

Lecture 6

3D Modeling in Blender

98-127: *Game Creation for People Who Want to Make Games* (S19)

Written by Adrian Biagioli

Instructors:

Adrian Biagioli (abiagiol@andrew.cmu.edu)
Carter Williams (ncwillia@andrew.cmu.edu)

1 Objectives

By the end of this lesson you will be able to:

- Understand the basics of the Blender user interface
- Build 3D models using local mesh editing tools
- Utilize global mesh editing tools (Modifiers)
- Map a texture to a 3D model using UV unwrapping
- Export 3D models in a Unity-optimized format

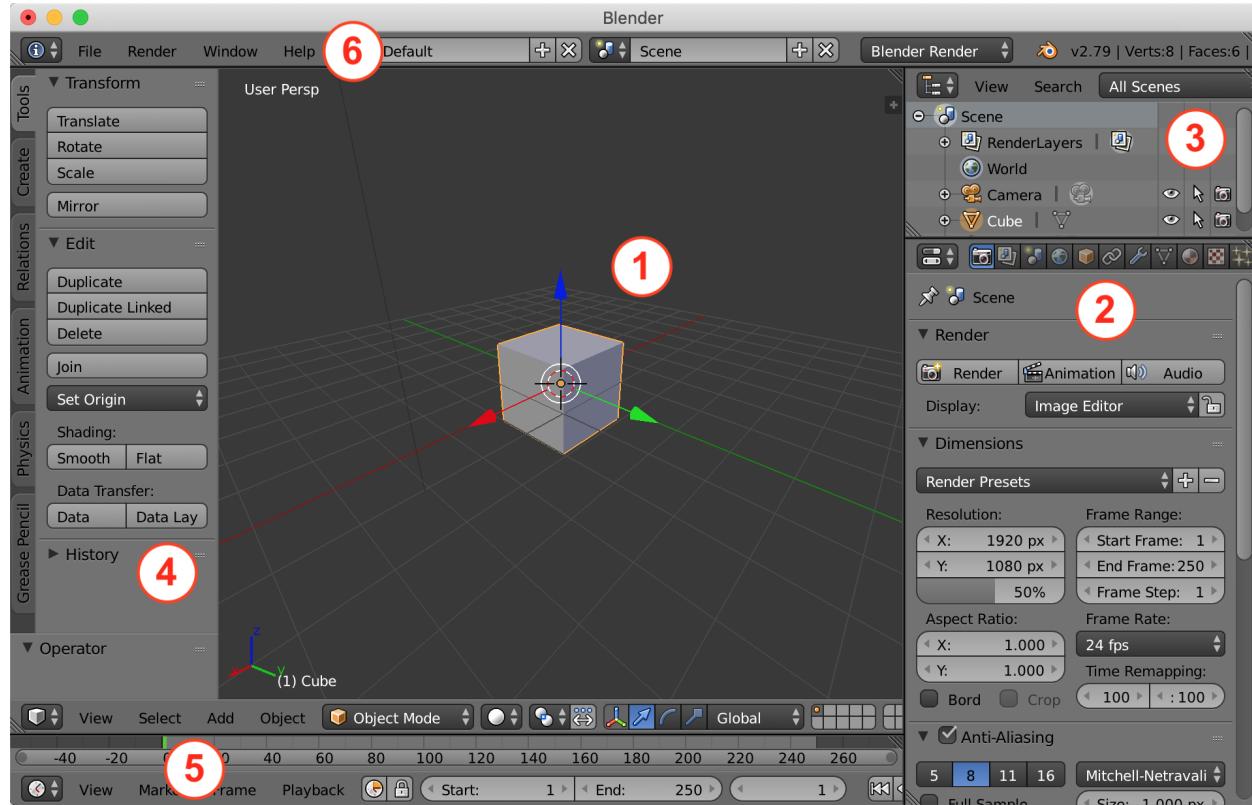
These lecture notes were written for **Blender 2.79b**. Keep in mind that at the time of writing, a new version with a vastly different user interface, Blender 2.8, is in beta. If you're reading this after 2.8 is officially released, you can still enable a mode that uses keybindings from earlier versions of Blender. More info [here](#).

We will also be looking at blender with a comparative eye towards Unity. It may be helpful to have some background on using Unity before looking at these notes. Check out the Intro to Unity notes [here](#).

2 Downloading Blender

Blender is a free and open source 3D modeling package that can be used to build assets for your games. You can download blender at blender.org – make sure to download version **2.79** for compatibility with these lecture notes. The primary alternative to blender, and a common option in industry, is **Autodesk Maya**. However Maya is very expensive for commercial use, so we will not use it in this class. For all but the most advanced tasks, blender has feature parity with Maya, however you may need to use Maya if you get a job at a studio. That being said, the techniques you learn here are transferable to Maya; you'll just need to learn the new interface. You can download the Maya student edition (note: the student edition can not be used for any games that you plan to sell) [here](#). You are going to want to use a mouse for this tutorial, as a mouse makes navigating blender's complex interface much easier.

3 Blender Interface Basics

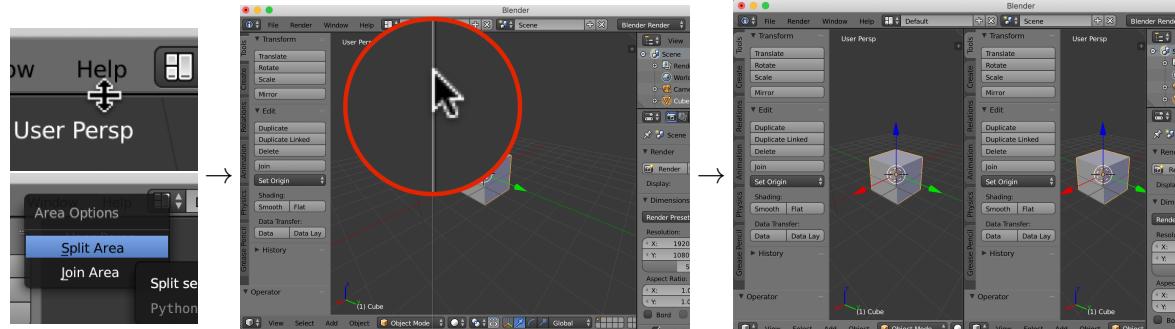


This is the default configuration of Blender when you open it for the first time. Each panel that you see in the above screenshot is labeled as follows:

1. The **3D View** is your main view into the scene you are editing. Unsurprisingly, this is where most of your work happens. We will talk more about how to navigate the 3D view in the subsection below. This is similar to the scene view in Unity.
2. The **Properties** panel allows you to edit various properties of your scene, including the camera, the currently selected object, the currently selected mesh, textures, materials, and so on. We will often use the Properties panel throughout this tutorial. The Properties are analogous to Unity's inspector.
3. The **Outliner** lists all of the objects, meshes, armatures (we'll get to this in a later lecture), and so on in your scene. The outliner is most similar to the hierarchy view in Unity.
4. The **Tool Shelf** contains shortcuts for common editing operations. You can toggle the tool shelf on and off with **T**. If you ever forget a shortcut, the tool shelf is your friend!
5. The **Timeline** is used to scrub through animations (we will talk about 3D animation in a later lecture).
6. The **Info** panel contains common menus like **File** and **Window**, and allows you to change between preset layouts. If you ever want to reset the blender UI, use the dropdown: **Default** and select “Default”.

You may have noticed that at the corner of each panel there is a dropdown. This dropdown allows you to change the *Editor Type* of each panel. For example, we can change from the 3D view to the text editor by clicking on the dropdown on the bottom left of the 3D view and selecting “Text Editor.” Fun fact: the entire blender front-end is written in python and is fully scriptable and extendable. The text editor is used to write python scripts for blender – but we won’t be using it now. Switch back to the “3D View” using the same dropdown. Also notice how all of the panels discussed on the last page are available here. For example, it may not be all that useful, but you can change the 3D view into a huge timeline.

You can resize each panel by clicking and dragging at the borders. If you want to create more editor views than the default, **right click** on the border of two panels and select **Split Area**. Then move your mouse over the panel you would like to split and **left click** to apply. This process is detailed below:



You can also select “Join Area” and click on one of the two bordering panels to join them together.

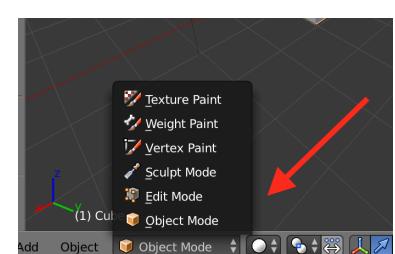
3.1 Navigating the 3D View

You can navigate the 3D view using the following controls:

- To **orbit** (rotate) the view, hold **Middle Mouse Button** while dragging the mouse.
- To **pan** (translate) the view, hold **Shift** + **Middle Mouse Button** while dragging the mouse. You can also hold **Shift** + **Ctrl** + **Midle Mouse Button** to “Dolly Zoom”, which moves the camera forward and backwards.
- To **zoom** the view, hold **Ctrl** + **Middle Mouse Button** while dragging the mouse, or scroll **Mouse Wheel**. This is different than the dolly zoom because the viewport camera does not move.
- If you ever get lost, press **Numpad .** (period) to zoom and pan to the currently selected object.

3.2 Working in Object Mode

By default, when blender opens you will be in **Object Mode**. You can verify what mode you are in by checking the *Mode Switcher* at the bottom of the 3D View. Each **Object** in blender (analogous to a *GameObject* in Unity terminology) has a **Mesh** attached. Just like in Unity, you can transform Objects. Below



The Mode Switcher

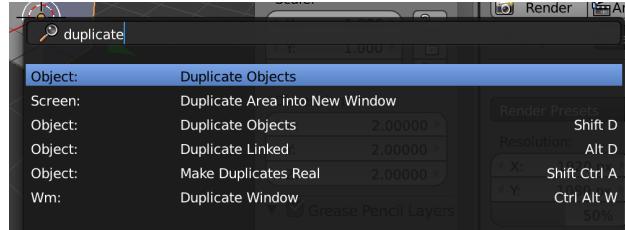
are some common shortcuts to manipulate objects in Object Mode (*Important: Make sure your mouse is on top of the 3D view when using these shortcuts, because shortcuts are different depending on which view is “active.”*)

- Press **Right Mouse Button** to select an object. **Important: Left Mouse button does not select objects in blender!** This is a common point of confusion for new users.¹
- Press **Left Mouse Button** to move the *3D cursor*. The 3D cursor is used for many 3D operations as a reference point – for now, though, you can largely ignore it.
- Press **G** (“Grab”) to Move an object, **R** (“Rotate”) to Rotate an object, or **S** (“Scale”) to Scale an object. You can move the mouse to control the transformation; click **Left Mouse Button** or press **Esc** to apply. While grabbing / rotating / scaling, you can press **X** / **Y** / **Z** to constrain the transformation to a single axis. Press the same key again to switch from Global axes (relative to the origin) to Local axes (relative to how that object is rotated). Also, if you press **Shift + X**, you will constrain the transformation to every axis *except for* the X axis—this is useful if you want to move an object along another face.
- Alternatively, you can use the on-screen 3D manipulators to move objects. You can change the type of manipulator on screen by using the selector ( Global ) located at the bottom of the 3D View. You can also use the pivot switcher ( ) to change the pivot point of any transformation. One useful feature is the ability to change the pivot to the 3D cursor. This allows you to (for example) rotate an object *about* the 3D cursor
- Press **X** to delete an object.
- Press **Shift + D** to duplicate an object. You can also press **Alt + D** to “Duplicate Linked,” which shares the same mesh between the two objects. This is similar to the idea of prefabs in Unity: editing the mesh of either object will affect the appearance of both. This is especially useful because this link will actually persist when you import to Unity – cool!
- Press **B** to box-select and **C** to circle-select objects.
- Press **T** to toggle on and off the Toolshelf. The toolshelf has clickable buttons for many common editing operations (including all of the shortcuts listed above).
- Press **N** to toggle on and off the **Properties** panel. The properties panel allows you to directly manipulate the position / rotation / scale of whatever you have selected, as well as various miscellaneous settings for how the mesh displays in the 3D View.
- Press **A** to select all or deselect all (depending on if anything is selected).

Blender has an incredibly robust shortcut system – almost *every* common editing operation has a corresponding shortcut. So while it may be difficult to get used to blender’s UI, mastery of it will allow you to

¹I know this is really confusing, but there is hope: in the Blender 2.8 beta, the developers have changed the default to left-click select. As you can imagine, this has been a source of debate and nerd wars in the blender community for *decades*!

do things very quickly. For example, rotating an object 45 degrees in the Z axis is as easy as **R** > **Z** > **4** > **5** > **↵**. That being said, it is still very easy to forget all of the random incantations–er–key combinations you need to use to do what you want. Luckily, pressing **Space** at any time will bring up a search bar that allows you to search for any operation. It will also show any relevant key combinations. You can then hit **↵** to apply the operation.



Searching for Shortcuts

3.3 Creating Objects

You can create new blender objects via the **Add > Mesh** menu at the bottom of the 3D View. There are plenty of other types of objects in the **Add** menu, but since we are creating a video game we have to use polygonal meshes that can be imported to Unity. If you were, for example, interested in making a film, or if you wanted to create a very high-detail model that will be later converted a game-ready mesh, you may want to look into the other surface types. Check out the Blender Manual for more info! Also, as always if you don't want to trawl through menus you can use ex **Space** + "Add Cube" + **↵** to add a cube to the scene. Note that new objects will be placed at the *3D cursor* location. If you can't find your 3D cursor, you can change its location in the Properties panel (**N**).

3.4 Parenting Objects

Just like in Unity, you can parent two objects to each other in blender. Any transformations done to the parent will “trickle down” to the children in an hierarchical fashion. You can parent objects in two ways:

1. In the outliner, **left click and drag** on the **icon** to the left of one object, and drag the icon on top of another object
2. Select the child object(s), then the parent, and press **Ctrl** + **P**.

Conveniently, parented objects will import to Unity with the same relative transformations.

4 Local Mesh Editing

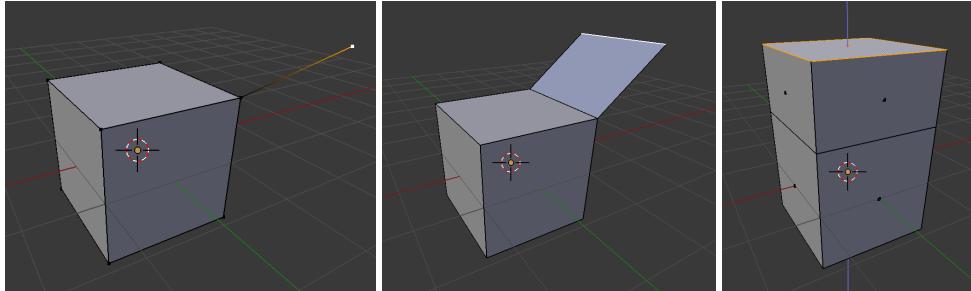
Well it's been 5 pages and we haven't even started modeling yet–blender is a pretty complex piece of software! Luckily, the UI is consistent (albeit complex) so editing meshes is very similar to editing objects. To edit an object's mesh, select it and switch to **Edit Mode**. The shortcut to switch between Object and Edit Mode is **Tab**. You can also use the Mode Switcher (mentioned above in “Working in Object Mode”) to

switch to Edit Mode. That being said, you will be switching between Object/Edit so often that `Tab` is worth memorizing. *Important: make sure you have an object selected or you can not go into Edit Mode.*

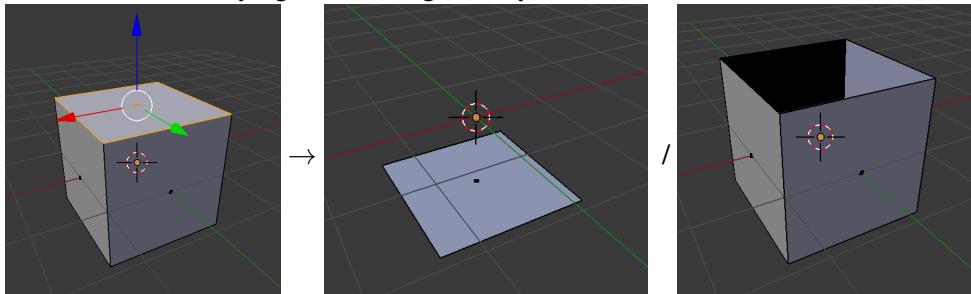
You'll notice a few things when switching from Object to Edit mode. First, the individual vertices of the mesh will be selected. A **vertex** (plural *vertices*) is an individual point on the mesh. Vertices are connected by **edges**, and edges are connected by **faces**. You can select vertices with `Right Mouse Button`, just like how you selected objects before. In addition to this, the tools at the Bottom of the 3D view as well as the Toolbar / Properties panels update according to the current mode. It's important to be aware of what mode you're in so that you don't get disoriented with the UI.

At the bottom of the 3D view, click on the buttons that look like this:  to change between Vertex, Edge, or Face selection. As always, use `Right Mouse Button` to select each type. You can hold `Shift` to select multiple elements of your mesh. Just like in object mode, you can press `G` / `R` / `S` to Move / Rotate / Scale parts of your mesh (see “Working in Object Mode” above). More complex operations can be accessed via the `Mesh > Vertices`, `Mesh > Edges`, or `Mesh > Faces` menus. The shortcut for these are `Ctrl + V`, `Ctrl + E`, and `Ctrl + F` respectively. There are lots of very useful features in each of these menus (and other submenus), but I will cover the most common / useful below.

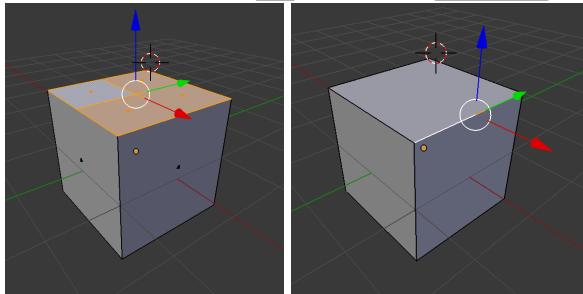
- **Extrude** (Shortcut `E`): Takes whatever you have selected, duplicates it, and connects what you selected to the duplicate. For convenience, transitions into grab mode, as if you pressed `G`, so that you can move the extruded object. This has slightly different implications depending on if you selected a Vertex, Edge, or Face (see below pictures, respectively):



- **Delete** (Shortcut `X`): Deletes a vertex, edge, or face. When you press `X`, a menu will appear with many different options for what exactly you want to delete, and how. For example, selecting `Vertices` will delete all selected vertices (and all connected geometry), but `Faces` will only delete the selected faces without modifying connected geometry:

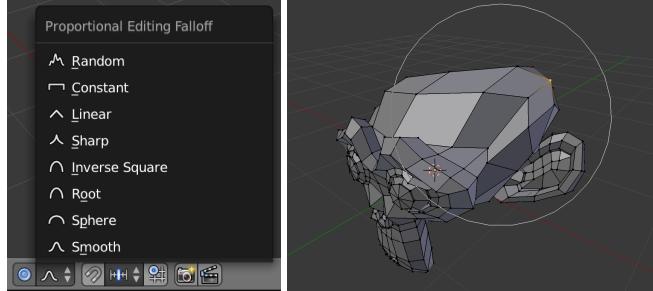


- **Subdivide** (Shortcut **W**, then click **Subdivide**): This command divides a face or edge into equal sections:



- **Merge** (Shortcut **W**, then click **Merge...**, OR **Alt + M**): Merges all selected vertices, edges, or faces into a single vertex. If you are selecting vertices, you have a lot of control as to where the merged vertex ends up (for example, you can set it to merge to the center of all points you selected to merge, or you can set it to merge to the first or last point you selected, etc).
- **Snap** (Shortcut **Shift + S**) will snap the current selection to the cursor, other selected objects, or *the grid*. The snapping grid is useful to keep your selection on a uniform grid – you can modify the size of the snapping in the **Display > Grid Scale** submenu of Properties (shortcut **N**). You can also snap all transformations to the grid via the snap button (at the bottom of the 3D View). You can also use **Shift + S** to snap the 3D cursor to things—notably, you can snap the 3D cursor to the origin, which is a fast way to reset the 3D cursor position when you left click by accident 😊
- **Loop Cut** (Shortcut **Ctrl + R**) will create a ring around the mesh. Using loop cuts to create new vertices is highly recommended as it maintains good topology of the mesh (you generally want to avoid triangles, especially very obtuse ones as they animate badly).
- **Select Edge Loop** (**Alt + Right Mouse Button**) while hovering an edge) selects all the edges that are in a contiguous loop. This is useful in many situations, for example you could select all of the edges around a character's torso.
- **Knife Cut** (Shortcut **K**) will allow you to simply cut through the mesh using the mouse.
- **Vertex Slide** (Shortcut **G G** in Vertex select mode) slides a vertex along one of its edges.
- **Wireframe Mode** (Shortcut **Z**) allows you to see through the mesh and select vertices / edges / faces that are behind other faces.

- **Proportional Editing** (Shortcut **O**) allows you to “pull” a neighborhood of Vertices / Edges / Faces in a weighted way, based on a falloff distance from what you have selected. You can change the falloff settings at the bottom of the 3D View ( when enabled); open the dropdown to change the falloff type. You can change the falloff distance with **Mouse Scroll Wheel**:

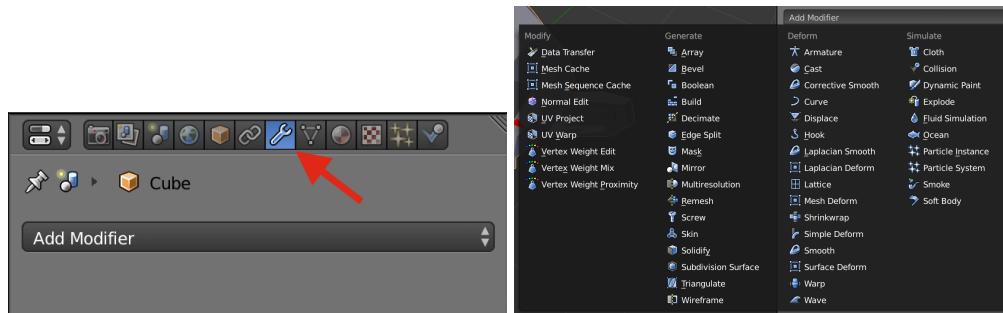


Proportional editing applies to Move, Rotate, and Scale operations in Vertex, Edge, or Face select.

5 Using Modifiers for Global Editing Operations

The operations that we have talked about so far are *Local Operations*. This means that they only affect the Faces, Edges, or Vertices that you specifically select. However, oftentimes you want to perform *Global Operations* on the entire mesh. The easiest way to perform “object- wide” operations on meshes is to use **Modifiers**. Modifiers can be accessed from the Properties panel when you have an object selected. *Note: you must switch to Object Mode before adding Modifiers to your mesh. If you don’t you may not see the results of applying the modifier until you switch out to Object mode.*

As a test, I’ll add in the **Monkey** primitive (**Add > Mesh > Monkey**), which is useful as a semi-complex object to play with. Her name is *Suzanne*, and she is the official mascot of Blender.² Open up the Modifiers tab in the Properties panel. You can click “Add Modifier” to add a new modifier to the object:

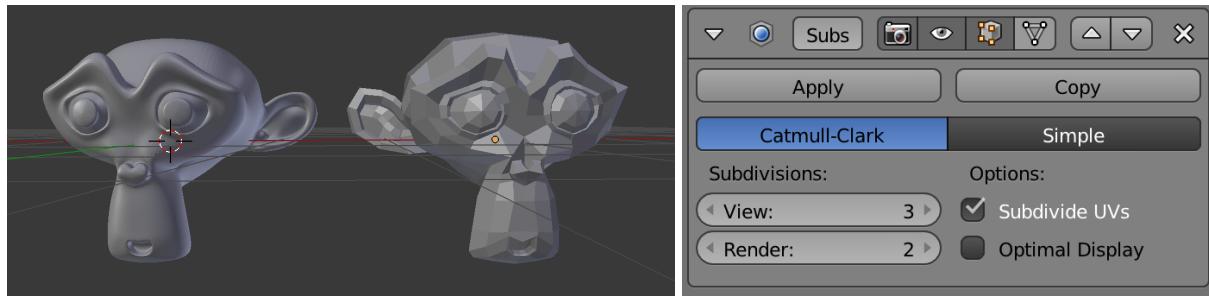


I’ll go through some of the most useful modifiers below. You can find a more comprehensive reference of all of the modifiers in the Blender manual.

²Why a monkey? Suzanne is one of a few very popular Test Models, alongside the “Stanford Bunny” and the “Utah Teapot,” which you’ll tend to see in many Computer Graphics research papers.

5.1 Subdivision Surface Modifier

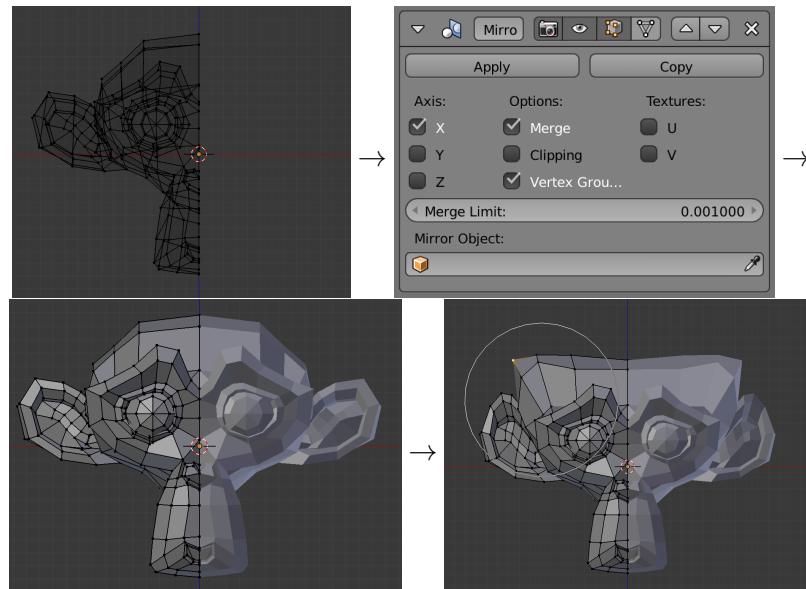
The **Subdivision Surface** modifier subdivides every face in the entire mesh (see the **Subdivide** operation above) and computes a carefully-weighted average of them, resulting in a much smoother version of the original mesh. Change the **View** parameter of the subdivision surface modifier to increase the resolution. Below is suzanne with and without the subdivision surface modifier applied.



You may recognize the algorithm name, “Catmull-Clark,” from Edwin Catmull, the President of Pixar. In fact, Ed Catmull invented the algorithm along with Computer Scientist Jim Clark in 1978. The Catmull-Clark algorithm is what allowed early Pixar movies to represent complex smooth surfaces, and was pivotal in creating convincing renderings of humans in their movies. More info here on the algorithm.

5.2 Mirror Modifier

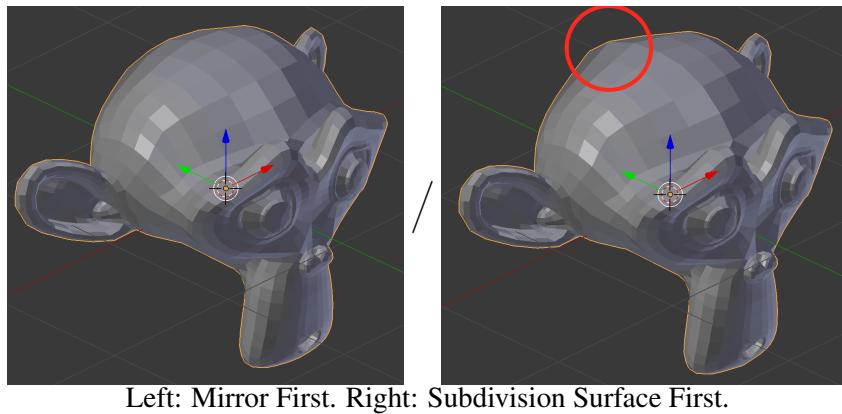
The **Mirror** modifier allows you to mirror a mesh along one of the principal (XYZ) axes. This is useful if you want to edit a symmetrical mesh, such as a character. Here I deleted half of Suzanne using the box select tool (**B**) and used the Mirror modifier to enable a symmetric editing workflow:



5.3 Nondestructive Editing

A very useful feature of Modifiers is that they are **Nondestructive**. For example, you can edit the mesh after a mirror or subdivision surface modifier are added to an object, and the final result will update as you would expect. This is in contrast to simply performing the changes in Edit mode, in which case the changes are permanent. You can also click the **Apply** button at the top of every modifier to destructively apply it to the mesh. This is useful if, for example, you would like to tweak the output of a Subdivision Surface modifier, or if you would like to add small asymmetric details to a character after blocking it out with the mirror modifier.

You can also add multiple modifiers to an object. The modifiers are applied in the order that they are added. For example, you would want to apply a subdivision surface modifier after a mirror modifier. Otherwise, you will get weird boundaries between the two halves:

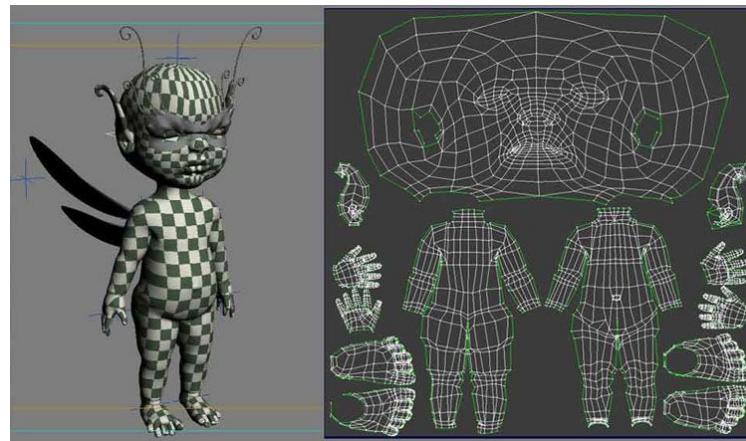


Left: Mirror First. Right: Subdivision Surface First.

You can use the buttons at the top of each modifier's UI to re-order modifiers.

6 Texture Mapping

Up to this point we have only talked about changing the shape or form of a 3D model. What do we do when we want to change the model's color? Ideally, we want to be able to paint over the model as if it were real-life clay model. It would be great if we could use Photoshop for example to define the model's color. So how do we do this? The solution is to use a technique called **Texture Mapping**. The essential idea is to create a **map** between an image file (a "texture") and your model. In practice, this is done via something called a **UV Map**. It's called a UV map because just like X, Y, and Z are used to describe the 3D axes in computer graphics, we use U and V to describe the 2D axes on a texture. It's just the common convention.

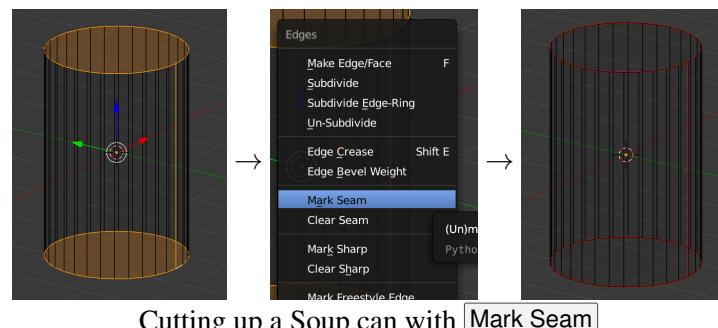


A UV map of a character (Source)

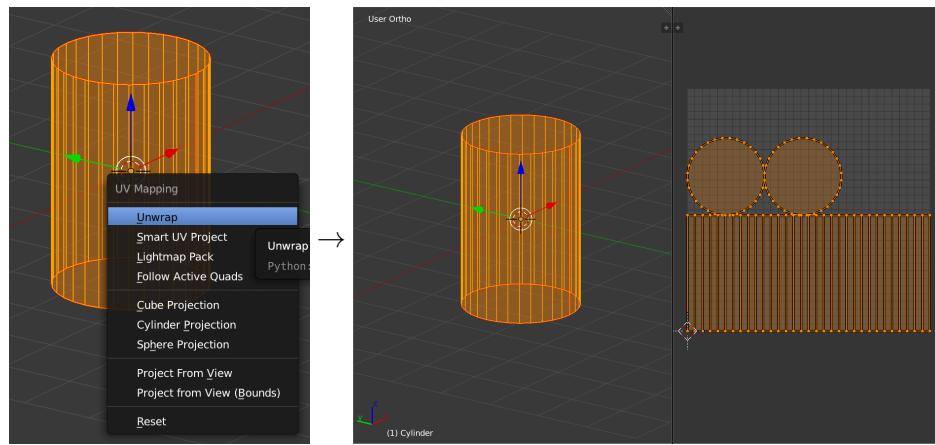
You can think of a UV map as if you were wrapping the texture around your model, sort of like wrapping a present. Or you could think about it in the inverse: how would we “skin” the model, like skinning an animal’s pelt, so that we could lay it flat? We need to somehow tell Blender how to skin our model. Let’s walk through an example.

6.1 UV Unwrapping a Campbell’s Soup Can

Let’s create the textures and UV map for a simple model, a Campbell’s Soup Can. To begin, add a Cylinder to the scene via **Add > Mesh > Cylinder** in Object Mode. We need to think about where we would “cut” the model so that we could lay it out as flatly as possible on the UV map. For a cylinder, we would cut out the top and bottom circles, and then we would cut straight down the sides. This would allow us to lay down the two circles and “label” flat next to each other with no stretching. We can explicitly tell Blender to cut along specific edges using the **Mark Seam** command. You can get to this in edit mode via the edges menu (**Ctrl + E**) then select **Mark Seam**:

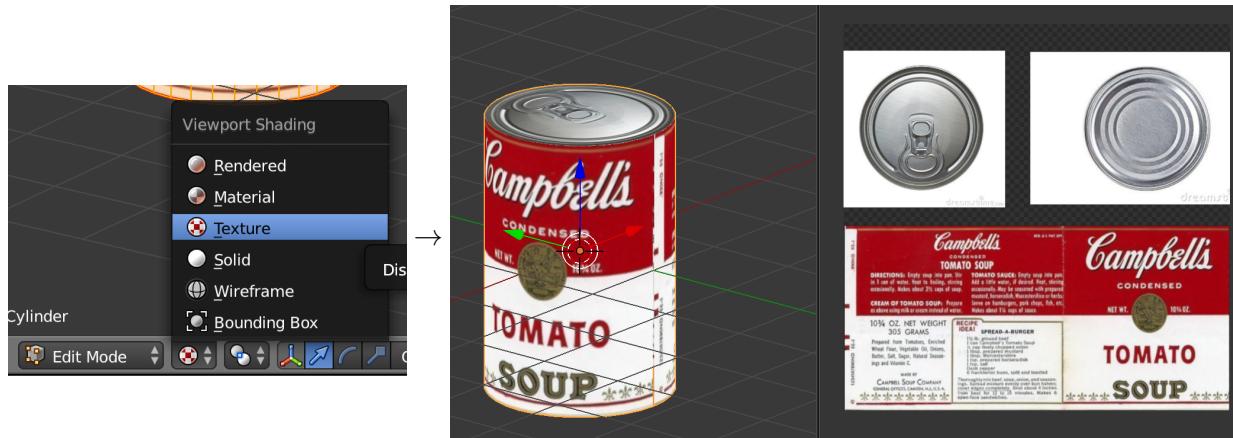
Cutting up a Soup can with **Mark Seam**

Next, we need to go ahead and create the UV Map. First, create a new panel (see the section “Blender Interface Basics” above) and open the **UV/Image Editor** View such that the 3D View and the UV/Image Editor are placed side-by-side. Recall that you can change the view of each panel using the dropdown located in the corner of the panel. Next, while hovering the mouse over the 3D View in edit mode, press **A** to select all, then open the **UV Mapping Menu** with the **U** key. Select **Unwrap**, and you should see our soup can fully unwrapped:



You can click the button at the bottom of the UV/Image editor to allow you to select vertices / edges / faces in 2D. Just like always, you can use , , and to transform each face in UV space. Select all faces in the UV/Image Editor by hovering your mouse over it and pressing , and then navigate to at the bottom of the editor. This allows you to save the UV layout as a PNG file. We can now use this as a reference in a standard 2D image editor, like Photoshop or GIMP.

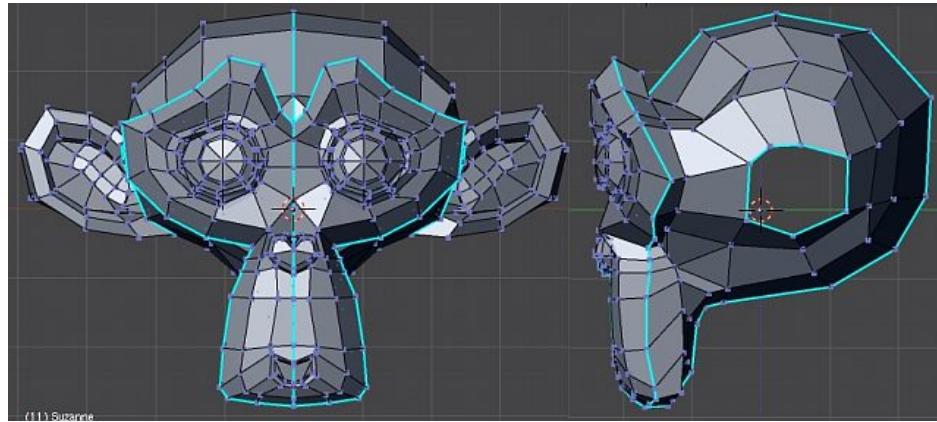
I'll leave creating the texture as an exercise for you. When the texture has been created, enter edit mode again and select all faces on the model (). Then open the texture via the "Open" menu at the bottom of the UV/Image editor (). Once again, make sure all faces on the model are selected before applying the texture, and make sure you are in edit mode. Otherwise the texture may not actually apply to the model. Next change the "Render Mode" of the 3D View to via a dropdown at the bottom. Here's what I managed to come up with:



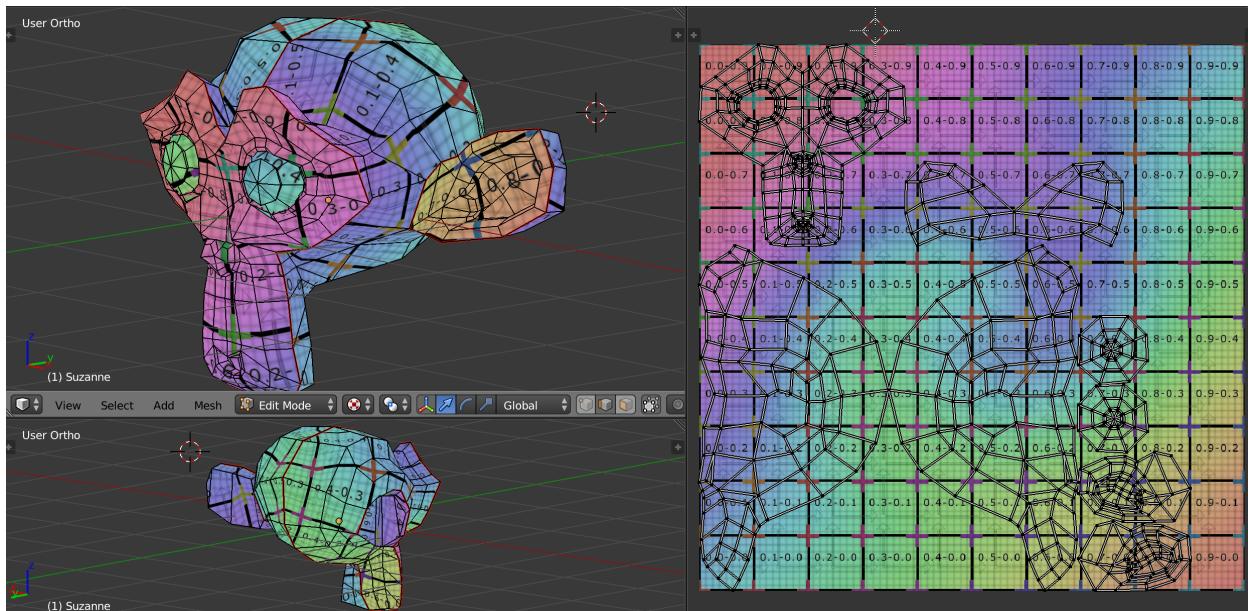
6.2 UV Mapping Complex Models

UV Mapping is a very complicated topic, and many people build their careers around being great at making these textures / maps. Below I've included where the seams might be placed in the Suzanne model. Generally, you want to put the seams in places where it isn't a big deal to see discontinuities in the texture. Good candidates include behind the ears, in the back of the head, and so on. You also want to minimize stretching, so you should pick your UV "islands" such that they are easy to unwrap. One way of "debugging" your UV

maps is to use a UV test texture. You can find plenty of these with a simple Google image search for “UV Test Texture”.



One way of configuring Suzanne's Seams–Source: Blender documentation
(<https://docs.blender.org/api/html/x5336.html>).

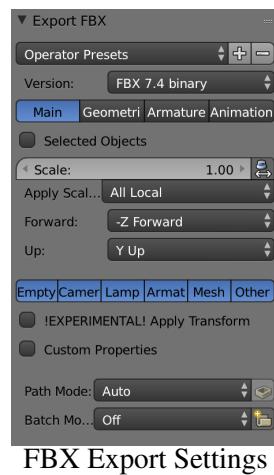


Unwrapping Suzie and applying a UV Test texture

7 Exporting to Unity

Unity’s native format for 3D Models is .fbx. This means that, without any extra software installed, Unity can load .FBX files by simply placing them in the Assets folder. Unity can also import .blend files (the blender file format), but you have to have blender installed to successfully import them. Similarly, you need to have Maya installed to import .ma or .mb files. For this reason, I recommend that you always export your models in blender to .fbx format before importing them into Unity.

You can do this via the menu `File > Export > FBX (.fbx)` at the top of the screen. You can configure the export settings on the left side of the file explorer that pops up:



FBX Export Settings

More info on these settings can be found at this guide here. You can now simply drag the created FBX file into the Project view in Unity. You will also need to import whatever textures you have created and apply them to a Unity material (more on this in a later lecture).

8 Exercise

The exercise for this class is to **Model, UV, and texture a container** (What is a container? Up to you...). Export the model and import it into Unity!