

# A Simple Spriting Tutorial

*An attempt to speed up the creative process by Helios*

## Introduction:

For the last few years I have been working, sporadically, on a material output from 3DSMax8 that renders directly into a passable sprite. I'll cover the inner workings of the scene and material with as much depth as possible to get the technique across. I am not covering the actual creation of a mesh; you have the internet for that. Unsurprisingly, no quick fix is one hundred percent perfect, so some minimal edit work is necessary here and there after the final render.

Even though this tutorial was made using 3D Studio Max 8, Paintshop Pro 6, zBrush 3.1 and Windows XP newer versions of these applications should suffice as well (save for newer versions of PSP that lost palette functionality).

## What you'll find in your typical render scene:

As some of you know 3DSMax8 (no, this tutorial isn't old, I just never upgrade if I don't have to) only renders toon outlines properly in the 640x480 Perspective view, so that means you will need a camera setup. The scene also requires lighting, so I'll give a few pointers on that too. So this is what you'll find in a scene:

- 1.) Your (potentially rigged and animated) mesh.
- 2.) The materials and application thereof.
- 3.) Camera.
- 4.) Lights.

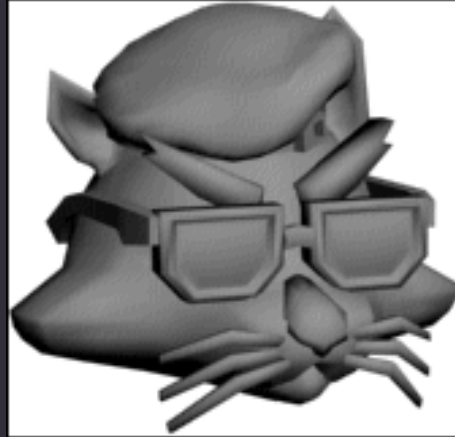
I'll teach you how to create your materials, cameras and lights to get the desired effect.

So shall we dive right in?

## Your mesh:

Now I'm not going to ream anyone for not using original work here, because quite frankly I'm probably talking to the 2D generation (my generation). We all want to quickly slap together the perfect rigged and pose-able mesh from scratch, but sometimes that simply isn't possible due to time constraints or lack of knowledge. So go ahead and be as macabre as you want; tear limbs, ears and heads from commercial meshes to get to grips with this tutorial's techniques, then go back and learn mesh modelling.

Well, with that out of the way I'll do the exact opposite and make my own mesh... For the purpose of this tutorial I made a quick head mesh in 3DSMax:



Meet Frank, the elitist dictator fox (he looks like a cat now, I know). He graciously offered us his head to be used, as required, for science. *Thanks Frank.*

Idiocy aside, that particular mesh brings up a few interesting points.

Firstly: Notice how everything is crafted to stand out quite a bit? When you render your mesh scaled down to suit whatever resolution your application requires you'll notice how some details simply vanish if they aren't prominent enough. Sometimes - if 3D fails you - you may just have to fill things in by mouse per-pixel, after the render, just for detail and continuation's sake.

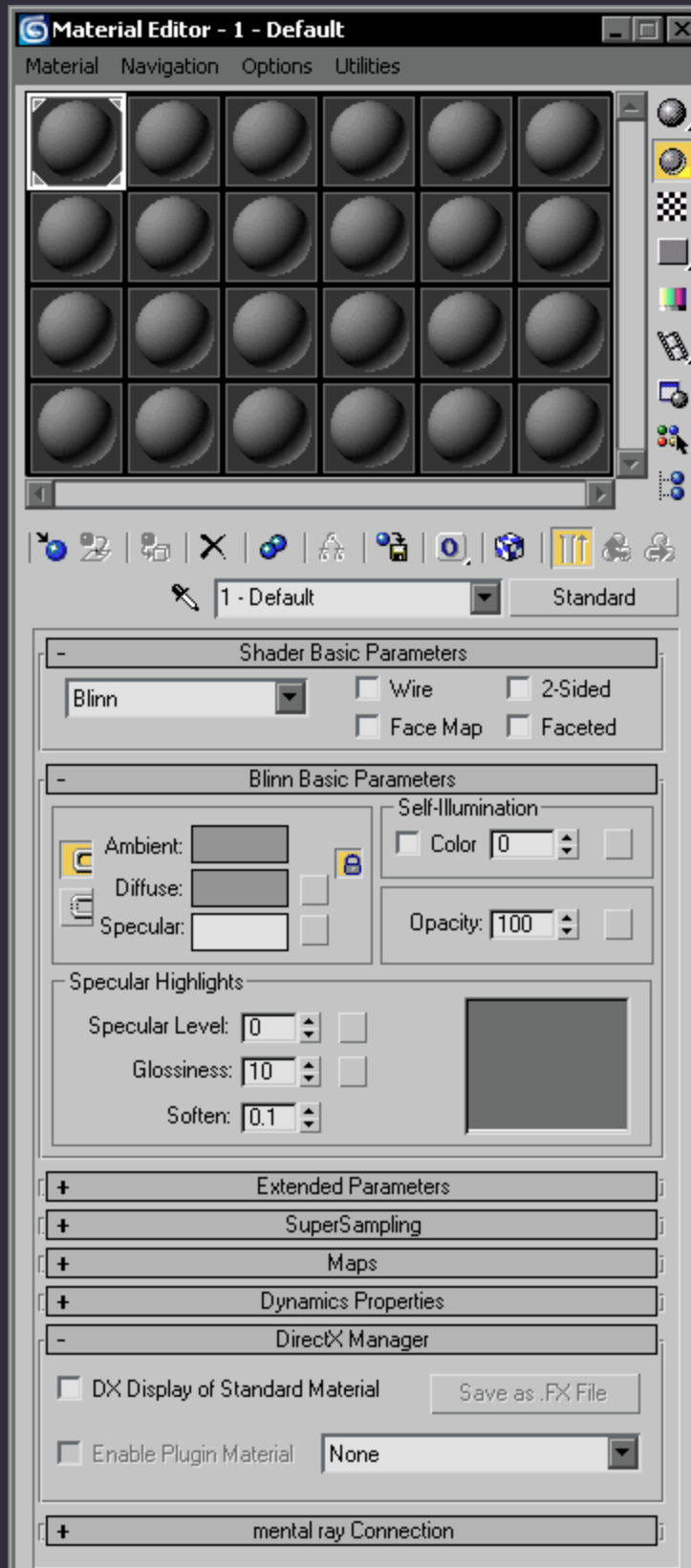
Secondly: You construct your mesh any way you want to. I'm not going to win any awards for my 3D work with any of my made-for-2D meshes. You work on something until the detail is there, regardless of poly count or imperfections. This means an absurd amount of artistic freedom limited only by your imagination (and PC).

Thirdly: Meshes need geometric detail, not texture detail. So unless you have a bump map or a specific look in mind you will likely not need to UV map for any of your work. Sadly if you're using a mesh that is made for textures you might have to tweak it a bit to suit your 2D needs, like adding eyebrows or editing lashes and clothing. So learning just a bit of 3D might be worth the effort.

Lastly: Not all of my work is done in 3DSMax8, that hat is a stupidly high poly mesh that I whipped up in zBrush (which is another fine tool if you can afford it). It only really shines with organic or round matter, though. Metallic or rigid objects are best left to low poly programs like 3DSMax.

### **Materials (The first waltz):**

Okay, it's time to have our first round with materials. Press the M key in 3DSMax to open up the Material Editor which looks a little something like this:




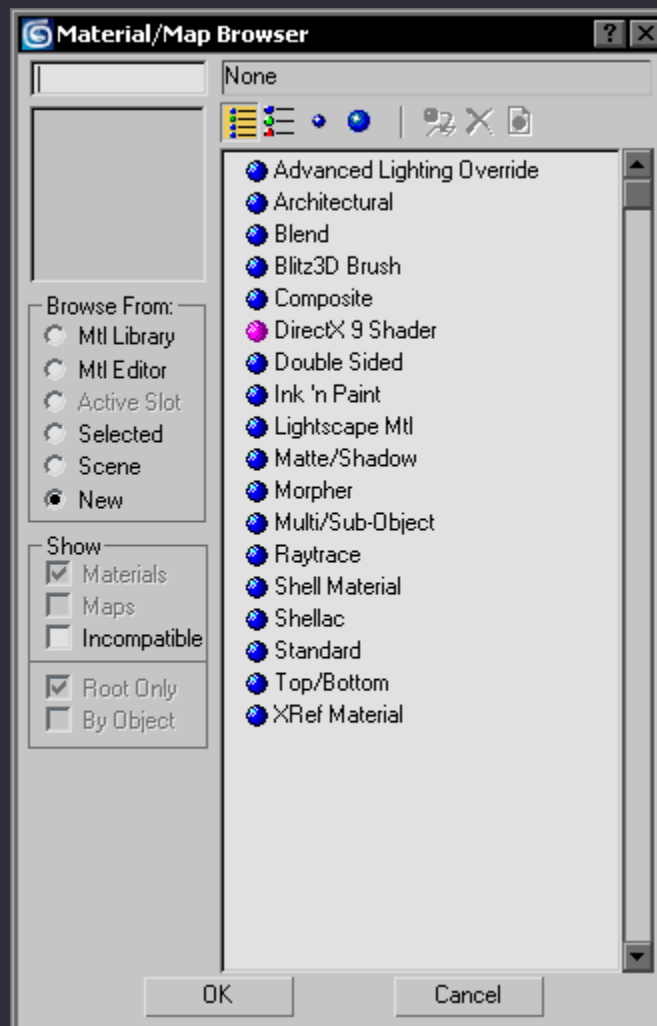
Yes, it deserves an entire page.

You'll notice that there are rows and rows of grey balls, each one representing a material that can be used on a mesh in your scene. A material is essentially the paint for your canvas if you aren't familiar with the 3D terms I've been using (with the mesh being your canvas, of course). You may want to mess around with some of the parameters to get a feel for what does what by adjusting values or clicking on a block of colour and tweaking it. The more comfortable you are handling this window the better. After you're done with that we can move on to...

### Materials (Getting that sprite look #1):

Grab one of those grey balls and drag it on to your mesh in a Perspective viewport. You'll notice your mesh gets changed to that dreary grey colour as well. Making any changes on said ball's parameters will change the look of your mesh as well to some degree (some specific parameters won't though).

Now, what we want to do is click on that  button at the bottom-right of those balls. It will bring up the following menu:



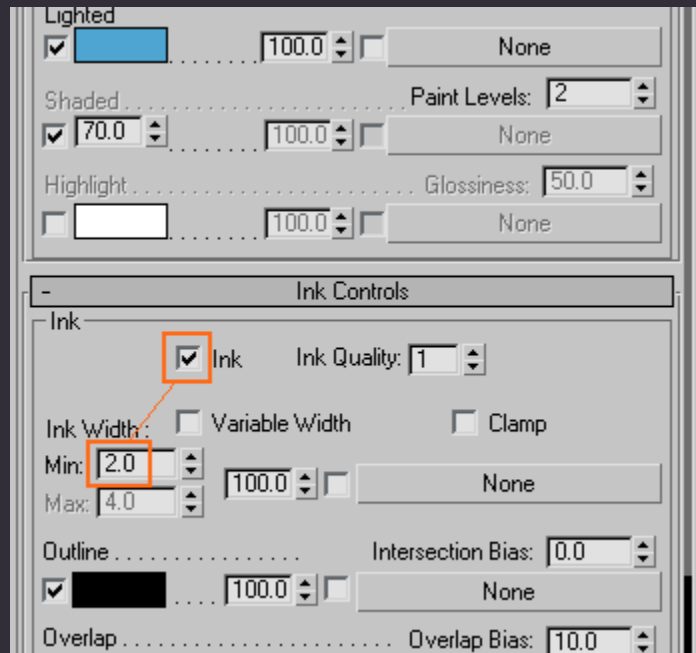
Now click on the “Ink ‘n Paint” menu item and click OK (the eighth one from the top).

If you applied the filter to the material that corresponds to your mesh’s material you’ll see that your mesh just went blue in the Perspective viewport. This means that you applied the Ink ‘n Paint material to your mesh. If your mesh didn’t turn blue you simply need to drag your toon material that you just created on to your mesh in the Perspective viewport.

You can now press F9 if you have your Perspective view port highlighted (click anywhere in the 3D view) to render your Perspective viewport’s scene. You’ll see a blue toon-shaded image with thick black outlines which generally looks pretty damn ugly. So let’s just close that window and move on.

Now, from here on in it becomes a matter of preference... I will be showing you the way that \*I\* like doing my toon material, you don’t have to do anything this fancy, you can even go for a two-tone shade if you want. The sky is the limit.

Okay, so firstly I adjust a few things on my base toon material’s settings. If you scroll down the parameters of the material you’ll see this:



As you can see I’ve drawn attention to two areas on the previous image. The first being “Ink” which pretty much governs the lines on your image. I leave this ticked on 99% of the time, unless I want a very specific look. I always leave variable width turned off. Next is the ink width, I put my Min on 1.0 (unlike 2.0 which is the default thickness).

If you press F9 now your mesh will likely look a bit closer to a sprite’s general look (thinner lines can make all the difference). Also, if you’re wondering if you can get rid of the anti-aliasing in and around your render: you can. I will be handling the rendering options in the next section.

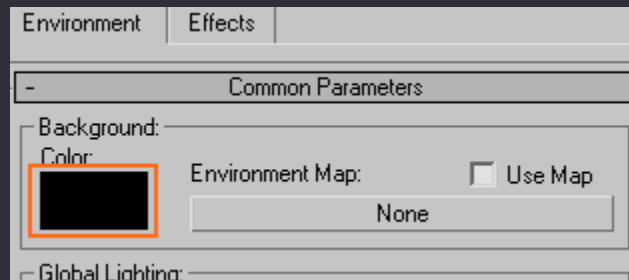
## The scene (Round 1):

That finishes the preparation phase for the material, we now move on to the rest of the scene. I do things this way so that I can render early prototypes and scale things as required to properly align the sprites to whatever target resolution I might have.

In this section I will be handling all of the little things that make sprites what they are, both easy to handle and palletize.

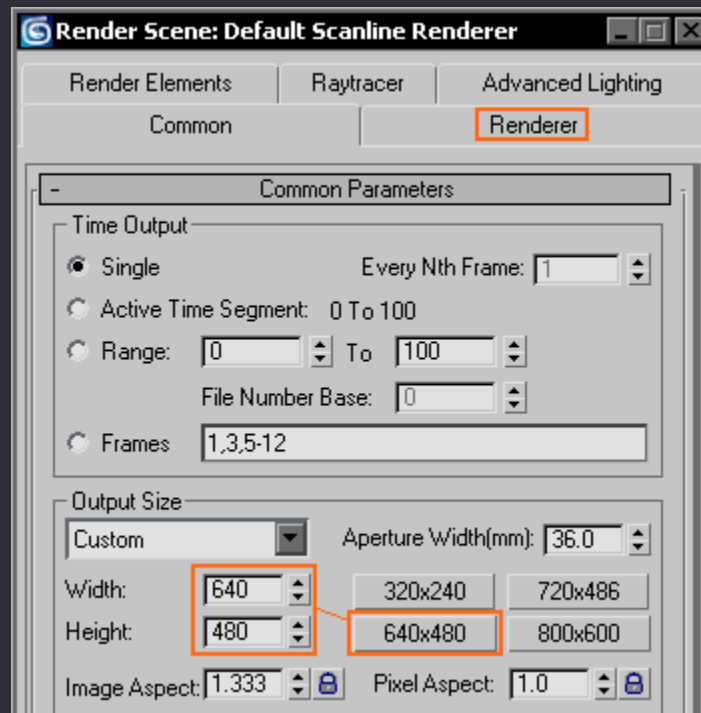
If you've been paying attention you'd have noticed that when you press F9 3DSMax just spews out a render of your currently selected viewport without you having any control over how things are actually rendered.

We'll do the background colour first. Press 8 (not F8) to bring up the "Environment and Effects" window, this window can also be accessed via the "Rendering" menu at the top toolbar.



Clicking on that highlighted black box will bring up a colour mixer where you can either click a colour you want or enter specific RGB values for the background colour you want. When you are done with that, click the "Close" button to apply your settings. Then you can close the "Environment and Effects" window as well.

Next up is the screen filter, because we really want to kill the anti-aliasing. Press F10 to bring up the "Render Scene" window which looks like this:

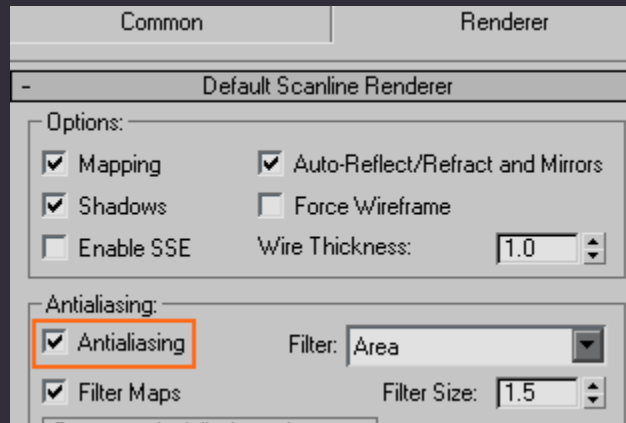


Earlier in this tutorial I mentioned that ink lines only render properly in a 640x480 output size. This does not mean that lines won't render at all in any other resolution, it just means that the lines will scale themselves based on the resolution you enter, which means either tweaking of the lines or actual mouse work after rendering.

For example, if you decide to render your scene at 1280x960 (which is 640x480\*2) you'd have to edit your lines to be half as thick as a 640x480 render (0.5 as opposed to 1.0).

The results are rarely 100% accurate when you do things like that though, you'll find quite a few rogue pixels in and around your mesh's lines. So just go with 640x480 for now.

Clicking the "Renderer" tab brings up the following options:



Removing the tick next to "Antialiasing" will finally get rid of the pesky anti-aliasing. Now when you click the "Render" button at the bottom of that window things should look a heckovalot closer to the intended target.

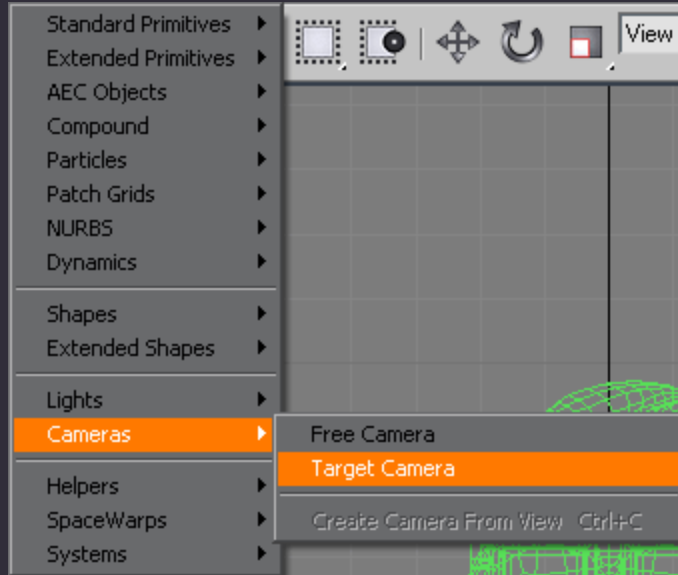
This is how Frank is looking so far (just ignore the anti-aliasing caused by the PDF conversion):



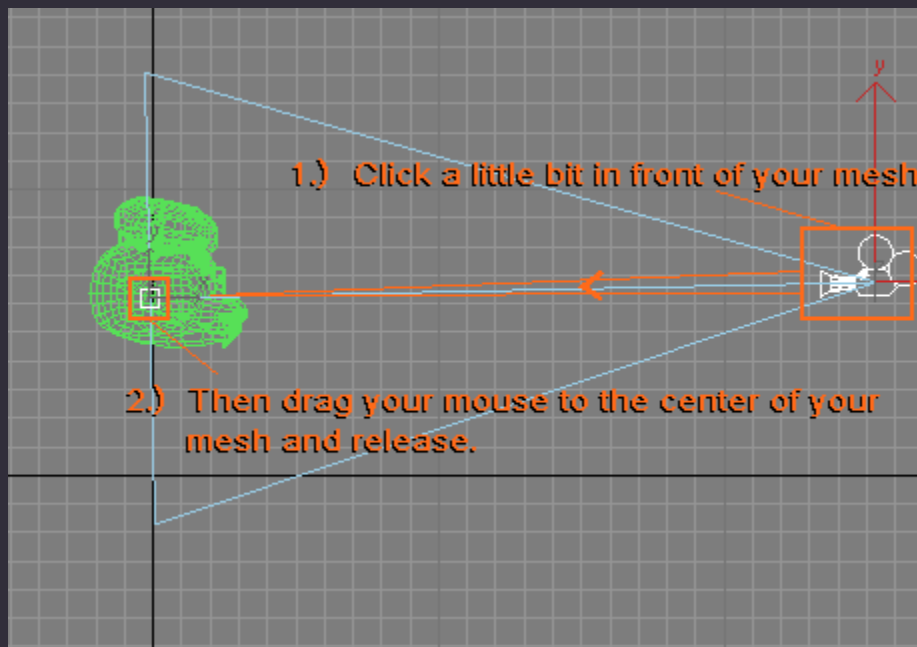
Now that's all fine and dandy, but how do we get the sprite to size up to other sprites and have the proper 2D perspective with just a tap of F9 (F9 renders the previously rendered viewport with your current rendering settings, by the way)? Simple. We just need to add in a...

### Camera (Getting some perspective #1):

Click anywhere in your "Left" viewport to activate it as your current viewport (you will notice a yellow outline appear around the currently selected viewport), then click the "Create" menu item in your top toolbar and create a new Target Camera as shown below:



Now this might freak some people out somewhat. We now have to place an **actual** camera. Here's a picture that hopefully explains the process a little better than words:

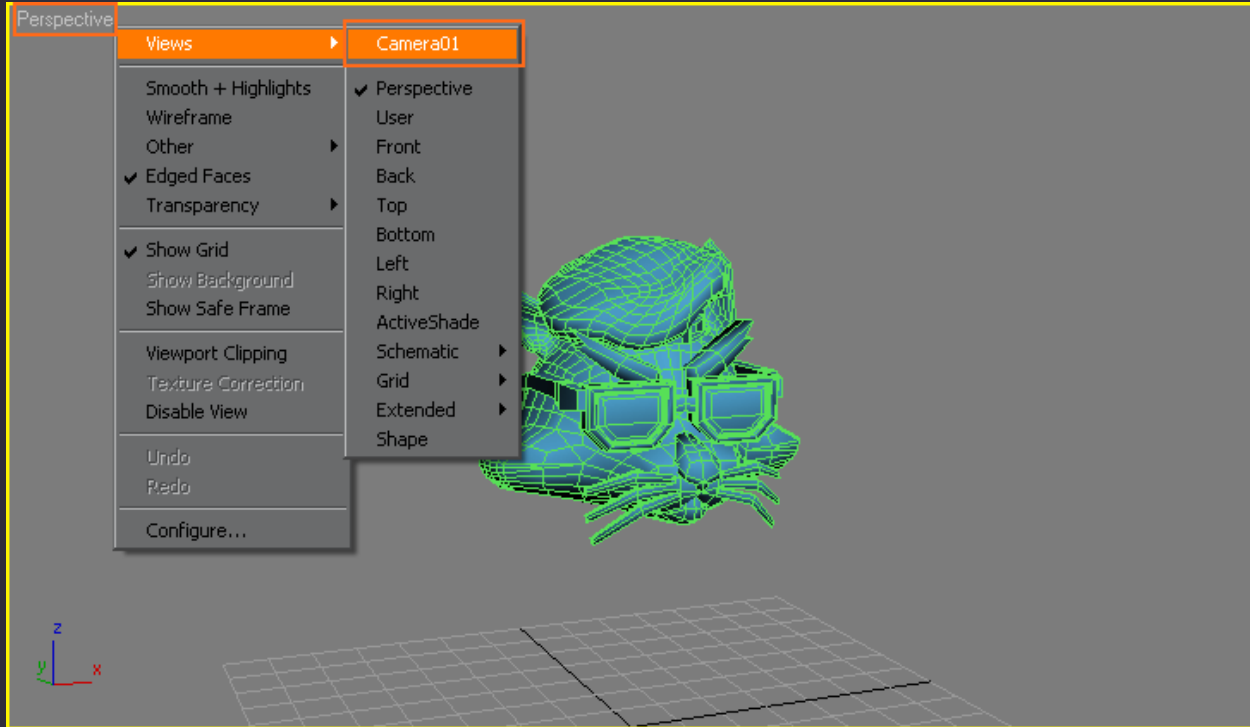


Now, the line from your camera doesn't have to be perfectly horizontal, in fact in some cases you actually don't want it perfectly horizontal at all, but doing this in the Left (or Right) viewport ensures that your camera centers up perfectly to the middle of your mesh. That is, if your mesh is perfectly aligned to the middle of your scene, naturally. I'm just going to assume that it is.

So how can we make use of this new camera exactly? We just have to assign it to one of your viewports, that's how.

You just right-click on the text that in one of your currently selectable viewports and change the viewport to reflect the view of your newly created camera:





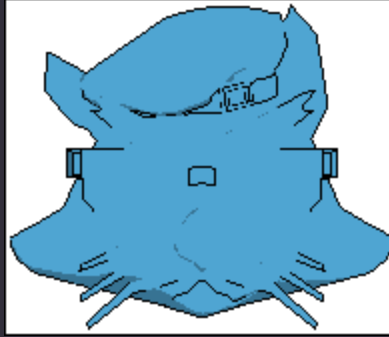
This will change that viewport to reflect the field of view of your camera. As you can see on that image you can also use this method to change any viewport to any angle you want.

But why not just use an existing view that comes with 3DSMax? I'll show you why not. Here is a render with the "Front" view enabled:



Notice how the lines totally screw up haha, he looks like Saddam? That's because the default non-perspective views flatten your meshes prior to rendering it. So why does this do anything to your lines? Well, the lines are actually a double of your mesh that gets scaled based on the parameters you enter into your "Ink" on your "Ink 'n Paint" material. It conforms to your mesh scaling itself just a little bit bigger than your mesh, so when everything gets smashed flat like that the geometry behind your values can't compute properly. And why doesn't this happen to your camera? 'Cause your camera is a non-flattened, focused perspective view. That's why.

When you render your scene through your camera viewport you will get something that looks a little something like this:



Much better. The lines are properly aligned now.

Doing things this way brings up a few interesting concerns. Firstly the perspective on your sprite might be all wrong, as in, things that are closer to the camera are much too big. Secondly, your sprite might be much too big or too small for the setting that it is planned for.

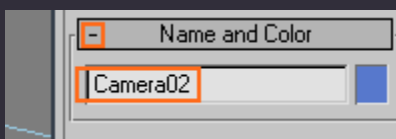
So let's handle things after a bit of tutelage.

### The tools (Knowing your stuff #1):

Here you will get your hands quite a bit dirtier in the world of 3D.

To move around in a viewport you can highlight your viewport of choice by clicking anywhere in it, then hold the middle mouse button or mouse wheel and move your mouse around. By holding down the Alt key while doing this you can rotate your view. Rotating the mouse wheel zooms in and out. And lastly, to reset your view, simply select a view from the right-click menu where you selected your camera view earlier.

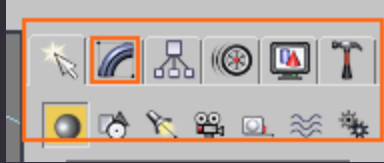
Click back on any viewport that has the camera object visible in it and left click on the camera itself, not the block that is attached to it by a line (the block being the camera target). You'll notice that something changed on the right hand side of your viewports. The name of that particular camera has now appeared within a text area with a coloured cube next to it.



This is the name of your currently selected object. You can change it to whatever you want if you want to organize things a certain way.

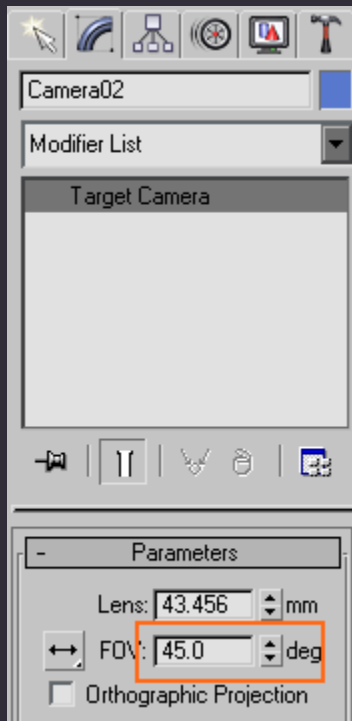
Just above it you see the title of the drop down called "Name and Color". If you click on the "-" of the drop down it will close up, and if you click on it again it will drop down again. Not all drop downs are open by default, so I thought we'd cover that now to avoid confusion later on.

Okay now above that there are a few tabs:



Clicking on that blue rainbow will open up the “Modify” options for that particular camera (or whatever object you may have selected in your scene). The “Modify” tab is pretty much the heart of any object in 3DSMax. You can edit meshes’ geometry, tweak lights and cameras’ parameters, and apply all kinds of interesting modifiers to meshes and so on. Awesome.

Now, with your camera selected and the Modify tab selected you’ll notice that you can edit quite a bit of things. However, very few of these things really apply to what we want to do to create our 2D renders. The only thing that honestly matters is your lens FOV:



So why does this matter? Because not all sprites are flat by perspective and you’d want control over that. I’ll give an example:

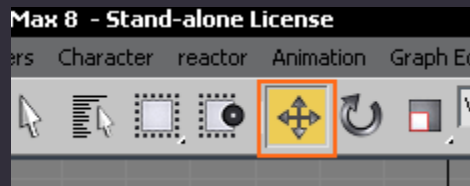


Notice how one of Mariah’s legs are in the background in image A as opposed to image B’s totally flat perspective? This can all be controlled by your camera’s FOV (field of view) and placing in your own sprites. The lower you decide to make your FOV, the flatter your image becomes. I generally make mine around 15-ish to get a balance between

images A and B's styles. So, for now, enter 15 in your FOV and press enter (as a side note; ticking Orthographic Projection will give you flat image, like image B, regardless of your FOV).

After pressing enter another problem promptly presents itself. Your camera viewport is zoomed in really close to your mesh and I seriously doubt you want it to be (I know I don't). We need to fix that by moving the camera around a bit, so I'll explain how to do just that.

Highlight your Left viewport again by clicking in it if it isn't highlighted anymore, next click on your camera if it isn't still selected. Now you need to move it around by using the Move gizmo placed at the top of the program's interface:



This gizmo allows you to move around whatever you currently have selected in your scene and move it around as you please. We can use this new knowledge and apply it to your...

### Camera (Getting some perspective #2 - Conclusion):

Move your camera (by holding left click on it and dragging it) away from your mesh as far as you want until your mesh scales up with your intended resolution. With the right combination of FOV and camera placement you can even make isometric sprites for a Diablo clone or an RPG if you wanted to (notice how the target stays in place when you move your camera around? It means your mesh, or target, always stays in focus, which is very handy).

I moved the camera away from my mesh until it lined up like this:



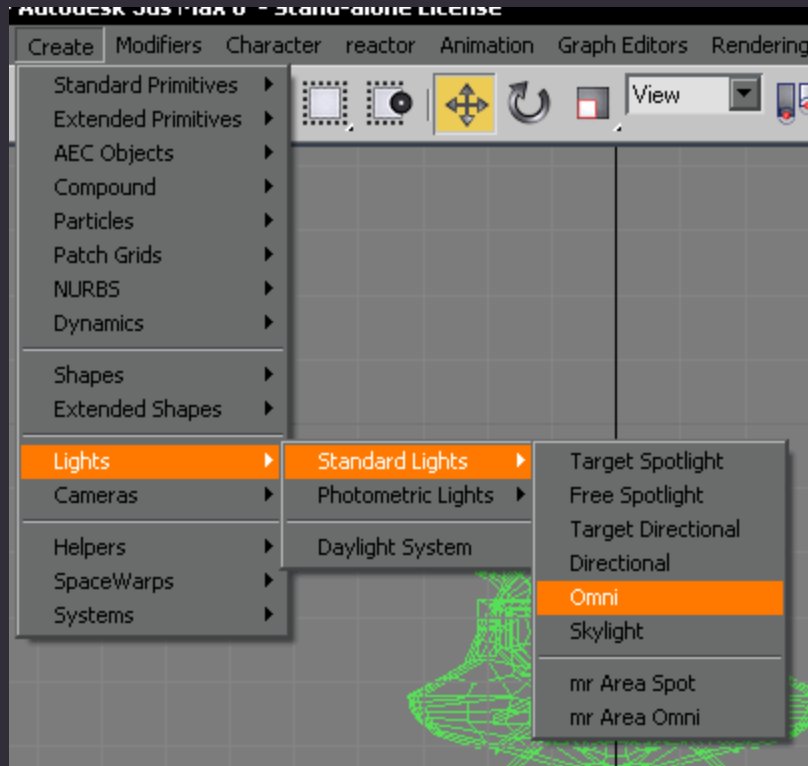
There we go; this is just an example of course. You'll likely need an entirely different scale of sprite.

With that out of the way you pretty much know everything you'll ever need to know about cameras in 3DSMax in terms of 2D rendering. Let's move on.

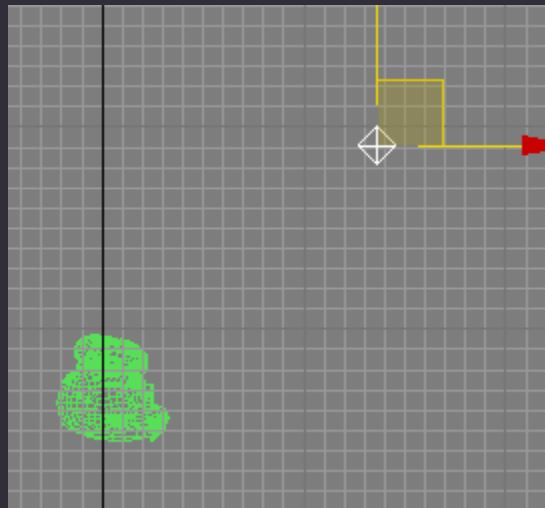
### Lights (Creating the illusion):

As you probably have noticed by now, different commercial companies shade their sprites in different ways. Lights play a very big part in recreating a look or in creating your own look.

With that said, let's get a light in the scene, in the "Create" menu at the top of the program's interface (the one we used to create our camera) and create an Omni light as shown below:



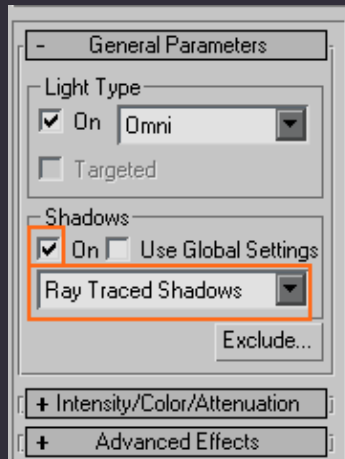
Then place it in your scene (I use a side viewport for this):



This creates a spherical light source. If you want a more direct light you can also use the Target Spotlight which functions like the camera in terms of placing it in the scene (click and drag). I only use these two while working with 2D renders, the others simply aren't as quick to set up.

Do a few quick renders (F9) while moving your light around to find a light source angle that suits your needs. When you are happy you can set up shadows. I only use ray traced shadows for this type of rendering, it is the cleanest way to do it.

Click on the Modify tab of your light and activate its ray traced shadows by ticking the "Shadows" on and setting your shadow type to "Ray Traced Shadows" as pictured below:



And there you go; doing a quick render on Frank had the following result:



As you can see things are coming along quite nicely, the shadows really add to the detail of the sprite.

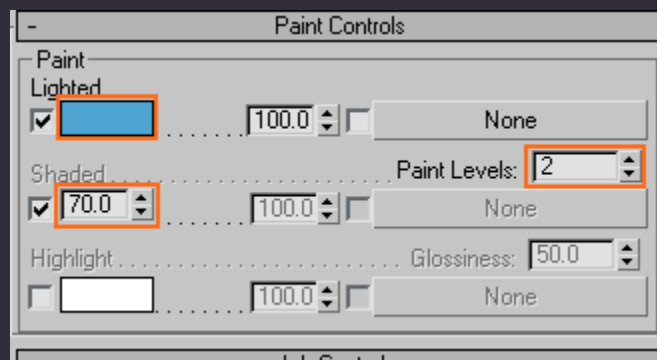
That covers lighting as far as you should require it to for 2D projects. In some cases you might need more than one light to light up a mesh like you want to (with both of them running on an intensity that's less than normal), but not particularly frequently.

Now that we have all of that done we can get back to...

### Materials (Getting that sprite look #2):

Are you getting sick of the two-tone blue fox? I know I am. So we're going to fix it.

Press the M key again to open the Material Editor if you've closed it earlier; if not just click on the material you've assigned to your mesh and I'll show you a few tricks.



If you click on the first blue box you can change the "Lighted" tone of your material. The "70.0" is the "Shaded" tone.

Adjusting the 70 will adjust just how dark the darkest shade is on your mesh. For example, making your "Lighted" tone pure white and your "Shaded" tone "0.0" will create a black shaded area on your material (with 100.0 being the exact same shade as your "Lighted" tone).

Un-ticking the tick next to the "70.0" will allow you to input your own colour for your shaded tone. There is no way to control what tones you get in between the two tones using the default Ink 'n' Paint material, but I'll get to that later.

Now on to that "2"... It controls just how many tones are present from your lightest tone to your darkest. Change it around a bit and render a few times to see what I mean. I'm going to change it to 3 on Frank.

This is how he looks after a colour and level adjustment:



A few clicks made a pretty drastic change. The lines, however, are still an ugly black, but never fear, we can edit that as well. Clicking on the black box next to a particular line enables you to change its colour.

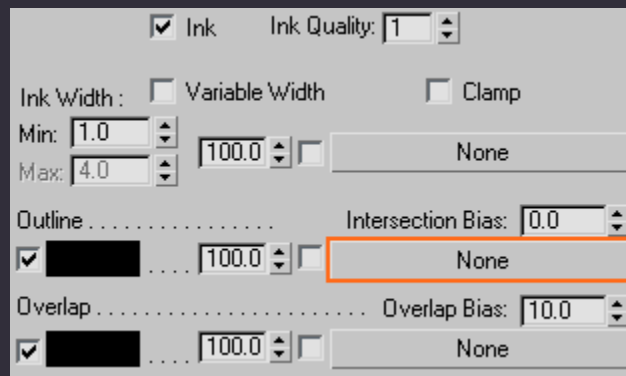
Alas, there is yet another problem. The lines can only be one colour at any given time, whereas the rest of your sprite has a couple of tones. **Completely unacceptable**, but thankfully it can be fixed.

I didn't think it could be fixed for quite some time during this method's developmental phases until I discovered...

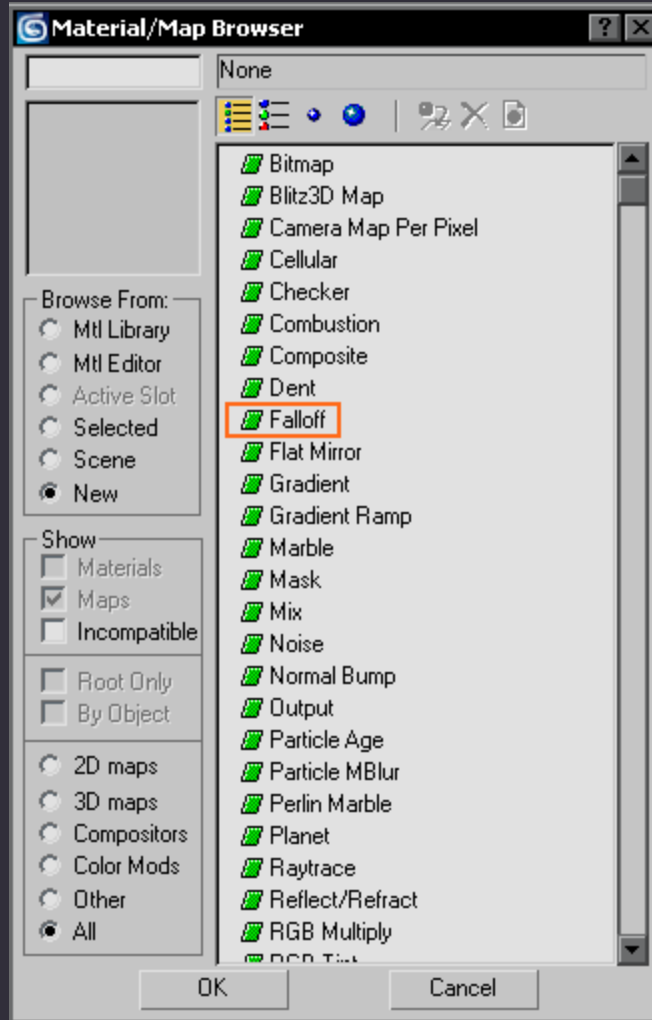
### Nested materials (Getting that sprite look #3):

Well, in all honesty I knew about nested materials for a long time, I just didn't know how to properly apply them to 2D as it was required. I tried all kinds of plugins for 3DSMax to try to achieve what I needed and ended up finding a solution in vanilla 3DSMax. Figures.

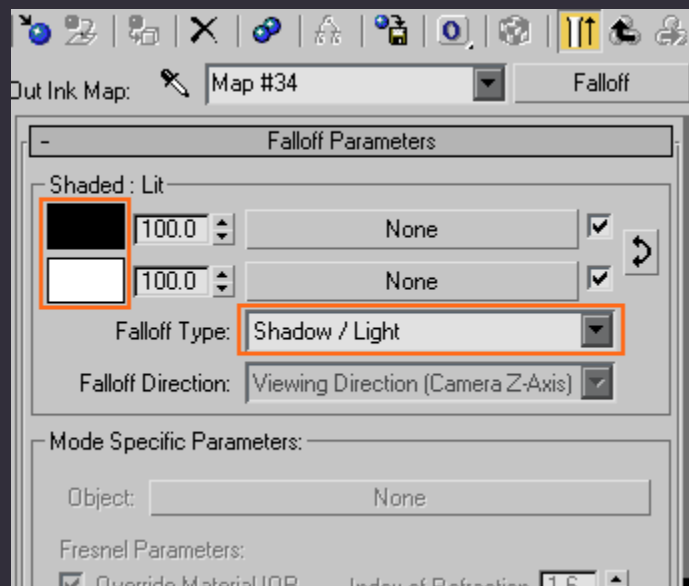
Clicking on the area indicated below...



...Brings up this window...



Double click on "Falloff" to apply a falloff material to the outline of your "Ink 'n' Paint" material.



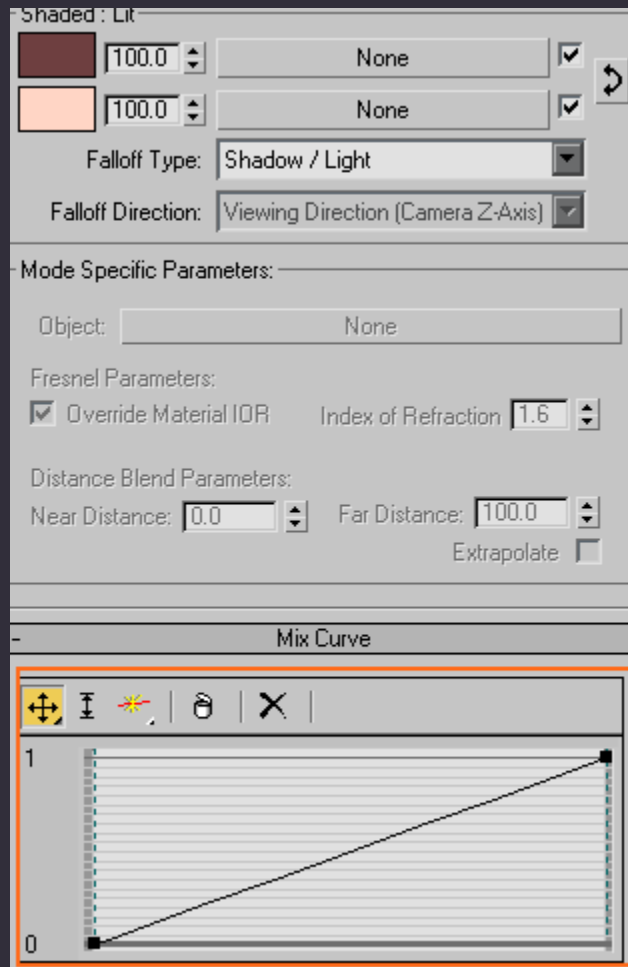


Explanation time.

Change the "Falloff Type" to "Shadow / Light" as shown in the picture, this creates a gradient that originates from the point on your mesh with the most light and works it's way down to the point with the least light. Also, as you can see, you can customize your colours as you see fit.

Perfect? Almost.

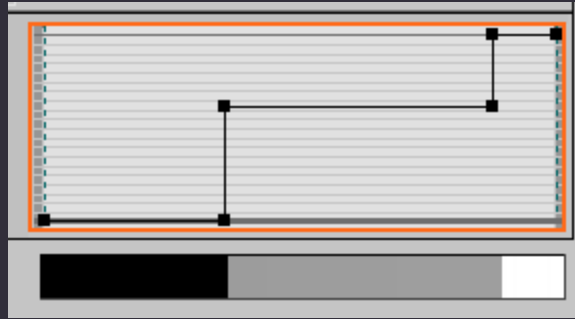
Having your mesh shaded in a few layers and having a gradient for your lines just doesn't look right. It looks better than boring monotone lines, but still *not quite right*. This is how you fix it.



See that black line (Also notice that I changed my colours)? It controls how the gradient, called the Mix Curve, flows. You can make quite a bit of adjustments to it with the tools provided above the line itself.

The first tool moves an existing point around, a "point" being those black boxes that are found within the mix curve. The second tool moves a point vertically only. Tool number three creates a point in the line, and then the next tool deletes the currently selected point so long as the point isn't one of the starting points. The final tool resets the line to default.

So how can we use it to create the look we want? Well, by creating and moving points to the right places in the mix curve we can recreate the "Ink 'n' Paint" layered look for our lines. This is done by editing the line into a staircase-like form as pictured below:



This is how Frank's mix curve ended up looking. Just create and move points around whilst doing the occasional quick render until you are happy. The gradient below the mix curve will give you an idea of how things will look.

Vary rarely this kind of mix curve will create the odd light pixel; this unfortunately needs to be fixed by hand after the render. Either by fixing it by mouse, or by palette adjustments.

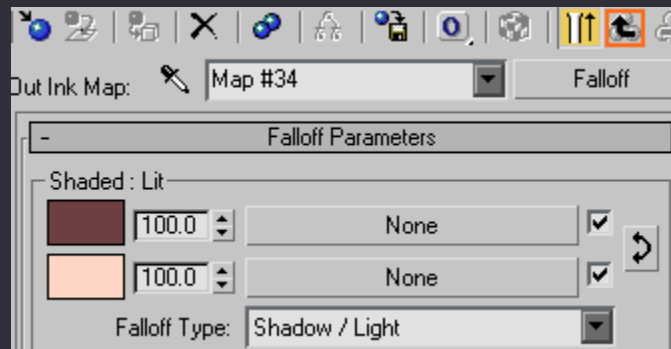
I occasionally (well, pretty much always actually) use a nested "Falloff" material for my main shade layers, it's the only way to directly control when and where the layers will occur. It's also a way to add a bit of anti-aliasing between colours for a smoother colour transition. We'll get to that a little later.

This is how Frank looks with the new nested material:

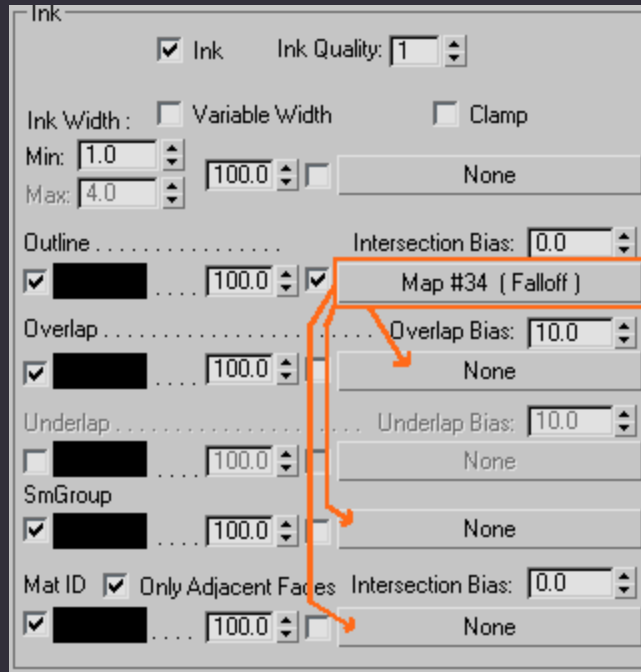


I made sure that my lines were coloured a bit darker than the colours used for the main mesh's shading, and it looks good, but there are still some lines that are black. We'll fix this in a few steps.

Firstly we need to go back to the main "Ink 'n' Paint" material by clicking here:



Then right-click on the nested material and paste it in the other active Ink material slots as pictured below:



I'll explain what the other lines are.

Outlines affects only the outermost lines of your geometry, you can see exactly what I mean on the previous Frank render.

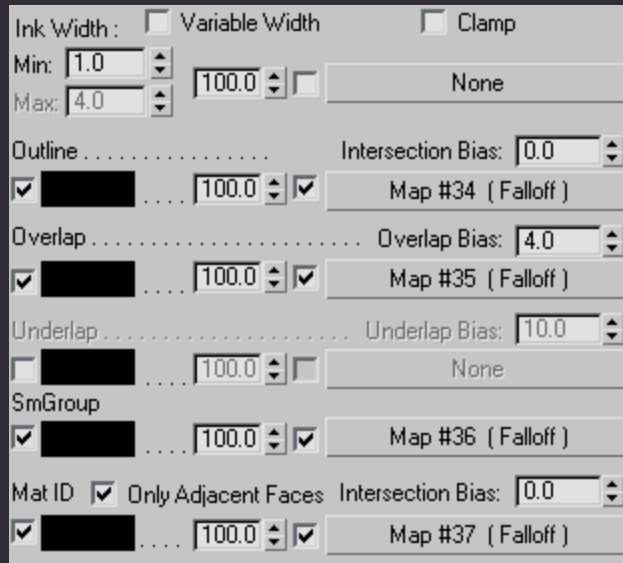
The Overlap lines are activated any time when a part of your mesh that uses the same material as another part of the mesh overlaps each other from your camera's perspective. The Overlap Bias controls just how far these lines will go, a value of 0.0 will create the maximum amount of lines, and 100.0 will do the opposite. I generally hover around 4.0 to 7.0 for my work. For Frank I went with 4.0.

The Underlap lines work the same way as the Overlap lines, but they only show when a part of a mesh Underlaps another. I alternate between Underlap and Overlap, but never use them together.

The SmGroup will create lines on smoothing groups, if your mesh has any. A smoothing group controls the general look of your mesh, for example: A shoe might have a round sole and a flat bottom for that same sole; this is usually achieved via smoothing groups. Having SmGroup lines activated will draw a line where the two smoothing groups meet on your mesh's geometry.

Mat ID will draw a line when two different materials meet on your mesh's geometry. We will actually be doing this later, so I don't have to talk about it a whole lot.

The image below shows how my Ink setup looks so far. The Falloff has been pasted and the Overlap Bias has been set to 4.0:



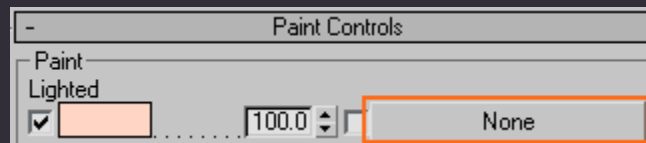
And this is how Frank looks with all of that applied to his material:



That kind of material setup should work for most sprites, but I like taking things a bit further. We can do a lot more with a proper nested Falloff material and that is exactly what I'll be covering in...

### Falloff (Getting that sprite look #4):

Click on your Lighted shade's nested material of your Paint Controls:



Add a nested Falloff material in there and adjust your colours as you see fit with a "Shadow / Light" Falloff Type.

If you use the old staircase Mix Curve like we did with our ink you can recreate the Paint Levels of the "Ink 'n' Paint" material. So why do that anyway? Because it gives you quite a bit of freedom as opposed to how the default paint levels flow on your mesh's geometry.

I'll give an example on a sphere:

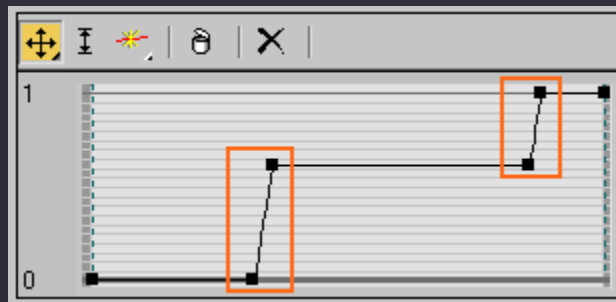


This is a plain sphere with Frank's material applied to it as it looks now. Next up is a sphere with a falloff material applied to both the Lighted and Shaded material slots (Activate the nested material by ticking it on in the box just left of it):



This is already more freedom than the default material allows, but we can do even more with this material.

Changing your Mix Curve from a rigid staircase to this (notice the tiny ramps as opposed to rigid angles)...



...Will create the following effect of localized anti-aliasing:



This just smoothes out the transition between two colours, but it can make things look a lot better if that's the look you want. Naturally you can apply this information to your ink lines as well (and you probably should to ensure that things conform properly, just copy-paste it and change the colours again).

You can create some interesting effects with your Mix Curve. The higher a point on the mix curve the lighter that particular area in the gradient becomes.

A Mix Curve like the one pictured below...



Would create a material that looks like this:



It is just a matter of creating and moving points on your curve.

Now that we have the 2D-ish materials figured out we can move on to learning how to apply more than one material to your mesh in...

### The tools (Knowing your stuff #2):

We've gotten the material creation process down, now we need to learn how to colour different segments of your mesh with different materials.

For this purpose I have prepared a few materials to use on Frank as shown below:

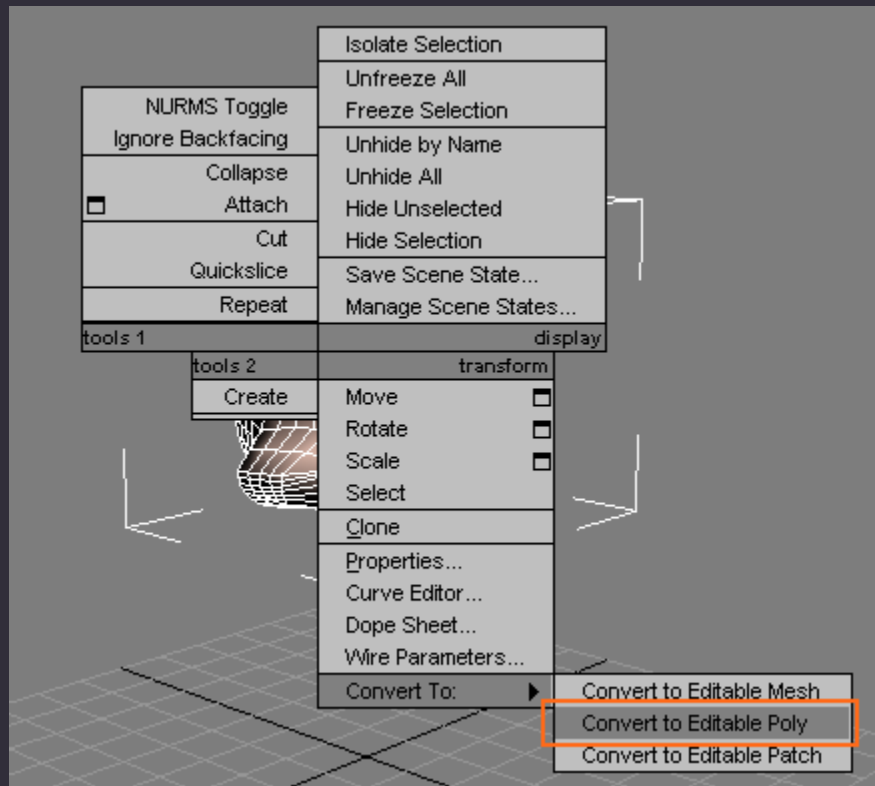


Remember to **give each material its own name** in the highlighted box so that neither you nor 3DSMax get confused.

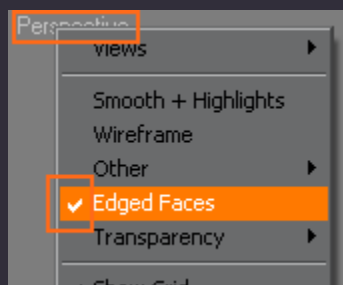
You'll also notice that some of my materials have a highlighted area on them; this is achieved by ticking on the "Highlight" box of your "Ink 'n' Paint" materials and adjusting the "Glossiness" value (They also support nested materials, as you can probably see).

Alright, now we learn how to apply different materials to parts of the same mesh.

Right-click on your mesh in any viewport and convert it to an "Editable Poly" mesh as pictured below:



And in your Perspective viewports turn on Edged Faces like so (Right-click on the word associated to that viewport's view):




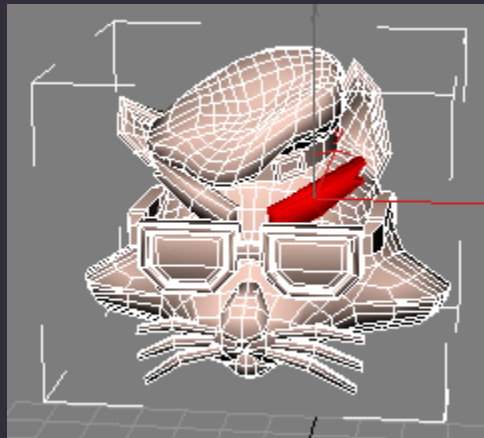
This allows you to see the wireframe of your mesh superimposed on it's shaded geometry. This is useful for polygon based selecting and editing work.

You may also have noticed that when you converted your mesh to an Editable Poly mesh, the mesh's modify tab popped up automatically. I will explain everything you need to apply colours as we go along.

You have the following options when selecting parts of your mesh (from right to left):



Firstly, on the rightmost side of those red options, is the "Element"  box. You can use this to select a part of your mesh that is not attached directly to the other parts. In this case I'm going to select an eyebrow of Frank as an example:




When you have something selected via your Selection tools it becomes that bright red colour. You can drag a Material on that specific selected area just like you did when you applied the first material to your mesh earlier on in the this tutorial.

Dragging another material on it and doing a quick render has the following result:



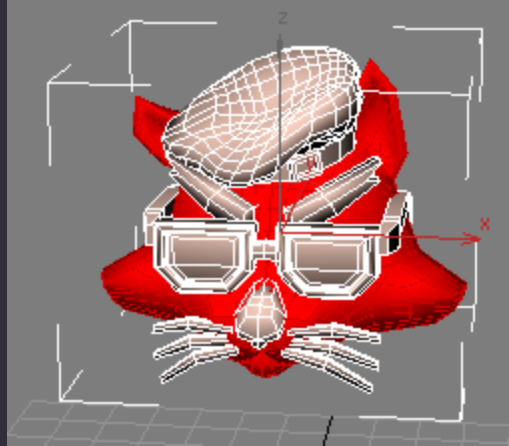
Quick and easy. You can also probably see why it is best to make your own made-for-2D meshes though. If a mesh is made properly it only takes a couple of minutes to colour it up into a usable sprite.

Next up is the "Polygon"  selection tool. This is used to select a polygon at a time. Holding Ctrl whilst clicking on a polygon selects that polygon plus anything you've had selected so far. Holding Alt whilst clicking on a polygon deselects that particular polygon. This also applies to hopping between selection tools. For instance, when you have an element selected and want to remove a couple of polygons from that selection you can select it with your Element selection tool, hold Ctrl, and click on the Polygon tool. The element will still be selected, ready to be tweaked on a polygonal level.

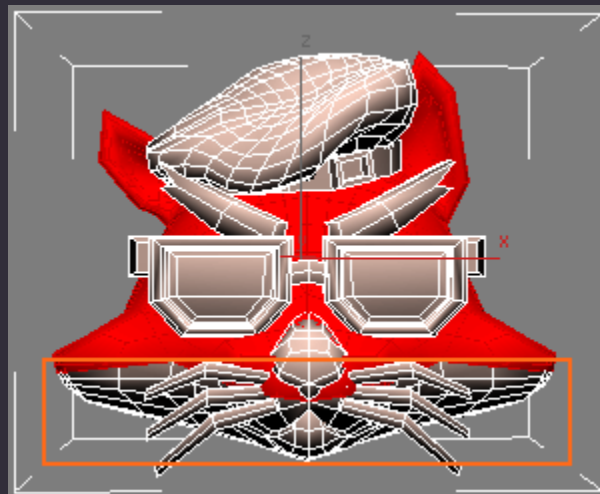
I'll demonstrate.

I'll select Frank's head via the Element tool:





Then I will hold Ctrl and click on the Polygon tool. With the element still selected I hold Alt and deselect as many polygons as I need to:



Then I apply my material and it renders to:

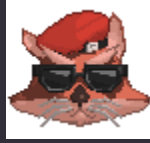


As you can see the two connecting materials create lines on the mesh itself. This is due to the Mat ID lines being active on both of the materials that I am using in this example. Mat ID lines are useful for various effects (like lines on a tracksuit, etc), but in this case I'm going to tick them in the off position. It now renders to this:



The following three selection tools aren't used as frequently when applying materials to meshes. I'll let you explore them on your own. Hovering over any tool in 3DSMax tells you what it does anyway.

This is Frank with all of his materials (that were pictured earlier on) properly assigned:



He actually looks like a fox now, and is a totally usable 2D sprite.

### The finale (25 pages later):

In retrospect the tutorial wasn't that simple.

However, when you have a pose-able mesh that is in the proper scene with the required materials assigned to it you'll cut hours/days/weeks from your spriting time if you have a big project lined up.



The above sprites were less than a minute worth of work (with the light moved around a bit). Not too shabby.

I hope this tutorial was at least of some use to some of you, I'll be around if any of you need a couple of pointers or have something to add to this tutorial.

### Credits (People that helped):

Thanks go to my proof readers: Fou Lu, Rolento, XCB, Kung Fu Man (not KFM), Abster, Artius, Xar, AntMagnum.

