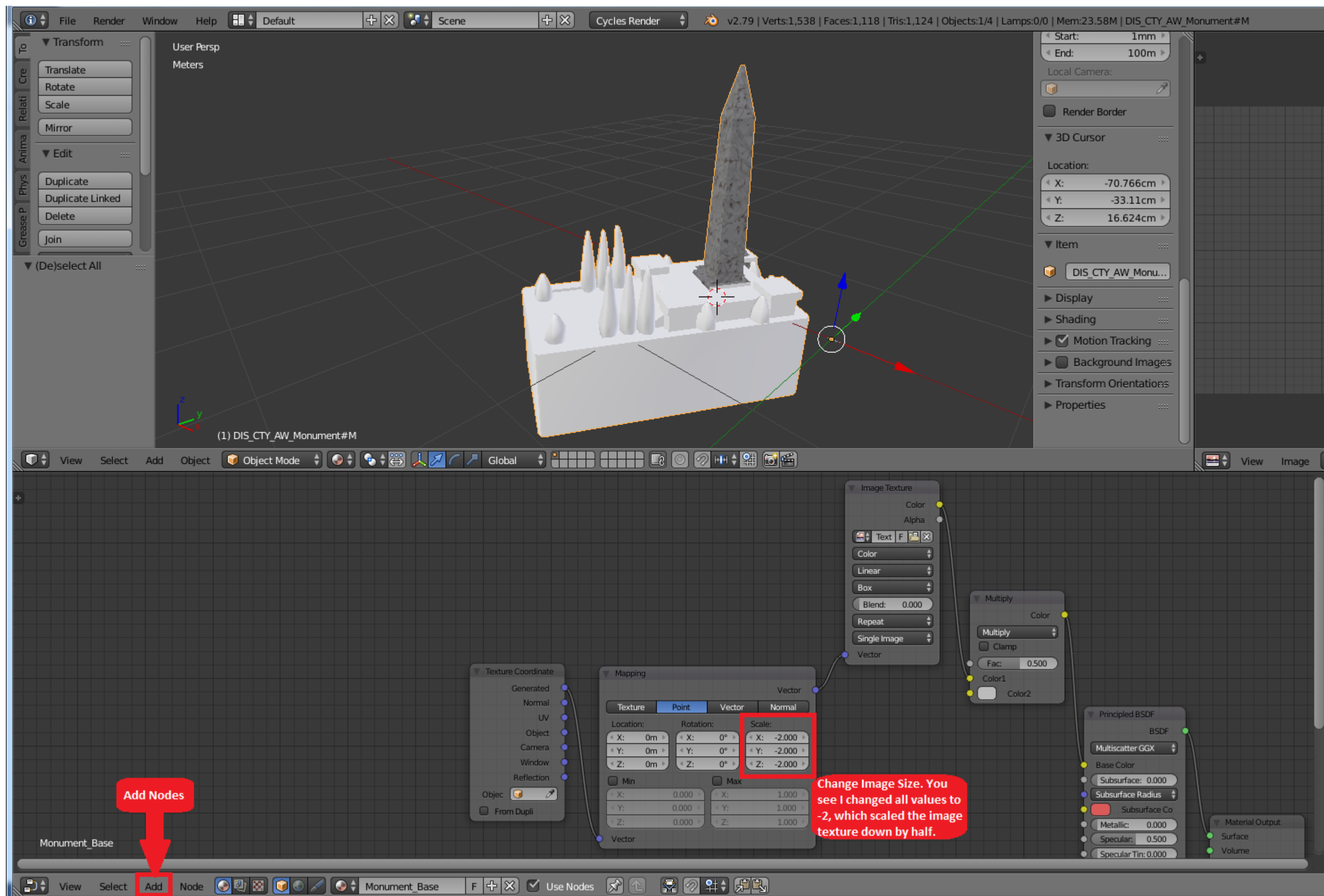
- Note: If you want to change the color of your texture, you would add another MixRGB node between the Image Texture node and the MixRGB (Multiply) node. You can drag and drop the node in between the two and it will auto connect them. Then just click on the white square labeled Color2 and select what color you want it to be. If you want to completely remove the original color, just click the first black button titled Color and select No-Color Data in your Image Texture node.
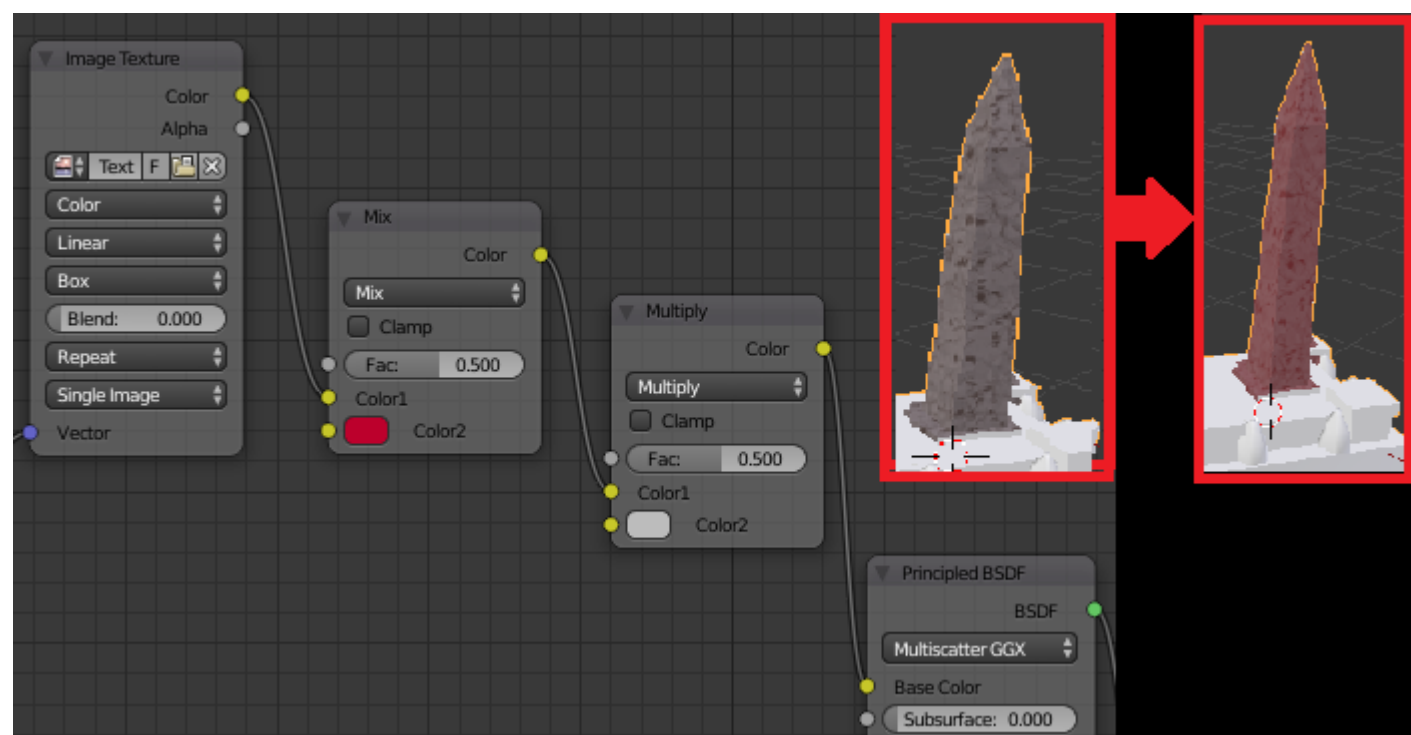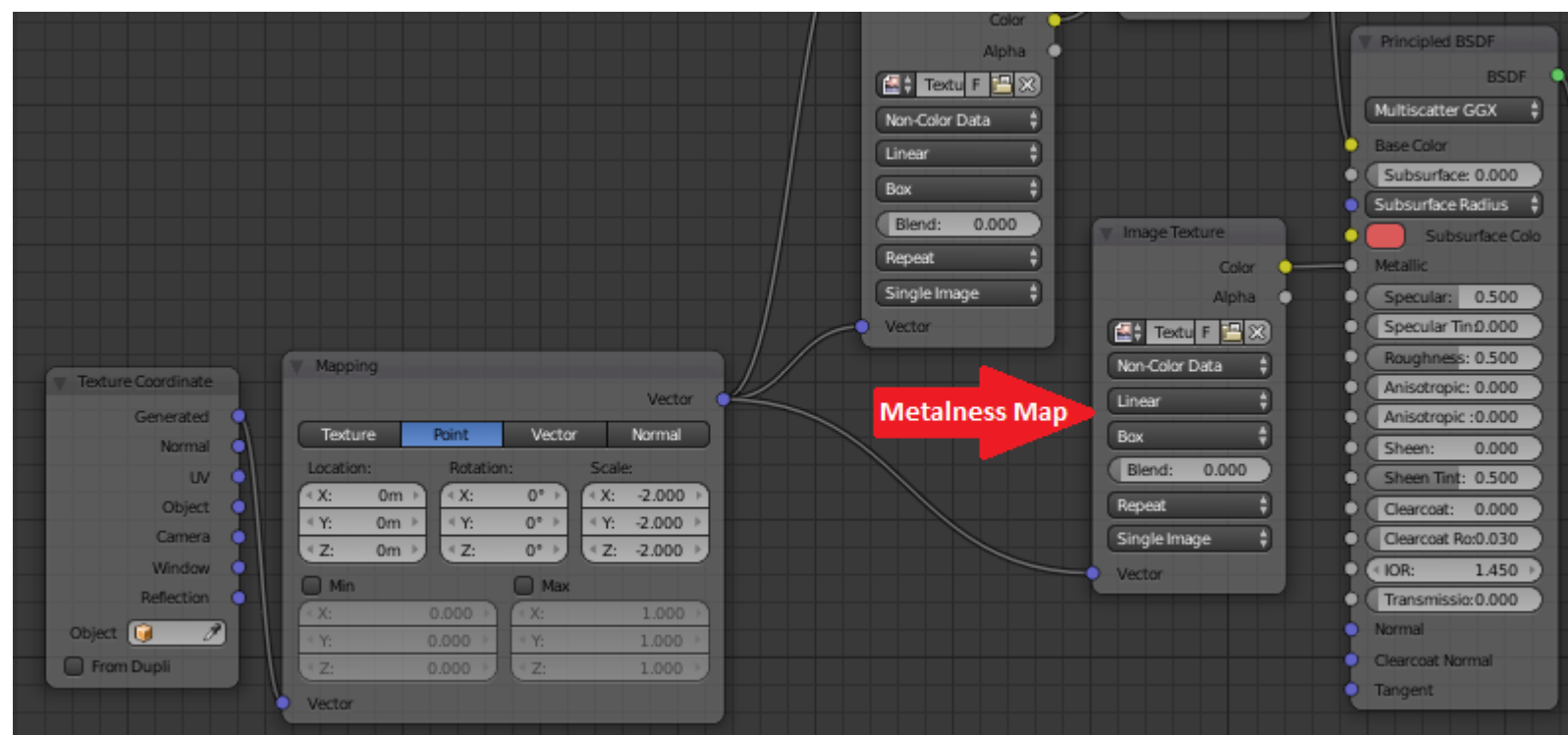

5) In the Node Editor panel. To add your Ambient Occlusion Map PBR texture, hover your mouse cursor over Texture, and select Image Texture. In the Image Texture node click the first black button titled Color and select No-Color Data (Image File Color Space), click the third black button from the top titled Flat and select Box, then click the grey button titled Open, navigate to the folder you have your PBR textures in, select the Ambient Occlusion Map of the texture you want to use for this material, and click Open Image. Connect the small blue dot labeled Vector (on the right side) in the Mapping node to the small blue dot labeled Vector in the Image Texture node. Connect the small yellow dot labeled Color in the Image Texture node to the small yellow dot labeled Color2 in the MixRGB (Multiply) node. This is the same node that your Color Map is attached to in the Color1 slot. Tip: The only Image Texture node with a Color Image File Color Space is the one for the Color Map, all of the other ones will be set to No-Color Data.
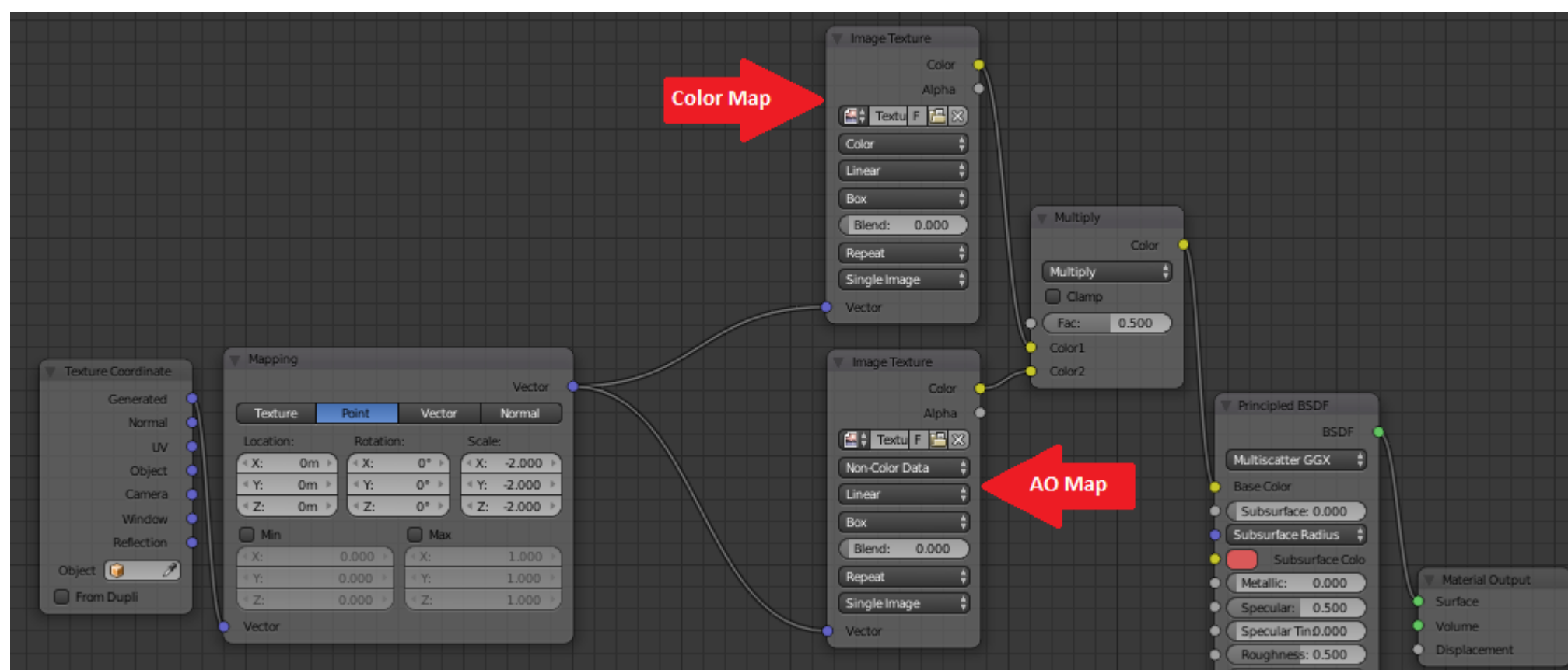

- Note: If you do not have an Ambient Occlusion Map just skip this step, it is not required. You will want to delete the MixRGB (Multiply) node though, and connect the Image Texture node directly in to Base Color in the Principled BSDF node.

6) In the Node Editor panel. To add your Metalness Map PBR texture, click Add, hover your mouse cursor over Texture, and select Image Texture. In the Image Texture node click the first black button titled Color and select No-Color Data, click the third black button from the top titled Flat and select Box, then click the grey button titled Open, navigate to the folder you have your PBR textures in, select the Metalness Map of the texture you want to use for this material, and click Open Image. Connect the small blue dot labeled Vector (on the right side) in the Mapping node to the small blue dot labeled Vector in the Image Texture node. Connect the small yellow dot labeled Color in the Image Texture node to the small grey dot labeled Metallic in the Principled BSDF node.


- Note: If you do not have a Metalness Map just skip this step, it is not required.
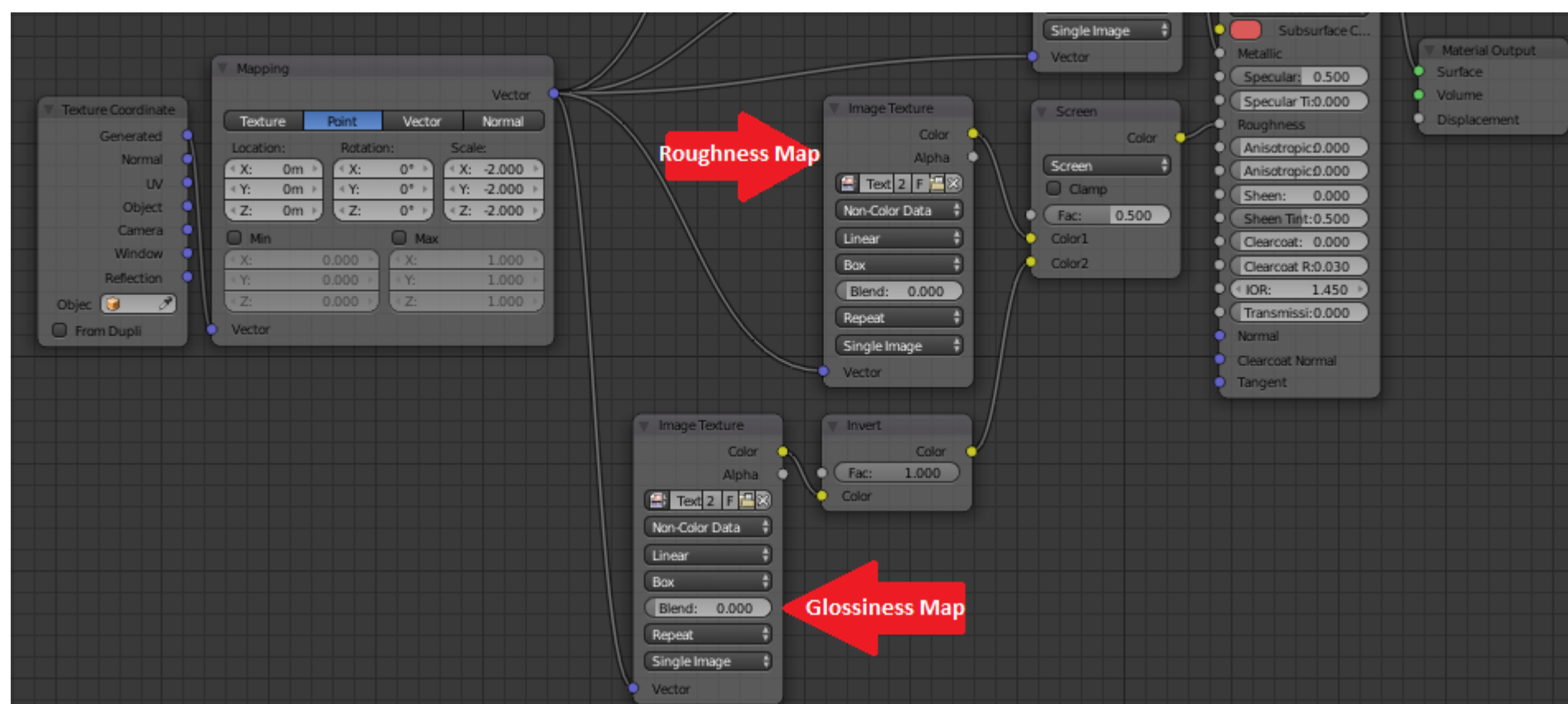
7) In the Node Editor panel. To add your Roughness Map PBR texture, click Add, hover your mouse cursor over Texture, and select Image Texture. In the Image Texture node click the first black button titled Color and select No-Color Data, click the third black button from the top titled Flat and select Box, then click the grey button titled Open, navigate to the folder you have your PBR textures in, select the Roughness Map of the texture you want to use for this material, and click Open Image. Connect the small blue dot labeled Vector (on the right side) in the Mapping node to the small blue dot labeled Vector in the Image Texture node. Connect the small yellow dot labeled Color in the Image Texture node to the small grey dot labeled Roughness in the Principled BSDF node.
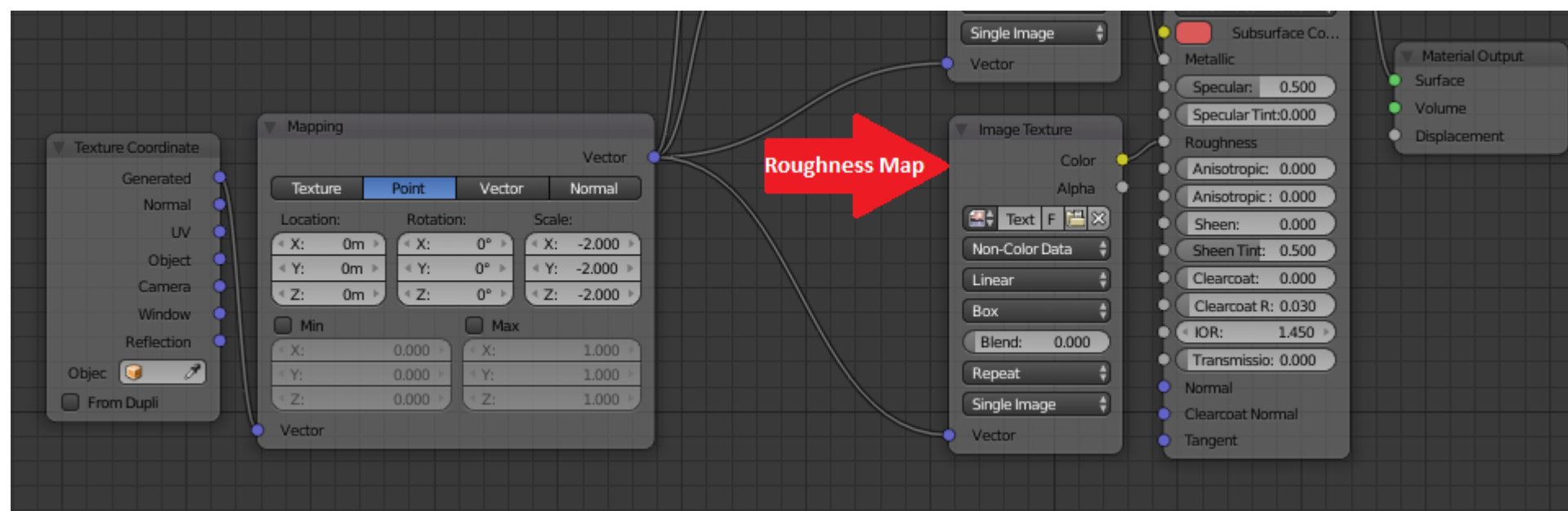
- If you have a Glossiness Map instead. In the Node Editor panel, click Add, hover your mouse cursor over Texture, and select Image Texture. In the Image Texture node click the first black button titled Color and select No-Color Data, click the third black button from the top titled Flat and select Box, then click the grey button titled Open, navigate to the folder you have your PBR textures in, select the Glossiness Map of the texture you want to use for this material, and click Open Image. In the Node Editor panel. Click Add, hover your mouse cursor over Color, and select Invert. Connect the small blue dot labeled Vector (on the right side) in the Mapping node to the small blue dot labeled Vector in the Image Texture node. Connect the small yellow dot labeled Color in the Image Texture node to the small yellow dot labeled Color (on the left side) in the Invert node. Connect the small yellow dot labeled Color (on the right side) in the Invert node to the small grey dot labeled Roughness in the Principled BSDF node.
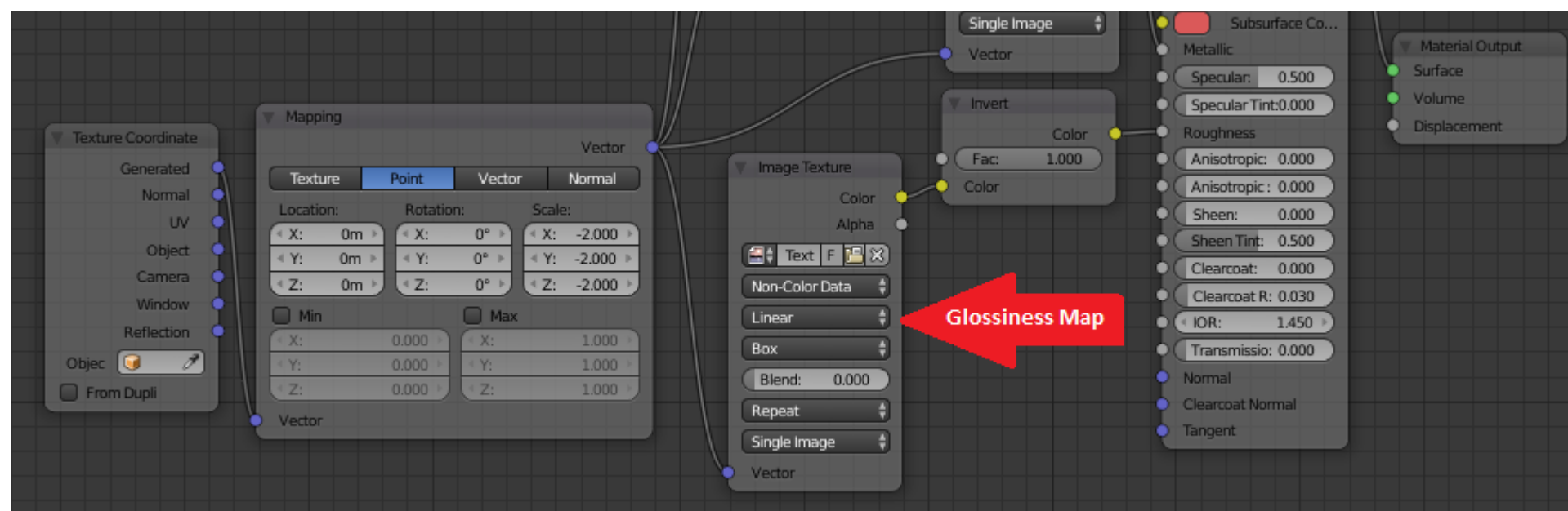

- If you have both a Roughness Map and a Glossiness Map, I recommend just using one because both are basically an inverted version of each other. I mean, I am not saying that they are an exact inverted copy of each other, just you will not really notice any difference with what we are doing here. But if you want to use both anyways here is how. Do the above step to add a Roughness Map and the above step to add a Glossiness Map, but do not connect them to the Principled BSDF node. In the Node Editor panel. Click Add, hover your mouse cursor over MixRGB. In the MixRGB node click the black button titled Mix, and select Screen. In the Node Editor panel, connect the small yellow dot labeled Color in the Image Texture node of your Roughness Map to the small yellow dot labeled Color1 in the MixRGB node (Screen). Connect the small yellow dot labeled Color in the Invert node to the small yellow dot labeled Color2 in the MixRGB node (Screen). Connect the small yellow dot labeled Color (on the right side) in the MixRGB node (Screen) to the small grey dot labeled Roughness in the Principled BSDF node.


- Note: If you do not have a Roughness Map or a Glossiness Map just skip this step, it is not required.

8) In the Node Editor panel. To add your Normal Map PBR texture, click Add, hover your mouse cursor over Texture, and select Image Texture. In the Image Texture node click the first black button titled Color and select No-Color Data, click the third black button from the top titled Flat and select Box, then click the grey button titled Open, navigate to the folder you have your PBR textures in, select the Normal Map of the texture you want to use for this material, and click Open Image. In the Node Editor panel. Click Add, hover your mouse cursor over Vector, and select Normal Map. Connect the small blue dot labeled Vector (on the right side) in the Mapping node to the small blue dot labeled Vector in the Image Texture node. Connect the small yellow dot labeled Color in the Image Texture node to the small yellow dot labeled Color in the Normal Map node. Connect the small blue dot labeled Normal in the Normal Map node to the small blue dot labeled Normal in the Principled BSDF node.
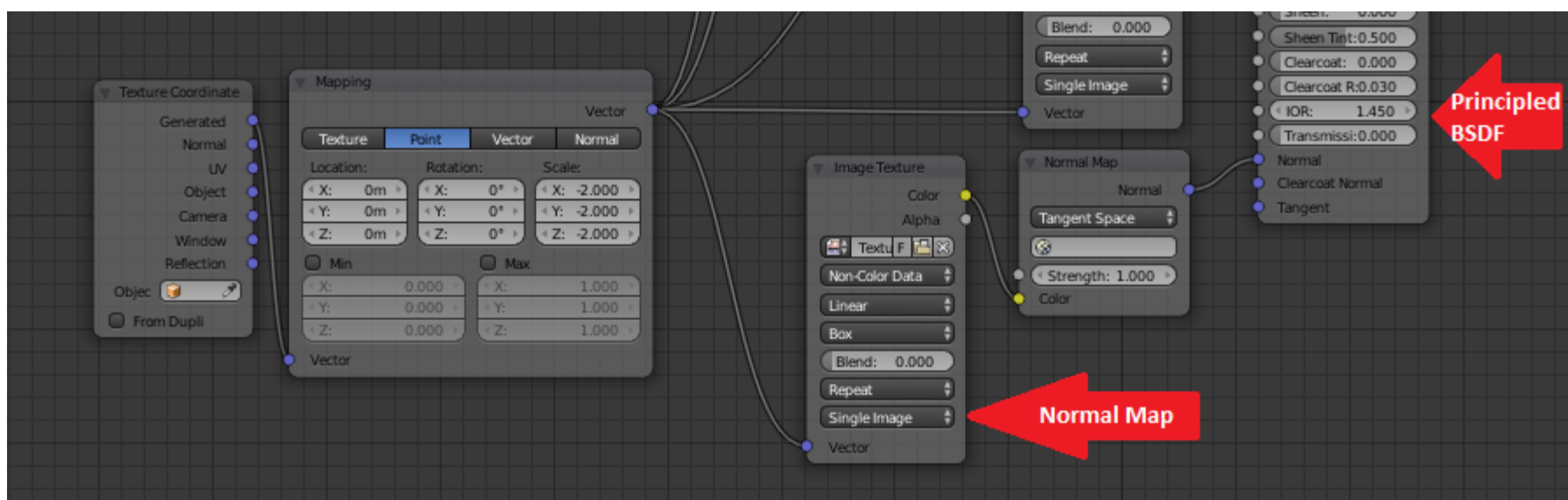
- If you have a Height Map instead. In the Node Editor panel, click Add, hover your mouse cursor over Texture, and select Image Texture. In the Image Texture node click the first black button titled Color and select No-Color Data, click the third black button from the top titled Flat and select Box, then click the grey button titled Open, navigate to the folder you have your PBR textures in, select the Height Map of the texture you want to use for this material, and click Open Image. In the Node Editor panel. Click Add, hover your mouse cursor over Vector, and select Bump. Connect the small blue dot labeled Vector (on the right side) in the Mapping node to the small blue dot labeled Vector in the Image Texture node. Connect the small yellow dot labeled Color in the Image Texture node to the small grey dot labeled Height in the Bump node. Connect the small blue dot labeled Normal in the Bump node to the small blue dot labeled Normal in the Principled BSDF node.
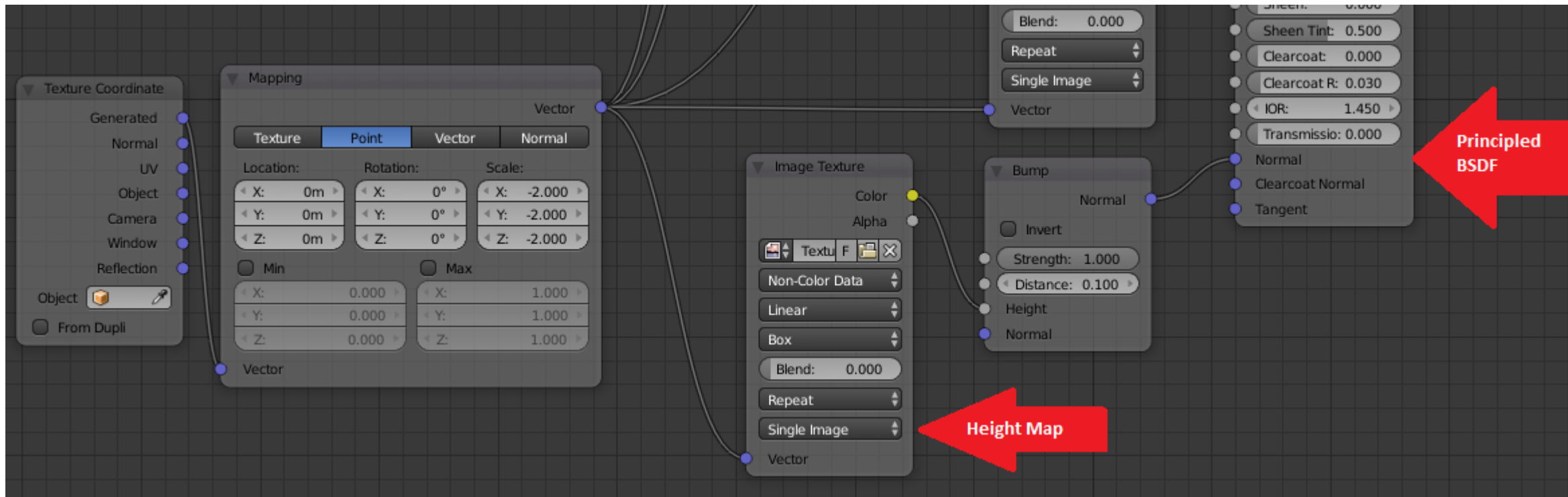

- If you have both a Normal Map and a Height Map, do the above step to add a Normal Map and the above step to add a Height Map, but do not connect them to the Principled BSDF node. In the Node Editor panel, connect the small blue dot labeled Normal in the Normal Map node to the small blue dot labeled Normal in the Bump node. Connect the small blue dot labeled Normal in the Bump node to the small blue dot labeled Normal in the Principled BSDF node.
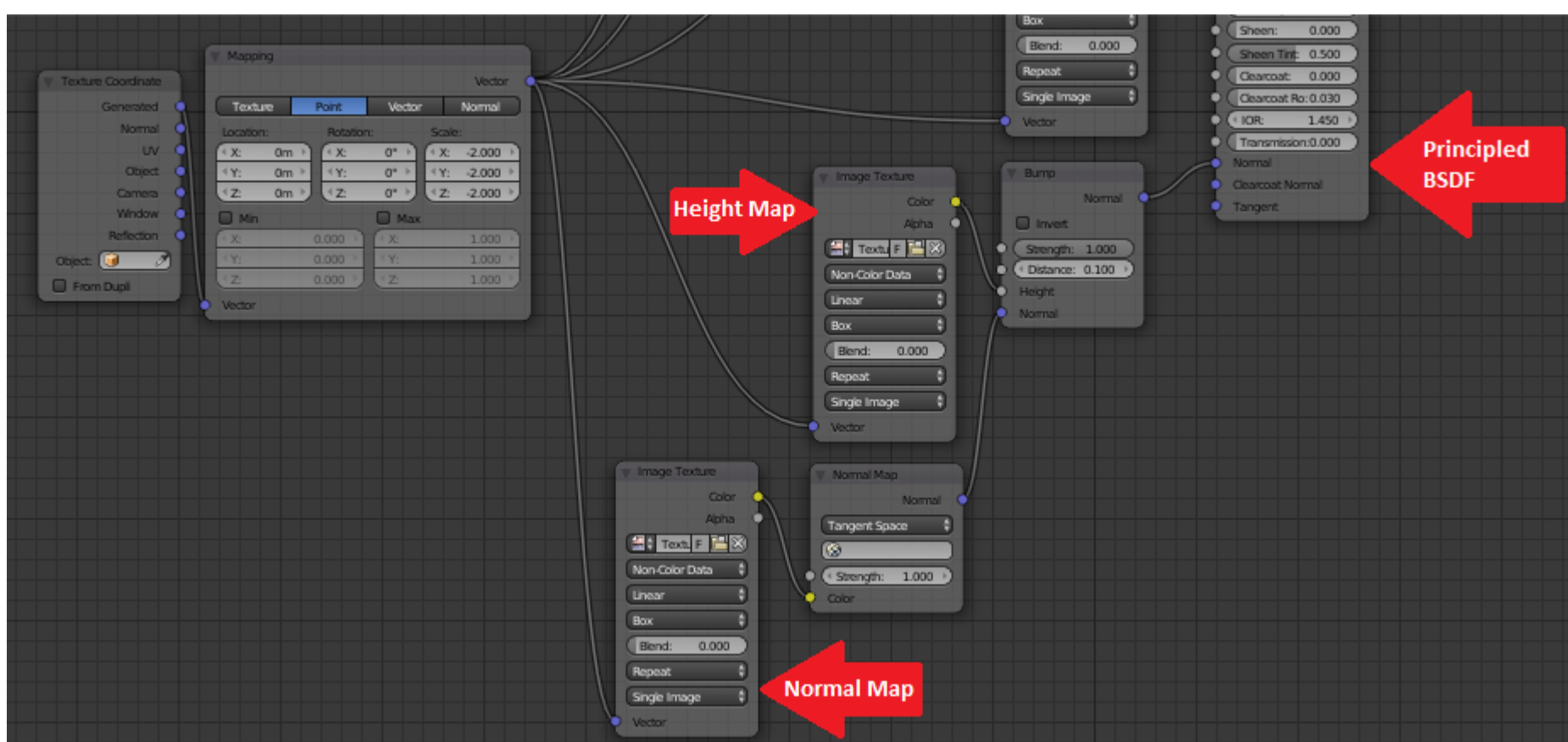

- Note: If you do not have a Normal Map or a Height Map just skip this step, it is not required.
9) Your first material is now fully textured. Repeat steps 3 through 8 for each one of your materials until your whole 3D Model (Mesh) is fully textured. Other than the Color map, you do not have to have every type of PBR map, just add and connect whatever came with your PBR material download, and leave the other slots on the Principled BSDF node empty. Remember that you use the Mapping node to scale, rotate, or move your textures. The purpose of this node setup is to get a preview of how the combination of all of your textures will look on your 3D Model (Mesh) so you can notice issues and make changes if necessary. It is also the same node setup that you use to bake your Color Map. Below is an image of a node layout for a metallic material that you can use as a reference.
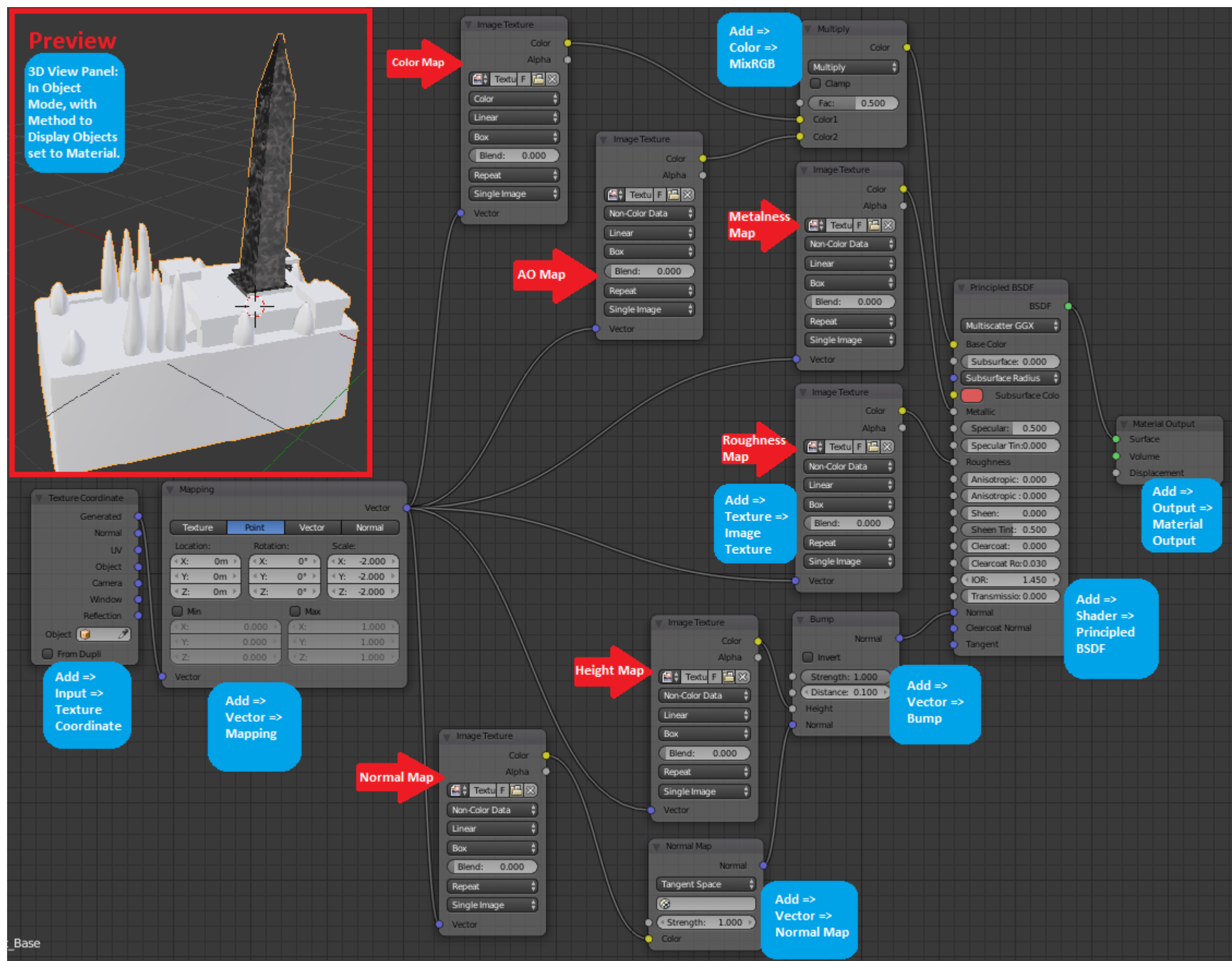
- Note: How it looks in the 3D View panel with Method to Display Objects set to Material is just a rough preview, it will look better rendered.

10) Now that your whole 3D Model (Mesh) is fully textured and general looking how you want it to look in game, it is time to bake the textures on to your UV map. Baking will permanently imprint all of your materials, as currently displayed in the 3D View panel, on to your UV map. You will need to do a bake for each type of PBR map that was used in any of your materials. For example, if you did not use a metalness map in any of your materials then you will not need to do a bake for metalness. However, if you used a metalness map in just one of your materials, then you will need to do a bake for metalness. The first bake we will do is for the Color Map. In the 3D View panel, select your 3D Model (Mesh), go in to Edit Mode, select your entire 3D Model (Mesh). In the UV/Image Editor panel make sure you can see the UV map you made earlier. If you imported a base game 3D Model (Mesh), there will be multiple UV maps in your project so just make sure the one you unwrapped your UV map to is active, which should have been UV1, you can see the name of the active UV map on the bottom menu bar to the far right. Now on the bottom menu bar click the light grey button titled New, type bakeColorMap for the Name field, make sure that Width and Height are both set to 1024px and that Generated Type is set to Blank, and press OK. This will be the image that the textures will bake to.
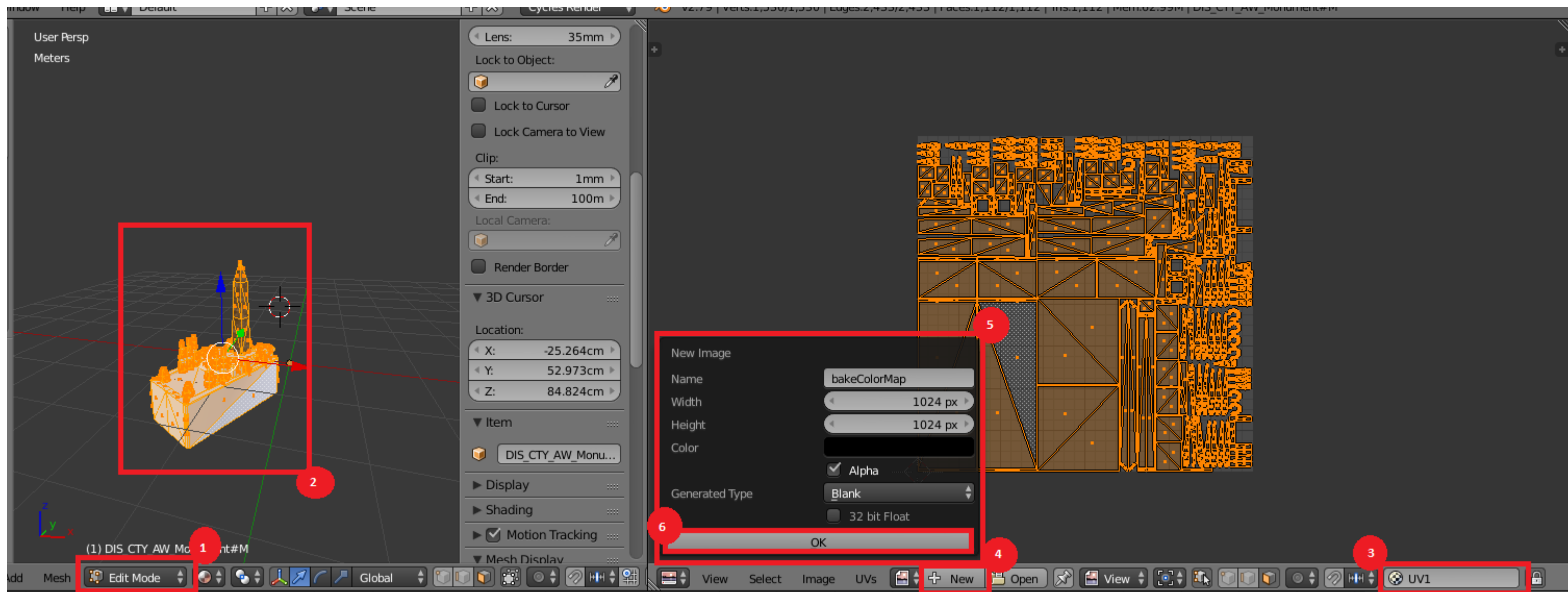
11) In the Node Editor panel, add an Image Texture node, do Not attach it to any other nodes. In the Image Texture node, click the little icon that has a sheet of paper on a black background to the left of the Open button, select the bakeColorMap image you made in step 10, and make sure this node is select and is the only node selected in the Node Editor panel (very important). Repeat for every one of your materials, so every one of your materials will have an unattached Image Texture node, with its image set to bakeColorMap, and be the only node selected. In the Properties Menu, click the Object Data button, under the section labeled UV Maps, make sure the UV map you unwrapped to is highlighted in blue and that the little camera icon (render icon) next to it is active (NOT partially greyed out). Tip: Selected nodes are outlined in orange.
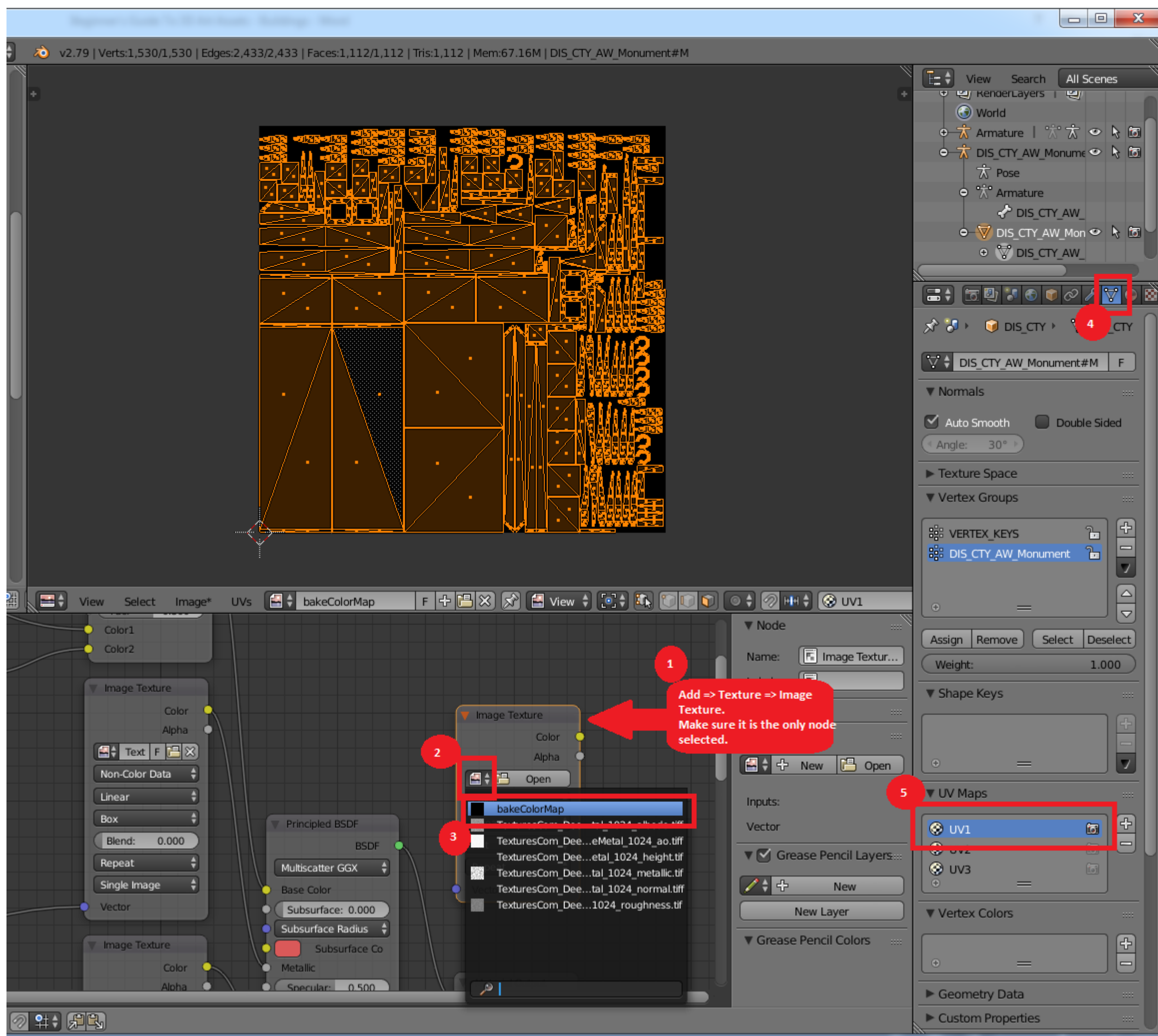
In the Properties Menu, click the Render button, expand the section labeled Bake, it is near the bottom. Click the black button labeled Combined, and select Diffuse (this field is known as the bake type). Click the dark grey button labeled Direct and the dark grey button labeled Indirect. Now check the image below and make sure your screen looks similar. Click the big grey button labeled Bake. If you set everything up correctly the bake process will begin and you will see a little progress bar pop up on the very top menu bar, if there is something wrong then you will see an error message pop up on the very top menu bar instead. If you got an error message, go back through steps 10 through 12 and make sure you did not miss anything. When the baking is finished, check the UV image in UV/Image Editor. Unselect everything and zoom in to see it better. You should see all your materials painted on it in color, with some bleed between islands, and the active image should still say bakeColorMap. If there is an area that is all black or the active image in UV/Image Editor changed, then you have another node selected in one of your materials. Fix it, make sure bakeColorMap is the active image in UV/Image Editor, click Image and select Reload Image, this will reset the bakeColorMap image to its original blank state, and click Bake again.
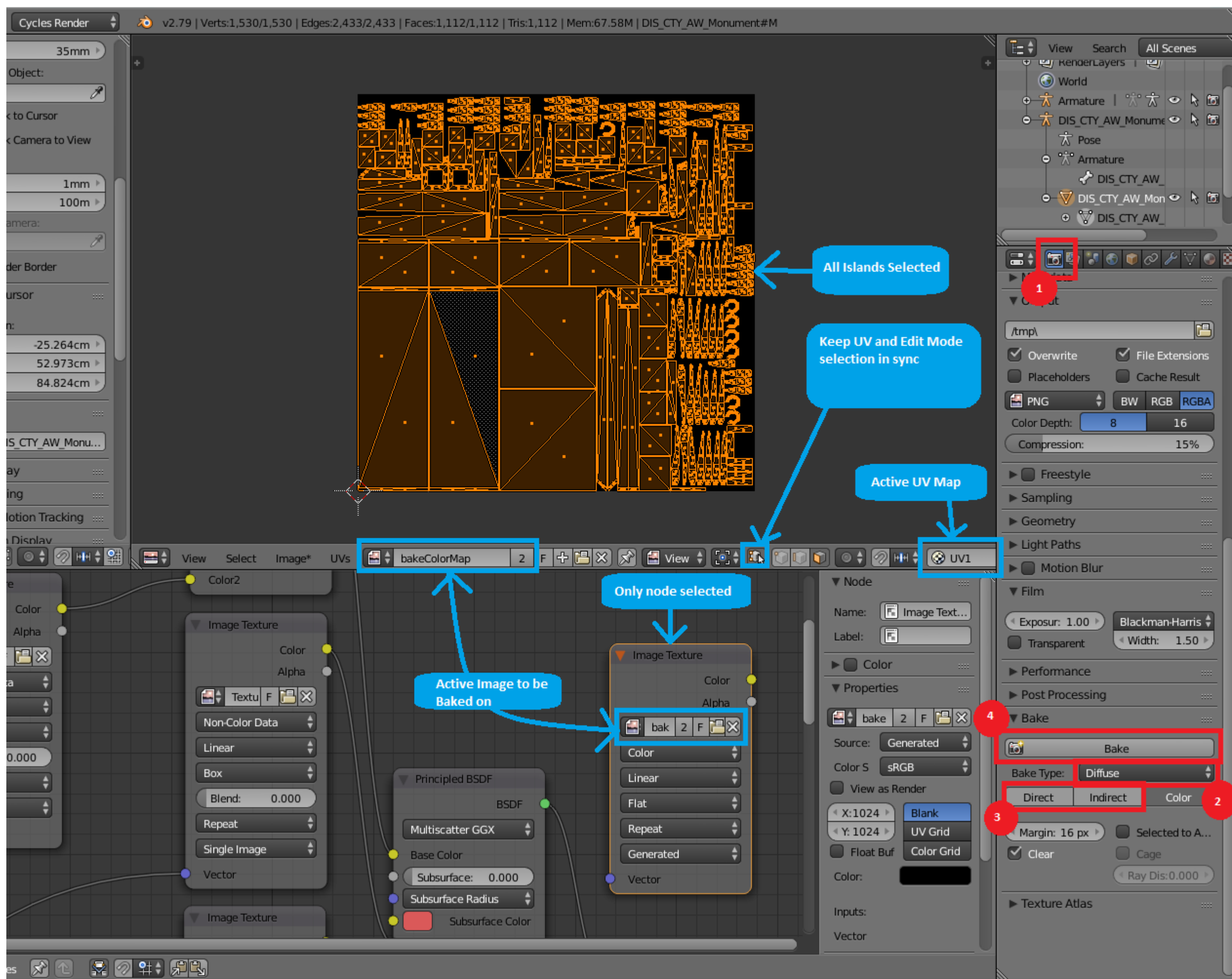
- Below is an example of how a Color Map properly baked on to a UV Map is supposed to look like. Yours should look similar.
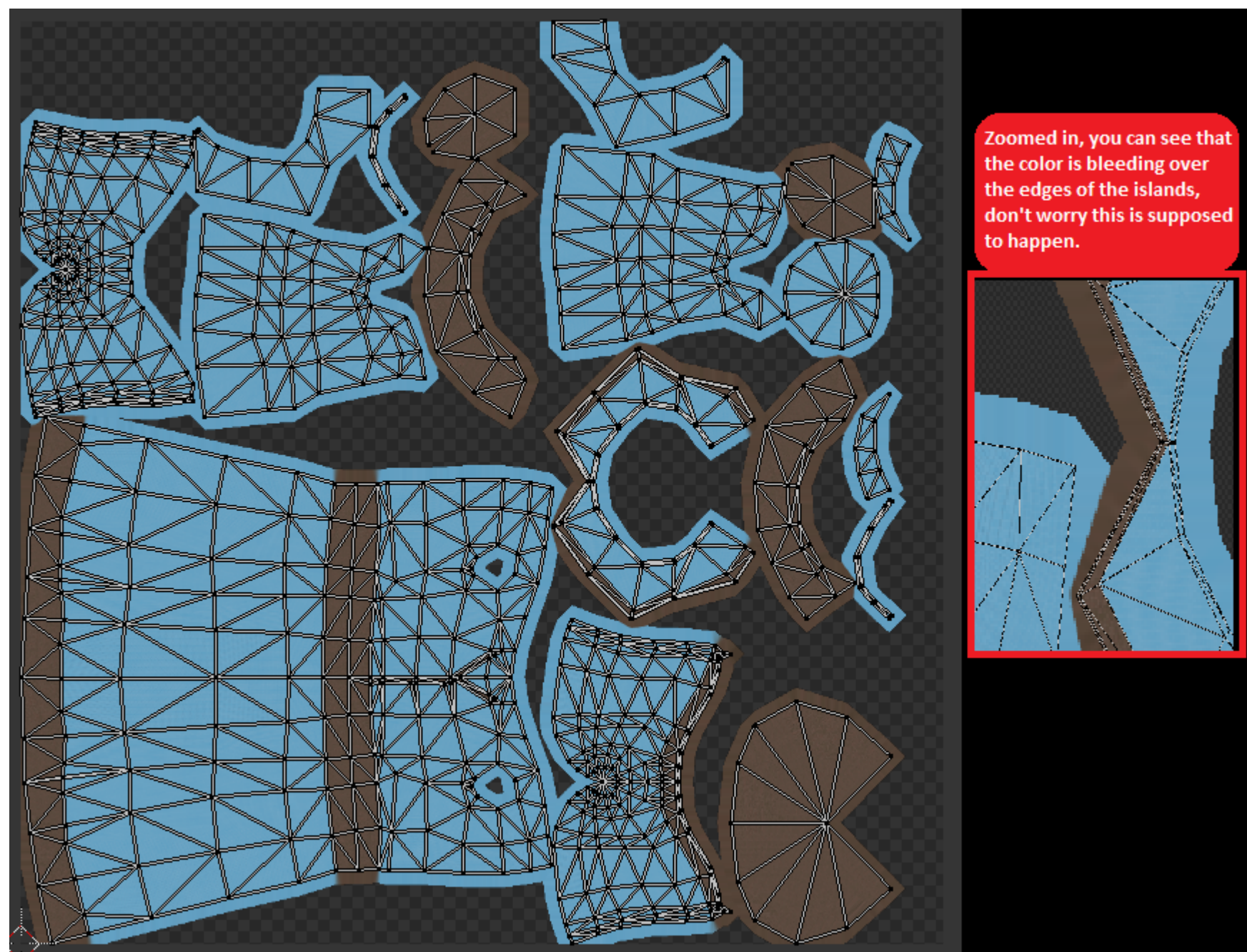

- If everything baked successfully and looks correct you need to save it now. Go ahead and make a new folder in your Blenders Projects folder called Textures. In the UV/Image Editor panel, click on Image, then select Save As Image, navigate to your Textures folder you just created (\Documents\Blender Projects\Textures), then press Save As Image.
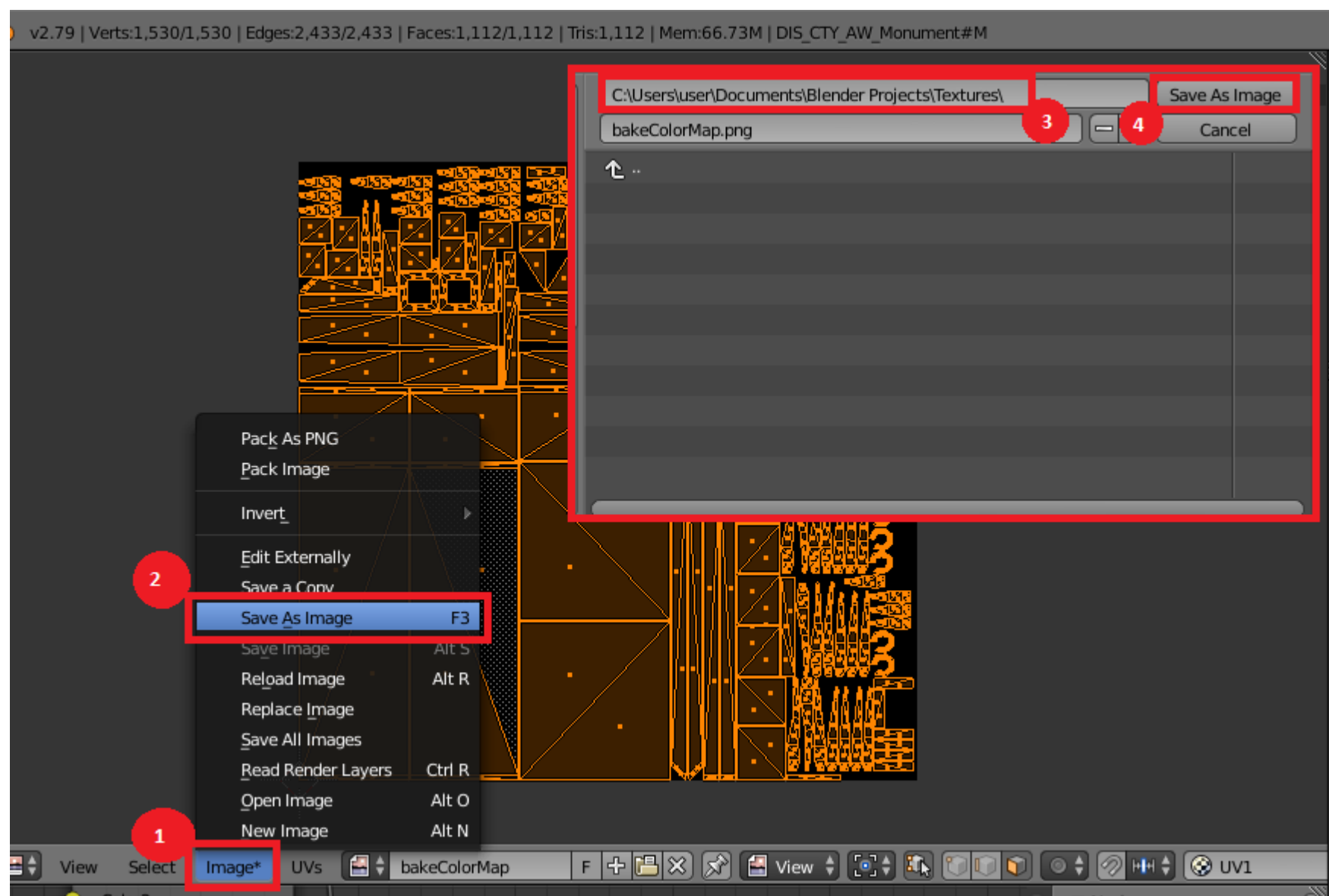
13) A lot of things have already been setup in steps 10 through 12 for baking so I will be focusing on what you need to change for the next bakes. Next we will be baking the Ambient Occlusion map. In the UV/Image Editor panel, click the small grey button with the white plus sign in it to create a new image to bake to, use the same settings as you did in step 10 but name it bakeAOMap and click OK. In the Node Editor panel detach the Image Texture node for Ambient Occlusion from the MixRGB (Multiply) node and the Principled BSDF node from the Material Output node. Click Add, hover your mouse cursor over Converter, and select ColorRamp. In the ColorRamp node click the field labeled Pos: and type 0.5 then press Enter. In the Node Editor panel, click Add, hover your mouse cursor over Shader, and select Emission. Connect the small yellow dot labeled Color in the Image Texture node to the small grey dot labeled Fac in the ColorRamp node. Connect the small yellow dot labeled Color in the ColorRamp node to the small yellow dot labeled Color in the Emission node. Connect the small green dot labeled Emission in the Emission node to the small green dot labeled Surface in the Material Output node. In the Image Texture node that is not connected to anything, change the active image to bakeAOMap image, the same way you selected the bakeColorMap image in step 11, then click the first black button labeled Color and select Non-Color Data. Repeat for every one of your materials. In the Properties Menu, click the Render button, expand the section labeled Bake, change the Bake Type to Emission, the same way you changed it to Diffuse in step 12. Now check the image below and make sure your screen looks similar. Click the big grey button labeled Bake. Check result after baking, if there is an issue, fix it, reload image, and bake again, like described in step 12.
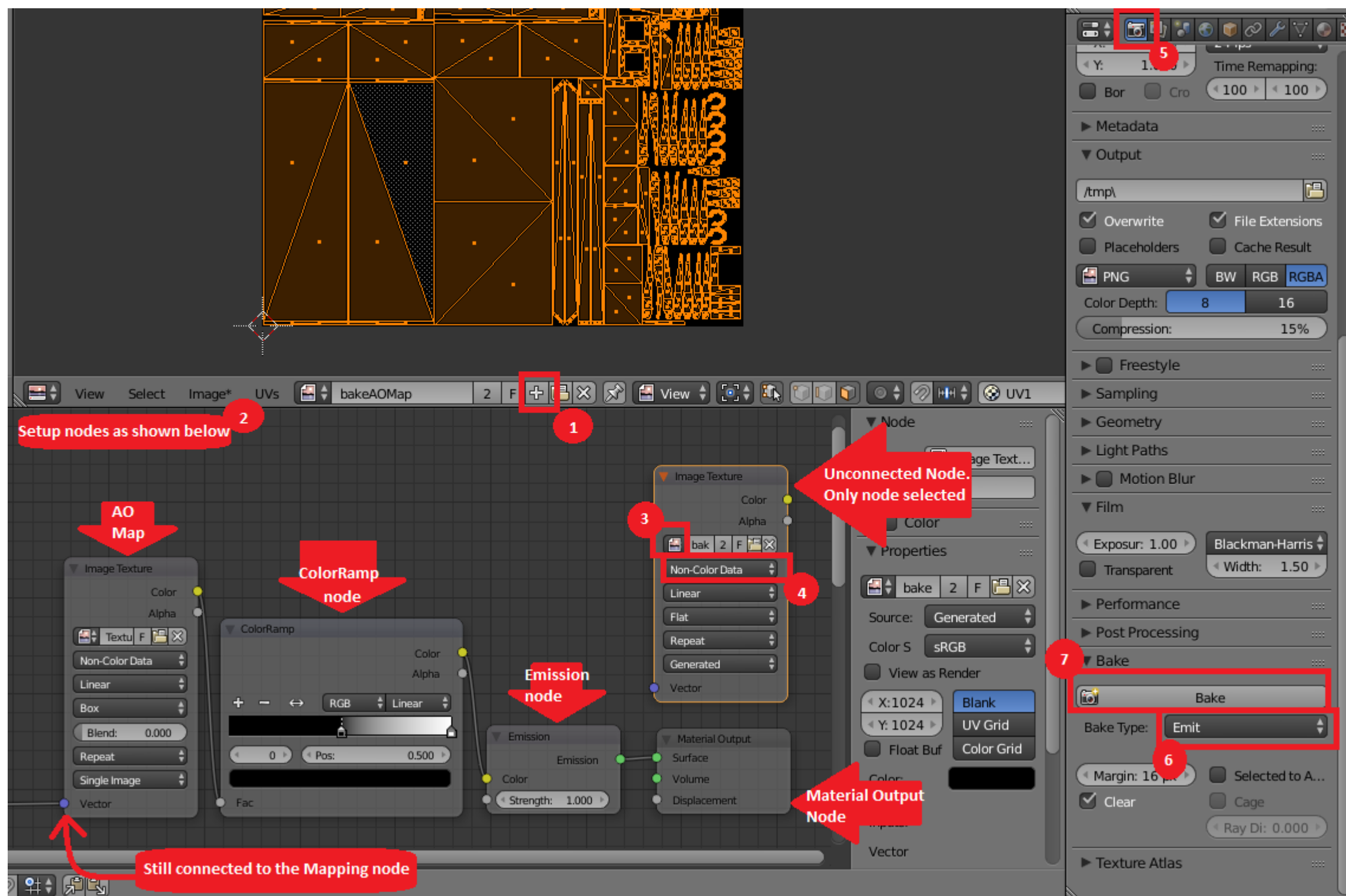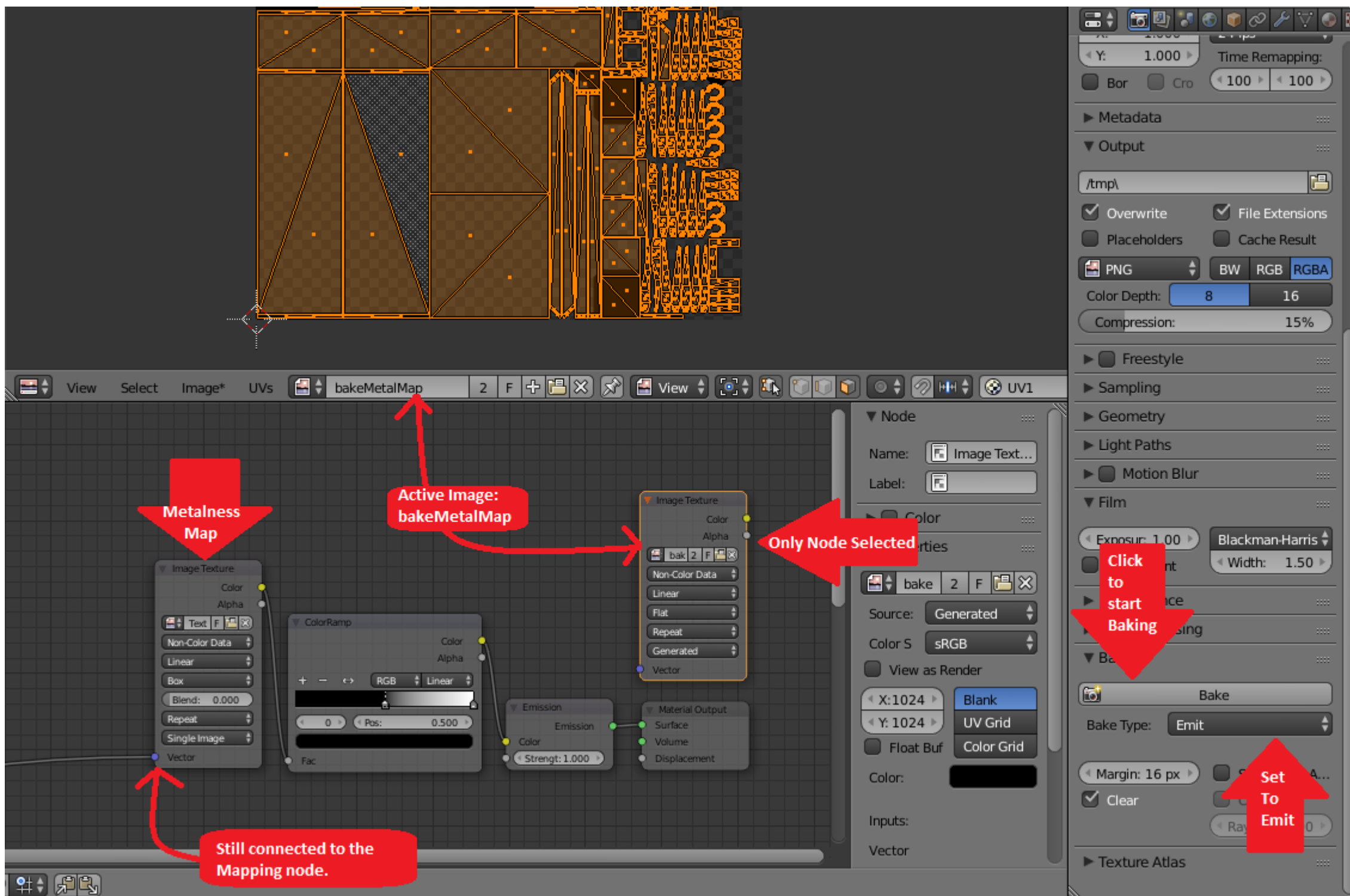

- If you did not have an Ambient Occlusion map in some of your materials, click Add, hover your mouse cursor over Input, and click RGB, In the RGB node set the color to white. Connect the small yellow dot labeled Color in the RGB node to the small yellow dot labeled Color in the Emission node. Exclude the ColorRamp node. Remember, if you don't have an Ambient Occlusion map in any of your materials you do not have to bake an Ambient Occlusion map period. The darker spots on an Ambient Occlusion map means that less light hits those spots, the white areas mean that they get full light exposure.
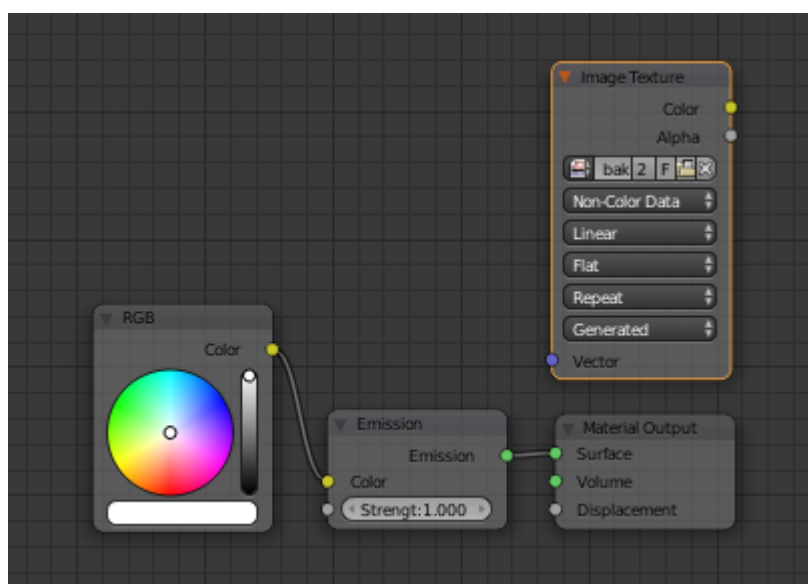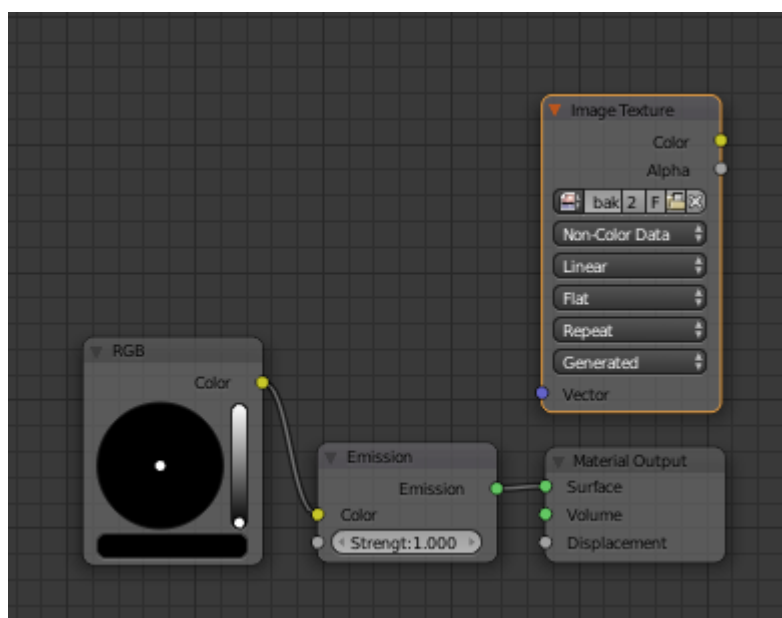

- Below is an example of how an Ambient Occlusion map properly baked on to a UV Map is supposed to look like. Yours should look similar. It can be kind of hard to see in the UV/Image Editor, but it should be white with some shadowed spots. Once you save it and actually look at it in Gimp you will be able to see the detail more clearly.
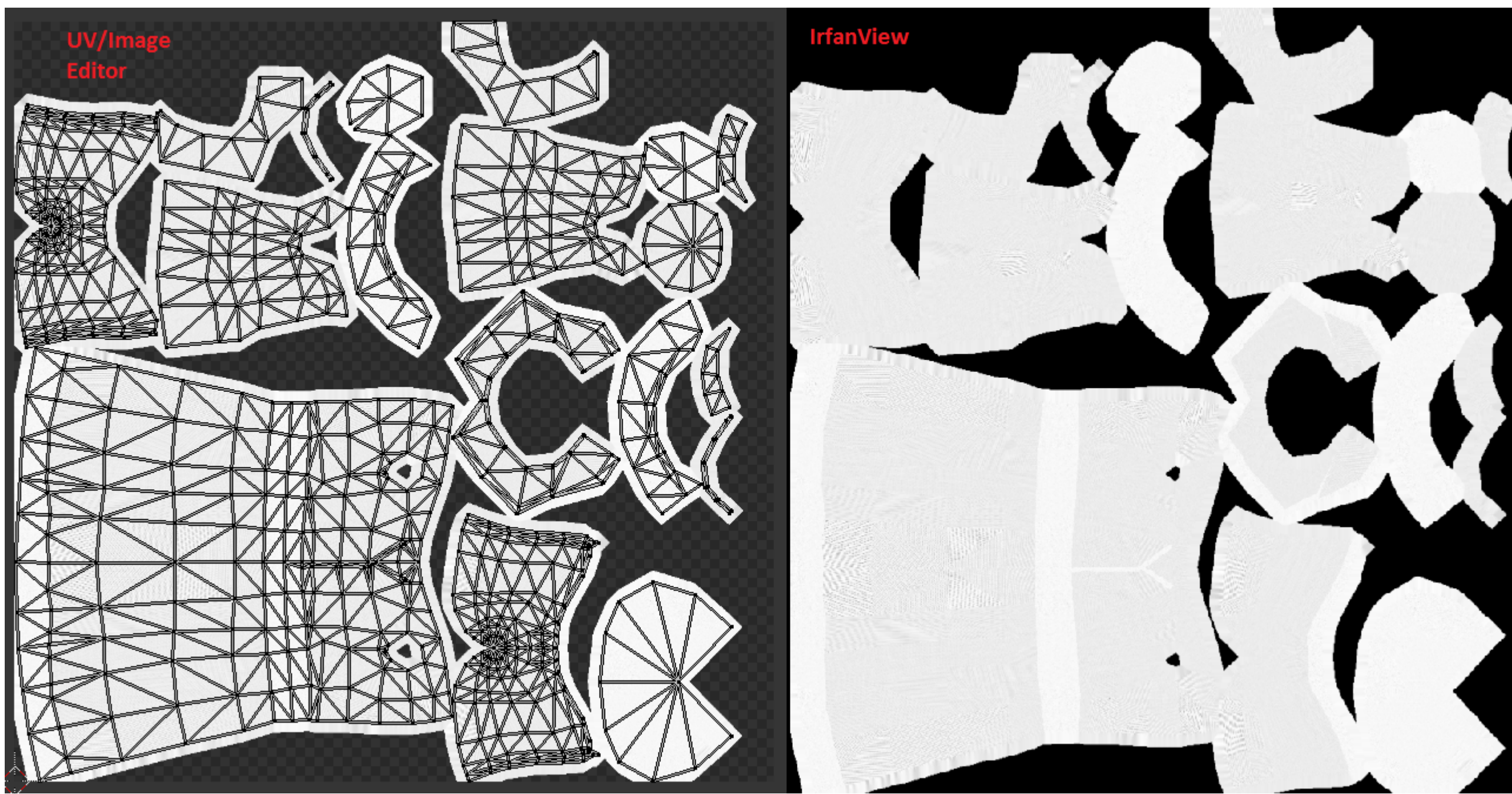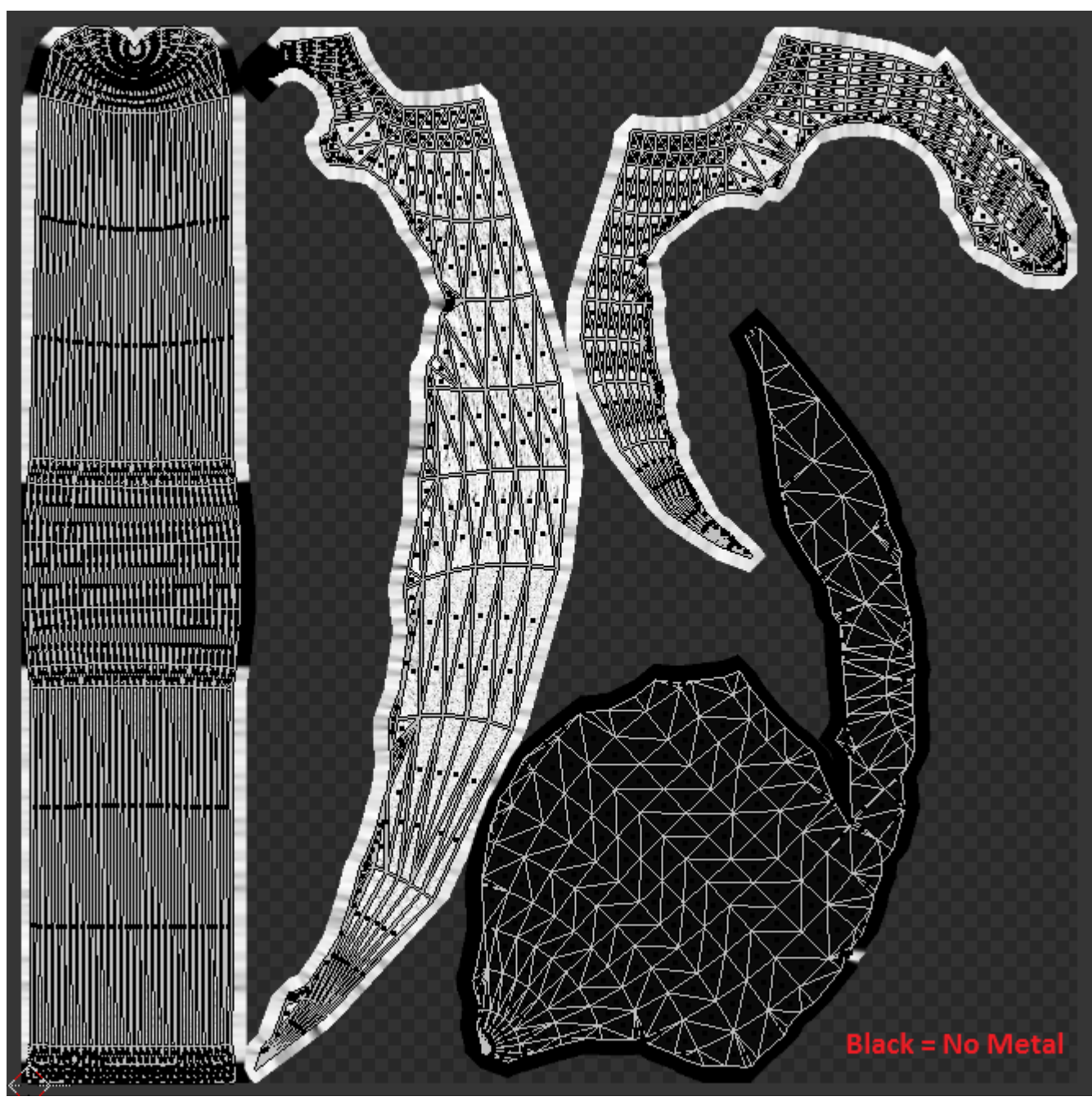
- If everything looks good, save it as described in the last part of step 12.
14) Next we will be baking the Metalness Map. Detach the Image Texture node for your Metalness Map from the Principled BSDF node. Do the exact same steps you did in Step 13, except name the new image to bake to bakeMetalMap, and connect the Image Texture node with your Metalness Map to the ColorRamp node instead. Once everything is set up correctly click Bake. See image below.


- If you do not have a Metalness map in some of your materials, click Add, hover your mouse cursor over Input, and click RGB, In the RGB node set the color to black. Connect the small yellow dot labeled Color in the RGB node to the small yellow dot labeled Color in the Emission node. Exclude the ColorRamp node. Remember, if you don't have a Metalness map in any of your materials you do not have to bake a Metalness map period.


- Below is an example of how a Metalness map properly baked on to a UV Map is supposed to look like. Yours should look similar. Black parts mean that this part is not metallic. Metallic parts will be different shades of grey up to white, which represent different types of metals.

Black = No Metal

- If everything looks good, save it as described in the last part of step 12.

15) Next we will be baking a Glossiness map, Civilization VI is setup to take a Glossiness map so if you have a Roughness map instead, we will be turning it in to a Glossiness map by running it through an Invert node. Detach the Image Texture node for your Roughness Map from the Principled BSDF node. Click Add, hover your mouse cursor over Color, and select Invert. Again, do the exact same steps you did in Step 13, except name the new image to bake to bakeGlossMap, and connect the Image Texture node with your Roughness Map to the Invert node, then to the ColorRamp node. Once everything is set up correctly click Bake. See image below.
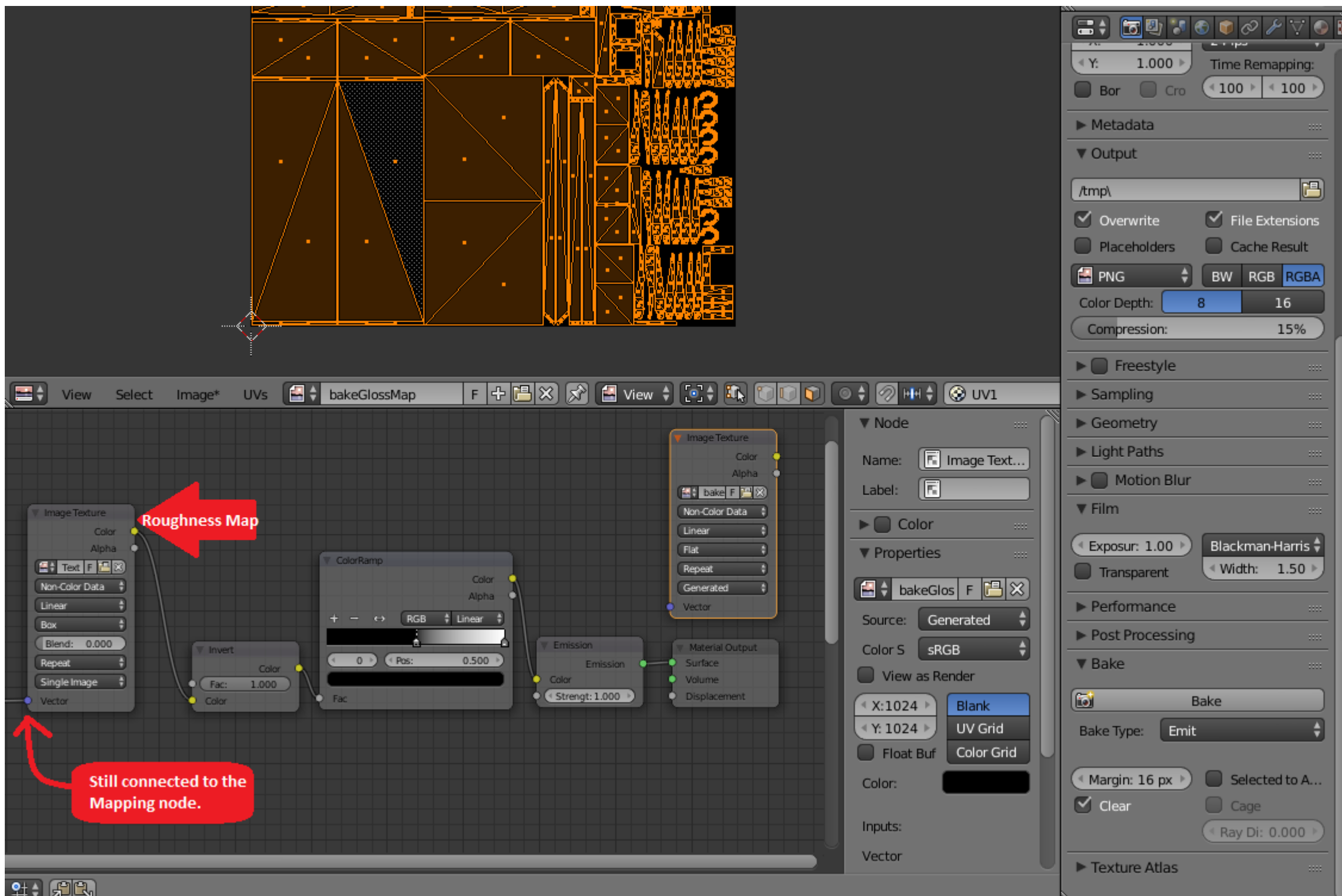

- If you just have a Glossiness map, detach the Image Texture node for your Glossiness Map from the Invert node, and just do same as above minus the Invert node by connecting your Image Texture node for your Glossiness Map directly in to the ColorRamp node. Once everything is set up correctly click Bake. See image below.
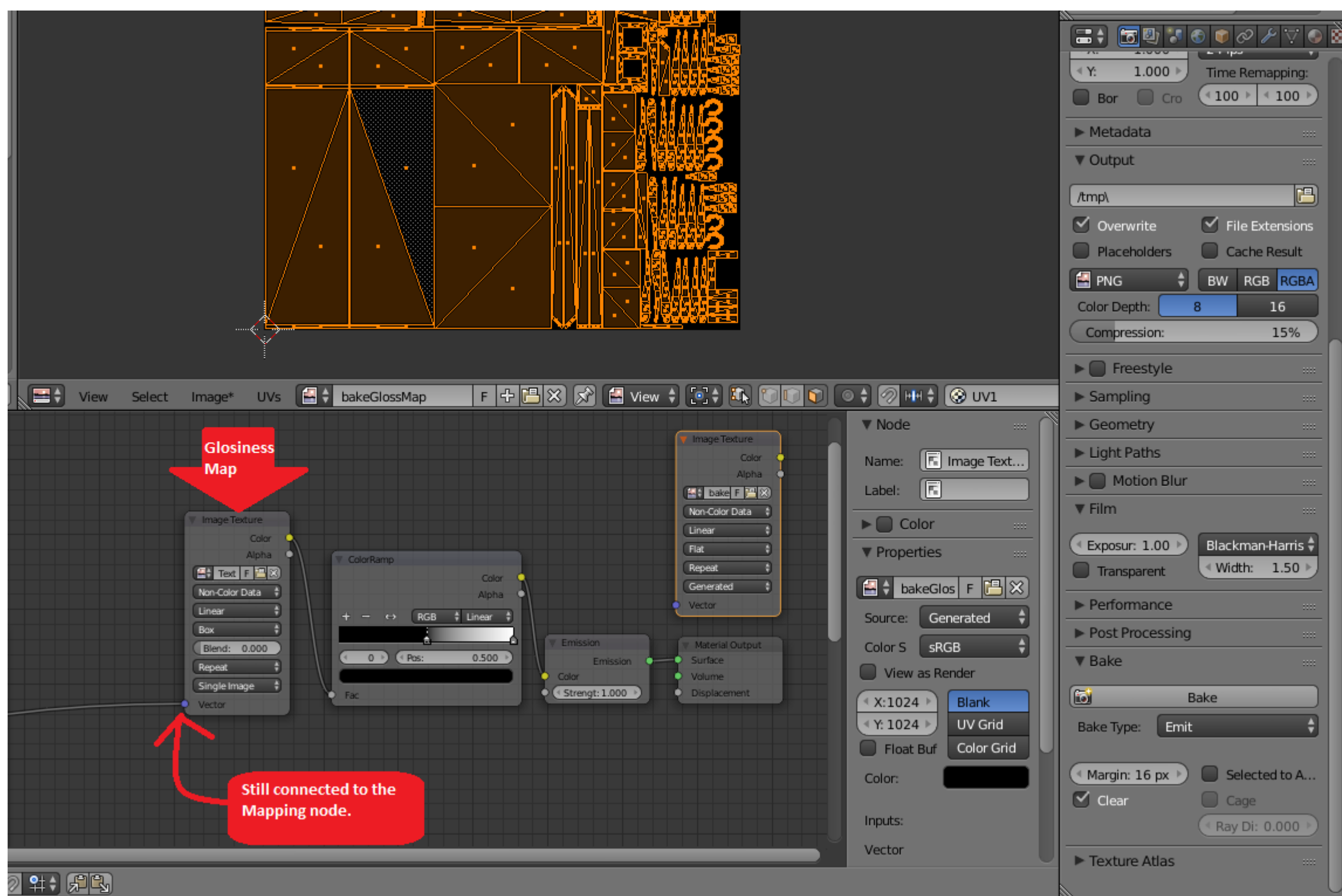
- If you had both a Roughness map and a Glossiness map, do the same steps as above, except disconnect the MixRGB (Screen) node from the Principled BSDF node, and connect it to the ColorRamp Node, then swap the Invert node so that the Roughness map is going through it instead, with the Glossiness map going straight in to the MixRGB (Screen) node. Once everything is set up correctly click Bake. See image below.
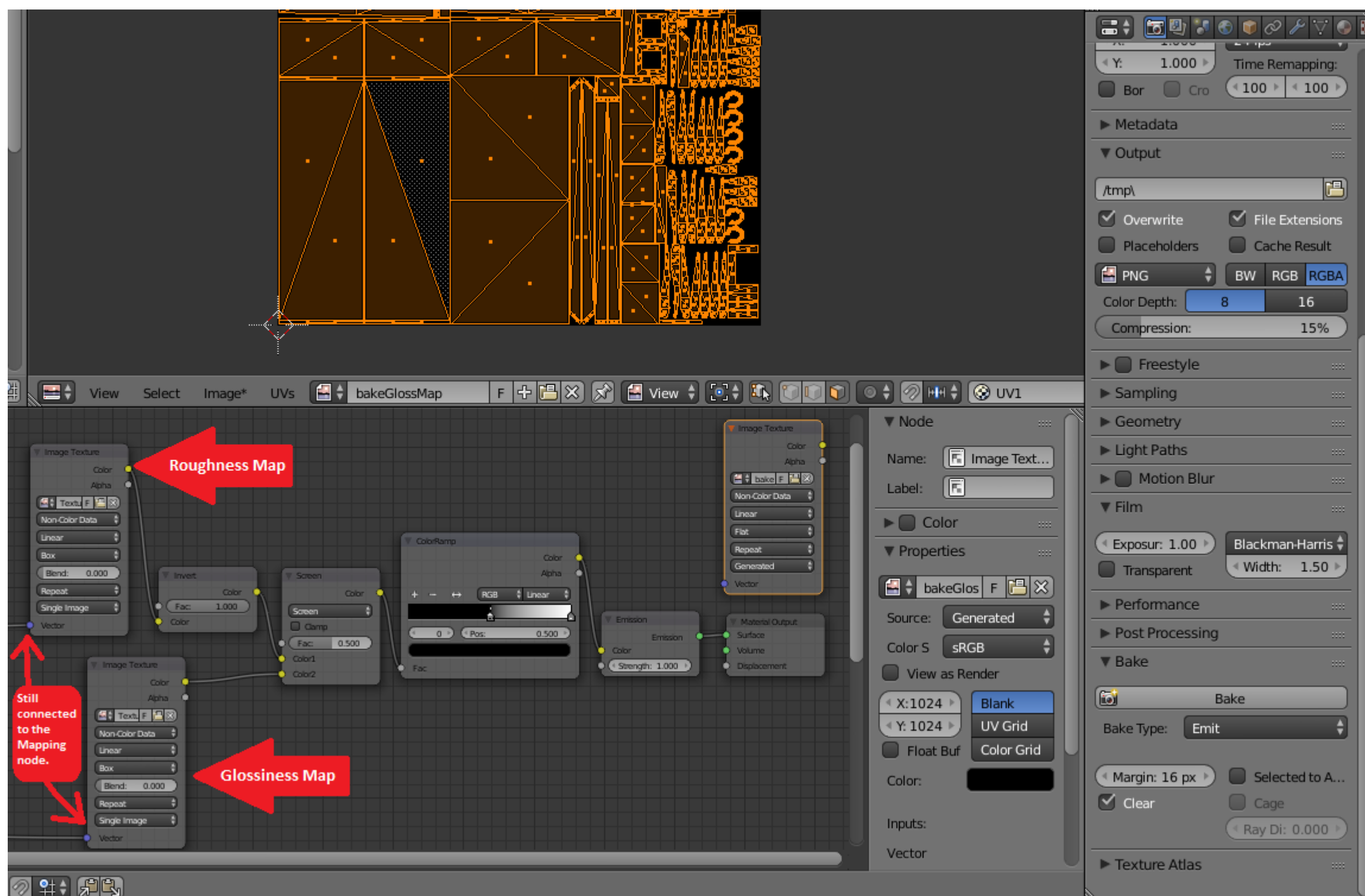
- If you do not have a Glossiness/Roughness map in some of your materials, click Add, hover your mouse cursor over Input, and click RGB, In the RGB node set the color to black. Connect the small yellow dot labeled Color in the RGB node to the small yellow dot labeled Color in the Emission node. Exclude the ColorRamp node. Remember, if you don't have a Glossiness/Roughness map in any of your materials you do not have to bake a Glossiness map period. See image below.
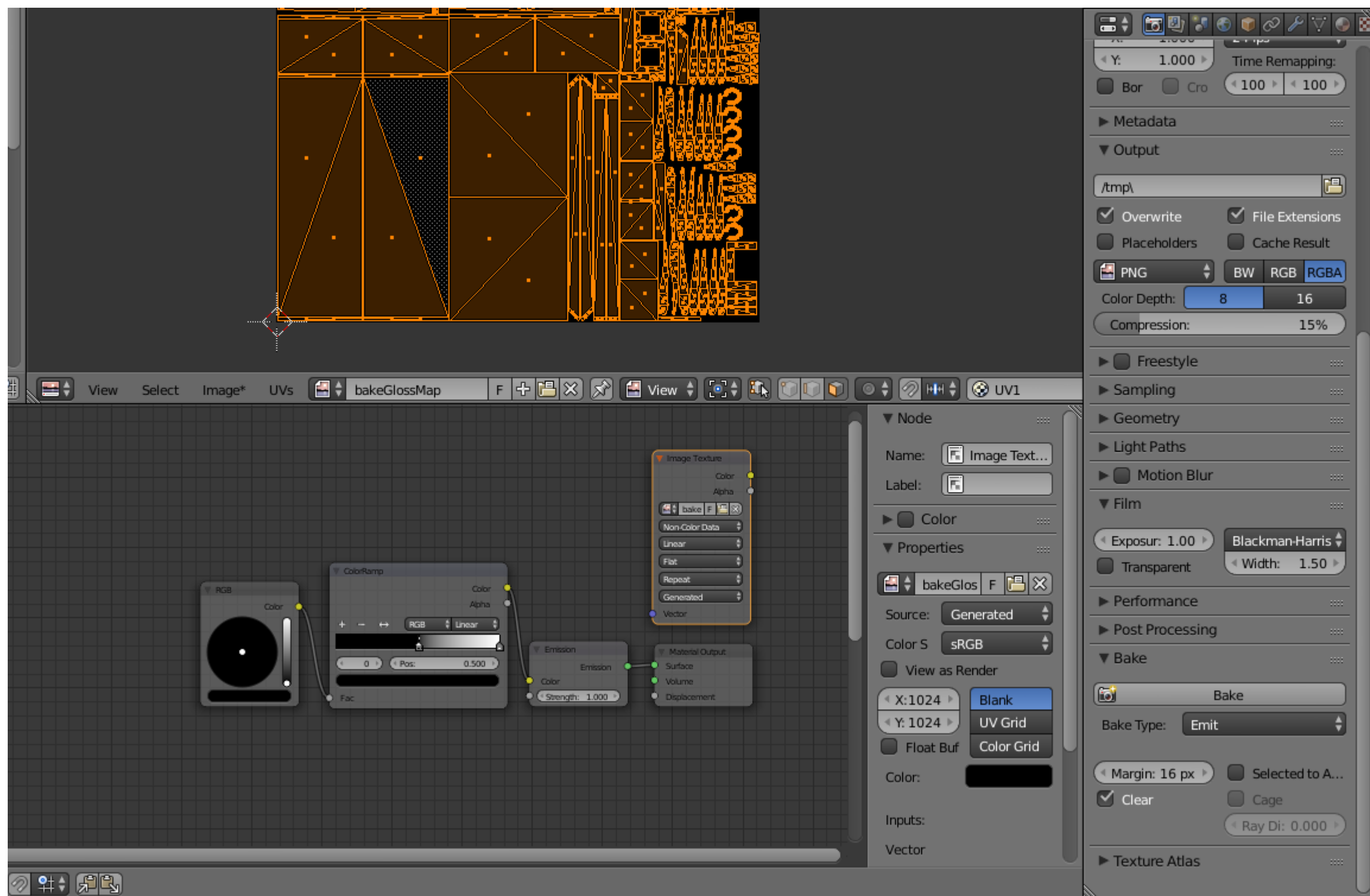
- Below is an example of how a Glossiness map properly baked on to a UV Map is supposed to look like. Yours should look similar. Black parts mean that this p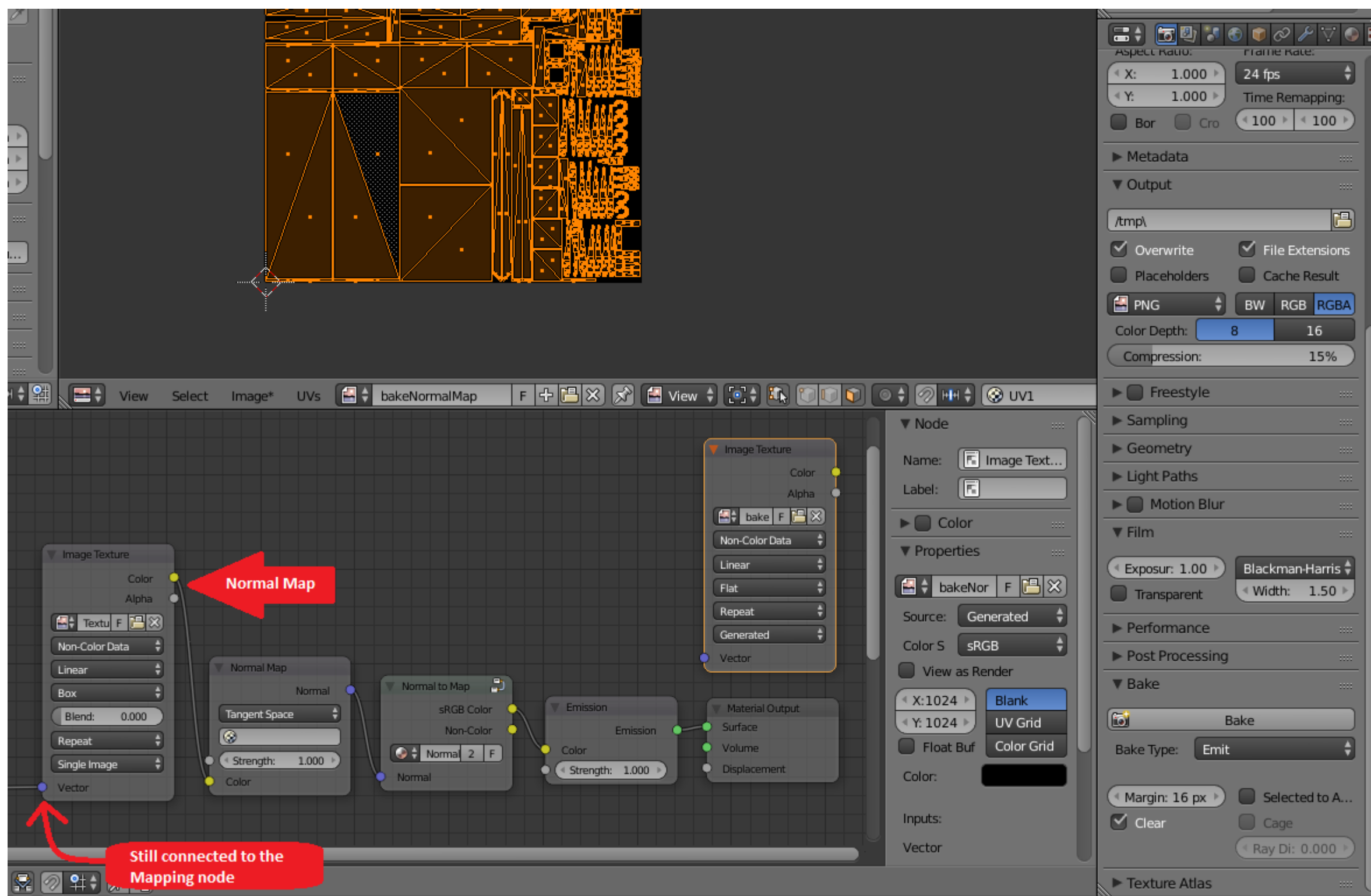art is not glossy/reflective. Glossy/reflective parts will be different shades of grey up to white, which represent how glossy/reflective it is, white being extremely glossy/reflective.

- If everything looks good, save it as described in the last part of step 12.

16) Finally we need to bake our Normal Map, Civilization VI is setup to take a Normal Map, so if you have Height map we will be turning it in to a Normal map. First we need to append that Normal to Map node group you downloaded. In the 3D View panel change to Object mode. On the very top menu bar, click File and select Append. Navigate to \Downloads\normal_to_map.blend\NodeTree, select Normal to Map and click Append From Library.
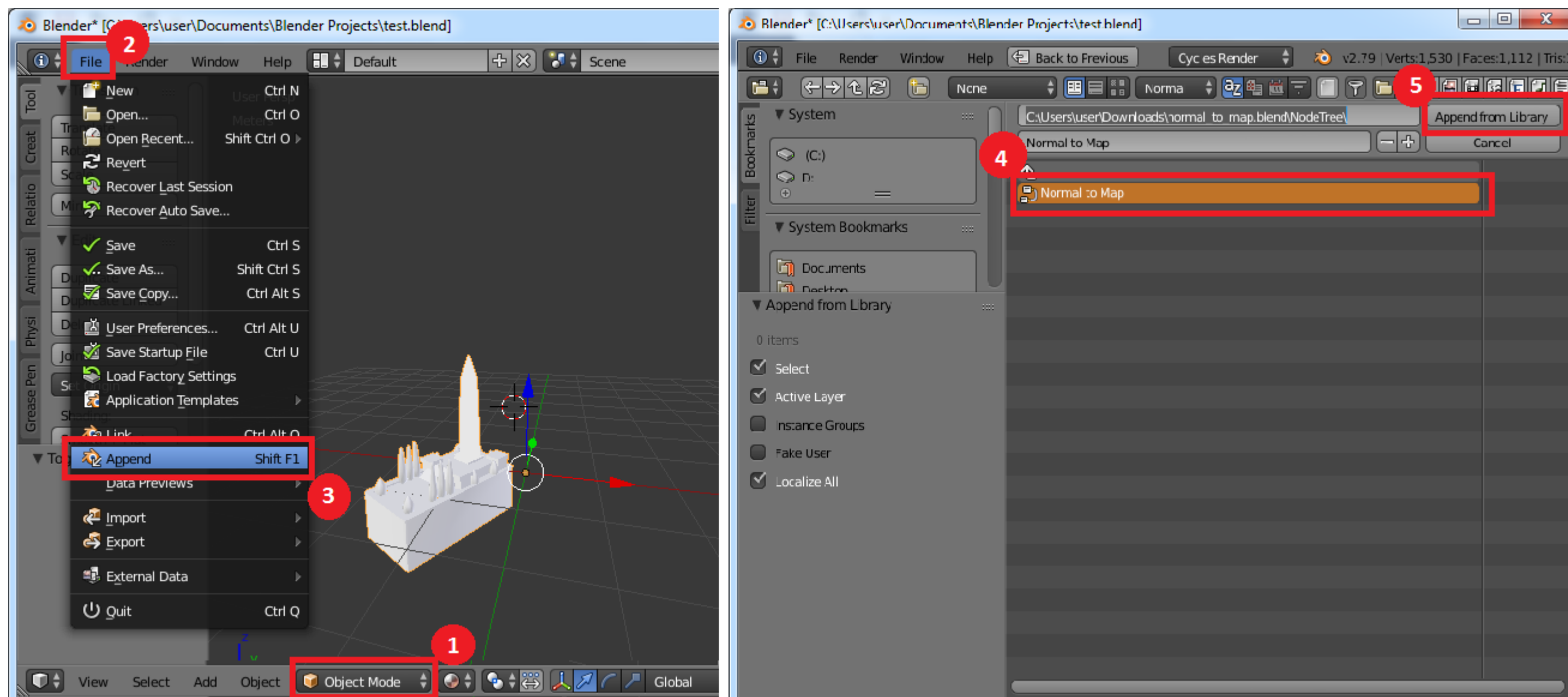
- If you just have a Normal Map, detach the Normal Map node from the Principled BSDF node, do the exact same steps you did in Step 13, except you will substitute the ColorRamp node for the Normal to Map node. Click Add, hover your mouse cursor over Group, and select Normal to Map (which will be there is you appended it correctly). Connect the small yellow dot labeled sRGB Color in the Normal to Map node to the small yellow dot labeled Color in the Emission node. Then connect the small blue dot labeled Normal in the Normal Map node to the small blue dot labeled Normal in the Normal to Map node. Your Image Texture node for your Normal Map should still be connected to the Normal Map node and to the Mapping node. Once everything is set up correctly click Bake. See image below.

- If you just have a Height Map, detach the Bump Map node from the Principled BSDF node, do the exact same steps you did in Step 13, except you will substitute the ColorRamp node for the Normal to Map node. Click Add, hover your mouse cursor over Group, and select Normal to Map (which will be there is you appended it correctly). Connect the small yellow dot labeled sRGB Color in the Normal to Map node to the small yellow dot labeled Color in the Emission node. Then connect the small blue dot labeled Normal in the Normal Map node to the small blue dot labeled Normal in the Normal to Map node. Your Image Texture node for your Height Map should still be connected to the Bump node and to the Mapping node. Once everything is set up correctly click Bake. See image below.

- If you have both a Normal Map and a Height Map, detach the Bump Map node from the Principled BSDF node, do the exact same steps you did in Step 13, except you will substitute the ColorRamp node for the Normal to Map node. Click Add, hover your mouse cursor over Group, and select Normal to Map (which will be there is you appended it correctly). Connect the small yellow dot labeled sRGB Color in the Normal to Map node to the small yellow dot labeled Color in the Emission node. Then connect the small blue dot labeled Normal in the Normal Map node to the small blue dot labeled Normal in the Normal to Map node. Your Image Texture node for your Height Map should still be connected to the Bump node and to the Mapping node, and your Image Texture node for your Normal Map should still be connected to the Normal Map node and to the Mapping node, and the Normal Map node should still be connected to the Bump node. Once everything is set up correctly click Bake. See image below.
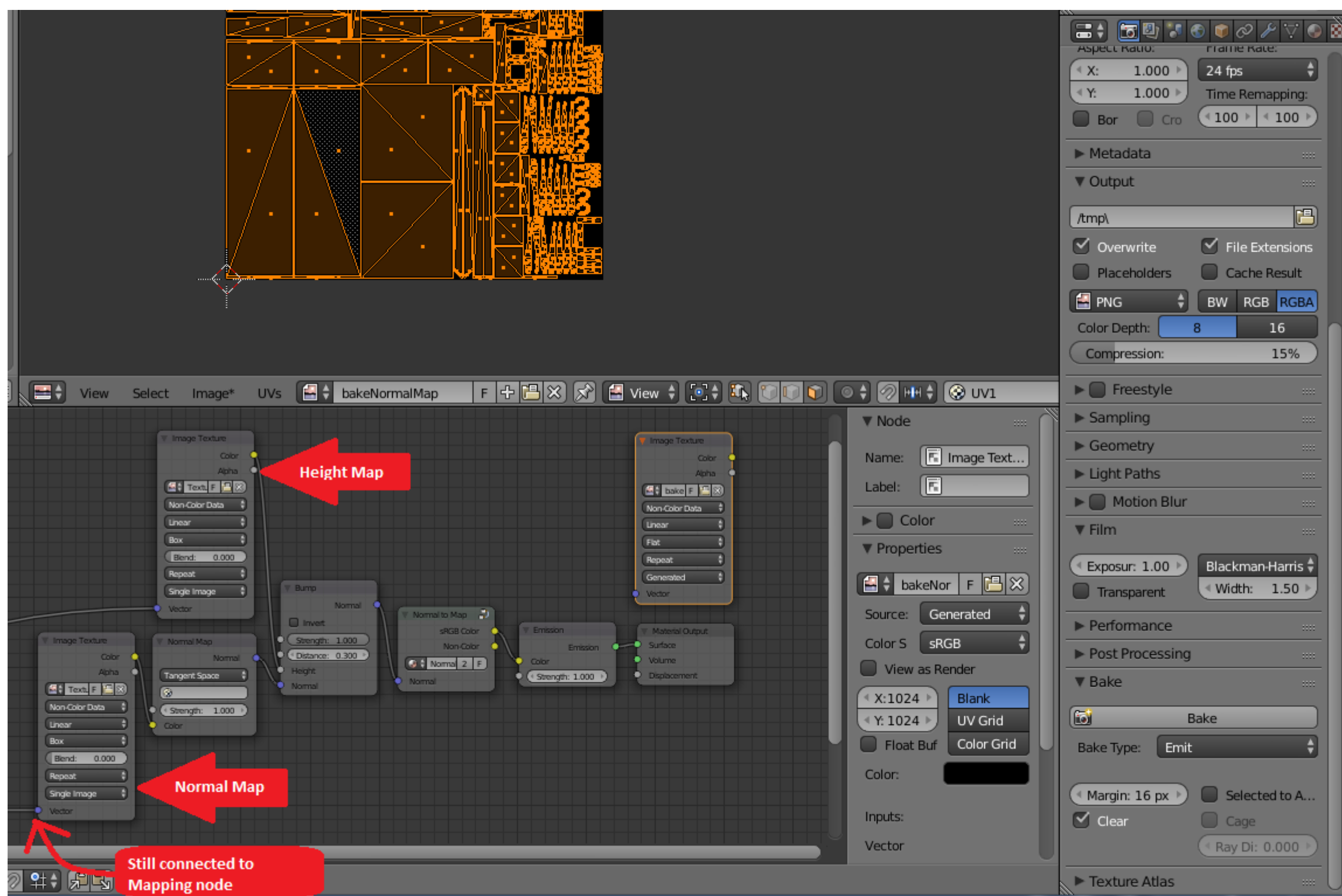
- If you do not have a Normal/Height Map in one of your materials, do the same steps as described above, except you will just connect the Normal Map node to the Normal to Map node. See image below.
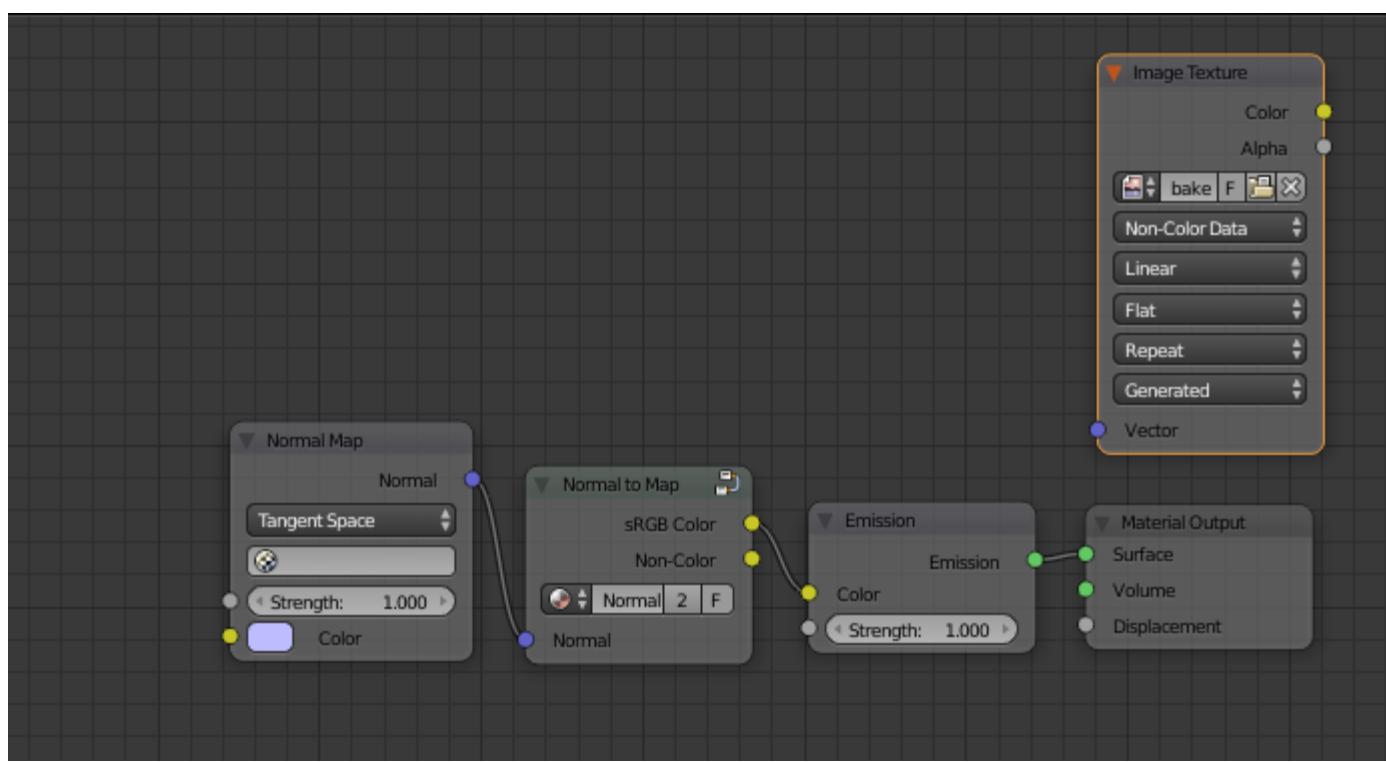
- Below is an example of how a Normal map properly baked on to a UV Map is supposed to look like. Yours should look similar.
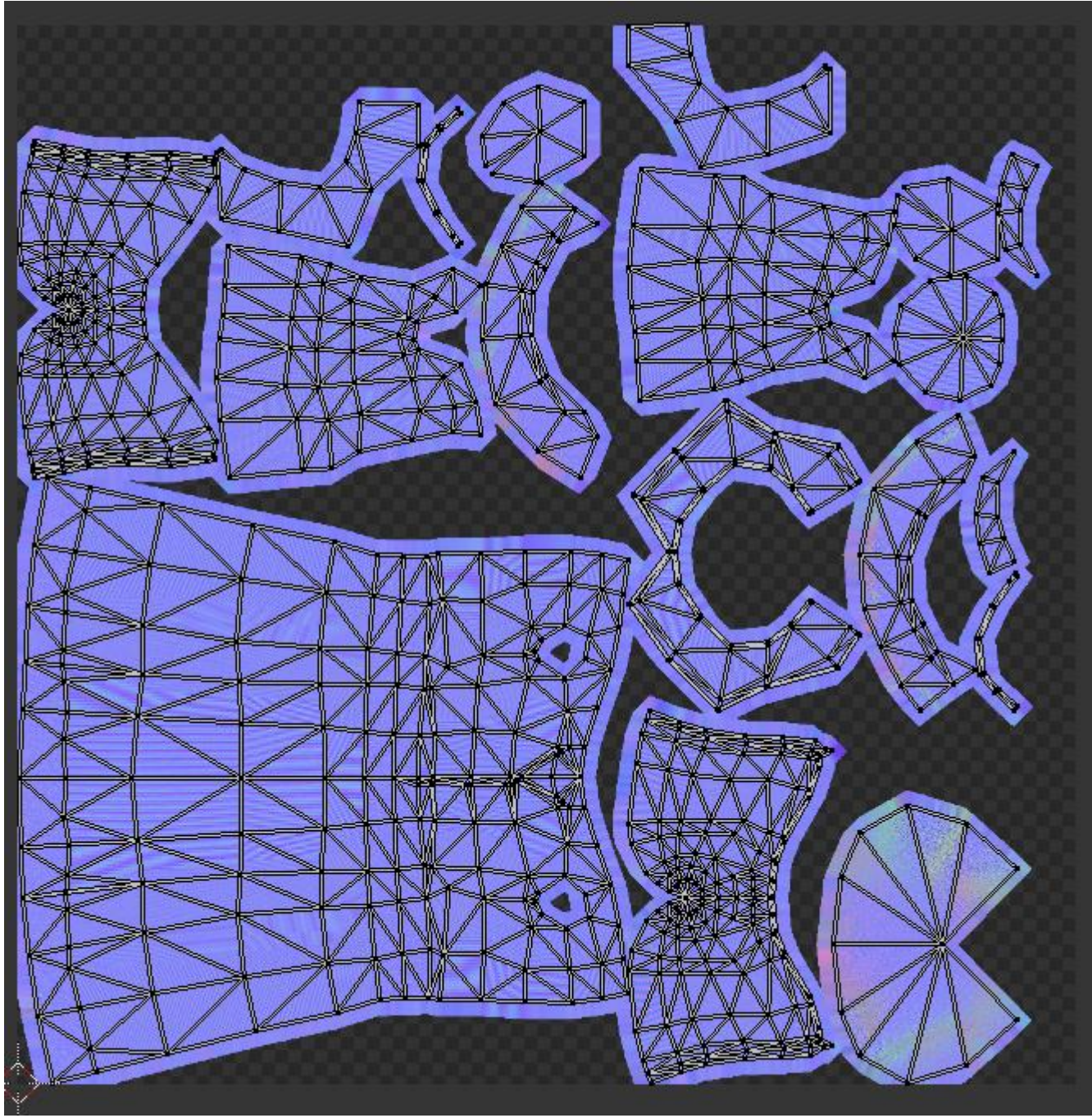

- If everything looks good, save it as described in the last part of step 12.

17) Now that you have all of your textured UV maps for your 3d Model (Mesh), you need to convert them in to a format that the game engine can use. Open the Gimp application. You could use any image editing software, you will just need a DDS format plugin for it. In this guide we will be using Gimp. The DDS plugin should already be setup if you followed the steps in the very beginning of this guide. Click File, select Open, navigate to \Documents\Blender Projects\Textures, select bakeColorMap.png, and click Open.
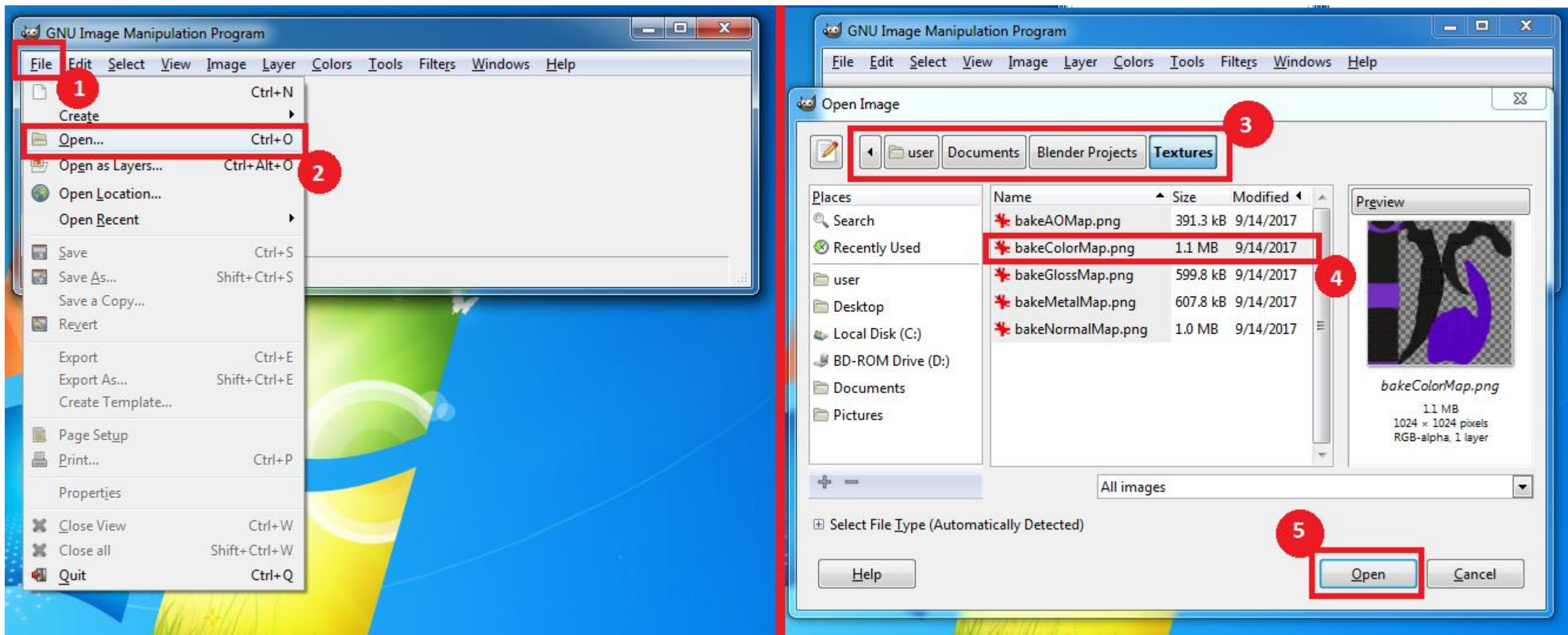
18) In Gimp, click File, select Export As, make sure the Save in Folder is \Documents\Blender Projects\Textures, click the plus sign next to Select File Type (By Extension), under the File Type column select DDS Image, rename similar to how you named your Project in Blender, but with the text ColorMap somewhere in there. For example if my Blender Project is named DIS_CTY_ESO_MagesGuild, I would name this image file ESOMagesGuildColorMap. ESO for esosorcdc, MagesGuild for the building's name, and ColorMap for the PBR map type. Click Export, leave compression on None, set Format to RGBA8, check the check box for Generate mipmaps, click OK. Now click File and select Close All. Repeat for all of your baked images. Note: If a window pops up saying that you have unsaved changes, just click Discard Changes.
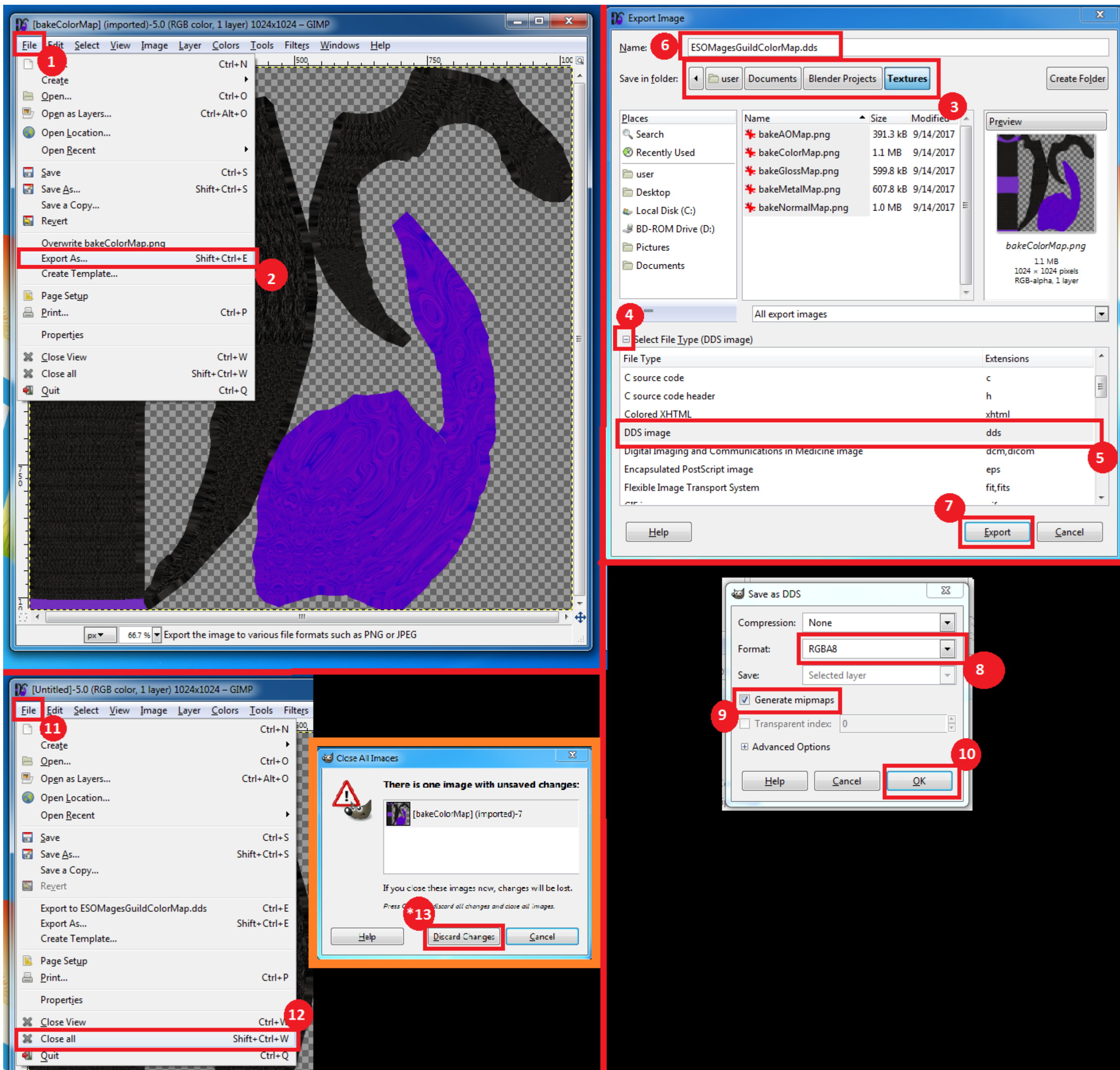
19) Open the Blender application and open your project, if you don't have it open already. In Object Mode select your 3D Model (Mesh). Under the Properties Menu click Material. Delete all of your Materials except one. Remember, to delete a material, click a Material to select it, and press the button to the right with the minus sign. You could delete all of your materials and create a new material, but it will save you time setting up nodes if you just reuse a Material you already have setup. Change the name of the material to the same name you named your Blender project except add Material at the end. For example, if my Blender Project is named DIS_CTY_ESO_MagesGuild, I would name my material DIS_CTY_ESO_MagesGuild_Material. In the 3D View panel, with your 3D Model (Mesh) selected, change mode to Edit Mode. Make sure your entire 3D Model (Mesh) selected. Under the Properties Menu click Material, if you are not already under Material, make sure your material is selected and press Assign.

Click material name to select material (selected material will be highlighted in blue), then click the button with the minus sign to delete.
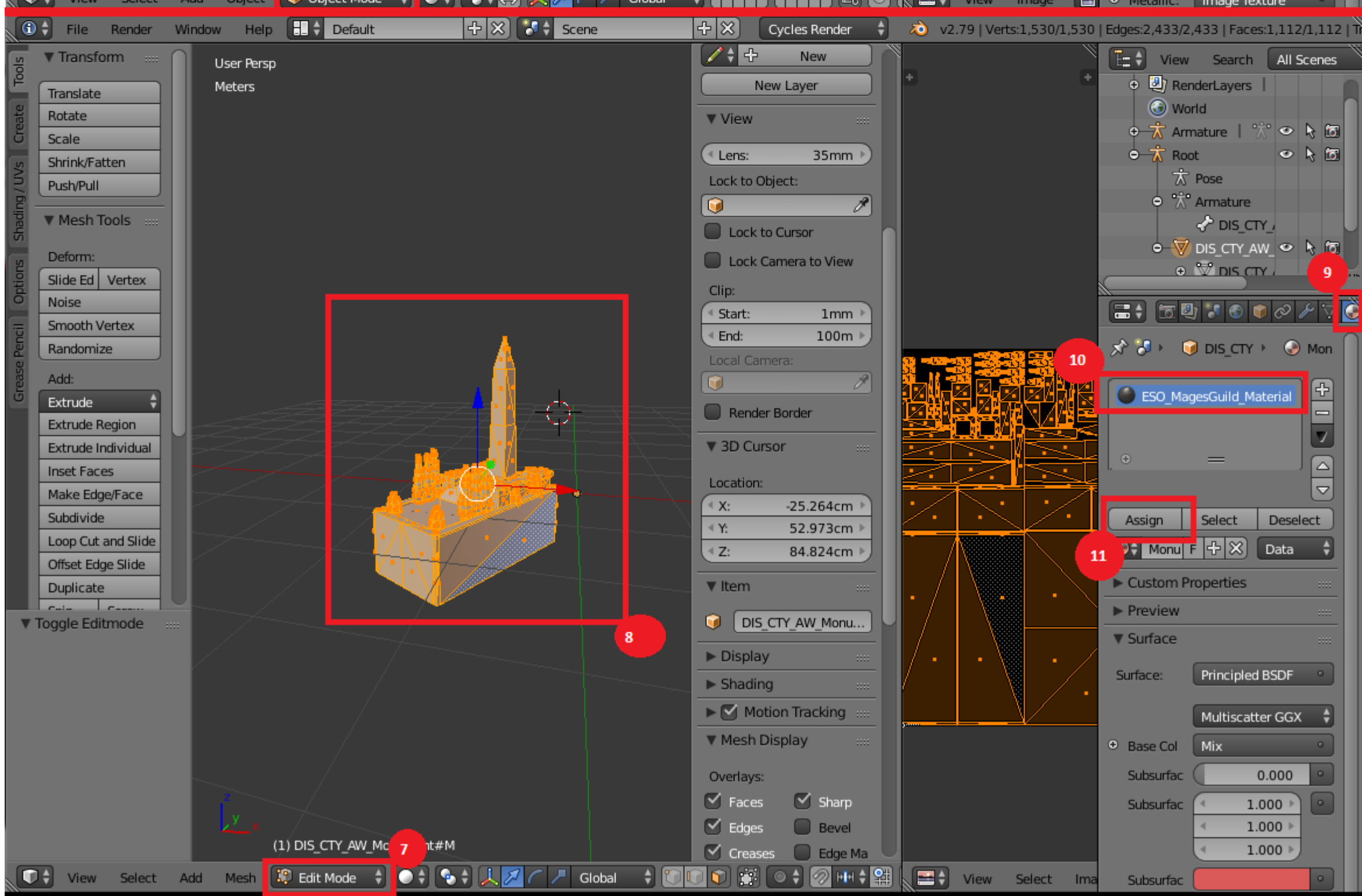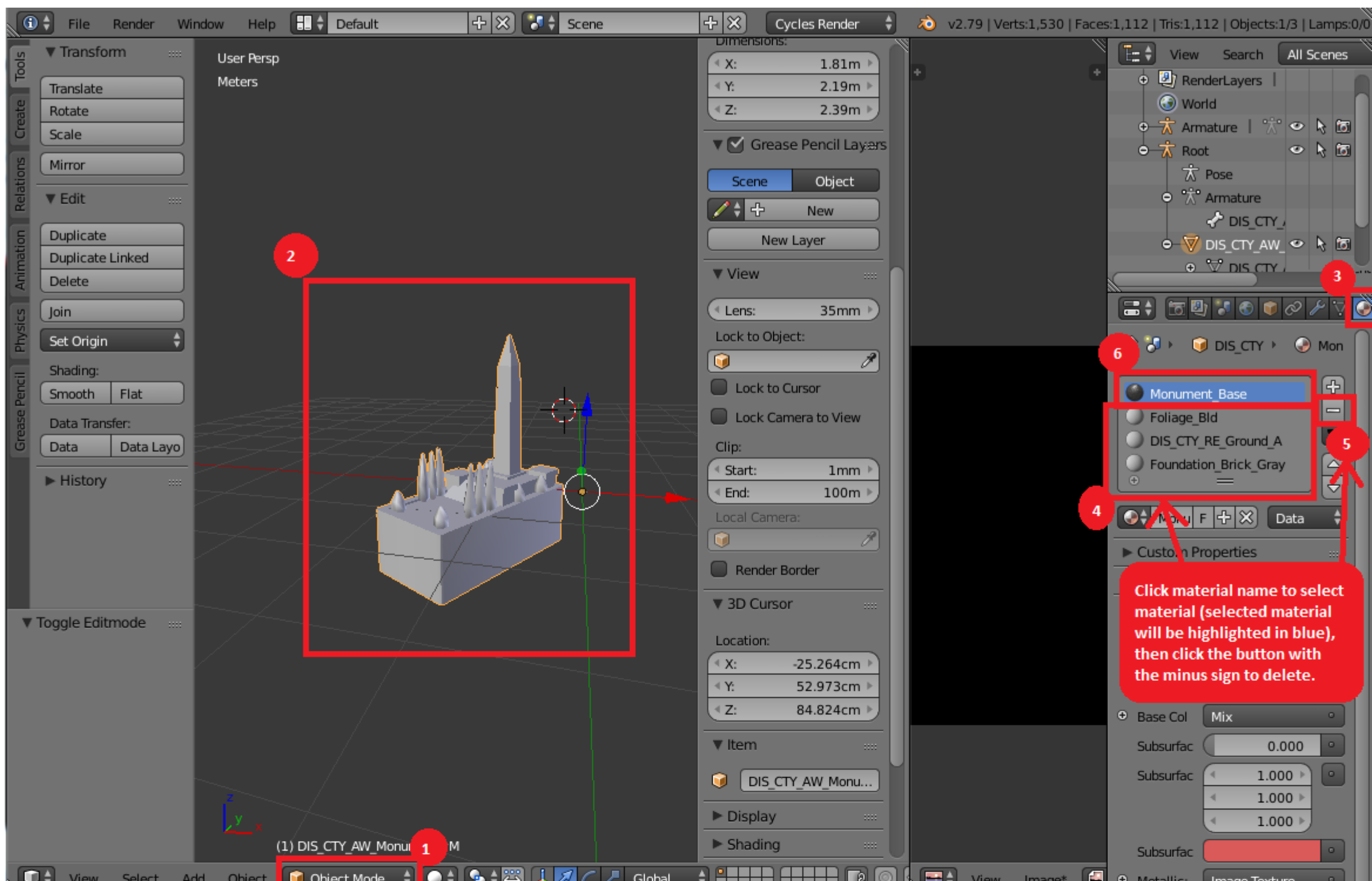
• Make sure your 3D Model (Mesh) is selected in the 3D View panel. Under the Properties Menu click Object. Click the text field to the right of the small black button with the tan cube on it, type the same name you used for your material minus the word Material, instead type Mesh. So referencing my example above, I would type DIS_CTY_ESO_MagesGuild_Mesh.
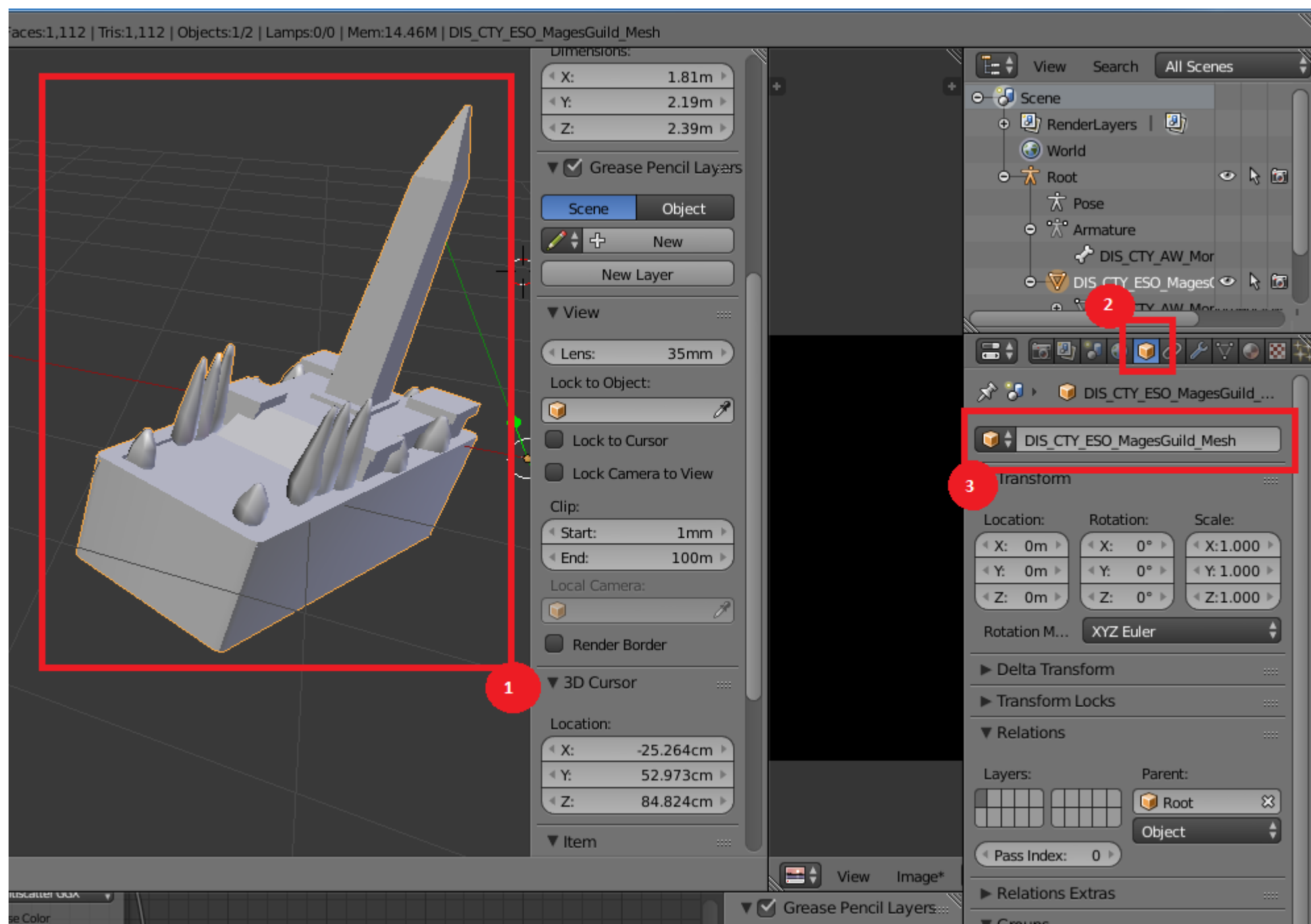
- Note: We specifically name our Material and Mesh because later we will use these names in ModBuddy and it will make identifying the Material and Mesh easier.

20) Your 3D Model (Mesh) is now ready to be exported. However, if you want to check how your baked textures look on your 3D Model (Mesh) this is how. In 3D View panel change mode to Object Mode and change your Method to Display Objects to Material. Under the Properties Menu click Material, make sure your material is selected. In the Node Editor panel, you will set up your nodes similar to the setup you had in step 12 when you baked your Color Map, except you need to delete some nodes, change the Method to Project setting in all of your Image Texture nodes to Flat (third black button from the top), add the Invert node to run your Glossiness map through (if you don't already have one), add the Normal Map node to run your Normal Map through (if you don't already have one). Add the UV Map node, click Add, hover your cursor over Input, and select UV Map. In the UV Map node click the grey button with the checkerboard circle and select your UV map, and connect it to all of the Vector inputs in all of your Image Texture nodes. If you imported a base game 3D Model (Mesh), be sure to select the UV map you unwrapped your 3D Model (Mesh) to, which will more than likely be called UV1. See Image below for setup.
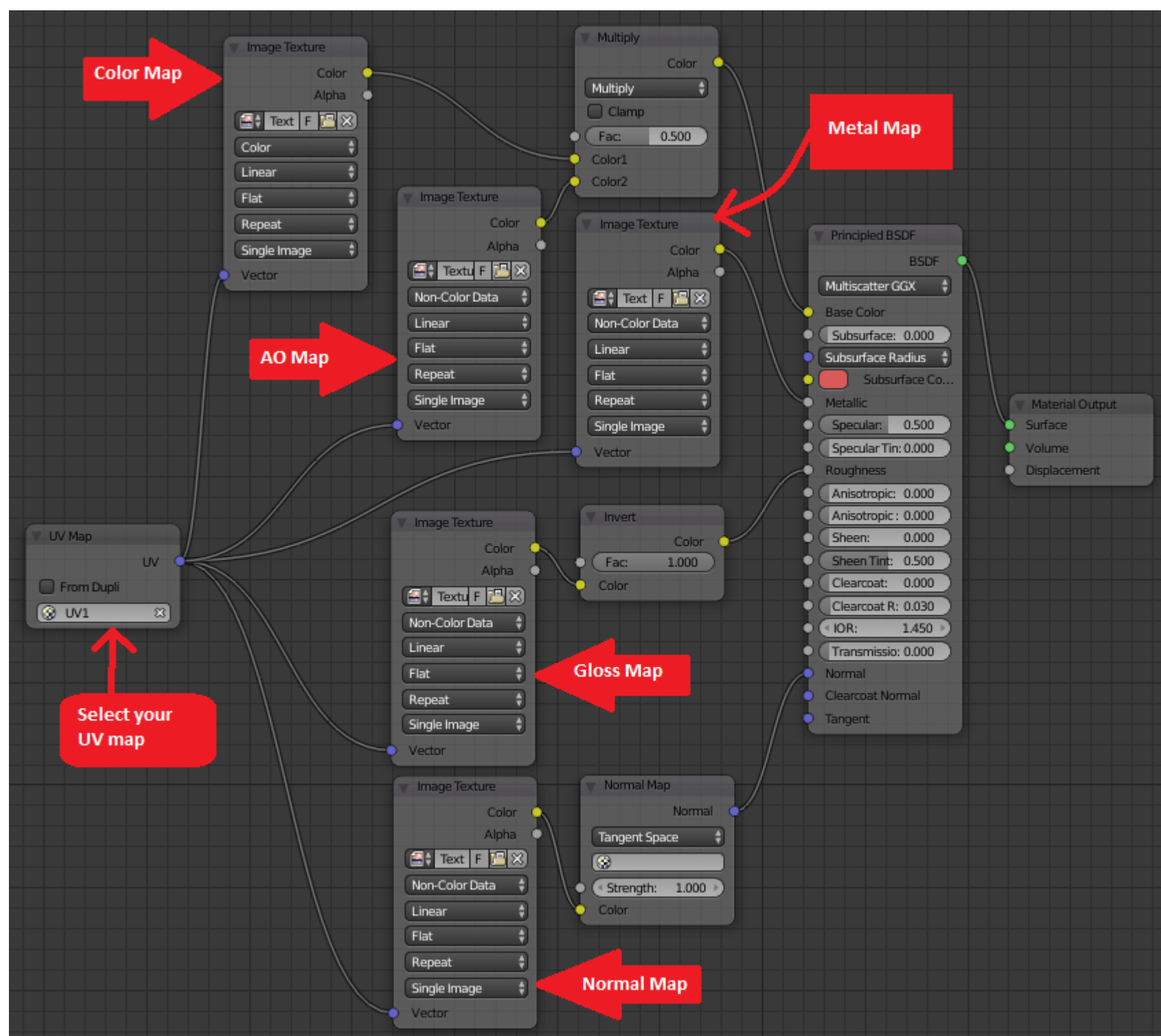
- If everything looks good, save your Blender project, and continue to exporting.

21) Lastly we need to export to CivNexus6 format (.cn6), click File, hover your cursor over Export, select CivNexus6 (.cn6), navigate to Documents\Blender Projects\CivNexus6, and click the Export CN6 button. Check the top menu bar for any error messages, if you don't see anything then your export was successful, if you see an error message then it did not export properly because there is an issue with your project.
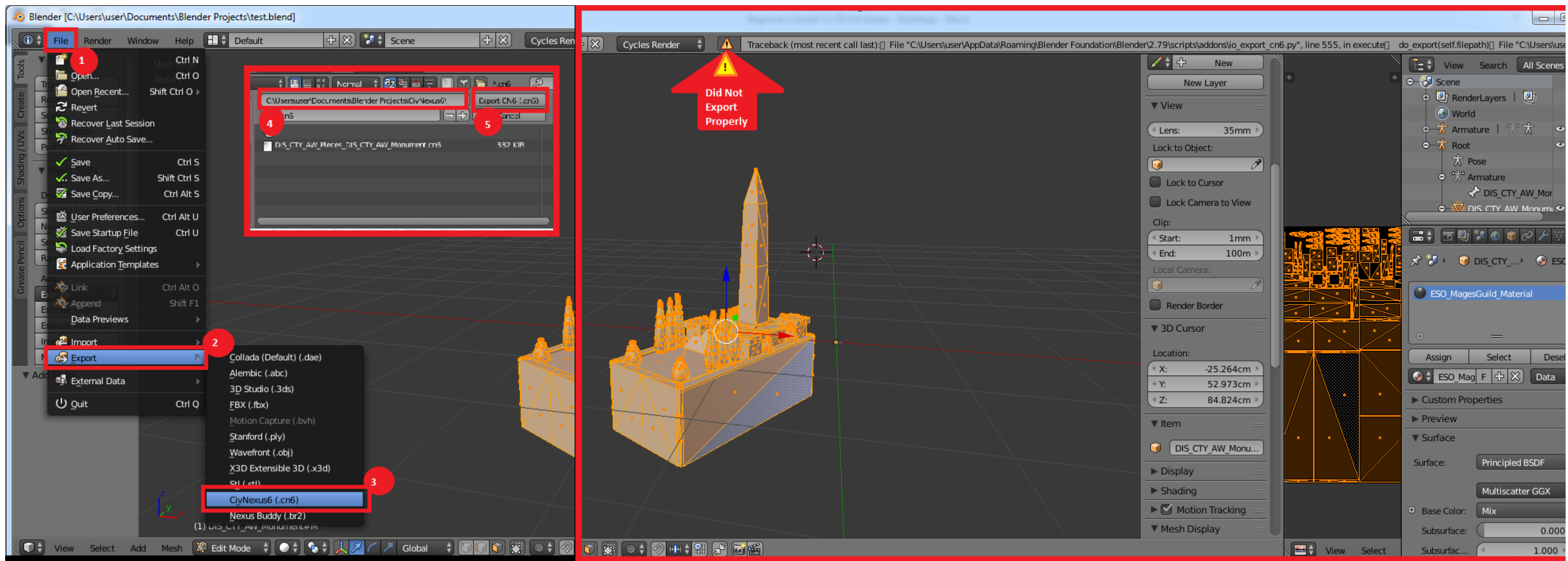
- If it did not export properly, 9 times out of 10 (or at least for me) it has something to do with the bones/armature. Just clear parent and re-parent the 3D Model (Mesh) and the bone, see Section 2 Steps 12 and 13 on how to do this. Save, and try to export again.
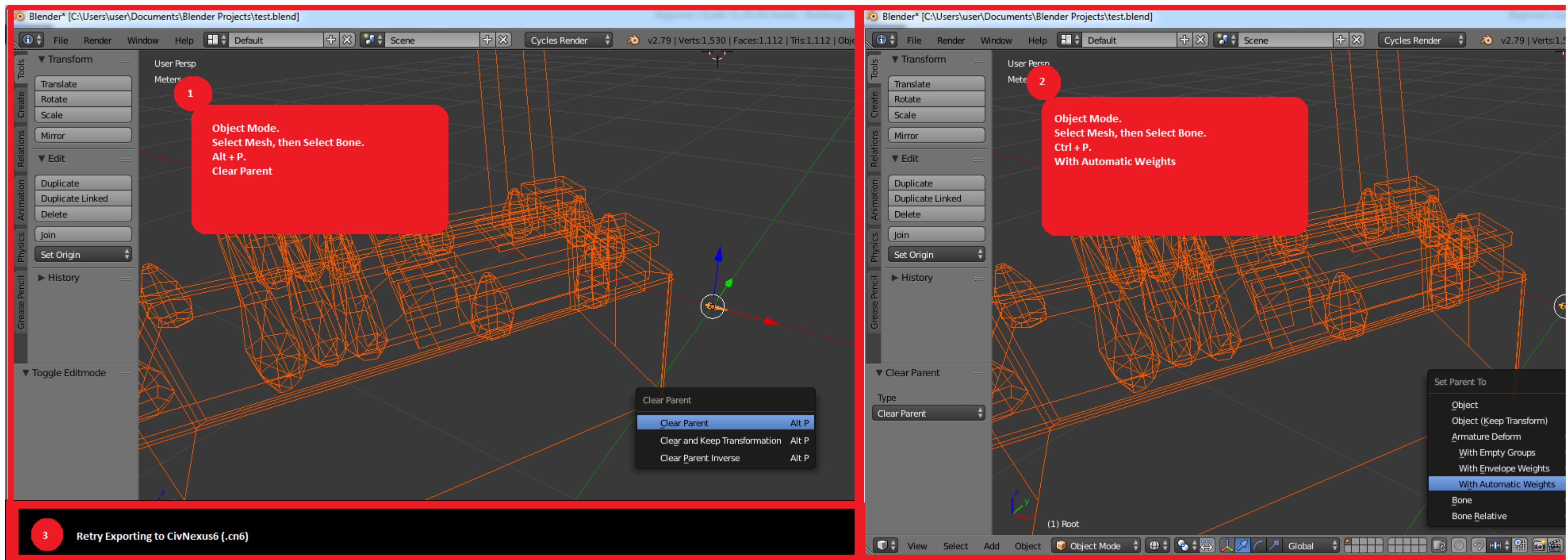
22) And that concludes Section 3, you now have all of the textured UV maps for your 3D Model in DDS format, and your 3D Model has been exported to CivNexus6 format.

Section 4   Creating and Getting Other Files You Need

In this section we will be creating other files you will need, copying a couple files from the pantry, and putting these files in to your ModBuddy project folders. The pantry is the other tool you downloaded from Steam called Sid Meier's Civilization VI SDK Assets. This is a folder that contains base game art assets that you can use. I will also talk about the Obstruction Profile at the end of this section.

1) Open the CivNexus6 application.
   - C:\Program Files (x86)\CivNexus6_1.2.3\CivNexus6.exe
   - Note: The folder you have CivNexus6 in may have a different name because you may have a newer version.
2) In CivNexus6, click the Import button, navigate to \Documents\Blender Projects\CivNexus6, select the CN6 format file of your 3D Model that you exported out of Blender in section 3, and click Open.
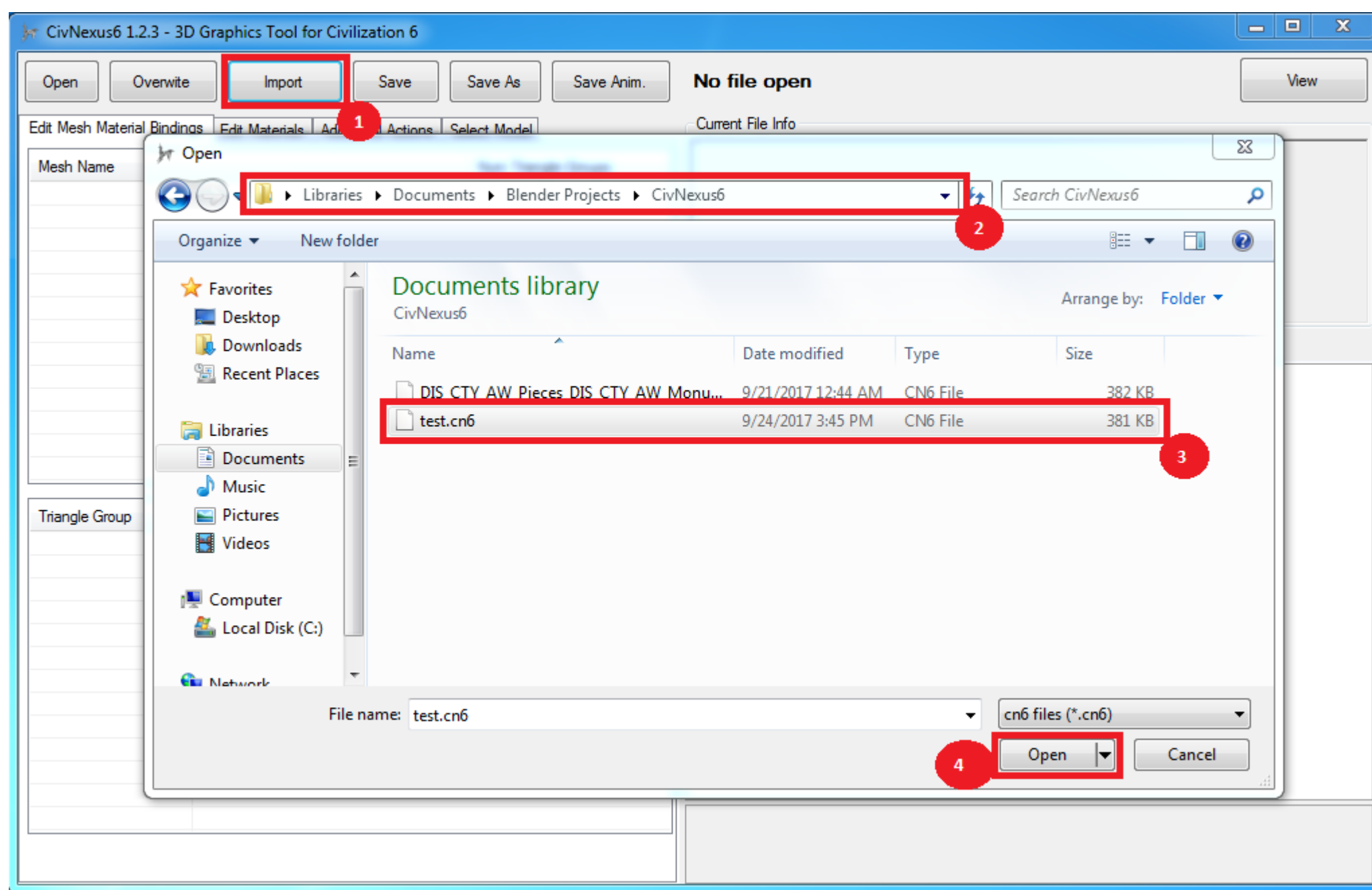
- Note: If you get an error pop up that says you have unsigned vertices, then you need to load your project back up in Blender (the .blend, not the .cn6) and make sure all of your vertices are assigned to your vertex group, then re-export it to CN6 format and try importing it in CivNexus6 again. Refer to Section 2 Step 14 on assigning vertices to a vertex group.
- Note: If you get an error pop up that mentions bones, armatures, or parenting, then you need to load your project back up in Blender (the .blend, not the .cn6), clear parenting between your 3D Model (Mesh) and bone, re-parent your 3D Model (Mesh) and bone, then re-export it to CN6 format and try importing it in CivNexus6 again. Refer to Section 2 Step 12 on clear parenting and Section 2 Step 13 on parenting.

3) In CivNexus6, click the Additional Actions tab, click the drop down menu labeled Class and select LandmarkModel, click the Save button. This will create the .fgx file for your 3D Model. Now click the button labeled Create Geometry Set xml for .ast file. This will create the GeometrySet.xml file for your 3D Model. Now click the button labeled Create Geometry/Animation (.geo/.anm) File. This will create the .geo file for your 3D Model. Tip: If you click the View button, another window will pop up that will give you an un-textured preview of your 3D Model (Mesh). If you see faces missing then you forgot to triangulate your 3D Model (Mesh).
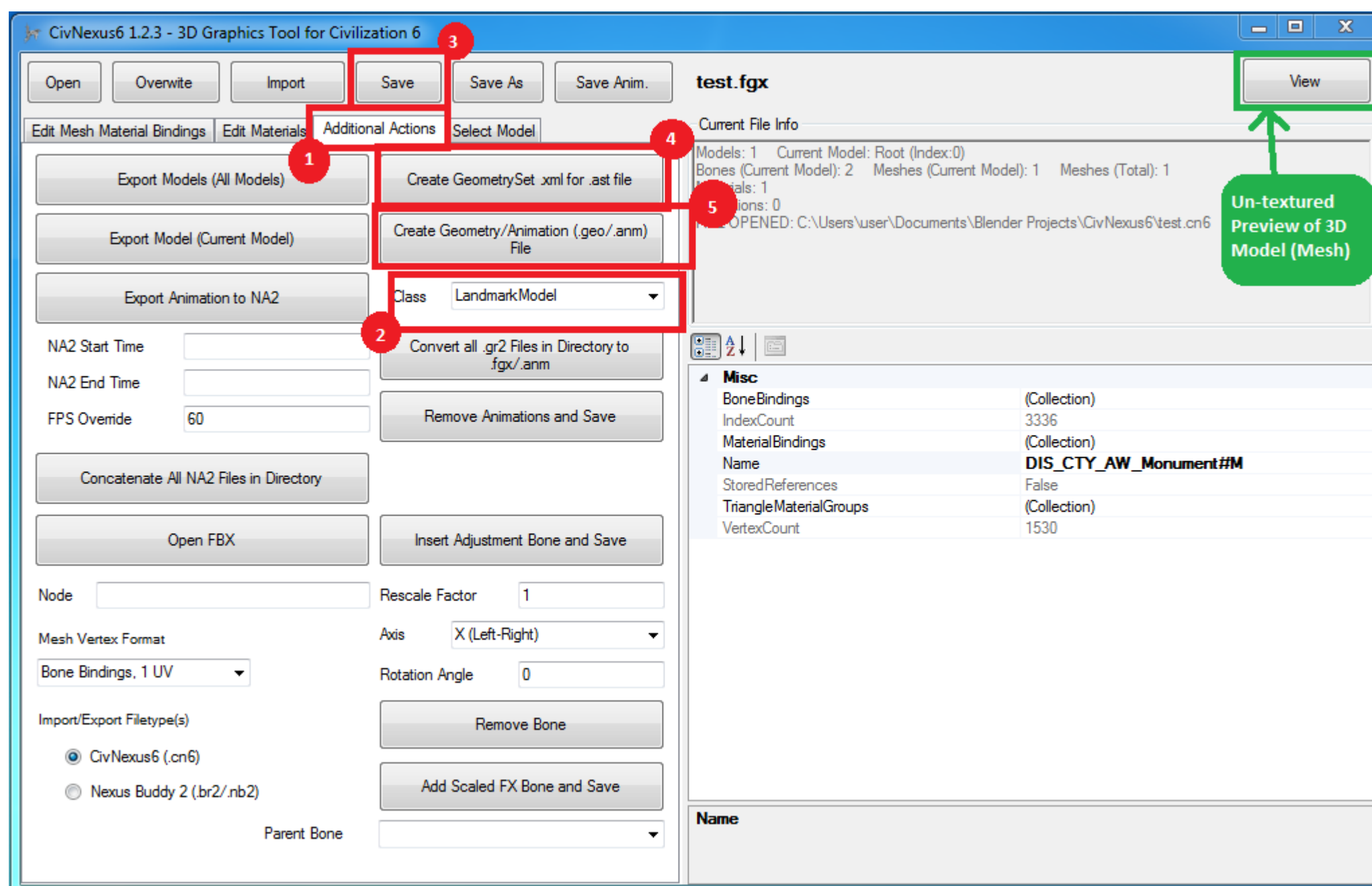
- Note: All three of the files created above were saved in the same folder as the CN6 file that you imported in step 2 (\Documents\Blender Projects\CivNexus6).

4) In Windows Explorer navigate to \Documents\Blender Projects\CivNexus6, select the three files you created in the step above, copy and paste them in to the Geometries folder of your ModBuddy Project, which should be located in \Documents\Firaxis ModBuddy\Civilization VI\YourProject\YourProject\Geometries. Create the Geometries folder if not already there. Navigate to the Assets folder in the pantry C:\Program Files (x86)\Steam\steamapps\common\Sid Meier's Civilization VI SDK Assets\pantry\Assets. Find a base game building's asset file, copy and paste it in to the Assets folder of your ModBuddy Project, which should be located in \Documents\Firaxis ModBuddy\Civilization VI\YourProject\YourProject\Assets. Then rename it the same name as your Blender project except add Assets at the end. For Example, DIS_CTY_ESO_MagesGuild would become DIS_CTY_ESO_MagesGuild_Asset. Navigate to the Assets folder in the pantry C:\Program Files (x86)\Steam\steamapps\common\Sid Meier's Civilization VI SDK Assets\pantry\Assets. Find a base game building's Base asset file, copy and paste it in to the Assets folder of your ModBuddy Project, which should be located in \Documents\Firaxis ModBuddy\Civilization VI\YourProject\YourProject\Assets. Then rename it the same name as your Blender project except add Base_Asset at the end. For Example, DIS_CTY_ESO_MagesGuild would become DIS_CTY_ESO_MagesGuild_Base_Asset. Tip: A building's Base asset file will usually have the word Base in it, though not always the buildings name. For example the base for the market is called DIS_COM_Base_A.
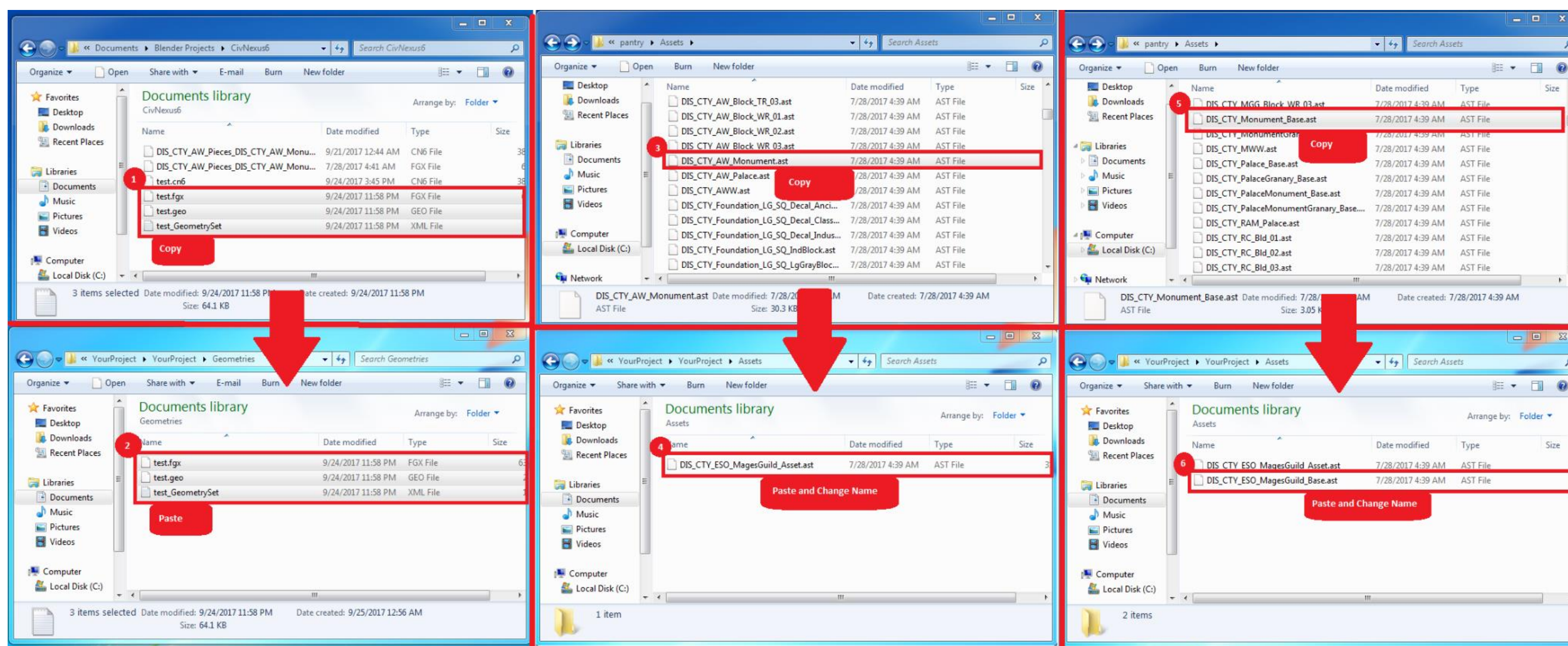
5) If your building is replacing and/or will be in the exact same spot as one of the base game's buildings, then you can skip this step because you will be using that base game building's Obstruction Profile. If your building will not be where a base game building already is then you will need to make a custom Obstruction Profile. The Obstruction Profile is like an invisible box around your 3D Model that prevents the smaller buildings that randomly spawn everywhere from spawning in or around it. In Blender, open your 3D Model's project, add a 2D plane mesh, position is directly under your 3D Model at Z= 0 and scale so that it is a little bigger than your 3D Model. Select the 2D plane and Ctrl + C to copy it, open a new Blender project while your other project is still open and Ctrl + V to paste it in the new project. It will paste in the same position that it was in the other project. Add a bone and parent to the 2D plane (Section 02 Step 11 and 13). Name the project the same as your 3D Model's project except with OB at the end. So DIS_CTY_ESO_MagesGuild would become DIS_CTY_ESO_MagesGuild_OB. Export to CN6 format, load in to CivNexus6 and do the same thing like in step 3 except with the Class set to LandmarkObstructionProfile. Copy the three files over to your ModBuddy Geometries folder like in step 4.
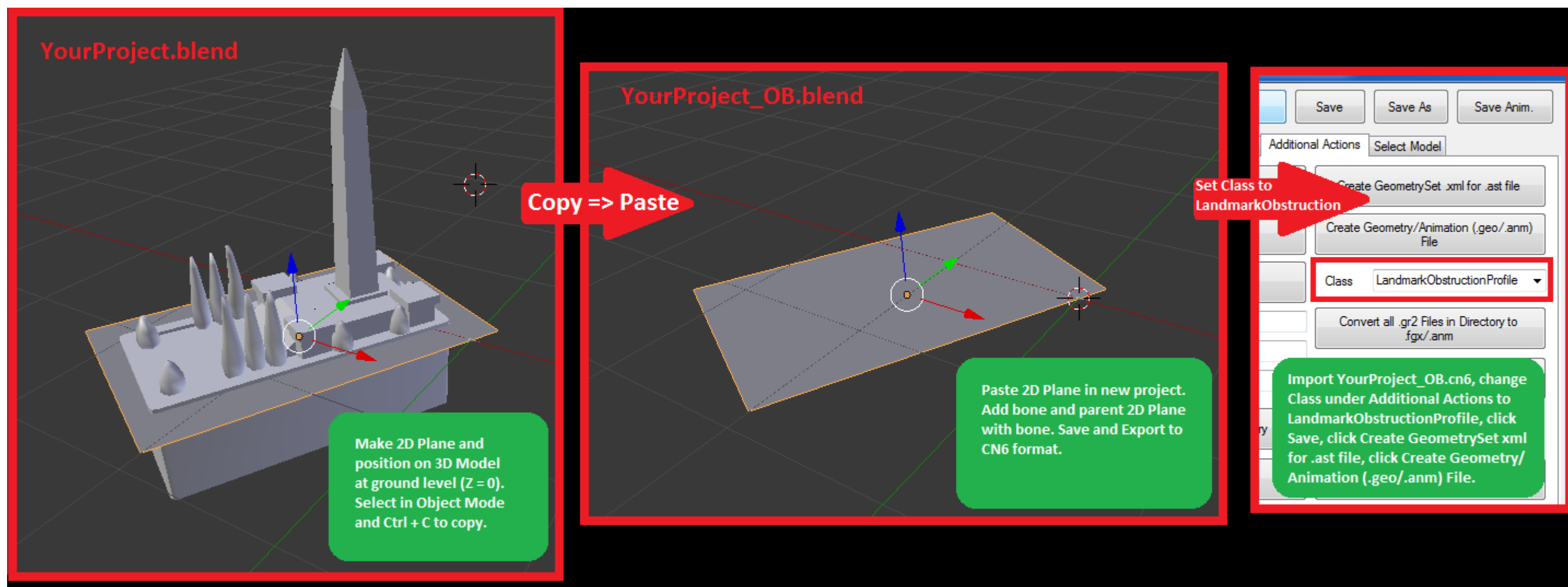
6) And that concludes Section 4, you now have all the files you needed to make or get outside of ModBuddy.

In this section we will be creating the other art assets you will need in ModBuddy's AssetEditor. We will also go ahead and add the files that we have created up until now to your ModBuddy project and edit the asset files. If you have never started up AssetEditor, know that it does take a few minutes to load for the first time. As I said in the introduction to this guide, I will assume that you already have a ModBuddy project started and that you have basic knowledge how to navigate ModBuddy.

1) Open the ModBuddy application and load your project. In the Solution Explorer panel right click your project, hover your mouse cursor over Add and select Existing Item. Navigate to your ModBuddy project's Geometries folder, select all of the files you copied to this folder in section 4 and click Add. In the Solution Explorer panel right click your project, hover your mouse cursor over Add and select Existing Item. Navigate to your ModBuddy project's Assets folder, select all of the files you copied to this folder in section 4 and click Add. You should now see all of the geometry and asset files you created up until now in the Solution Explorer under their respected folder.
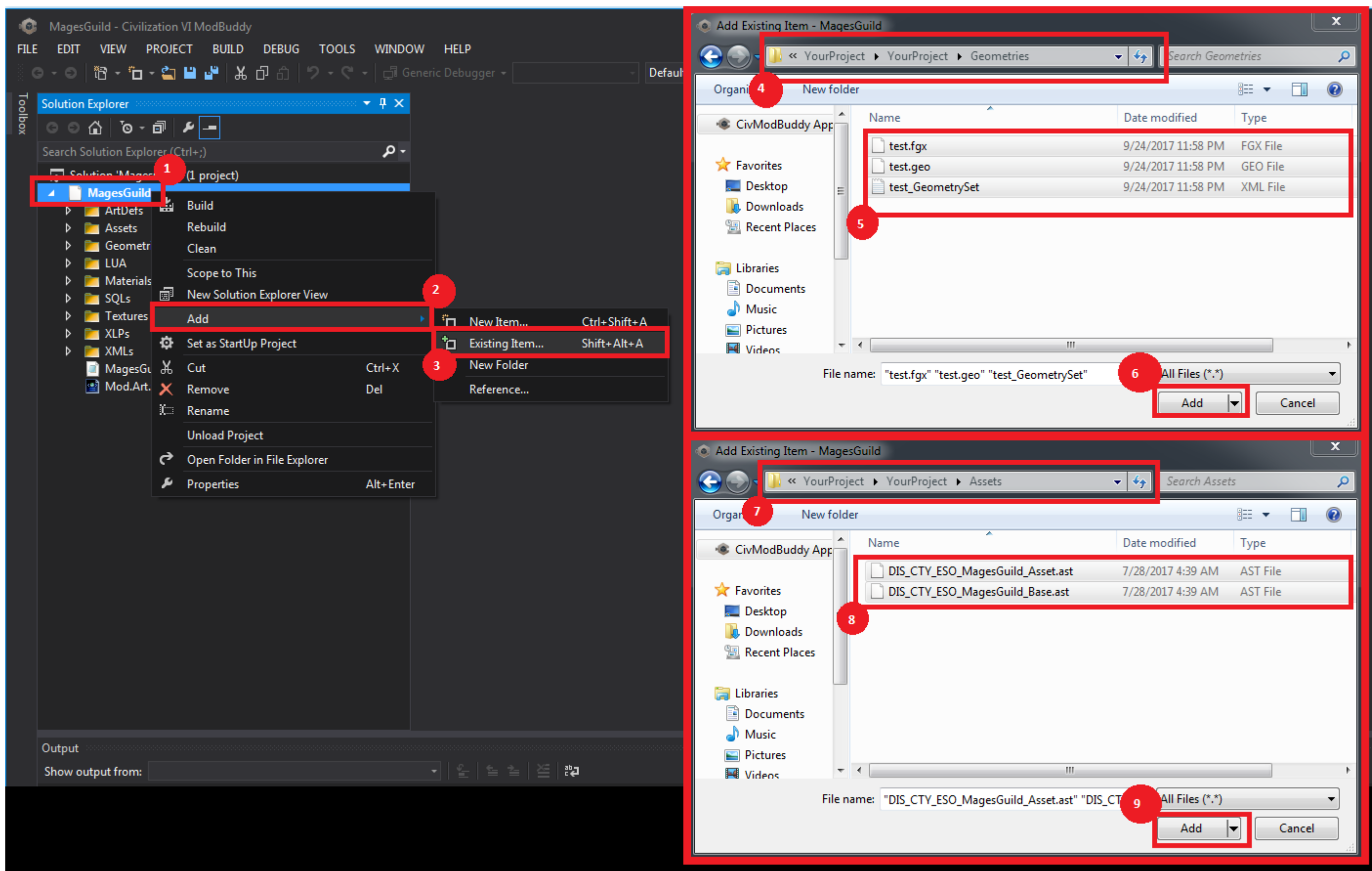
2) In ModBuddy, double left click your asset file to open it, the one for the building not the Base one. Near the top there will be an element named m_Name, select what is in the text field and run a find and replace (Ctrl + F), replacing it with the name of your Armature Object, then click the Replace All button. For the element named m_GeoName, select what is in the text field and run a find and replace (Ctrl + F), replacing it with the name of your Geometry file (the file with extension .geo in your Geometries folder), then click the Replace All button. For the element named m_ObjectName, select what is in the text field and run a find and replace (Ctrl + F), replacing it with the name of your Material (what you named your material in Blender), then click the Replace All button. For the element named m_GroupName, select what is in the text field and run a find and replace (Ctrl + F), replacing it with the name of your Material (what you named your material in Blender), then click the Replace All button. For the element named m_MeshName, select what is in the text field and run a find and replace (Ctrl + F), replacing it with the name of your Mesh (what you named your mesh in Blender), then click the Replace All button. Then scroll down to the end and for the element named m_Name, change the text field to the name of this file (the file you are currently editing). This element was change during your first find and replace, but it is not what you want so change it as described above, and do not run a find and replace for it, you just need to change this specific element. Now open your asset file for the Base, scroll down to the end and for the element named m_Name, change the text field to the name of this file (the file you are currently editing). Click File, then select Save All.
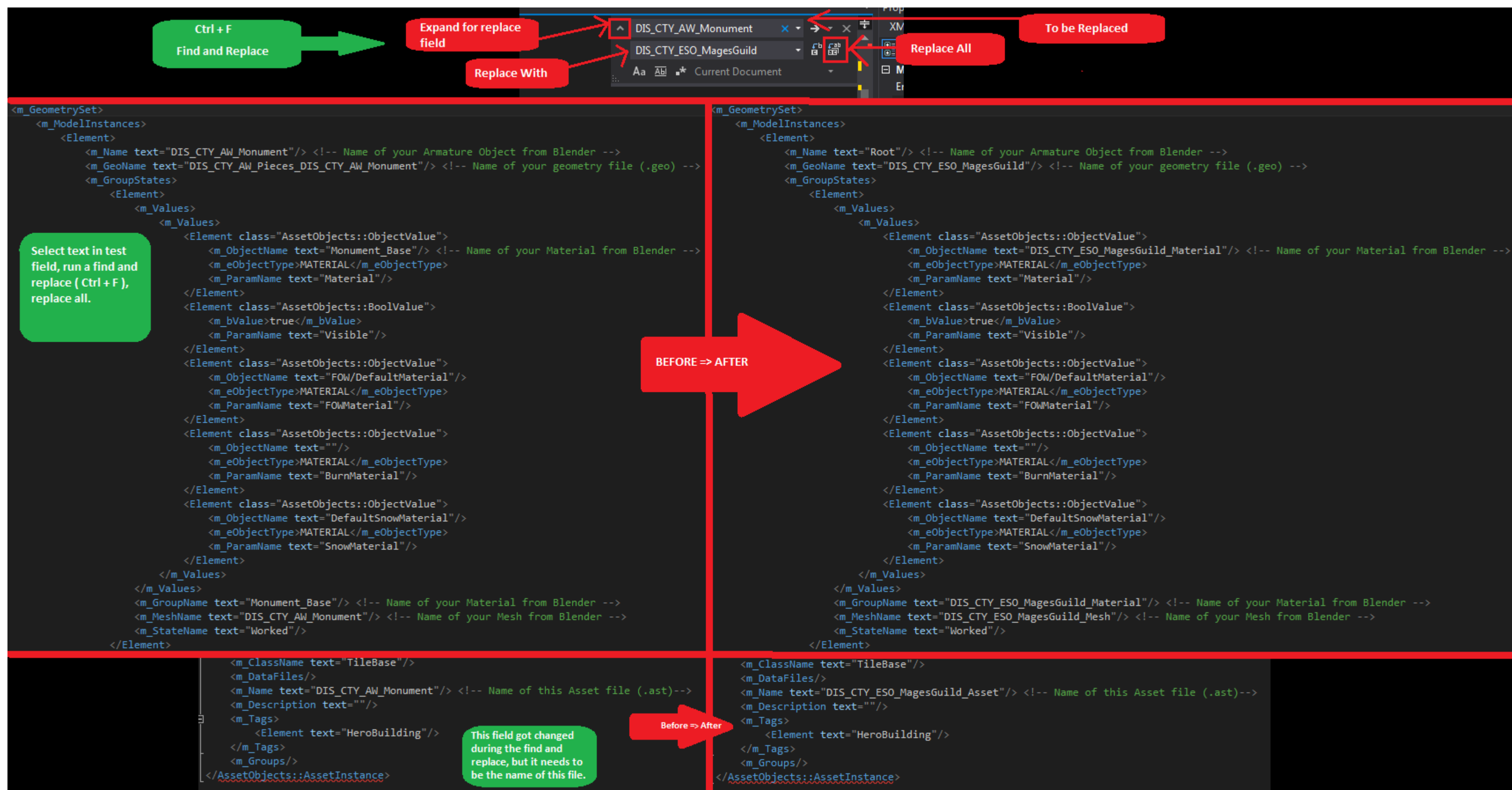
- If you made a custom Obstruction Profile, then scroll down to the end of the asset file, under element m_cookParams, for the element named m_ObjectName that is above the mParamName labeled Obstruction Profile, change the text field of the m_ObjectName element to the file name of your Obstruction Profile's geometry file (.geo). Do this to both the building's asset file and the Base asset file. Click FILE, then select Save All.

- Note: Enter in the file name only, no not include file extensions.

3) Open AssetEditor, click the TOOLS tab found on the top menu bar and select Launch Asset Editor. Click File, then select New, click Texture, then click OK. For Class Name select BaseColor from the drop down menu, for Source File Path navigate to \Documents\Blender Projects\Textures and select your DDS format UV ColorMap, then click the Save icon up top. Repeat for all of your other DDS format UV maps, with the Class Name set to its respected map type. Close the files after you are done (the small black x to the far right of the file's title tab. What we did was imported the UV maps and created the accompanying texture files.
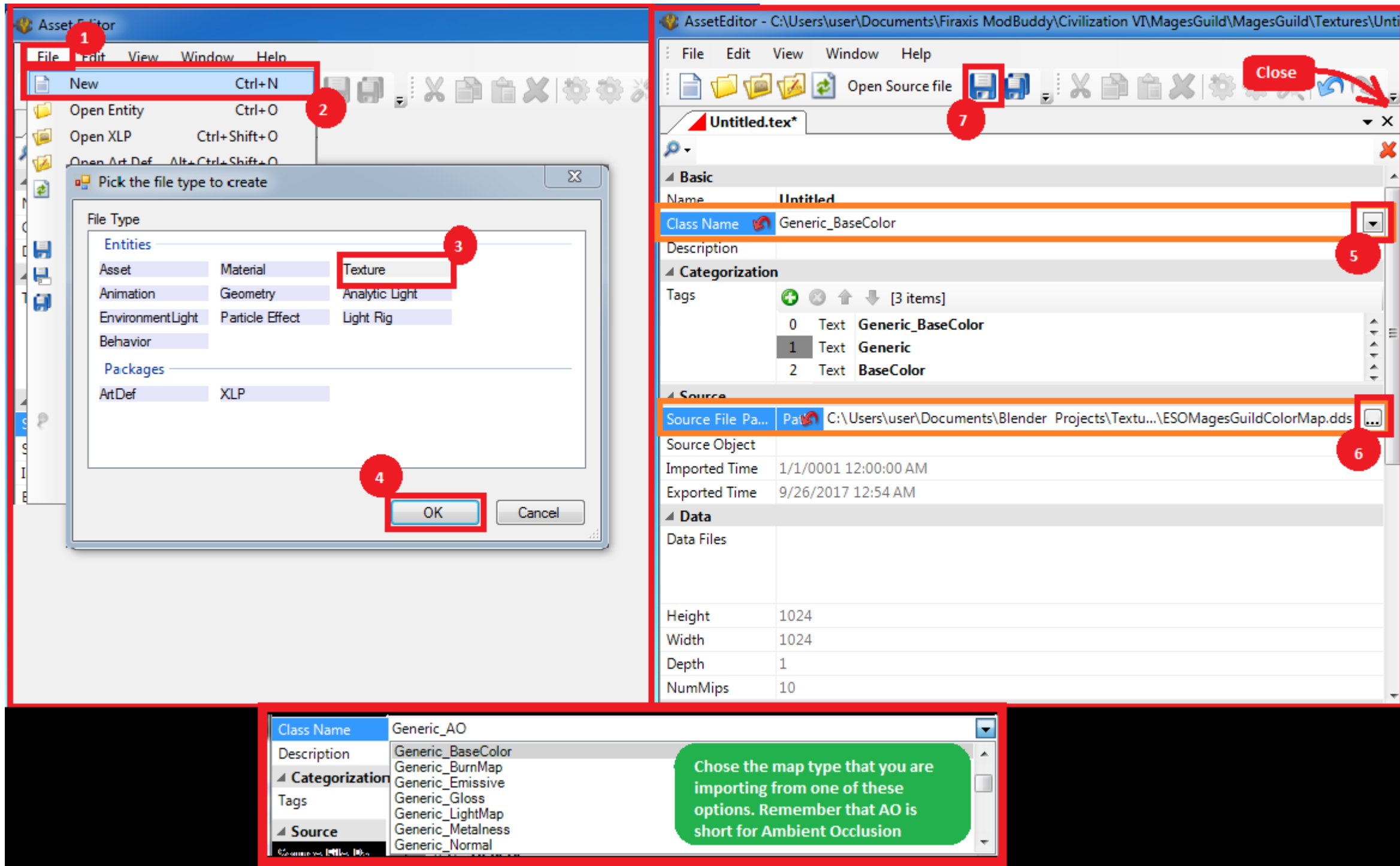
4) Click File, then select New, click Material, then click OK. For Class Name select Landmark. Add each one of the texture files you created in the previous step to its respected value under Cook Parameters by clicking the Add Existing icon, uncheck Civ6, select your texture file and click OK. Click in the Name field and name it the same name as you named your material in Blender. After you have added all of the textures you had, click the Save icon up top. Close the file after you are done.
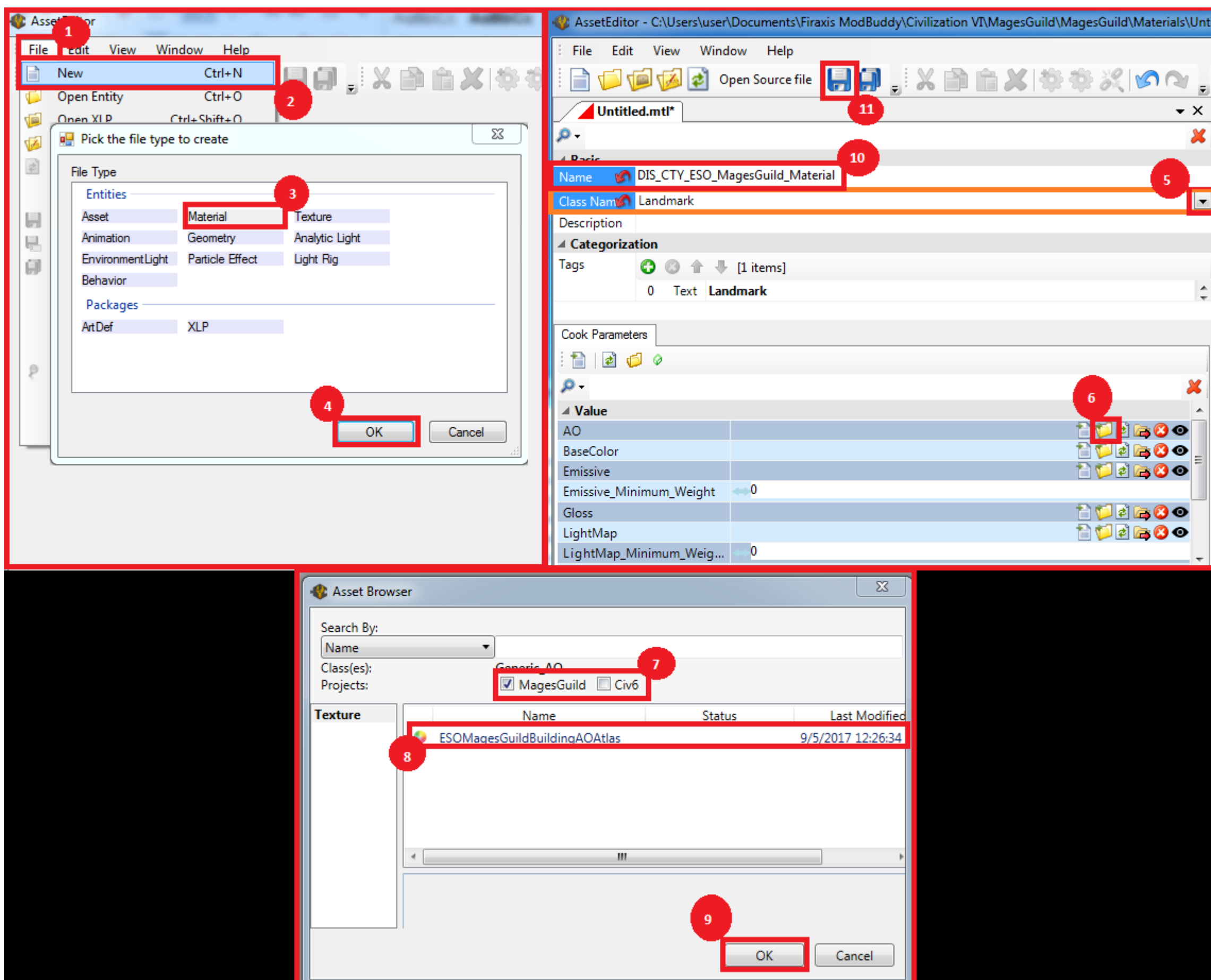
- Note: If your texture is not showing up in the Asset Browser when you go to add it, just set the Class Name, Rename it, and save it. You can add them by hand in ModBuddy. Just add the Material file (.mtl) to your project, just like you did in step 1. It will be under the Materials folder in your projects folder. Double left click to open it, and you will change the m_ObjectName to the file name of your texture file with no extensions, that is above the m_ParamName with the respected map type. After you add all of your textures, click File and select Save All.

5) Click File, then select New, click ArtDef, then click OK. This will be the ArtDef that will be in your project. Click File, then select Open Art Def, navigate to C:\Program Files (x86)\Steam\steamapps\common\Sid Meier's Civilization VI SDK Assets\pantry\ArtDefs, select Buildings, then click OK. This is the base game ArtDef that you can use as a reference. In the base game Buildings ArtDef, expand Building, then find a base game building that is similar to the building you are making and expand it. Use the settings as an example for the following. Now go back to the new ArtDef you created, under Art Definition Template at the top, select Buildings. Below that, right click on Building and click Add Element, left click the new element that appeared, for the Name field type in the name of your building that you defined in your ModBuddy game data file (Buildings.xml). Click the Save icon up top.
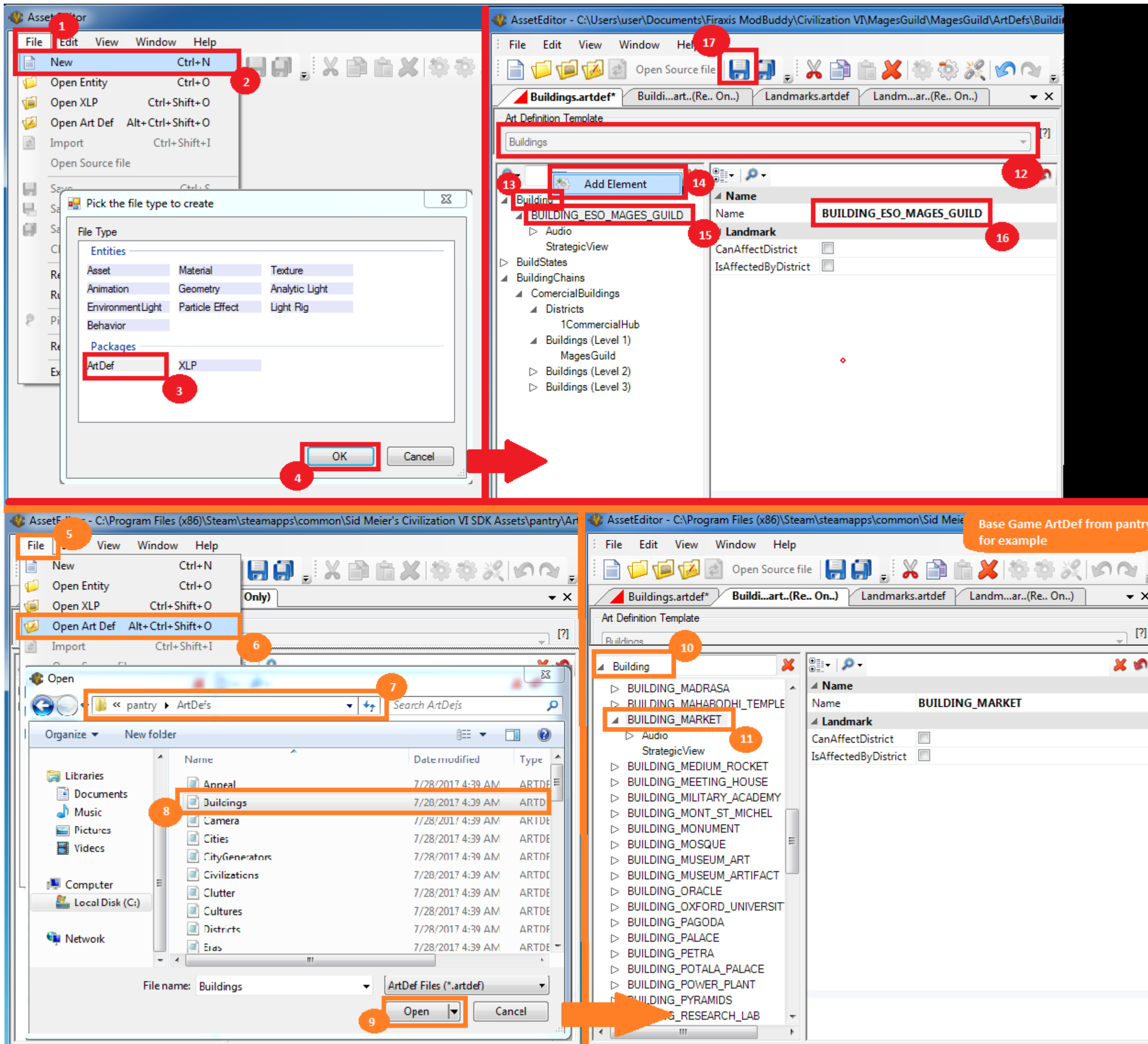
- In the new Buildings ArtDef left click StrategicView to select it, then right click it and click Add Element. You will need to add 3 elements. Then in the base game Buildings Art Def left click StrategicView to select it, look at the values that are in its 3 elements and set yours up the same way. Just left click in the fields to change them. See image below.
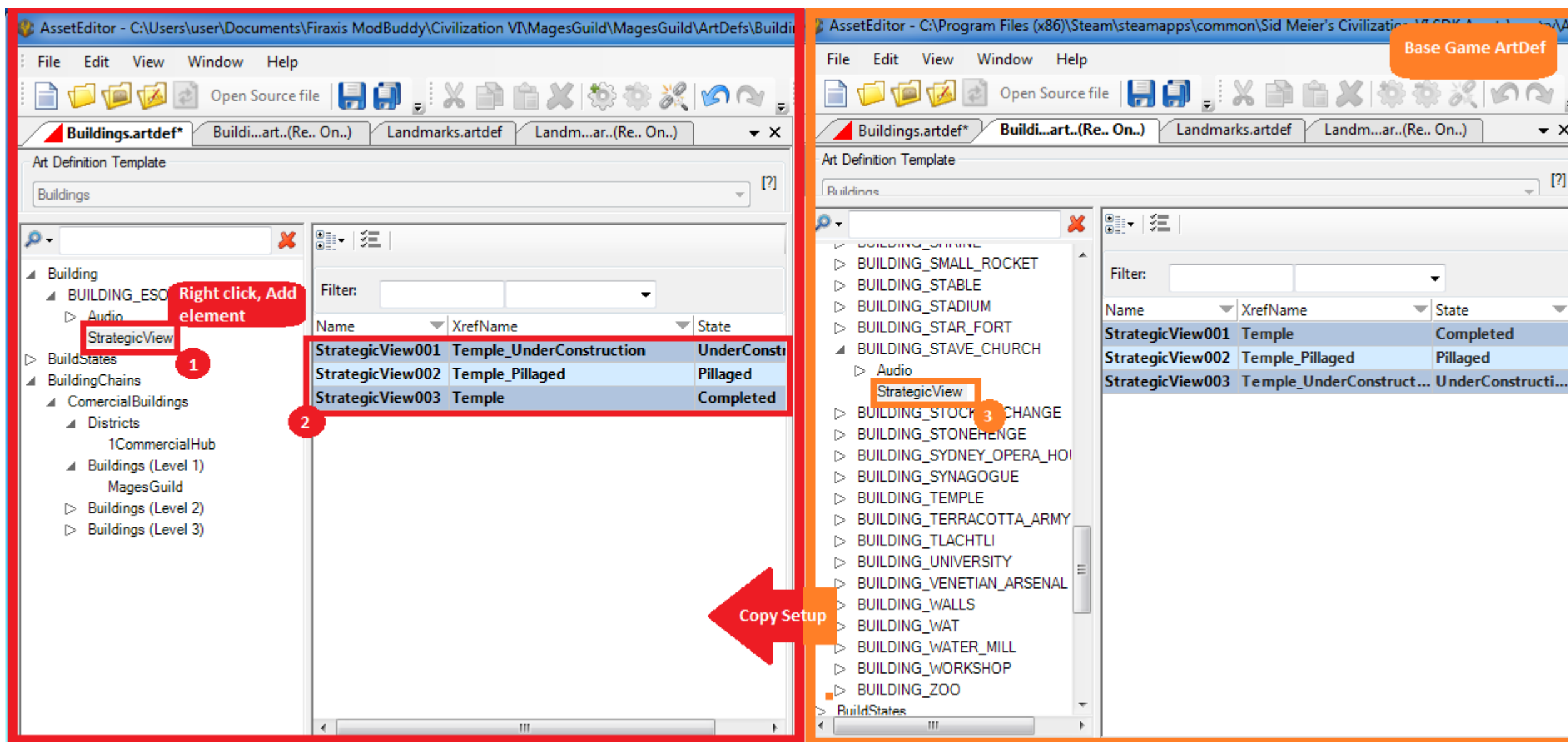
- If your building is a city center district building, skip this part. In the base game Buildings Art Def expand BuildingChains, find the district your building will be in and expand, expand Districts and whichever level your building will be in. Level 1 being the prerequisite building for level 2 and so on. In the new Buildings ArtDef expand BuildingChains, right click BuildingChains and click Add Element, in the Name field type the same name as in the base game ArtDef. Right click Districts and click Add Element, edit the Name field and the District field to be the same name as in the base game ArtDef. Right click which ever Buildings (Level#) your building is and click Add Element, edit the Name field to be a short hand of the name of your building, then for the Building field choose your building from the drop down menu. Click the Save icon up top.
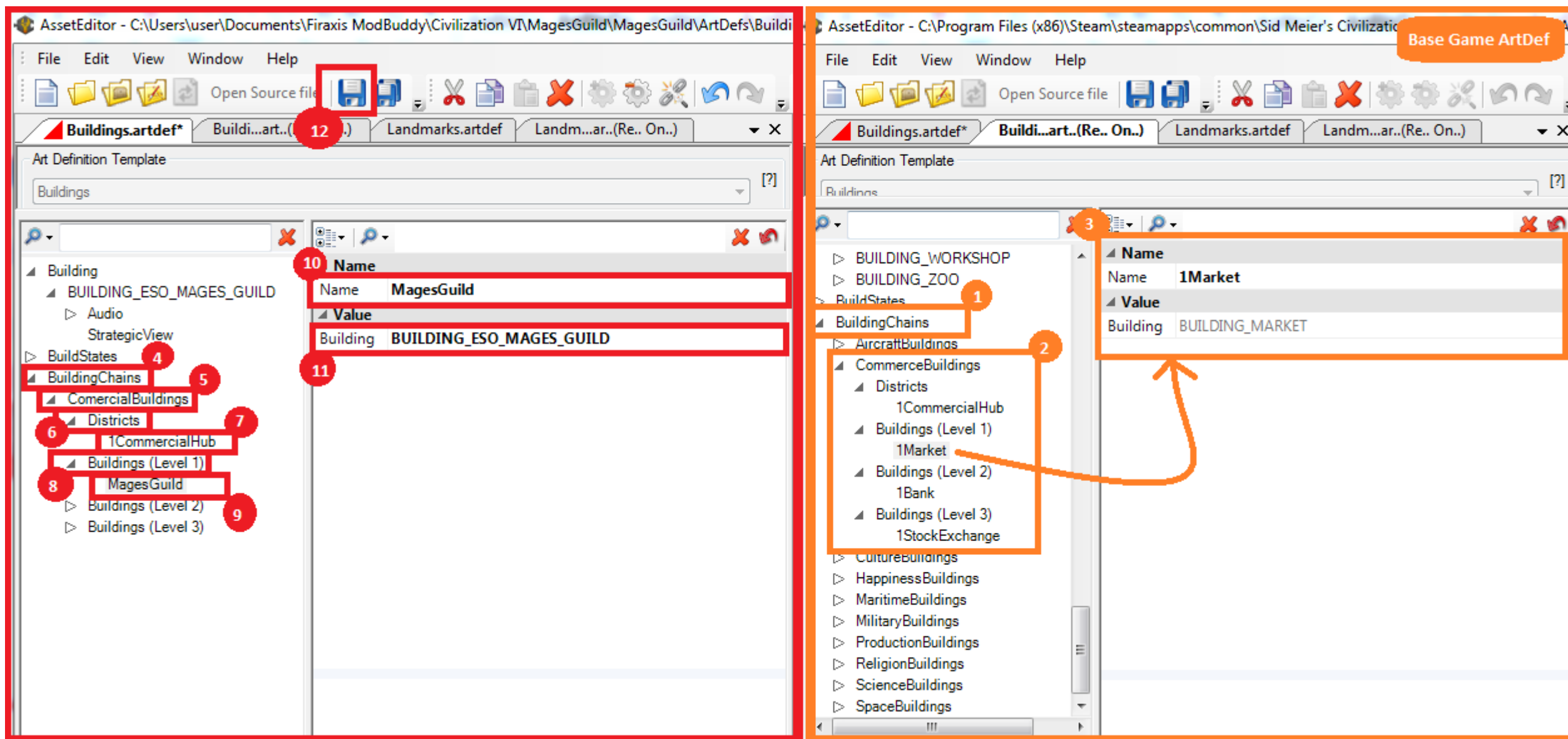
- Note: If your building did not show up in the drop down menu you can just add it by hand in ModBuddy. Just add Buildings.artdef to your project, just like you did in step 1. It will be under the ArtDefs folder in your projects folder. Double left click to open it, under m_CollectionsName BuildingChains, find the section where the mName field is the short hand name of your building you chose above, change the m_ElementName to the name of your building that you defined in your ModBuddy game data file (Buildings.xml). Click File and select Save All.

- We are now done created your Buldings.artdef file, you can close this file after saving it. You can also close the base game's Buldings.artdef, do not save.
6) Click File, then select New, click ArtDef, then click OK. This will be the ArtDef that will be in your project. Click File, then select Open Art Def, navigate to C:\Program Files (x86)\Steam\steamapps\common\Sid Meier's Civilization VI SDK Assets\pantry\ArtDefs, select Landmarks, then click OK. This is the base game ArtDef that you can use as a reference. In the base game Landmarks ArtDef, expand Districts, then find the district that your building will be in and expand it (the city center is a district). Use the settings as an example for the following. Now go back to the new ArtDef you created, under Art Definition Template at the top, select Landmarks. Below that, right click on Districts and click Add Element, left click the new element you just added to select, for the Name field, type in the name of the district your building will be in the same way it is in the base game ArtDef. For the DistrictGenerator field, select the same option as you see it in the base game ArtDef. Change to TintColor to the same as how it is in the base game ArtDef. Click the Save icon up top.
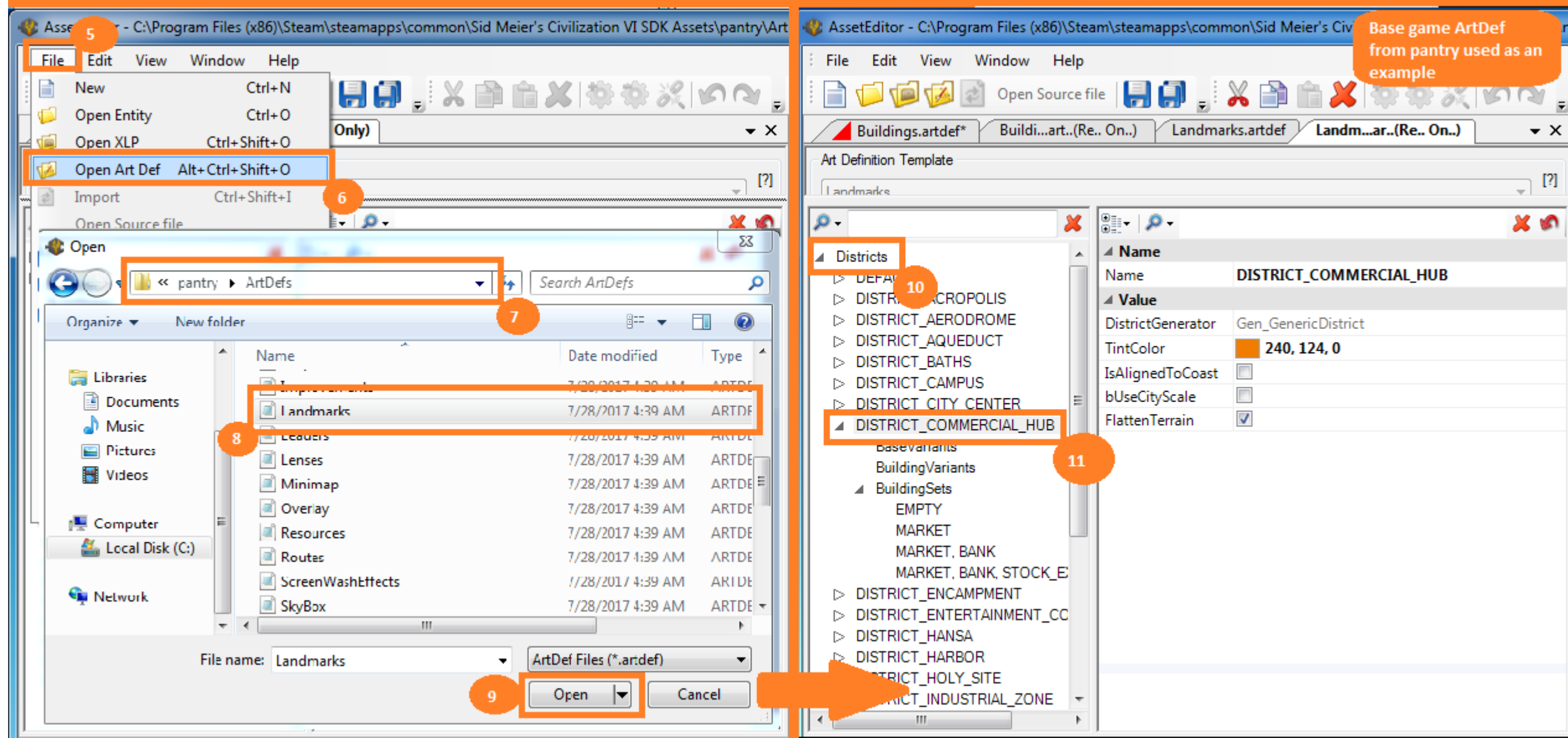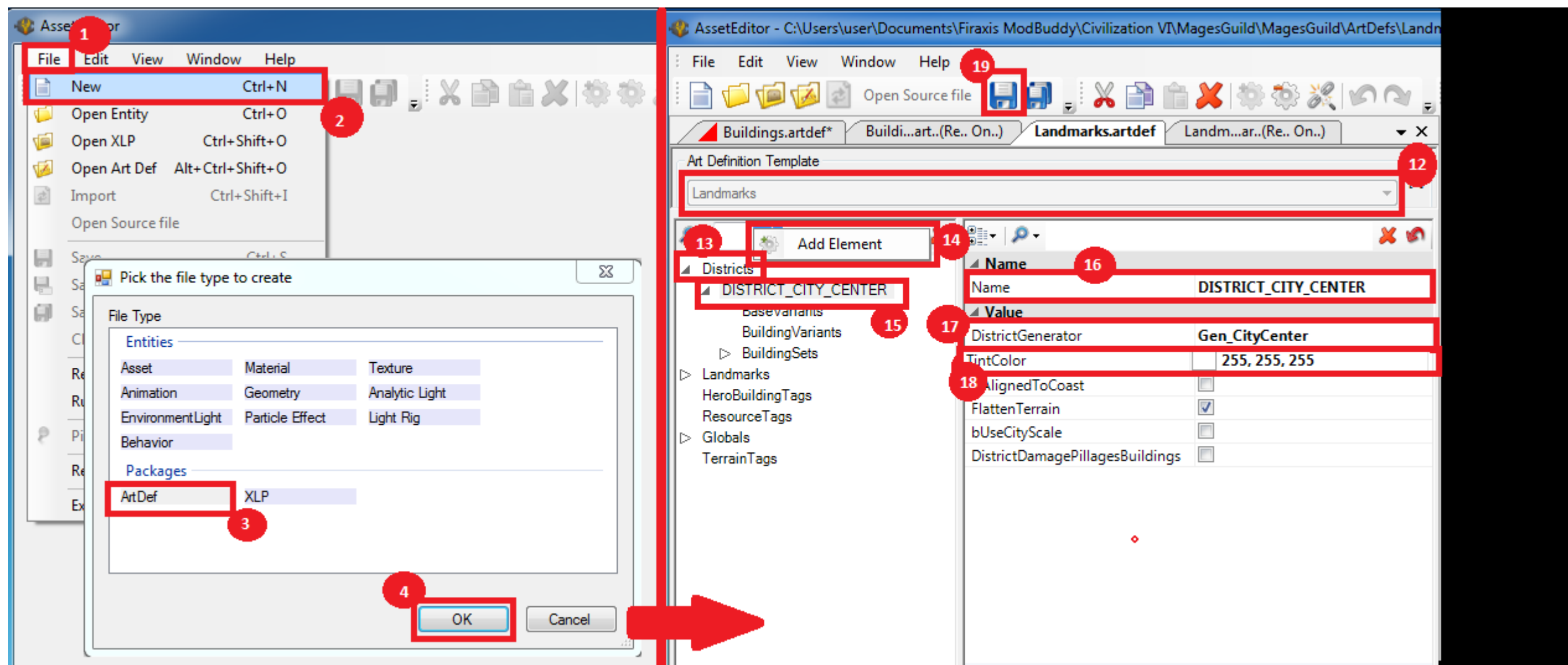
In the new ArtDef you created, right click HeroBuildingTags, click Add Element, for the Name field type in the name of your building that you defined in your ModBuddy game data file (Buildings.xml). Click the Save icon up top. In the new ArtDef you created, expand BuildingSets, right click it and click Add Element, left click the new element you just added to select, for the Name field type the short hand name for your building, click the green button with the plus sign to add a new element, select your building from the drop down menu. Click the Save icon up top.
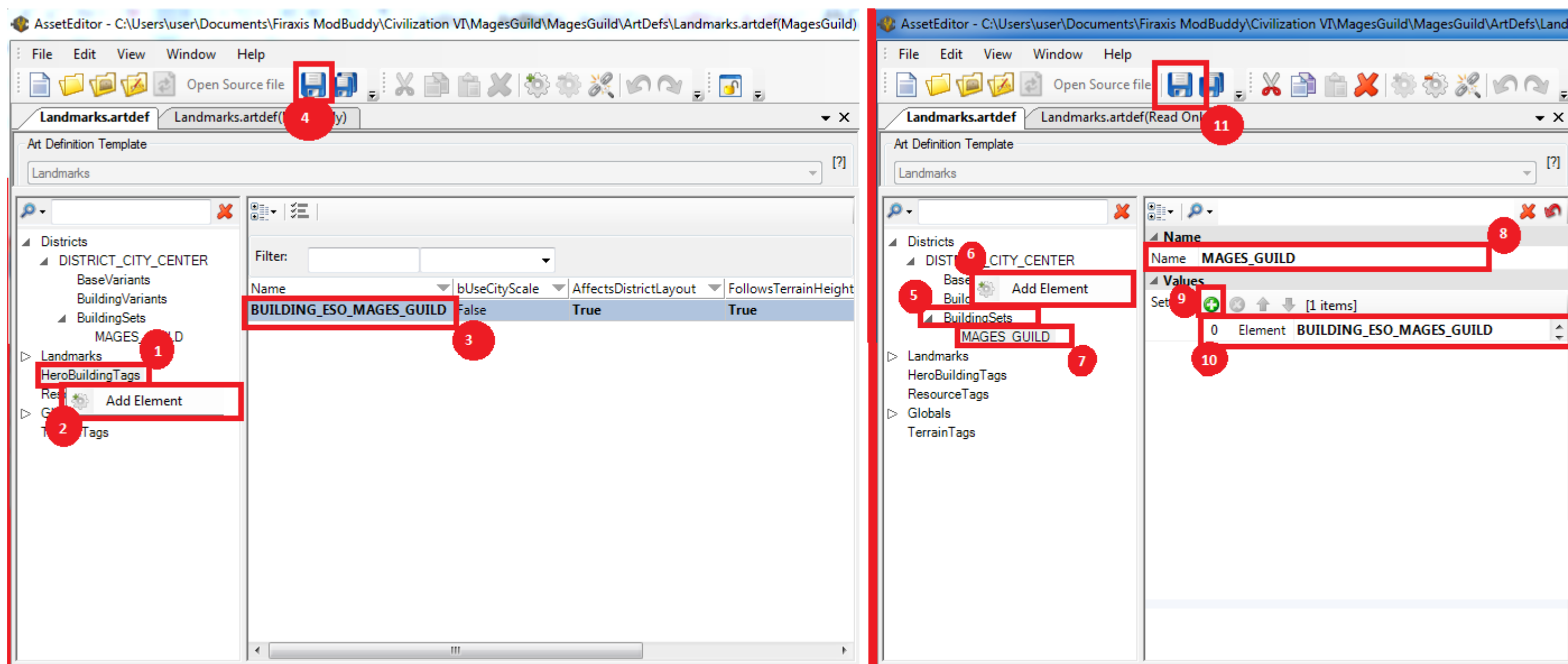
In the new ArtDef you created, right click BaseVariants, click Add Element, for Set_HeroBuildings field select the name of the element you made under BuildingSets from the drop down menu. For Asset choose the name of the asset file you made for the Base from the BLP Entry Browser. In the new ArtDef you created, right click BuildingVariants, click Add Element, for Tag_HeroBuilding field select the name of the element you made under HeroBuildingTags from the drop down menu. For Asset choose the name of the asset file you made for your 3D Model from the BLP Entry Browser. Click the Save icon up top.
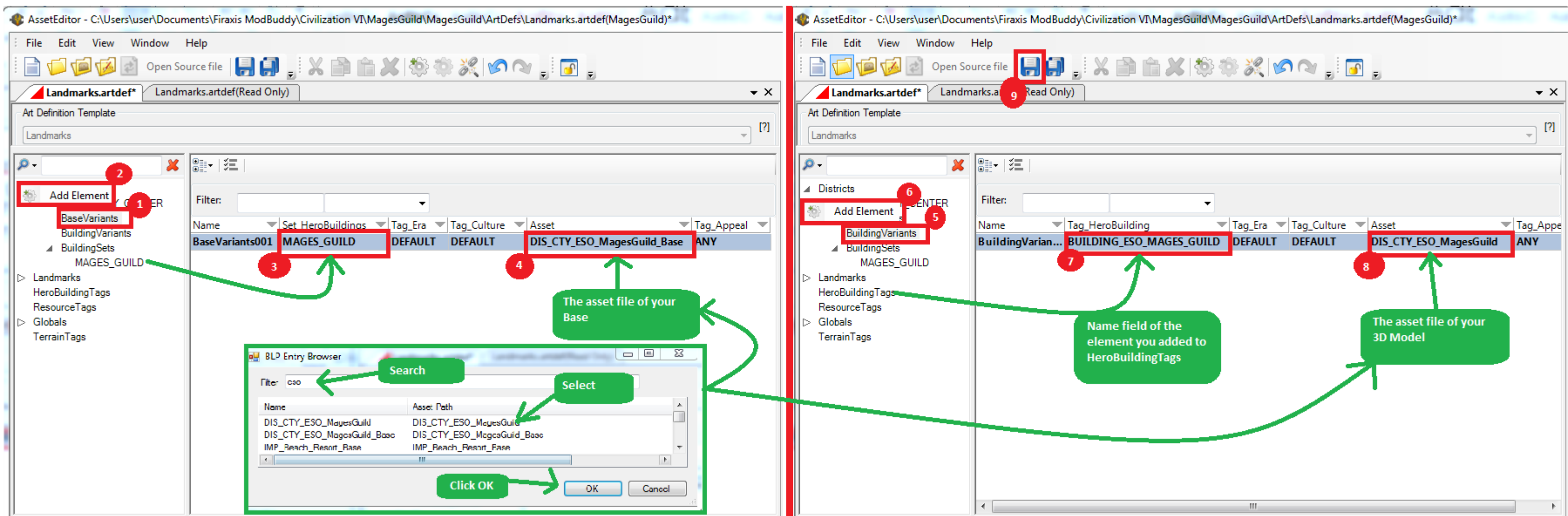
- Note: If your building did not show up in the drop down menu when adding the BuildingSets element, or in the Set_HeroBuildings field under BaseVarients, or the Tag_HeroBuilding field under BuildingVarients, or your assets in the BLP Entry Browser for the Asset field under BuildingSets and BaseVarients, you can just add them by hand in ModBuddy. Just add Landmarks.artdef to your project, just like you did in step 1. It will be under the ArtDefs folder in your projects folder. Double left click to open it, search for BuildingSets, for the m_ElementName field type in the name of your building that you defined in your ModBuddy game data file (Buildings.xml). Search for BaseVarients, for the m_ElementName field type the name of the element you made under BuildingSets. For the m_EntryName field type the name of the asset file you made for the Base. Search for BuildingVarients, for the m_ElementName field type the name of the element you made under HeroBuildingTags. For the m_EntryName field type the name of the asset file you made for your 3D Model. Click File and select Save All.

- We are now done created your Landmarks.artdef file, you can close this file after saving it. You can also close the base game's Landmarks.artdef, do not save.
7) Click File, then select Open Entity, under Assets uncheck Civ6, select the asset file for your 3D Model. Click the Geometries tab, one of the entries will be the geometry of your 3D Model and will have the name of the geometry file in it. Delete any other entries so that only the entry for your 3D Model remains. To delete an entry, left click it to select it, and click the red circle with the x in it above. If you made your own Obstruction Profile, click the Cook Parameters tab, under Obstruction Profile you should see the obstruction profile that you created (we added it to this file in step 2), under that make sure the Obstruction Profile AutoGeneration is unchecked. If you are using a base games obstruction profile, make sure it is set to the base game's building OB file and not the Base. Set DSG to Standard_Landmark. Click Save up top. Ignore any error messages, or what is displayed in Asset Previewer for now.
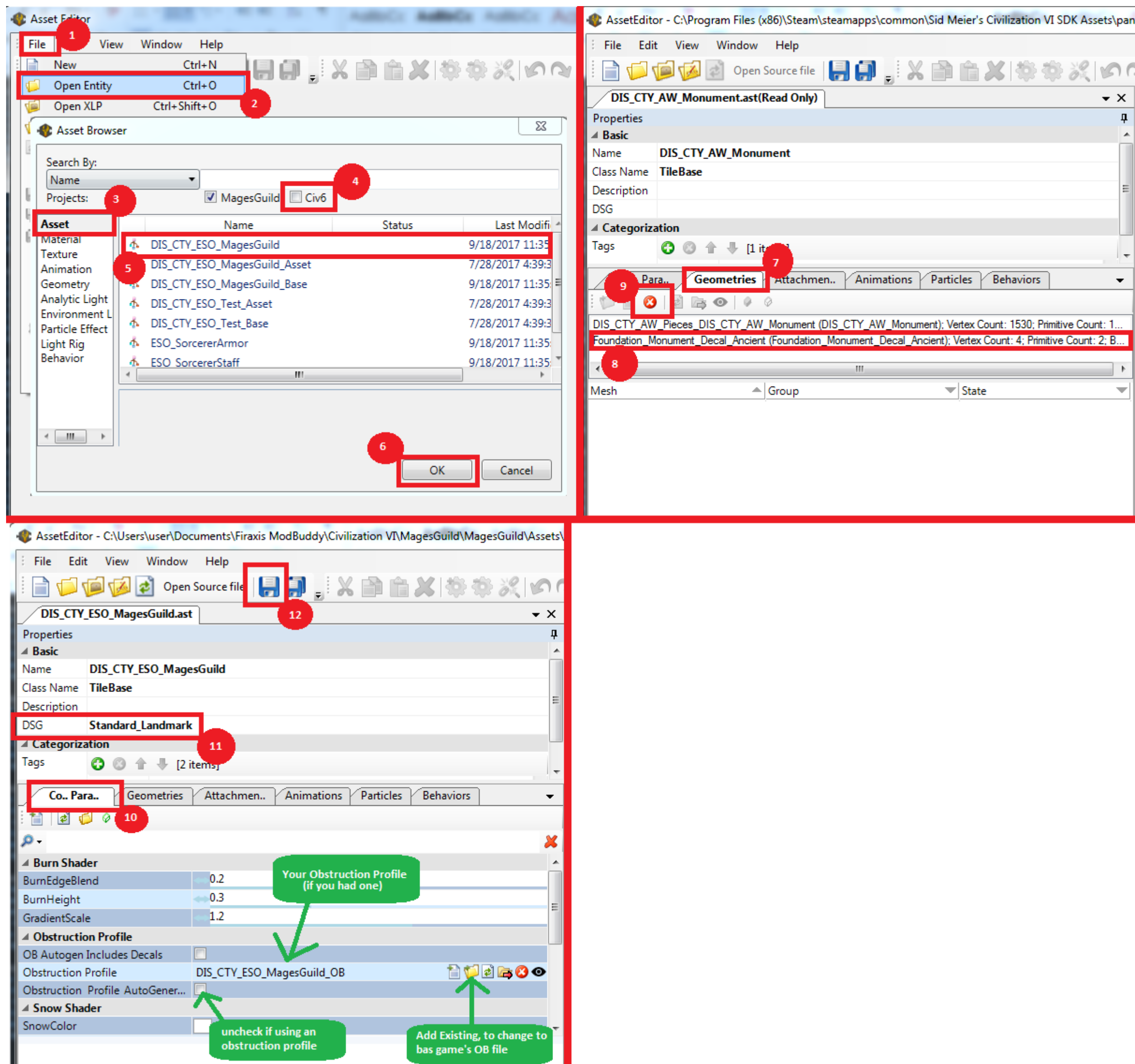
- Note: You do not need to mess with your asset file for your Base, it should already have the correct Obstruction Profile assigned to it, and if you had your own Obstruction Profile, it should have already been added in step 2.
8) And that concludes section 5, you now have all the files you need from AssetEditor.

Section 6 Setting up Your 3D Model in ModBuddy

In this section we will create and edit any remaining files you need to get your 3D Model working in ModBuddy. After you complete this section your 3D Model will show up in your game.

1) Add all of the files you created in AssetEditor to your project. Refer back to section 5 step 1 if you forgot how. You will be adding both the DDS format (.dds) and the TEX format (.tex) of your UV maps, find these under the Textures folder. Your material file (.mtl) will be under the Materials folder. Both of your ArtDefs (Buildings.artdef and Landmarks.artdef) will be under the ArtDefs folder. Your Solution Explorer panel should look something like the image below.
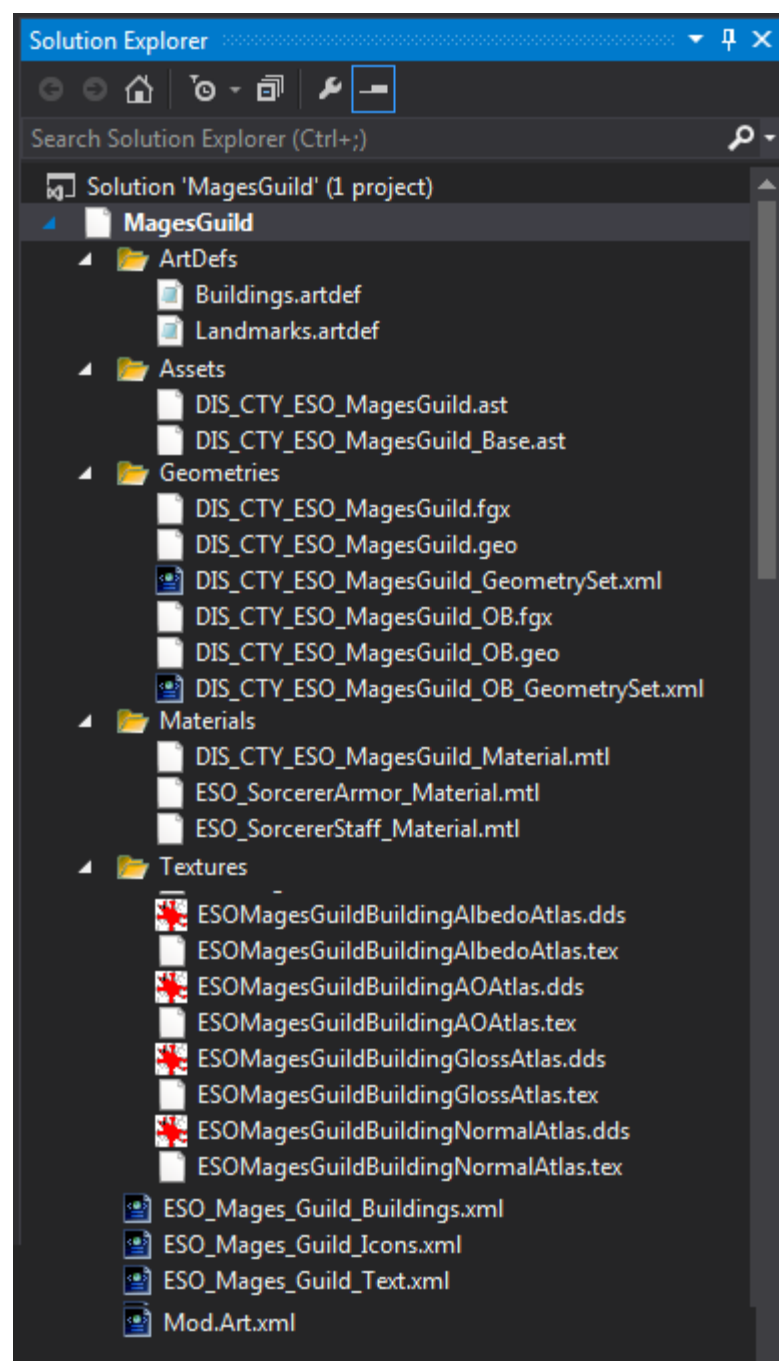
2) In the Solution Explorer panel, right click your project, select Add and click New Item. Select Text File, for the file Name type exactly hero_buildings.xlp and click Add. In the Solution Explorer panel, right click your project, select Add and click New Item. Select Text File, for the file Name type exactly tilebases.xlp and click Add. You now have two XLP files in your Solution Explorer. Double left click on each of them to open them.
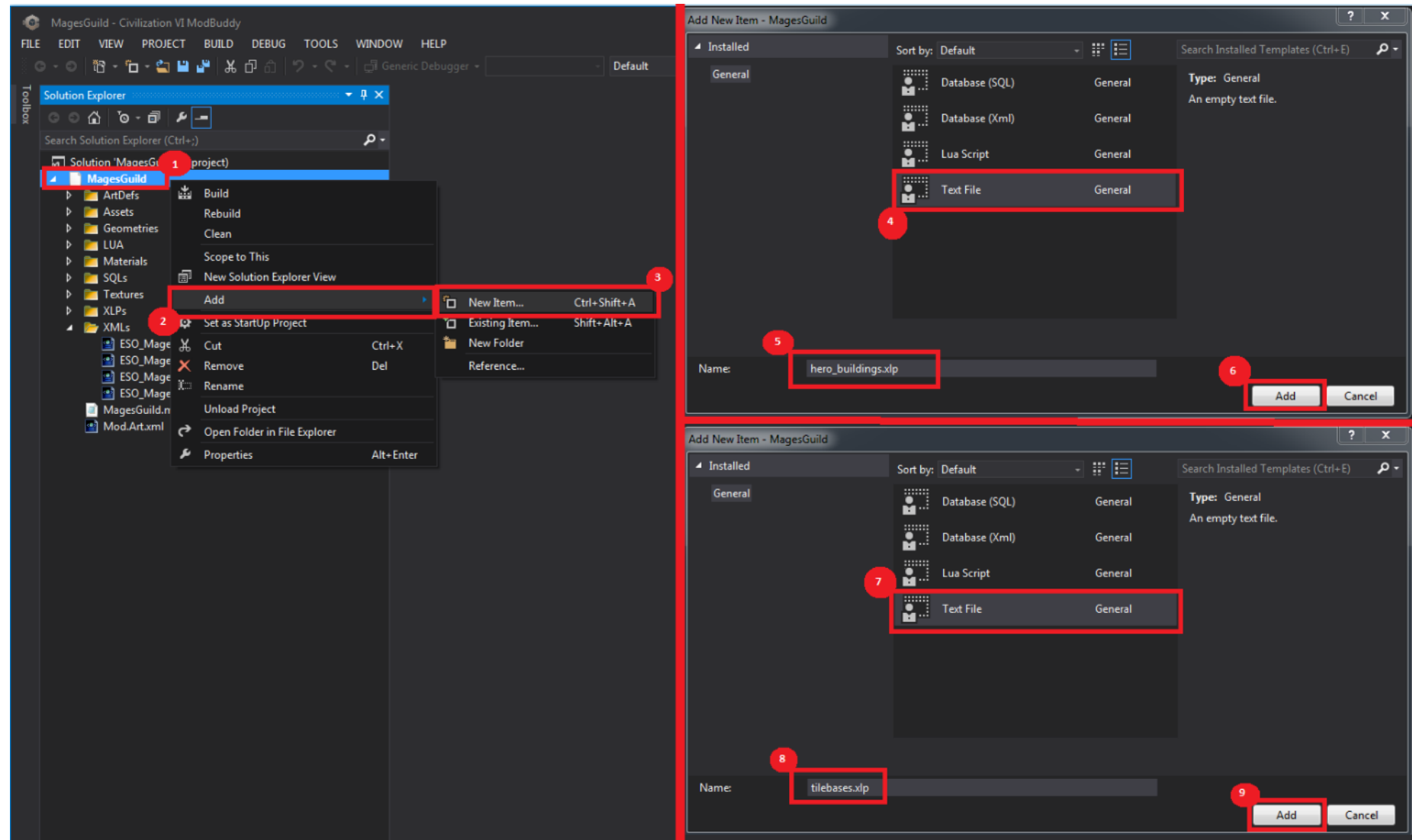
- For hero_buildings.xlp copy and paste the following code in to it, and change field for the m_EntryID and m_ObjectName to the name of your 3D Model's asset file (.ast), without file extension.

```
<?xml version="1.0" encoding="UTF-8" ?>
<AssetObjects::XLP>
    <m_Version>
        <major>4</major>
        <minor>0</minor>
        <build>260</build>
        <revision>754</revision>
    </m_Version>
    <m_ClassName text="TileBase"/>
    <m_PackageName text="landmarks/hero_buildings"/>
    <m_Entries>
        <Element>
                    <m_EntryID text="NAME OF YOUR 3D MODEL'S ASSET FILE"/>
                    <m_ObjectName text=" NAME OF YOUR 3D MODEL'S ASSET FILE "/>
        </Element>
    </m_Entries>
        <m_AllowedPlatforms>
            <Element>WINDOWS</Element>
            <Element>LINUX</Element>
            <Element>MACOS</Element>
            <Element>IOS</Element>
        </m_AllowedPlatforms>
</AssetObjects::XLP>
```

- For tilebases.xlp copy and paste the following code in to it, and change field for the m_EntryID and m_ObjectName to the name of your Base's asset file (.ast), without file extension.

```
<?xml version="1.0" encoding="UTF-8" ?>
<AssetObjects::XLP>
    <m_Version>
        <major>4</major>
        <minor>0</minor>
        <build>260</build>
        <revision>754</revision>
    </m_Version>
    <m_ClassName text="TileBase"/>
    <m_PackageName text="landmarks/tilebases"/>
```

```
        <m_Entries>
          <Element>
                      <m_EntryID text="NAME OF YOUR BASE'S ASSET FILE"/>
                      <m_ObjectName  text="NAME OF YOUR BASE'S ASSET FILE"/>
          </Element>
            </m_Entries>
            <m_AllowedPlatforms>
                    <Element>WINDOWS</Element>
                    <Element>LINUX</Element>
                    <Element>MACOS</Element>
                    <Element>IOS</Element>
            </m_AllowedPlatforms>
      </AssetObjects::XLP>
```

- Click FILE and select Save All. Below are examples of both files in ModBuddy.
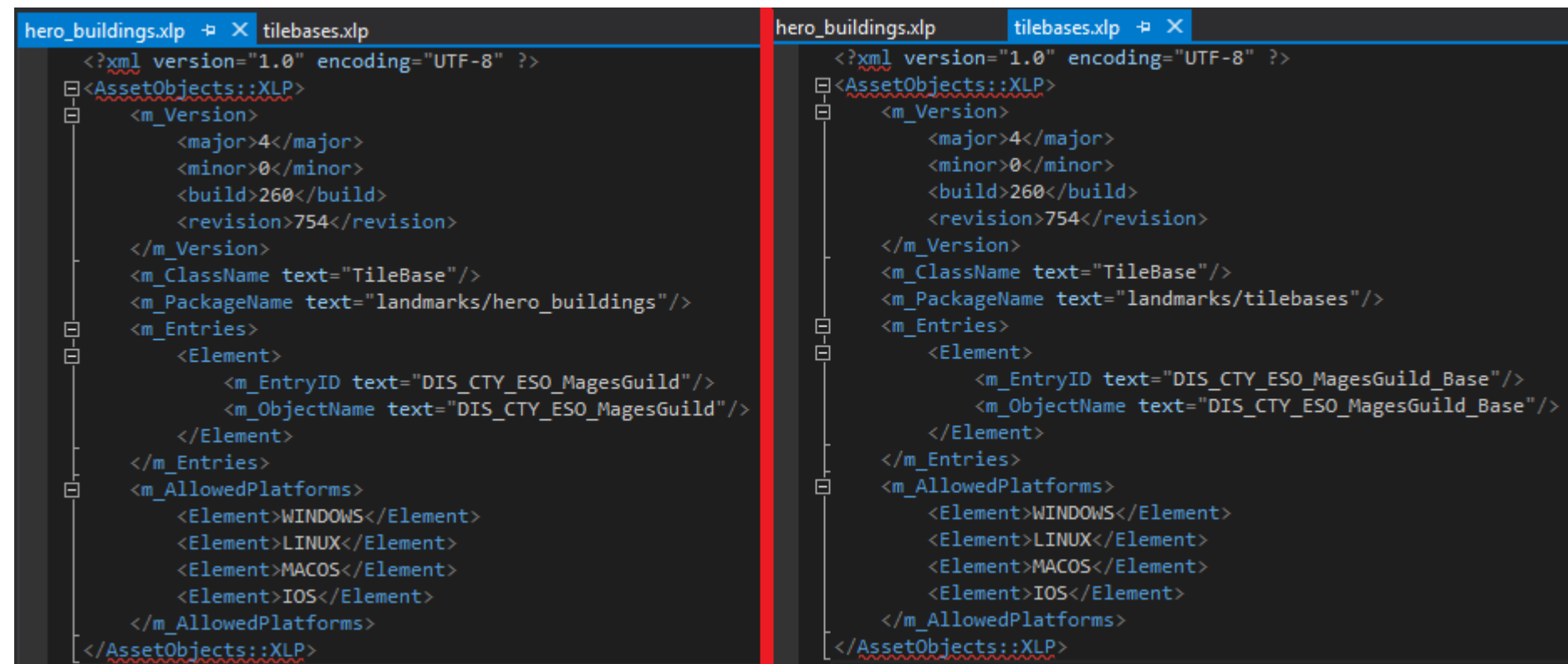
3) In your Solution Explorer, right click your project, select Add and click New Item. Select Database (.xml), type exactly Mod.Art.xml for the file name then click Add. You may already have a Mod.Art.xml, if you do then you don't need to create another. Double left click your Mod.Art.xml file to open. If you created a new Mod.Art.xml, or if you don't have anything you've edited in the one you already have, delete all of the file's existing code, then copy and paste the following code in to it. Just change the name filed to the name of your project, and the id field to your Mod's ID Number (find this under Project Properties in ModInfo). This code is set up for buildings, units, and icons. If you don't need the whole code, see part below the code.

```
        <?xml version="1.0" encoding="UTF-8" ?>
        <AssetObjects::GameArtSpecification>
              <id>
                    <name text="Name of Your Project"/>
                    <id text="Your Mod's ID Number"/>
              </id>
              <artConsumers>
                    <Element>
                          <consumerName text="Units"/>
                          <relativeArtDefPaths>
                                  <Element text="Units.artdef"  />
                                  <Element text="Unit_Bins.artdef"  />
                          </relativeArtDefPaths>
                          <libraryDependencies>
                                  <Element text="Unit"/>
                                  <Element text="VFX"/>
                                  <Element text="Light"/>
                          </libraryDependencies>
                          <loadsLibraries>true</loadsLibraries>
                    </Element>
                    <Element>
                          <consumerName text="Clutter"/>
                          <relativeArtDefPaths>
                          </relativeArtDefPaths>
                          <libraryDependencies>
                                  <Element text="Landmark"/>
                          </libraryDependencies>
                          <loadsLibraries>true</loadsLibraries>
                    </Element>
                    <Element>
                          <consumerName text="Landmarks"/>
                          <relativeArtDefPaths>
                                  <Element text="Landmarks.artdef"  />
                          </relativeArtDefPaths>
                          <libraryDependencies>
                                  <Element text="CityBuildings"/>
                                  <Element text="TileBase"/>
                                  <Element text="RouteDecalMaterial"/>
                          </libraryDependencies>
                          <loadsLibraries>true</loadsLibraries>
                    </Element>
                    <Element>
                          <consumerName text="Farms"/>
                          <relativeArtDefPaths>
                          </relativeArtDefPaths>
                          <libraryDependencies>
                                  <Element text="TileBase"/>
                                  <Element text="CityBuildings"/>
                          </libraryDependencies>
                          <loadsLibraries>true</loadsLibraries>
                    </Element>
                    <Element>
                          <consumerName text="GameLighting"/>
                          <relativeArtDefPaths>
                          </relativeArtDefPaths>
                          <libraryDependencies>
                                  <Element text="ColorKey"/>
                                  <Element text="GameLighting"/>
                          </libraryDependencies>
                          <loadsLibraries>true</loadsLibraries>
                    </Element>
                    <Element>
                          <consumerName text="StrategicView_Properties"/>
                          <relativeArtDefPaths>
                          </relativeArtDefPaths>
                          <libraryDependencies/>
                          <loadsLibraries>false</loadsLibraries>
                    </Element>
                    <Element>
```

```xml
            <consumerName text="StrategicView_Sprite"/>
            <relativeArtDefPaths>
                    <Element text="Buildings.artdef"/>
            </relativeArtDefPaths>
            <libraryDependencies>
                    <Element text="StrategicView_Sprite"/>
                    <Element text="StrategicView_DirectedAsset"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="StrategicView_Route"/>
            <libraryDependencies>
                    <Element text="StrategicView_Route"/>
                    <Element text="StrategicView_DirectedAsset"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="StrategicView_TerrainType"/>
            <libraryDependencies>
                    <Element text="StrategicView_TerrainBlend"/>
                    <Element text="StrategicView_TerrainBlendCorners"/>
                    <Element text="StrategicView_TerrainType"/>
                    <Element text="StrategicView_DirectedAsset"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="StrategicView_TerrainBlendCorners"/>
            <libraryDependencies>
                    <Element text="StrategicView_TerrainBlendCorners"/>
                    <Element text="StrategicView_DirectedAsset"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="StrategicView_TerrainBlend"/>
            <libraryDependencies>
                    <Element text="StrategicView_TerrainBlend"/>
                    <Element text="StrategicView_DirectedAsset"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Terrain"/>
            <libraryDependencies>
                    <Element text="TerrainAsset"/>
                    <Element text="TerrainElement"/>
                    <Element text="TerrainMaterial"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="WorldViewRoutes"/>
            <libraryDependencies>
                    <Element text="RouteDecalMaterial"/>
                    <Element text="RouteDoodad"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="UI"/>
            <libraryDependencies>
                    <Element text="UITexture"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="FOW"/>
            <libraryDependencies>
                    <Element text="FOWSprite"/>
                    <Element text="FOWTexture"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="WonderMovie"/>
            <libraryDependencies>
                    <Element text="WonderMovie"/>
                    <Element text="TileBase"/>
                    <Element text="GameLighting"/>
                    <Element text="ColorKey"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="UILensAsset"/>
            <libraryDependencies>
                    <Element text="OverlayTexture"/>
                    <Element text="UILensAsset"/>
            </libraryDependencies>
```

```xml
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Overlay"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies>
                    <Element text="OverlayTexture"/>
                    <Element text="UILensAsset"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="VFX"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies>
                    <Element text="VFX"/>
                    <Element text="Light"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Water"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies>
                    <Element text="Water"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="ColorKeys"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies>
                    <Element text="ColorKey"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="ScreenWashEffects"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies>
                    <Element text="ColorKey"/>
            </libraryDependencies>
            <loadsLibraries>true</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Camera"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Terrains"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Features"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Civilizations"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Cultures"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Resources"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="Improvements"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="WorldView_Translate"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
    <Element>
            <consumerName text="StrategicView_Translate"/>
            <relativeArtDefPaths>
            </relativeArtDefPaths>
            <libraryDependencies/>
            <loadsLibraries>false</loadsLibraries>
    </Element>
```

```xml
<Element>
        <consumerName text="Audio"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
<Element>
        <consumerName text="LeaderLighting"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies>
                <Element text="LeaderLighting"/>
                <Element text="ColorKey"/>
        </libraryDependencies>
        <loadsLibraries>true</loadsLibraries>
</Element>
<Element>
        <consumerName text="Leaders"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies>
                <Element text="Leader"/>
                <Element text="LeaderLighting"/>
                <Element text="ColorKey"/>
        </libraryDependencies>
        <loadsLibraries>true</loadsLibraries>
</Element>
<Element>
        <consumerName text="LeaderFallback"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies>
                <Element text="LeaderFallback"/>
        </libraryDependencies>
        <loadsLibraries>true</loadsLibraries>
</Element>
<Element>
        <consumerName text="Lenses"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
<Element>
        <consumerName text="IndirectGrid"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
<Element>
        <consumerName text="AOSystem"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
<Element>
        <consumerName text="GenericObject"/>
        <relativeArtDefPaths/>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
<Element>
        <consumerName text="Wave"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies>
                <Element text="Wave"/>
        </libraryDependencies>
        <loadsLibraries>true</loadsLibraries>
</Element>
<Element>
        <consumerName text="DynamicGeometry"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies>
                <Element text="DynamicGeometry"/>
        </libraryDependencies>
        <loadsLibraries>true</loadsLibraries>
</Element>
<Element>
        <consumerName text="UIPreview"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
<Element>
        <consumerName text="SkyBox"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies>
                <Element text="SkyBoxTexture"/>
        </libraryDependencies>
        <loadsLibraries>true</loadsLibraries>
</Element>
<Element>
        <consumerName text="Minimap"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
<Element>
        <consumerName text="UnitSimulation"/>
        <relativeArtDefPaths>
        </relativeArtDefPaths>
        <libraryDependencies/>
        <loadsLibraries>false</loadsLibraries>
</Element>
```

```xml
        <Element>
                <consumerName text="RangeArrows"/>
                <relativeArtDefPaths>
                </relativeArtDefPaths>
                <libraryDependencies>
                        <Element text="OverlayTexture"/>
                        <Element text="UILensAsset"/>
                </libraryDependencies>
                <loadsLibraries>true</loadsLibraries>
        </Element>
</artConsumers>
<gameLibraries>
        <Element>
                <libraryName text="CityBuildings"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="ColorKey"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="DynamicGeometry"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="FOWSprite"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="FOWTexture"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="GameLighting"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="Landmark"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="Leader"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="LeaderFallback"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="LeaderLighting"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="Light"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="OverlayTexture"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="RouteDecalMaterial"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="RouteDoodad"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="SkyBoxTexture"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="StrategicView_DirectedAsset"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="StrategicView_Route"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="StrategicView_Sprite"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="StrategicView_TerrainBlend"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
        <Element>
                <libraryName text="StrategicView_TerrainBlendCorners"/>
                <relativePackagePaths>
                </relativePackagePaths>
        </Element>
```

```xml
<Element>
        <libraryName text="StrategicView_TerrainType"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="TerrainAsset"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="TerrainElement"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="TerrainMaterial"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="TileBase"/>
        <relativePackagePaths>
                <Element text="landmarks/hero_buildings"/>
                <Element text="landmarks/tilebases"/>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="UILensAsset"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="UITexture"/>
        <relativePackagePaths>
                <Element text="UI/Icons"/>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="Unit"/>
        <relativePackagePaths>
                <Element text="units/units"/>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="VFX"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="Water"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="Wave"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
<Element>
        <libraryName text="WonderMovie"/>
        <relativePackagePaths>
        </relativePackagePaths>
</Element>
        </gameLibraries>
        <requiredGameArtIDs>
        <Element>
                <name text="Civ6"/>
                <id text="cb2f71b7-843e-4af3-9ca7-992acda9c195"/>
        </Element>
        </requiredGameArtIDs>
</AssetObjects::GameArtSpecification>
```

- If you did not need the whole code, there are 3 sections you will need to edit. See image below. You can also copy the pieces of the code above that you need after identifying them in the image below.
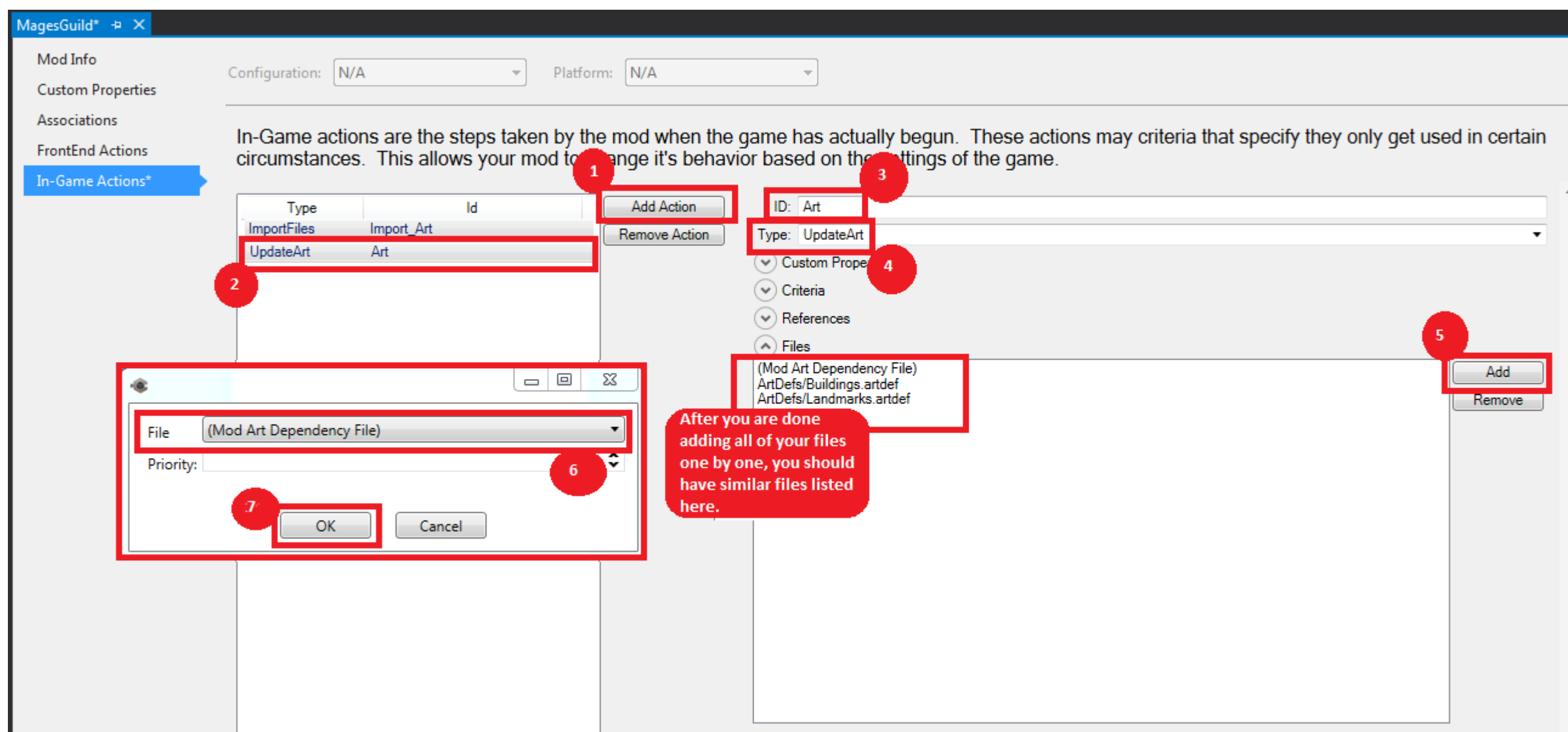

- Click FILE and select Save All.
4) On the top menu bar click the PROJECT tab, and select Properties (last option in the drop down menu). Click In-Game Actions, then click Add Action, left click the new action you just added to select, name it something like Import_Art and select Import for the Type. Under the Files section click Add, click the drop down menu for File and select one of your UV map texture files, then click Add. Repeat for every one of your UV map texture files, including both the DDS format one and the TEX format one. You also need to add your .fgx and your .geo geometry files of your 3D Model, as well as your .fgx and your .geo geometry files of your obstruction profile if you made one. Then add your 3D Model's Material file (.mtl). Finally add the Asset file (.ast) for your 3D Model and for your Base. See image below for a general list of what you should have added.
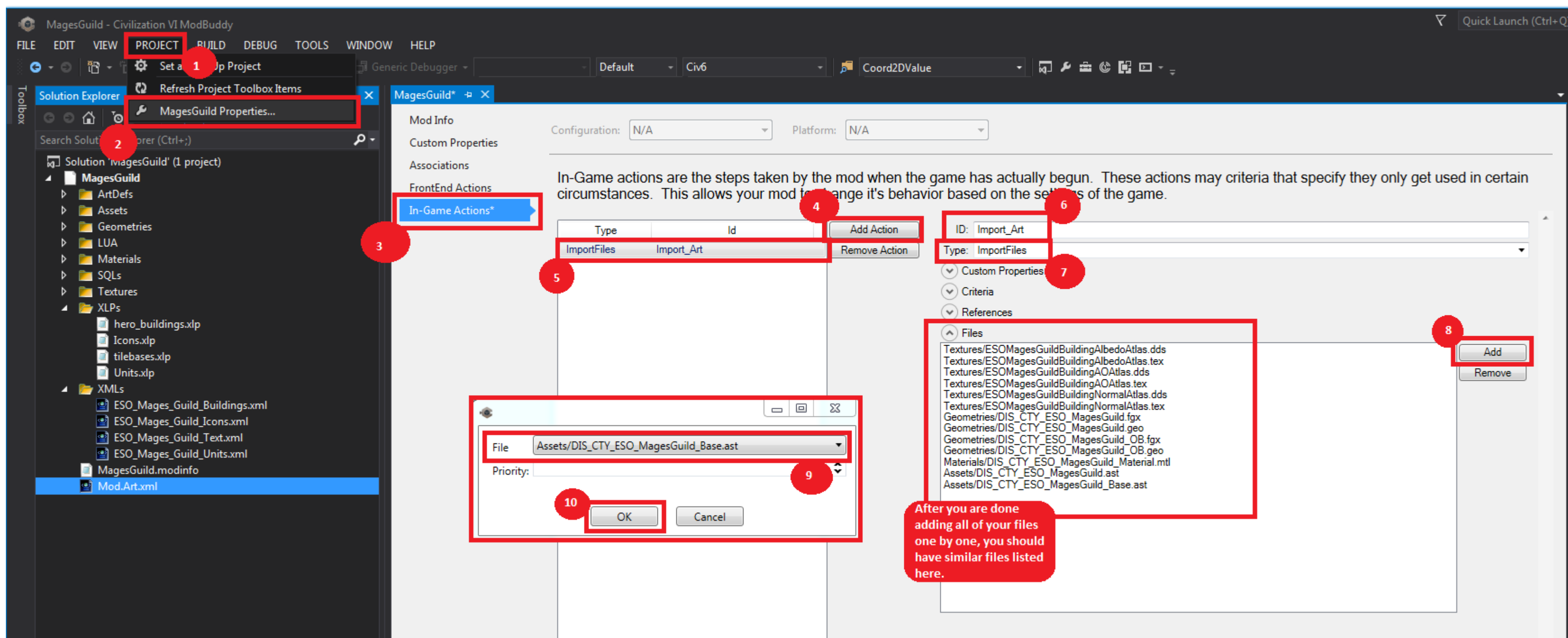
- Click Add Action, left click the new action you just added to select, name it something like Art and select UpdateArt for the Type. Like you did above, add a file called (Mod Art Dependency File), then add your Buildings.artdef and your Landmarks.artdef.

- Note: If you are wondering, you do not have to add the XLP files or the GeometrySet.xml files to In-Game Actions, those files are called automatically by other files.
- Click FILE and select Save All. Click BUILD and select Build Project.

5) The first thing you need to check now is if your 3D Model displays properly in Asset Previewer. Open AssetEditor again, click the TOOLS tab found on the top menu bar and select Launch Asset Editor. Click File, then select Open Entity, under Assets uncheck Civ6, select the asset file for your 3D Model. You should see your complete textured building to the right in the Asset Previewer. The pink cage around it represents the Obstruction Profile. If you did not build your 3D model off of a base game 3D model, then this is where you can see how your positioning in Blender translates in game. If you need to make any positioning changes, then you will need to go back in to Blender, reposition it in Edit Mode, then redo the exporting part of section 3 and all of section 4.
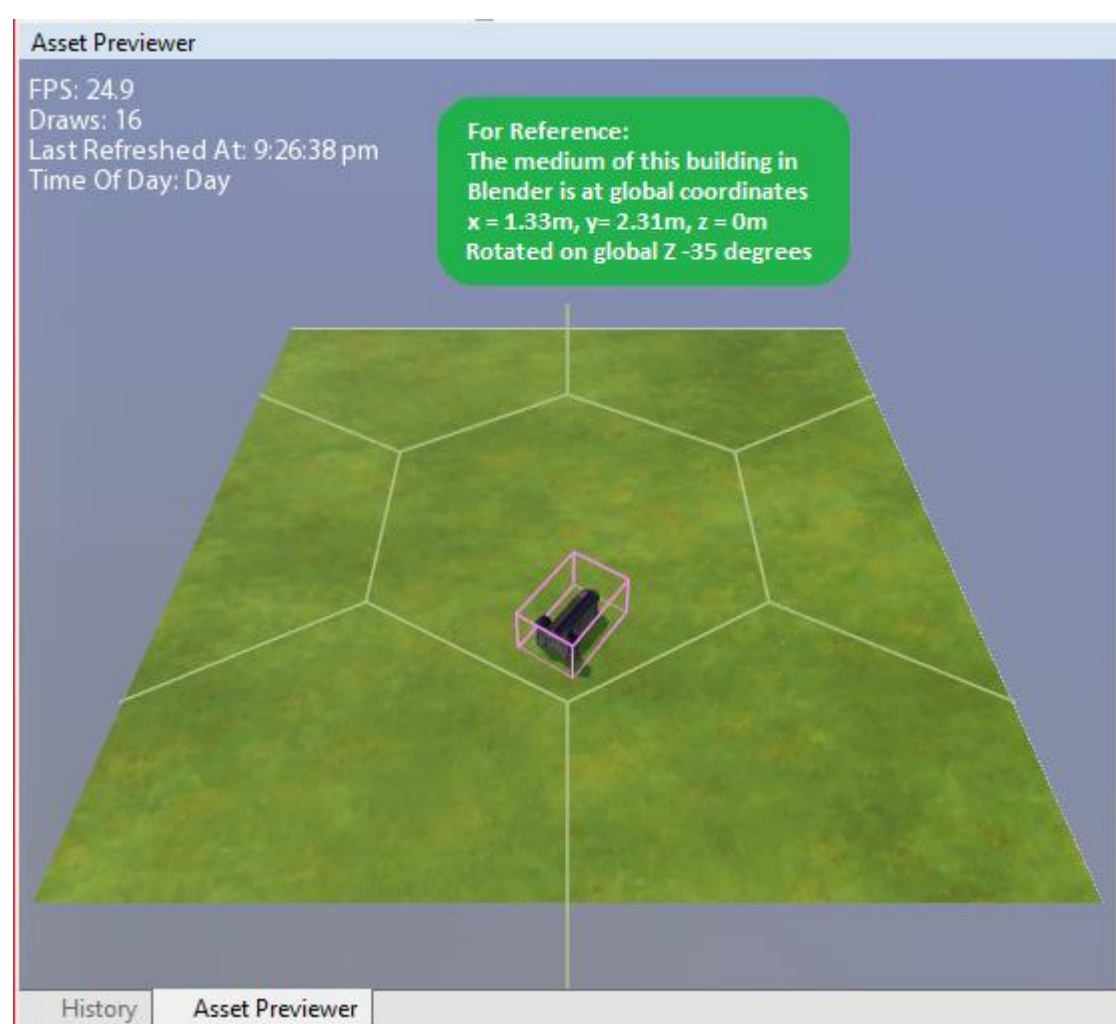
- Note: If you see a giant exclamation mark in Asset Previewer instead of your 3D Model then there is a problem. Go back through section 5 and make sure you set everything up properly. Make sure there is no other Geometries in the Asset file, other than your 3D Model's geometry. Make sure the name of your geometry starts with something like [DIS_CTY_ESO_MagesGuild (Root) Vertex...], Root being the name of your Armature Object, we set this value in step 2. Make sure there are no Animations, Attachments, Particles, or Behaviors in the asset file. Try removing the Obstruction Profile. If you still can't fix it post your issue on a Civilization VI modding form. Remember to save it every time you make a change, because your change is not reflected in Asset Previewer until you save it.
6) After you get your 3D Model to show up in Asset Previewer, go back in to ModBuddy, click FILE and select Save All. Click BUILD and select Build Project. Start up your Civilization VI game and see if your 3D Model shows up in game. If it is not showing up go back through sections 5 and 6 and make sure you did everything correctly. If you still can't fix it post your issue on a Civilization VI modding form. Remember to save and build your project every time you make a change.
7) And that concludes section 6, your 3D Model now shows up in game.

Conclusion

And that concludes this guide. You now know how to create your own 3D Art of a building and add it to your ModBuddy Project. Below in the special thanks section is a list of people on the civfanatics.com forum who helped me learn this stuff and are always there to help and answer questions. I also included some helpful threads they run as well where you can go and ask questions.

Special Thanks

- Deliverator
  - Deliverator's 3D Art Assets in Civilization 6: A Modding Guide
    - https://forums.civfanatics.com/threads/3d-art-assets-in-civilization-6-a-modding-guide.612050/
  - Deliverator's Rough Guide to Units.artdef
    - https://forums.civfanatics.com/threads/rough-guide-to-units-artdef.603687/
- sukritact
  - A Quick Intro to PBR/Civ 6 Textures
    - https://forums.civfanatics.com/threads/a-quick-intro-to-pbr-civ-6-textures.612055/
- CivilizationAce
  - CivilizationAce's How to make a civilization icon in way too many easy steps
    - https://docs.google.com/document/d/1fFGEAJwyFaLmY31b6JnkI1rmu8Q07zlhuJU_2DcJ4dE/edit
- LeeS
  - LeeS' Dummy Buildings Systems
    - https://forums.civfanatics.com/resources/lees-dummy-buildings-systems.26207/
- Gedemon
  - Gedemon's Lua Objects Spreadsheet
    - https://forums.civfanatics.com/threads/lua-objects.601146/
- RFormica
  - RFormica's Modding Help Thread
    - https://forums.civfanatics.com/threads/rformicas-modding-help-thread.621857/#post-14868738
- FurionHuang
- lasttry
- Wolfdog
- thecrazyscot