# 3D modeling for real-time visualization

## Table of Contents



## Introduction

The workshops seek to provide an overview of the best practices in 3D modeling aimed to produce detailed, clean, and light models for real-time visualization. Topics covered in the workshops range from manual (parametric) to automated and procedural modeling. Attendees will also learn how to import their optimized 3D models in Unity game engine and build an interactive walk-through application for the Virtual Reality headset Oculus Rift. Theoretical concepts will also be provided throughout the tutorial along with hands-on activities.

## Software and material

- **Blender** (Free download: https://www.blender.org/)
- **MeshLab** (Free download: http://meshlab.sourceforge.net/)
- **MeshMixer** (Free download: http://meshmixer.com/)
- **ReMesh** (http://remesh.sourceforge.net/)

Download tutorial material from: https://duke.box.com/v/3D-WS2



Dig@Lab | Duke University
Created: April 20, 2016
Revised: September 26, 2017
nevio.danelon@duke.edu

# Re-topology and simplification techniques

Editing very complex 3D objects could be extremely difficult and Constructive Solid Geometry techniques (see WS1 handout, p. 3) are not always successful when a mesh is made of too many polygons. In this case, organize your model as a set of intersecting independent objects. Then perform both **polygonal resampling** and **polygonal simplification** to produce a homogenous single surface.

## Polygonal resampling (re-topology)

**Re-topology** – also known as polygonal resurfacing or re-meshing – is the reorganization of the polygons in a mesh in order to make a model watertight for 3d printing or – in our specific case – to make it suitable for a perfect decimation and UV-mapping. Indeed, sometimes it is easier to make a new model around the old one instead of trying to fix the topology of a clumsy model.

## Polygonal reduction (decimation)

Once you have a perfectly closed mesh, you are also able to apply a homogeneous reduction preserving the general topology of the model. You can create different **Levels of Detail** (LODs, see WS1 handout, p. 2) changing the parameters controlling the decimation percentage (Figure 1).

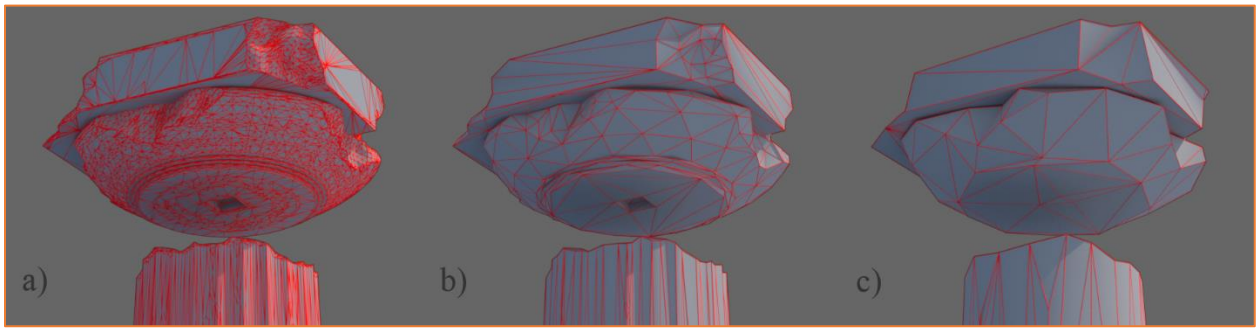



*Figure 1: Levels of Detail [High (a), Medium (b), and Low (c)].*

## Parametric reduction

Another way to create LODs of a 3D model consists in creating a model in a parametric mode and then obtain different instances of it just changing the interpolation steps of the Bezier curves (see WS1 handout, p. 15).

## Manual resurfacing

A much more time-consuming way to get your LODs is **Snap modeling**. This technique consists in resurfacing (rewrapping) your model with a handmade low-poly mesh, by setting snap to vertex mode. Snapping allows an object to be easily positioned in a 3D space by causing it to automatically jump to an exact position when the user drags it in proximity of another vertex. This technique is possible only if you have a quite simple underlying geometry.

## UV mapping

In order to bake the textures (see below) of your model, you would need to properly UV-unwrap and UV-map it. These operations are needed to make the baked texture perfectly fit your model. **UV-Mapping** specifies how textures are projected onto an object. While X Y Z coordinates refer to the Cartesian axes in which the object is placed, U V coordinates refer to the axes of the 2D texture and W is used only when assigning procedural materials to a 3D object.

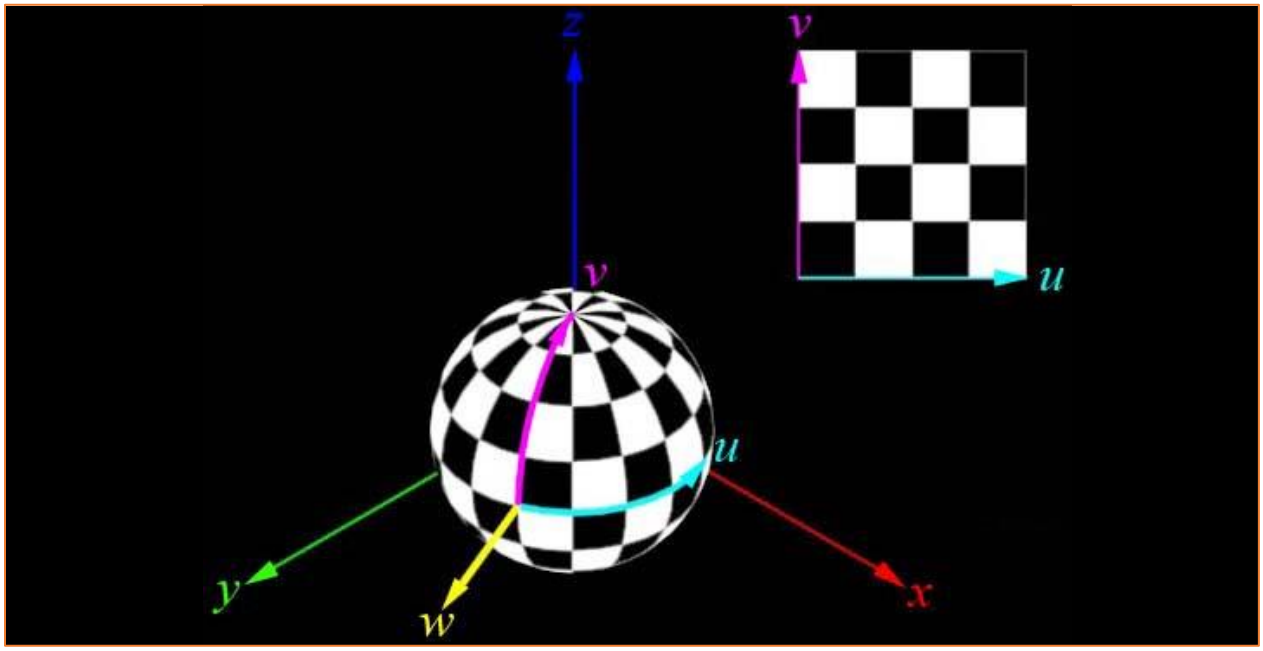



*Figure 2: UVW coordinates on a sphere.*

**UV-Unwrap** is an advanced mapping technique that consists in projecting a mesh onto a plane in order to make the texture mapping easier. This procedure creates a new set of UV-coordinates that will be used in baking process.

## Texture baking

"**Baking**" – also Render to Texture – generally refers to the process of mapping some surface details on an image texture. Complex meshes and materials can take longer to compute in real-time visualization than a texture. The baking operation is usually done once a high-poly model is finalized. The features of your 3D model that can be projected on a texture are: small geometric details (baked onto a bump or displacement map); complex materials (baked onto a specular or alpha map); and lights (direct and indirect illumination is backed onto a light map). The main types of textures involved in multi-texture materials are:

- **Color map** is an RGB image describing the visible color components of the object surface, without considering the effects of direct and indirect lights (shadows and ambient occlusion).
- **Normal map** is an RGB image where the color components correspond to the X, Y, and Z coordinates of each surface normal. This map affects how light is reflected from the model surface in order to simulate small geometric details.

- **Height map** – also displacement map – is a grey-scale image that represents the offset between the low-poly and high-poly surface of a model, with black representing minimum height and white representing maximum height. It can be used both for parallax and displacement effects.
- **Light map** is a grey-scale image that represents direct and/or indirect light effects on a surface. If the light sources in real-time rendering move (e.g. sun at different time/season), only indirect shading (ambient occlusion) must be baked and direct shadows will be calculated by the engine in real-time.
- **Specular map** is a grey-scale image that controls how much light is reflected from a surface. Materials like polished marble or metal can be simulated in this way.
- **Alpha map** is a grey-scale image used to create the appearance of partial or full transparency on an object surface. It is used to simulate glass surfaces or cutout objects (e.g. tree leaves).

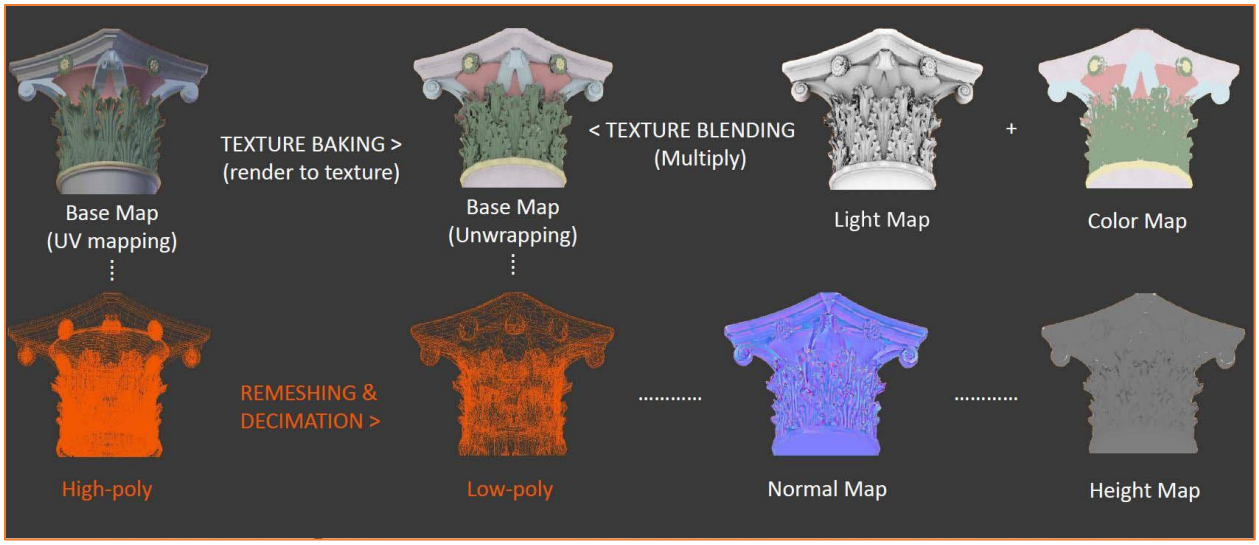



*Figure 3: Texture baking of a Corinthian capital.*

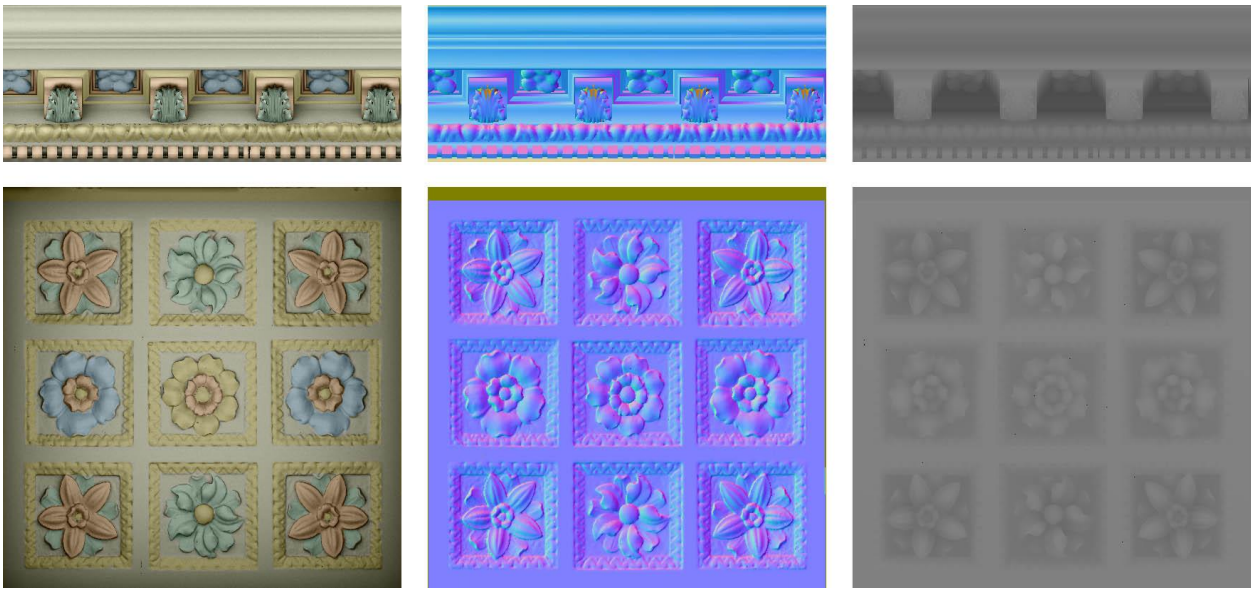
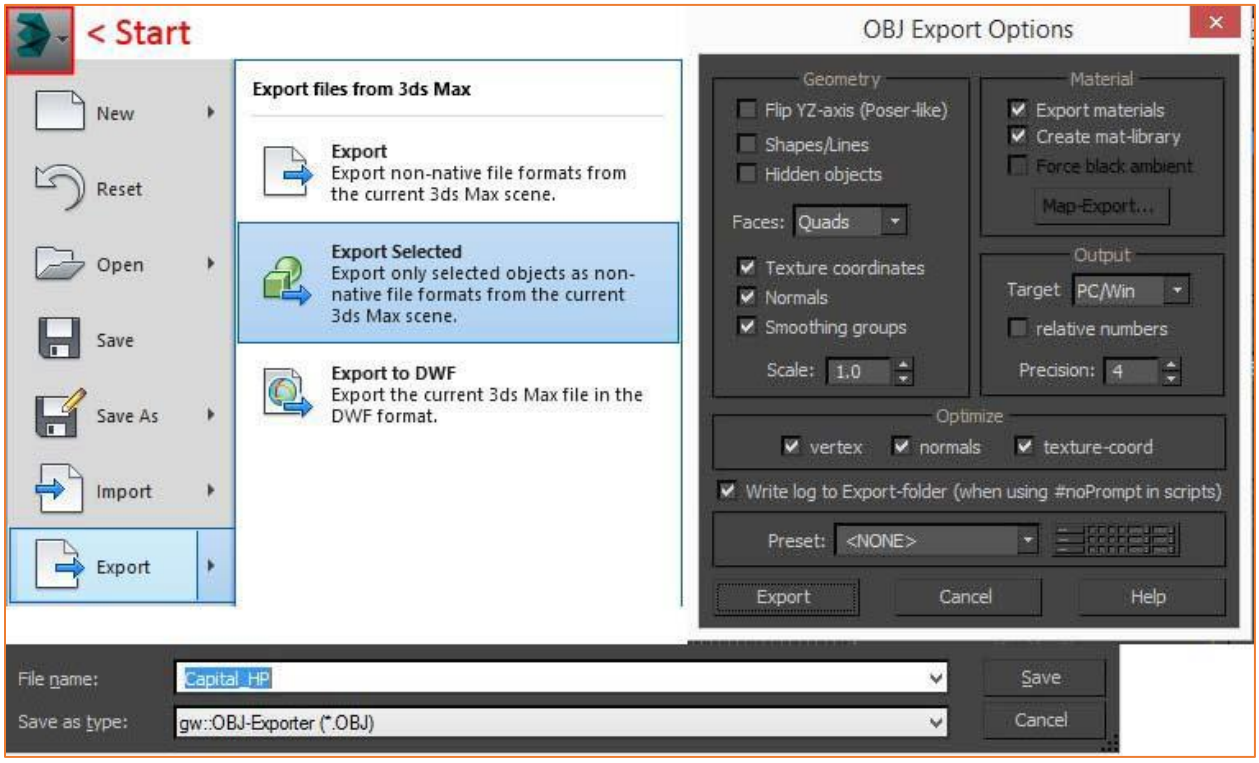


*Figure 4: Color, Normal and Height Map.*

# Tutorial 1: Resurfacing and simplification

This tutorial does not address 3D models creation at large, but focuses on the optimization process and low polygonal modeling techniques. Once you have created your high detailed model (hereafter referred as **High-poly** model), export it as .OBJ or other 3D file format.

1. Open the WIP / WS2_tut01 folder and double click ***Step01_ColumnMax***
2. In 3dS Max, Select the object ***Capital_HP*** (where **HP** stands for High-poly model)
3. Go to **Start** menu / **Export** / **Export Selected**
4. Save as "***Capital_HP***" and select the file format OBJ in the drop-down menu
5. In the **Export Options** panel, leave all the settings as they are and export the model in your project folder.

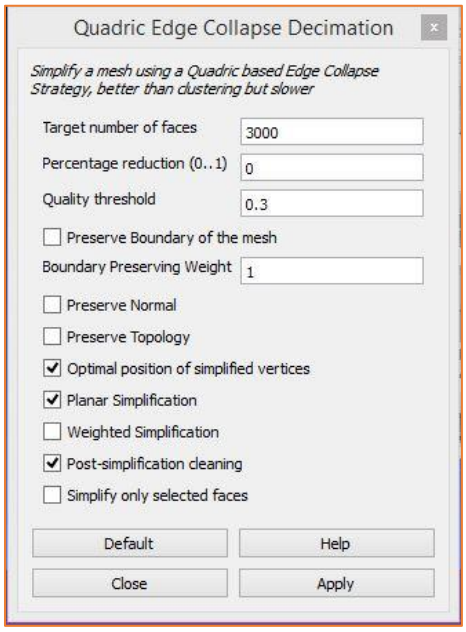



6. Open **Project Miller** software

**Project Miller** was a free Autodesk Labs technology preview that allowed you to resurface and optimize your design for 3d printing. Unfortunately, it was discontinued on July 10, 2014 and it is no more available for downloading. Potential replacements could be found in **MeshMixer** (http://meshmixer.com/) in **Blender** *Remesh* modifier (https://www.blender.org/manual/modeling/modifiers/generate/remesh.html) or in **ReMESH** *Resampling* algorithms (http://remesh.sourceforge.net/).

7. Select **Import** and browse your computer to the model ***Capital_HP.obj*** (created in 3dS Max)
8. Select **Re-surface** [chose option Medium (1024)]
9. Select **Clipping planes** to check if there are double faces or unwanted under-surface geometry
10. Select **Re-mesh** [chose Optimized 50K (50K = 50,000 triangles)]
11. Select **Export**, name your model "***Capital_LOD0***" and chose OBJ format

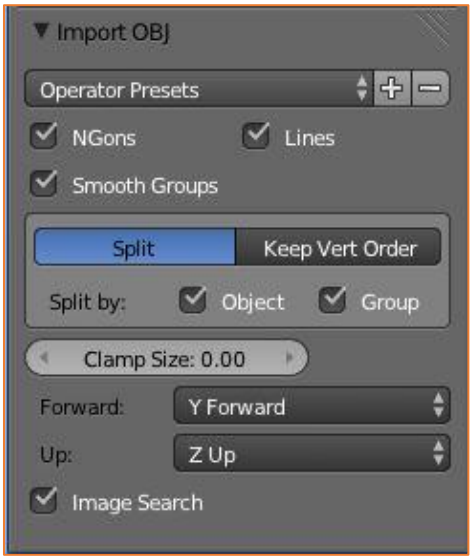
12. **Re-surface** again [chose option Coarse (512)]
13. Select **Export**, name your model "***Capital_LOD1***" and chose OBJ format
14. Open **MeshLab** software
15. Select **File** / **Import Mesh** and select ***Capital_LOD1.obj***
16. If your model looks dark, it means that its faces' normals are inverted and they must be flipped. To do so, go to **Filters** / **Normals, Curvature and Orientation** / **Invert Faces Orientation** and then **Save** your model (the edited model will replace the old one).
17. Now you want to further reduce the number of faces. Go to **Filters** / **Remeshing, Simplification and Reconstruction** / **Quadratic Edge Collapse Decimation** and set **Target Number of Faces** to **3000**. Check **Planar Simplification** option and **Apply**



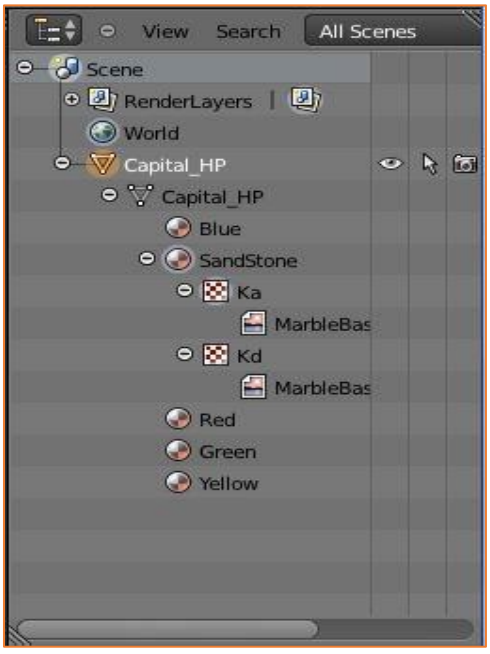

18. **Export** the decimated model as "***Capital_LOD2***" and chose OBJ format.

# Tutorial 2: Unwrapping and baking the textures ([1])

1. Open the WIP / WS2_tut01 folder and double click ***EmptyScene.blend*** file
2. **File** / **Import** / **Wavefront (obj)** and browse your computer and select ***Capital_HP.obj***
3. On the left side of the **Import OBJ** panel select **Y forward** (Z Up), as shown below:

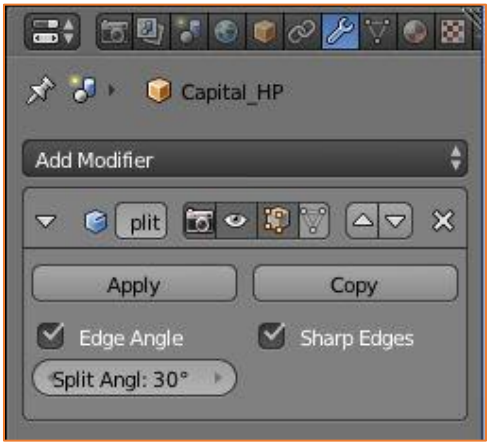



4. In the central (main) pane of the **Import OBJ** panel, double click the 3D model ***Capital_HP*** (the Import OBJ panel will close and the imported model will be displayed)
5. Now check that your model's settings shown in the Menu on the right are as follows:



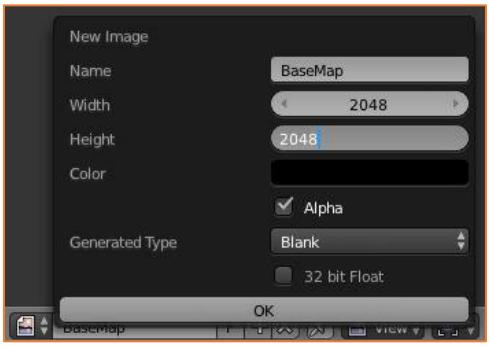

[1] The same procedure can be done in 3ds Max using the **Unwrap UVW** modifier and Rendering / **Render To Texture**.

6. If your model looks too smooth, apply **Edge Split** modifier in the **Properties Menu** shown on the right. See the following image as a reference:

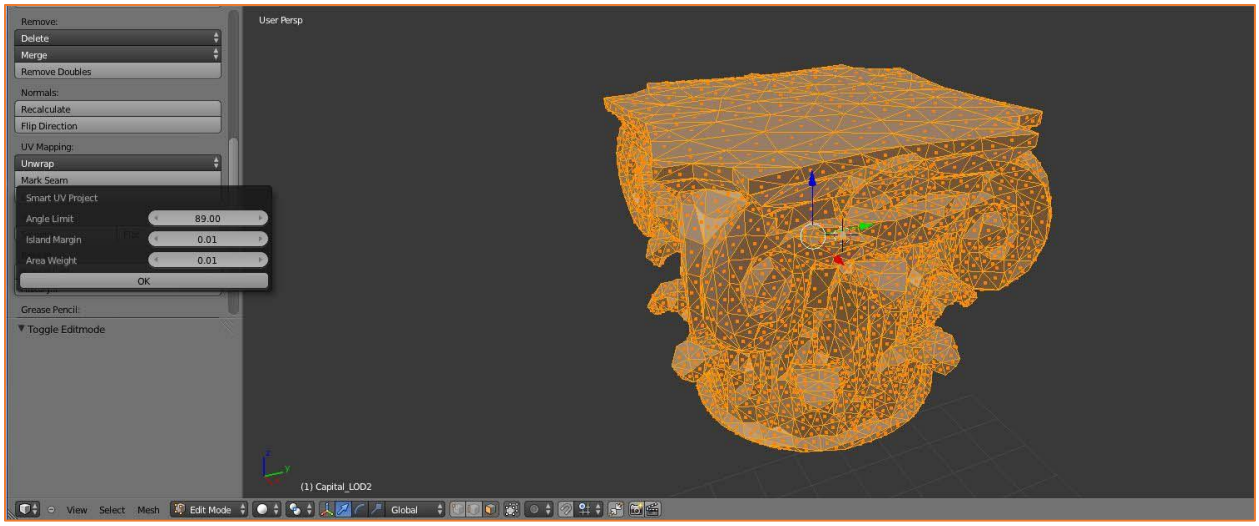



7. Click on a new layer (an empty gray square) on the 3D View lower header as shown below:

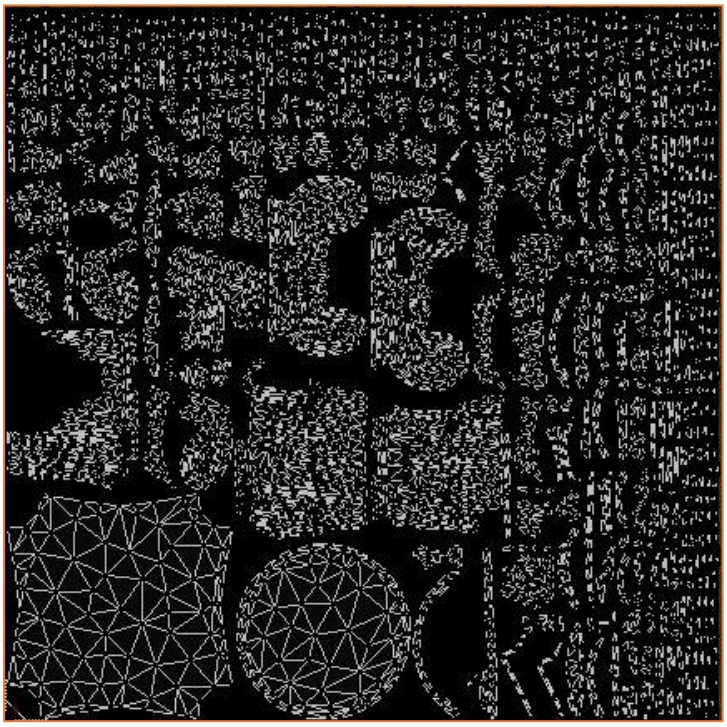
8. **File** / **Import** / **Wavefront (obj)** and browse your computer and select ***Capital_LOD0.obj***
9. Repeat step 7 and 8 above for ***Capital_LOD1.obj*** and for ***Capital_LOD2.obj***
10. You should now have four models in four separate layers. Otherwise, just locate in the WS2_tut01 folder the file ***Step02_CapitalBlend.blend*** and open the new scene which is ready for baking
11. Create a new image clicking the **+ New** button in the lower editor header and set the following values: Name = "**BaseMap**"**;** Size = **2048 × 2048** pixels ("Power of Two" data optimization rule) and press **OK**



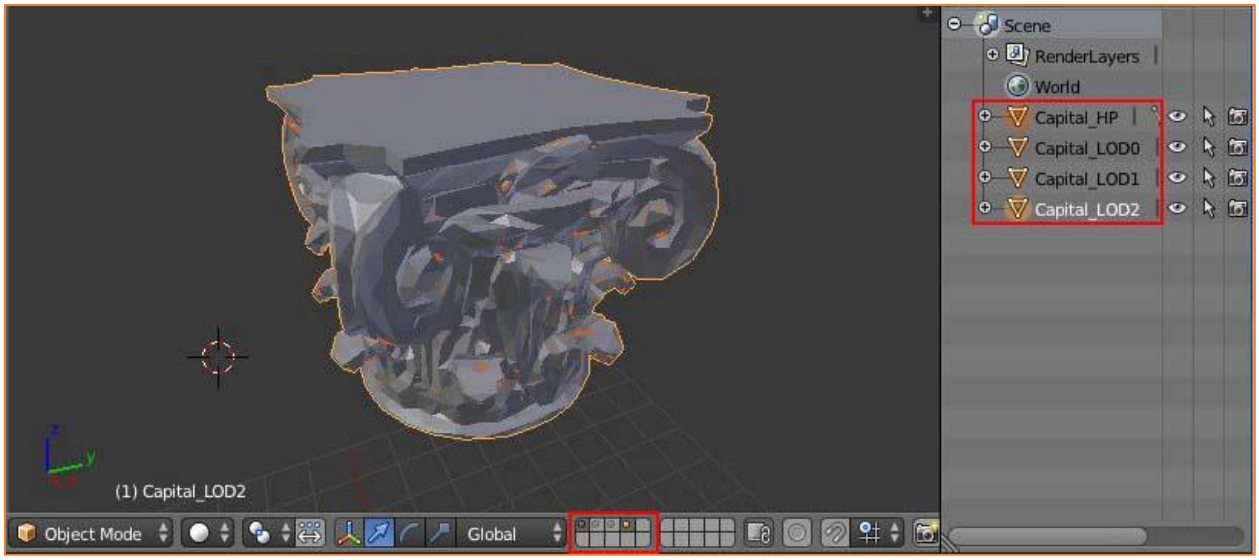

12. In the Layer viewer, select the **fourth gray square** where the model **Capital_LOD2** is stored (the main scene viewport will display **Capital_LOD2** model)
13. Make sure that only *Capital_LOD2* is selected (type "**A**" to select/unselect all), click on **Object Mode** in the 3D view header and select **Edit Mode** in the drop-down Menu (Blender has several modes for different types of editing)
14. In Edit Mode, all the faces of your model should be selected (visualized as orange triangles). Otherwise, press "**A**" again to select/unselect all.
15. In the panel **Shading/UVs** (on the left), scroll down to **UV Mapping** section and click on **Unwrap**. In the pop-up window, select **Smart UV Project** (make sure not to select Unwrap!). Set values as follows: Angle limit = **90** (89.00 is fine!); Island Margin = **0.01;** Area Weight 0.01 and click **OK**



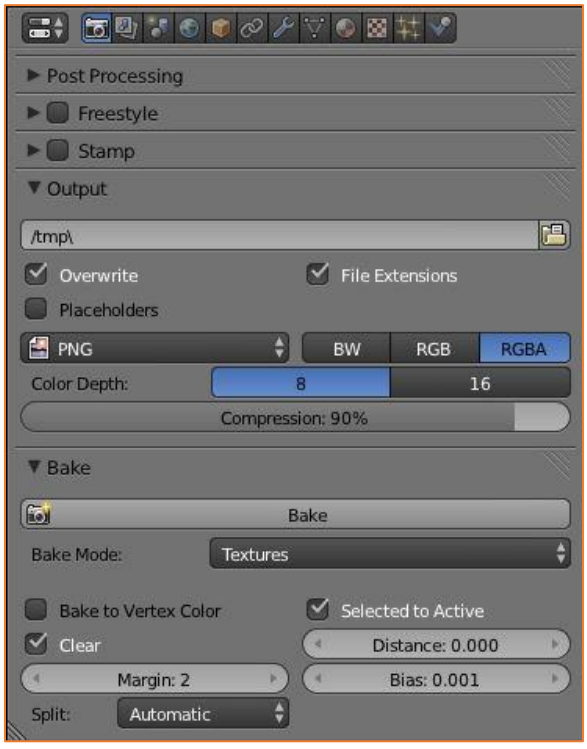

16. Now you should see your black *BaseMap* showing the unwrapped *Capital_LOD2*. Otherwise, select ***BaseMap*** from the image browser in the lower editor header as shown in the following picture

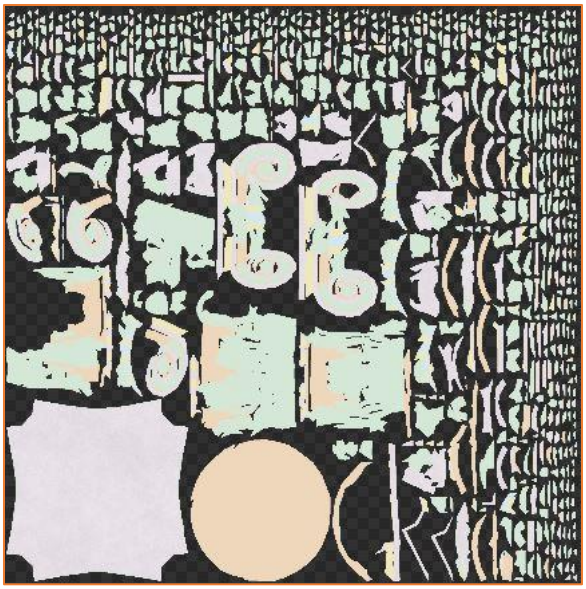
17. Switch back to **Object Mode** (see step 13 above), select the first layer in the Layer Viewer (*Capital_HP* model becomes visible) and then select ***Capital_HP model*** in the **Outliner** panel on the right or right-click directly on the object in the viewport
18. Holding down **Left Shift**, select the **fourth layer** and select ***Capital_LOD2 model*** in the **Outliner** panel on the right. Finally release **Left Shift** key. You should now see 2 models selected as shown in the picture below:



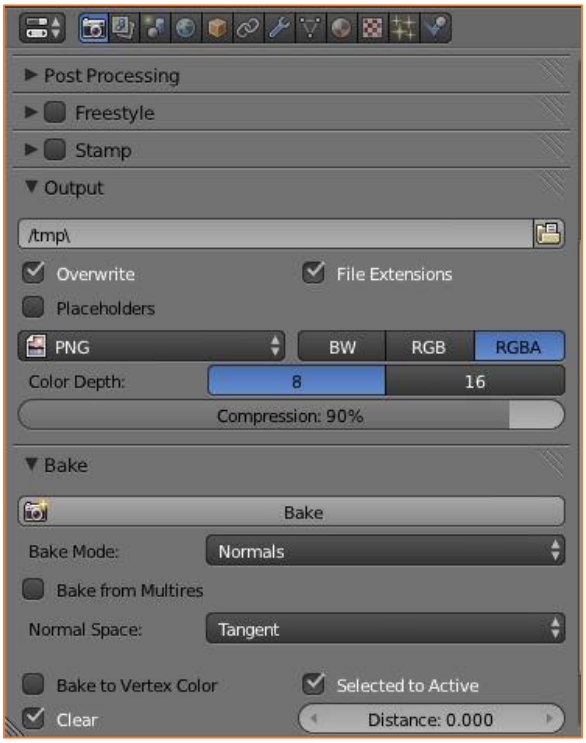

19. In the **Properties** panel (below the Outliner), click on the Camera icon (**Render tab**) and then expand the **Baking** section at the end. Then set the following values: Bake Mode = **Texture**; Margin = **2** (pixels); check **Selected to Active** (as shown in the picture below); and press **Bake** button



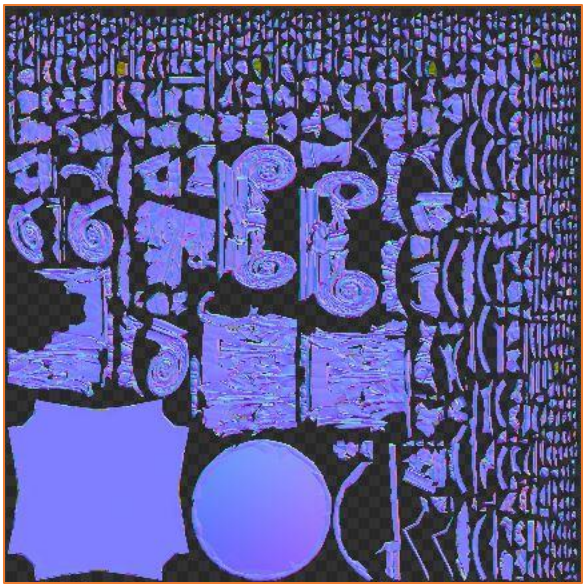

20. In the **UV/Image Editor** (look at the lower part of the screen), you should now see the *ColorMap* being created (see below)



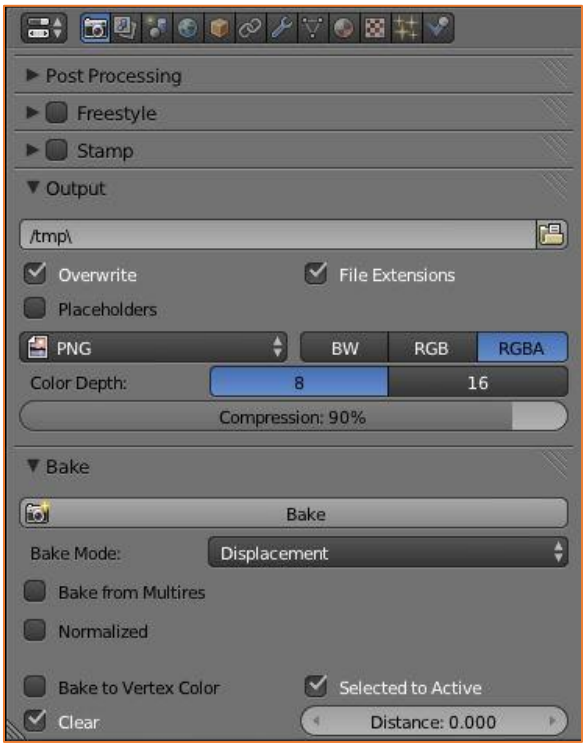
21. When completed, save it to your computer clicking the **Image*** button in the lower UV/Image Editor header and **Save As Image.** Name it "***Capital_LOD2_ColorMap***" choosing PNG as the file format
22. In the **Properties** panel, in the **Render** tab**,** expand the **Baking** and change **Bake Mode** selecting **Normals** from the drop-down menu. Make sure that the Normal Space is set to **Tangent** (see below) and then press **Bake** button again




23. In the lower UV/Image Editor, you should now see the NormalMap being created (see below)

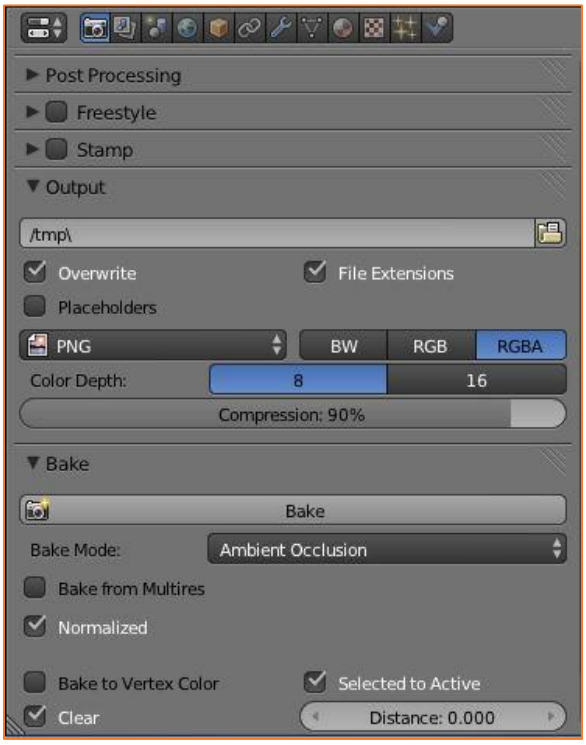


24. When completed, save it to your computer clicking the **Image*** button in the UV/Image Editor header**.** Name it "***Capital_LOD2_NormalMap***" choosing .PNG as the file format
25. In the **Properties** panel, in the **Render** tab**,** expand the **Baking** section, change **Bake Mode** selecting **Displacement** from the drop-down menu, and press **Bake** button again

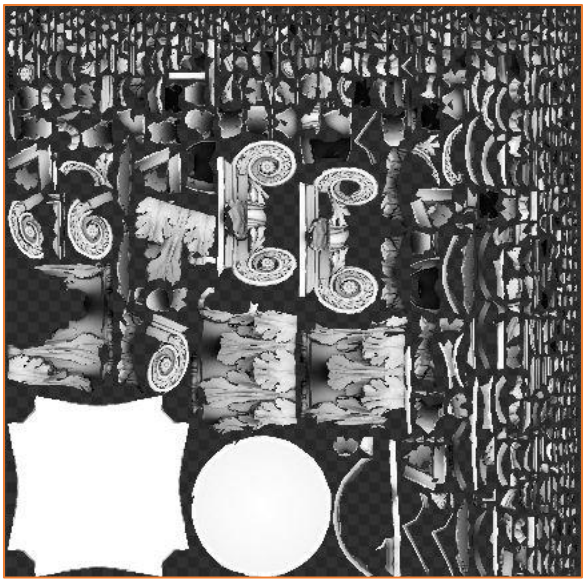



26. In the lower UV/Image Editor, you should now see the HeightMap being created (see below):

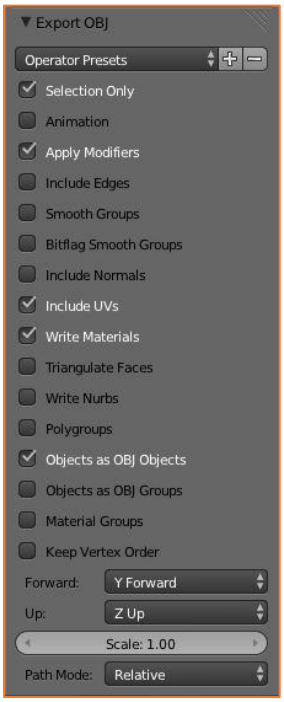
27. When completed, save it to your computer clicking the **Image*** button in the UV/Image Editor header. Name it "***Capital_LOD2_HeightMap***" choosing .PNG as the file format
28. In the **Properties** panel, in the **Render** tab**,** expand the **Baking** section and change **Bake Mode** selecting **Ambient Occlusion** from the drop-down menu. Then tick **Normalized** option and press **Bake** button again




29. In the lower UV/Image Editor, you should now see the LightMap being created (see below):



30. If the generated LightMap is too grainy, select the world icon (**World** tab) in the **Properties** panel and make sure that **Ambient Occlusion** option is checked. In the **World** tab locate the **Gather** section and set **Samples** to **12** or more (the higher the number the longer it will take to create the LightMap)
31. When LightMap is completed, save it to your computer clicking the **Image*** button in the lower UV/Image Editor header. Name it "***Capital_LOD2_LightMap***" choosing .PNG as the file format
32. Select the **fourth layer** only and select the **Capital_LOD2** model in the **Outliner** or directly right-click on the object in the viewport
33. Go to **File** / **Export** / **Wavefront** (.obj), name the file "***Capital_LOD2_baked***" and make sure all of the following settings are ticked:



34. Repeat the entire process from step 12 (use the third and second layer) to 33 for **Capital_LOD1** and **Capital_LOD0**. Otherwise, open the WIP folder and double-click the file *Step06_ColumnBlend.blend* to find **all the LODs** already unwrapped and mapped for you. For highly detailed (LOD1) model, the Height Map is unnecessary; for very high-detailed objects instead, only Color and Light Maps are required
35. As a last step of the Baking process, create a BaseMap, merging Color- and LightMaps together in a single RGB image using your favorite image-editing program (e.g. Adobe Photoshop or Gimp). While merging the above maps, make sure that the blending option you will use are either **Multiply** or **Brightness**. You can find pre-made BaseMaps for all the LOD models /Materials/Files (**Capital_BaseMap_LOD0.png**, **Capital_ BaseMap_LOD1.png** and **Capital_ BaseMap_LOD2.png)**.

## Tutorial 3: Create a basic scene in Unity and import your models ([2])

Unity 3D is one of the most popular game development ecosystem available on the market for independent production and mobile production. This multiplatform software is available in two versions basic and pro, the first being downloadable for free and the second one being commercial. Unity 3D is modular and one can buy different plugins to develop games and real-time applications for Win, Mac, Web, Android, iOS, Xbox, Playstation, Windows Mobile, Blackberry, etc.

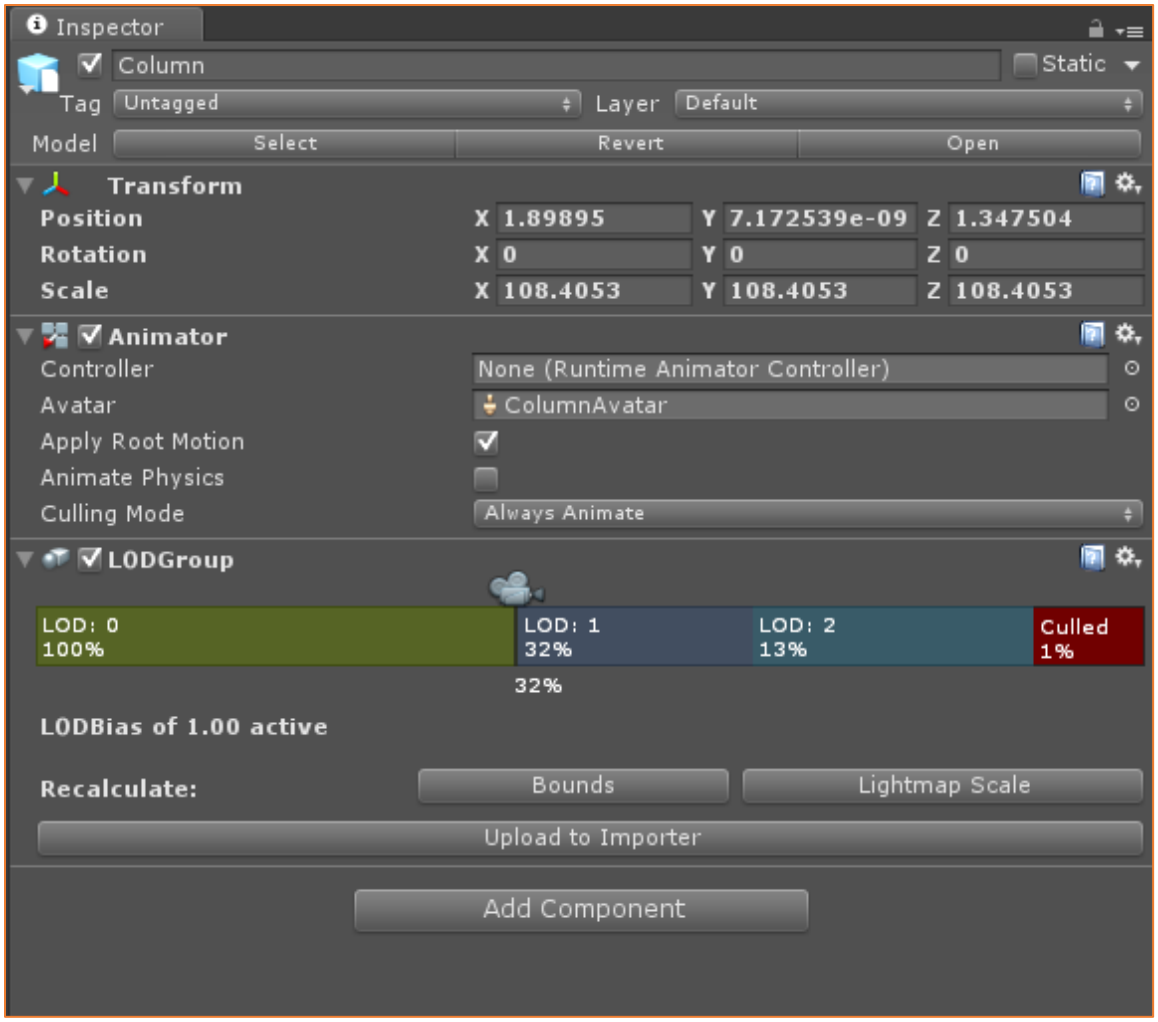
Middle VR is a third party middleware software that enables virtual reality (VR) apps development in Unity. Traditional VR functionalities such as user tracking, wand controller, OpenGL Quad Buffering (stereoscopic rendering) are available to Unity users via Middle VR. The interesting aspect of this middleware is that the VR app development is not device or platform dependent because the final application can be simulated within Middle VR and configured at runtime (e.g. the Unity app we provided in the folder /Unity_oculus can run on different platform such as Oculus Rift, DiVE, 3D projector, etc.). Middle VR Configurator and SDK are available for free download in basic version. A commercial license is needed to finalize application that need Middle VR pro advanced functionalities and Quad Buffered Rendering (stereo projectors and DiVE).

This tutorials does not address the basics of Unity 3D application development or how to import Middle VR scripts in your Unity project but guides you through a number of steps that allow you to import your models in a Unity project, verify the LODs of your models, create advances materials, and finally build your app for Oculus Rift head set.

### Setting the Levels of Detail (LoDs)

36. Launch Unity 3D (Start / All Programs / **Unity**)
37. Create a new project
38. **Asset** / **Import New Asset** and browse you folder to /Materials/Files/**Column**. Select **Import**
39. Select (single click) the newly imported **Column** asset in the **Project** panel
40. Change **Scale Factor** to 1 and tick **Generate Colliders**. Select **Apply**
41. Drag the **Column** asset in the **Visuals** folder
42. Select again the **Column** asset
43. Verify that the LODs are loaded correctly and change the distance settings for your LODs in the **Inspector** panel on the right until you get the desired balance between visual result and distance

[2] nicola.lercari@duke.edu.




44. Double click the **Visuals** folder
45. **Asset** / **Create** / **Prefab** / and name your prefab ***"Column_prefab"***

**Prefabs** are instances that allow you to have a large number of 3D objects in the Unity scene as copies of an imported model that do not affect the real-time performance (frames per second) of your app.

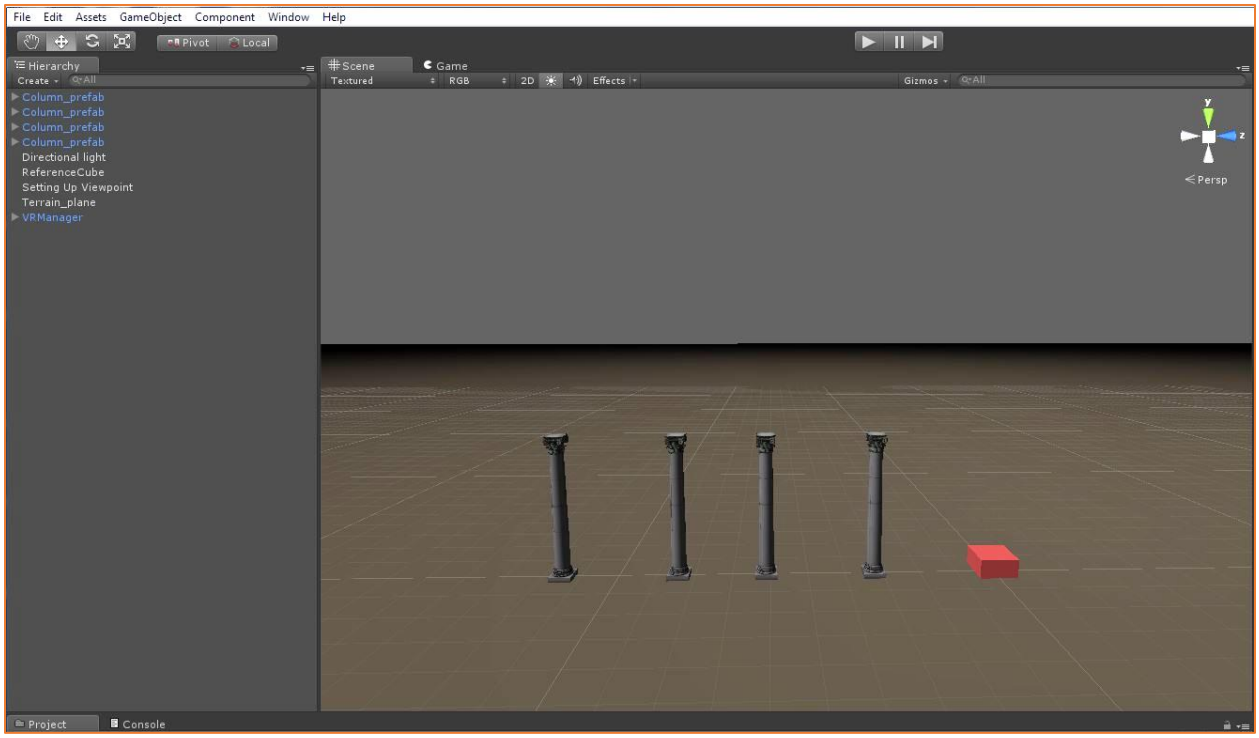
46. Drag the Column asset onto the **Colum_prefab** icon

This add the Column asset to the empty prefab container.

47. Drag the **Column_prefab** to the **Prefab** folder
48. Double click the **Prefab** folder
49. Drag the prefab **Column_prefab** in the **Scene** panel near the red reference cube
50. Press **CTRL**+**D**

This shortcut duplicates the selected asset in the scene.

51. Move the new instance along the X axis (in red)
52. Repeat step 15 a couple of times and create a colonnade as shown below




## Assigning advanced materials to the models

To add realism to your models you can assign advanced materials to your creation imported in Unity as FBX files and assign to them the baked textures created in Blender (e.g. Color Map with ambient occlusion, Normal Map, Displacement Map).

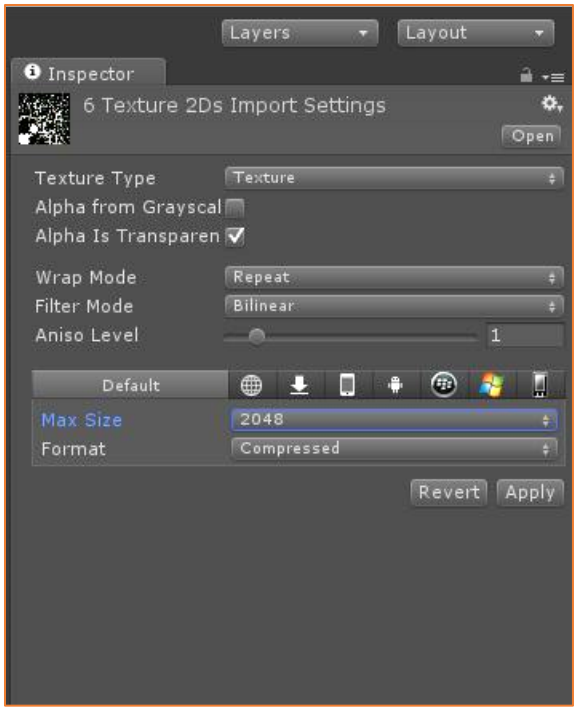
53. Double Click the **Materials** folder
54. Select **Capital_LOD2_BaseMap** material
55. Select **Parallax Diffuse** in the **Shader** drop-down menu

**Shaders** are advanced graphics special effects used to increase realism of textures and colors.

56. Click the empty texture slot on the right of **Normalmap settings**, start typing *"Capital_LOD2…"* and select the blue texture **Capital_LOD2_NormalMap**
57. Click **Fix** label under the texture slot
58. Repeat step 56 for the Heightmap but select **Capital_LOD2_HeightMap** from the list
59. Verify the new material on your **Column_prefab** model

To improve the resolution of your textures and avoid artifacts in the vertical lines of your models (e.g. column shaft) change resolution and the transparency settings for **Alpha Channel** texture in the folder containing the textures linked to your models.

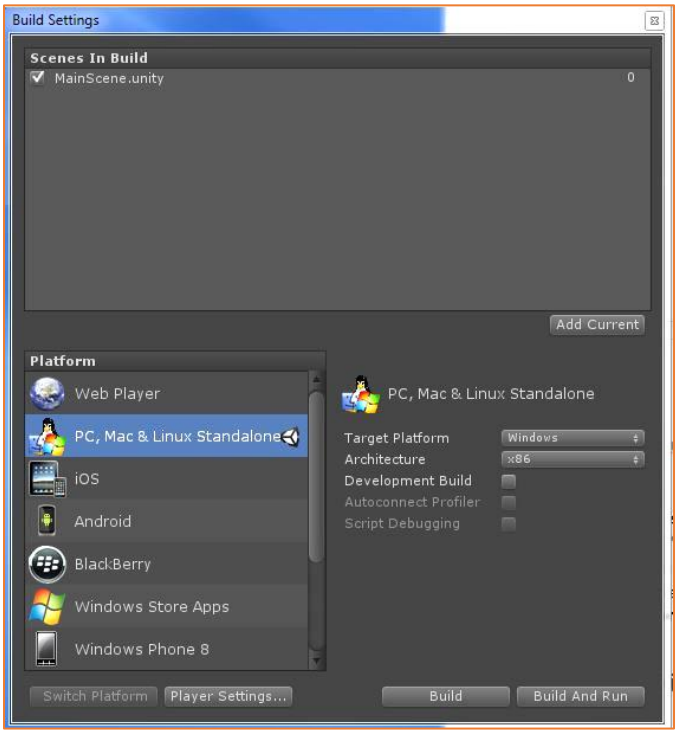
60. Double Click the folder that contains the textures
61. Select all the textures but the Normal Maps (the images in blue)
62. Tick **Alpha is Transparent** in the **Inspector panel**, change **Max Size** to 2048 (as shown below) and **Apply**




## Build an .exe/.dmg standalone realtime app

To use your 3D scene independently from Unity and deploy it as a standalone or web application you need to use the **Build** function in Unity.

63. In **File** / **Build Settings** select your favorite platform (e.g. Win, MacOS, Linux or Web) and make sure that the scene that you are editing in Unity is selected
64. Click on **Build** button, name your application and save it in a known folder in your computer



## Launch you Unity app using in Middle VR

65. Launch Middle VR Configuration (Start button /All Programs / Middle VR / **Middle VR Configuration**)
66. Click **Use Free Edition**
67. **File** / **Open** and browse your computer to C:\
68. Select **nico_oculus_2_monitors_with_gamepad** configuration file
69. Click **Simulations** tab
70. Click **+** button
71. Browse your computer and add your unity **app.exe** or **dmg**
72. Run (Oculus Rift and MS Xbox 360 Controller need to be installed prior to launching Middle VR)