4 - Textures and Mapping

In this Chapter:

- *Working With Textures*
- *Alpha Channels*
- *All About Materials*
- *UV Mapping*
- *Mapping Faces*
- *Mapping Tips*

Models always look better with textures. In my opinion, they look 10 times better. Nowadays, many people take the easy way out, and just "flat-texture" their models, by assigning a flat color to the body panels and only drawing and mapping the door lines. I have never liked this. I think that textures need some sort of static shading to make them look real. You can do this by incorporating photos in to the textures, which I do, or hand-drawing shading (which I've also done). However, that's not the purpose of this chapter. Unfortunately we lack both the time and the space to discuss how to make great textures.

Also, since this is a modeling tutorial, this sort of topic would be techincally off-subject. So we'll just discuss how textures relate to modeling. Therefore, in this chapter we will discuss how to work with textures, materials, and mapping.

Working With Textures

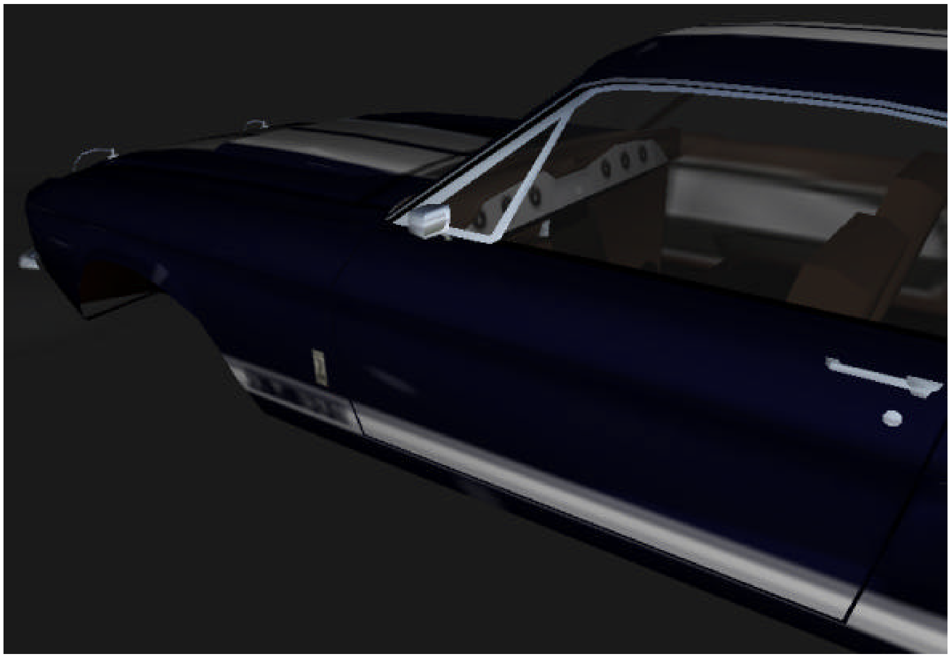
Before you can start to work with textures, you must understand what they are, what they do, and how you can create them. Textures are image files, like the pictures you can see on my web site. They are special because they contain the graphical details of a model. They are essentially pictures of the object from different angles, with little details all arranged on one canvas. Many people call texturing "skinning" because it adds a "skin" that stretches overtop to your mesh, giving it detail, color, and shading. Textures are affected by the lighting and shading of your mesh, but they are not totally at the mercy of the model's dynamic shading. They add invaluable lighting interest to the model. They can also be used to add details to it. Photos are a great resource for details, because it's nearly impossible to hand-draw a perfect taillight, for example. Therefore, even those who use hand-drawn textures resort to photos for textures of emblems, lights, and other details.

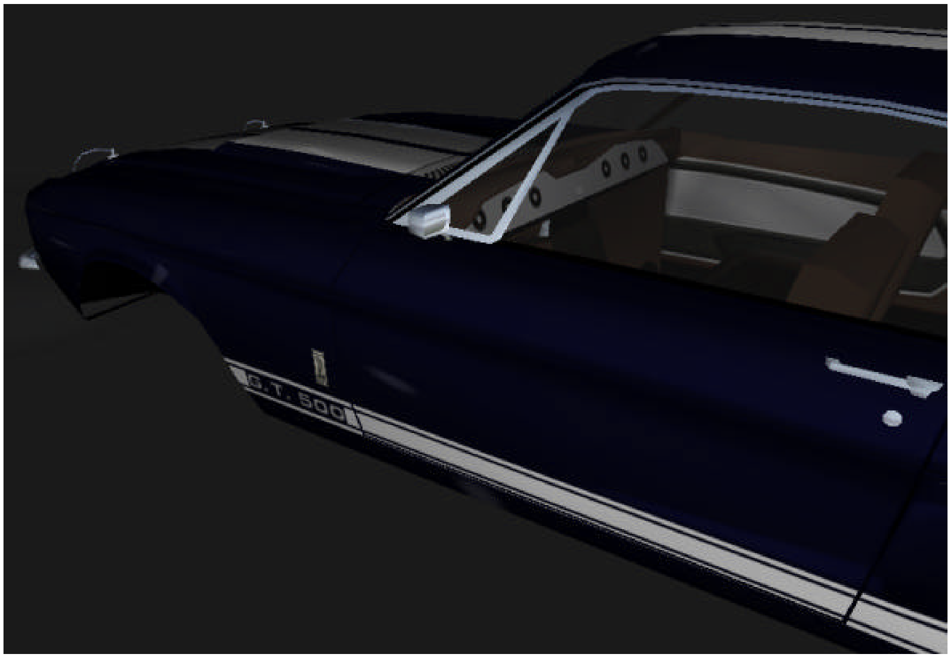
However, these images aren't just thrown at the mesh and expected to be correctly placed. That's why we have a process called *mapping*. Mapping is how you assigning an image texture to a portion of a mesh. Therefore, when we make a texture we MUST map it to the mesh, and do it correctly, so that it looks good.

Most of the time, textures are not just untouched photos of whatever it is that you're modeling. Instead, they are distorted, retouched, airbrushed, groups of pictures in one image. The beauty of mapping is that you can assign different parts of an image to different parts of a mesh. For example, you can have side, front, and back views of a car, all in different sections of the same image, and then you can assign, or *map*, the various parts of the car onto the corresponding parts of that image. In fact, one of the virtues of a good texture is that it makes the most use of the space on the canvas, and fits as many parts of the car on one image as is possible, all at the highest possible resolution.

This issue of resolution is another key concept when working with textures. A practice that takes time and experience to develop is knowing just how much resolution is needed to properly display the desired amount of detail. Therefore, you don't want to include a huge texture of a certain emblem on the car, but relegate the side texture to a tiny portion of the image. The fact of the matter is, if you zoom in close enough, *almost any* texture is going to look blurry and pixelated. This effect can be minimized, though, by a process of selection. You can take the portions of the model that are very large (and thus subject to a larger zoom factor) and make them have larger textures, and those parts which are small and less visible will have smaller textures. This is a balancing act which you will learn from experience.

Overall resolution of your texture is important as well. You have to work within the limits of whatever application you choose to make of your model. For example, those of you who still work with Need For Speed High Stakes are limited to a single texture with a size of 256x256 pixels. Those of you who work with Racer Free Car Simulation are not constrained, however, it is wise to keep texture sizes at or under 1024x1024 pixels, at least at the time of this writing. In the future, 128 Mb 3d cards will make 2048x2048 or even 4048x4048 sized textures possible. However, for now, 1024x1024 pixels is plenty enough space to draw sharp, clean, and stunning textures. On the next page you'll see the same shot of the same car. One has a single 256x256 texture, and the other has a single 1024x1024 texture.





You can readily see that the higher resolution texture (the one on the bottom) creates a sharper image.

Now, in many applications, you are allowed to have more than one texture per model. This is very useful, allowing you to have a different image for interior, trim, and/or tire textures and so forth. A quick pointer that I want to share with you is a way to optimize your car from the very beginning. If you want to make your model easier for people to run in a game, or if you want it to be quicker to render, or whatever, you should use fewer textures, at higher resolutions. Actually, loading many smaller textures degrades performance more than loading one big texture. If you keep the number of images used for texturing down, you can improve the performance of your model. At this point, I think it's my duty as a texture editing veteran to explain the most common point of confusion. Alpha Channels. Before you run screaming away, read the next section. They're not so scary after all.

The Dreaded Alpha Channel

One of the most common question I see asked is: "How do I edit an alpha channel?", which is sometimes preceded by "What's an alpha channel?" First let's define alpha channels in general.

To understand an alpha channel, you have to understand what a *channel* in an image is. Basically, an image is displayed using pixels, which are each colored. The colors on your screen are displayed by mixing red, green, and blue. If you look closely at your TV or monitor when it is off, you can see tiny red, green, and blue dots on it. By using these three colors to a greater or lesser degree, almost any color can be displayed. Therefore, Red, Green, and Blue have to be represented in an image file. This is done with *channels.* So, each image has a red, green, and blue channel which contains all the data on each respective color within the image. So if you looked at each channel alone, it would look like a partial image displayed in varying shades of one color. In modern computers, you can usually display 256 shades of red, green, or blue. This means that you could display 16777216 colors in any given pixel.

Alpha channels are also channels of an image. Some images have them, and some do not. In fact, only certain image formats allow you to add an alpha channel, like PNG or TGA. Thus, when working with JPG, GIF, BMP, or other files, you can't deal with an alpha channel. What does it do? Well, since every possible color of the spectrum can be displayed by combin-

ing Red, Green, and Blue, it has nothing to do with the color of your pixels. Instead, it determines the *opacity (transparency)* of your pixels. So, each pixel really has four channels that determine how it is displayed. Red, Green, Blue, Alpha.

So, how does an alpha channel determine the transparency of your image? An alpha channel is basically a black and white image. When working with an alpha channel, you paint black, white, or grey onto the image, and that determines how transparent that area is. When the alpha channel is black, it is totally transparent. When it is white, it is totally opaque. In between those two extremes are 254 shades of grey. The darker your grey, the more transparent it will be, and the lighter your grey, the more opaque it will be.

That's the concept behind alpha channels, but how on earth do we edit them? Photo editing programs have the capability to load, display, edit, and save alpha channels. Usually, this is done by means of a *mask*. A mask is just a photo editor's way of displaying an alpha channel. It lays on top of your image, and is usually visible by showing your image in varying shades of transparency, showing the checkerboard transparency behind it.

For starters, your image probably does not have an alpha channel. So to go about making one, you need to find the Create Mask command in your photo editor. You want to create a white one to start with, so create one that shows all. The specific command in Paint Shop Pro 7 is Masks...New>>Show All.

When you edit a mask, you will usually have to enable mask editing, so that you don't alter the Red, Green, and Blue channels. Once you have done that, you need to select areas that need to be something other than white. Then you need to decide how transparent they should be. Keep in mind the rule. The darker your alpha channel, the more transparent it is.

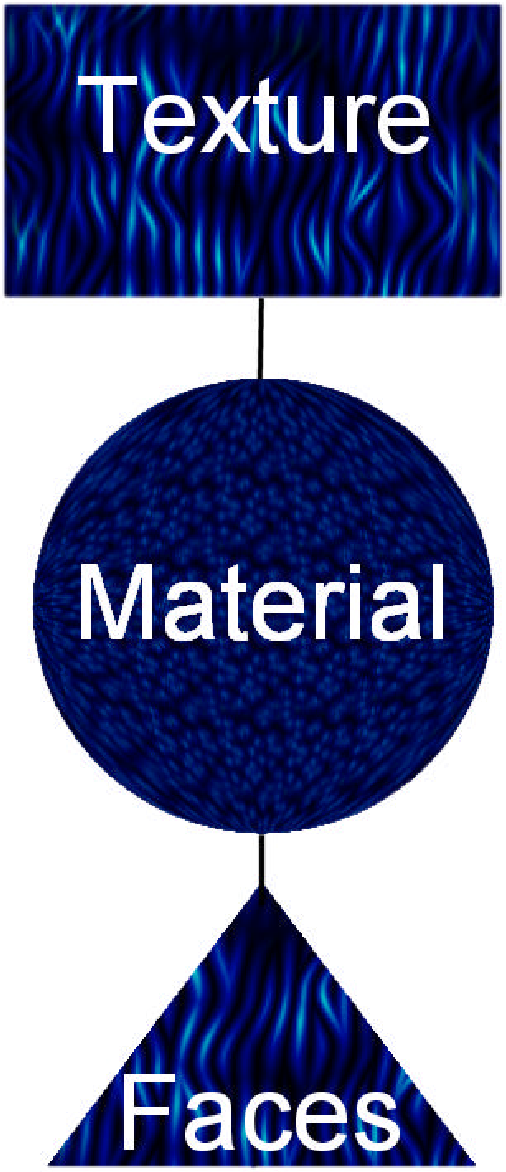
When you have finished editing your alpha channel, you have to remember to do two things to the mask. You have to first save it to the alpha channel. This is immensely important. You have created a mask, but that does not equate to an alpha channel, you have to put it there. The specific command for this is Masks...Save to Alpha Channel. Also, make sure you only have one alpha channel, so overwrite any alpha channels that are stored in the image. The second thing you need to do is delete the mask after you have saved it to the alpha channel. This is equally important. If you don't delete the alpha channel, it will be merged into your image, distorting it beyond repair. If you have to, write this on a post-it note and stick it to your monitor. (As you can tell from the tone of this warning, I've lost many a texture to this silly mistake.

So as you can see, Alpha Channels are very important, though not as scary as they seem at first. Now you're armed with the

knowledge you'll need to move on to our next section. Let's learn how Z-Modeler keeps track of the different textures assigned to a model. It does this with *materials*. Let's talk about what materials do for you, and what you can do with them.

All About Materials

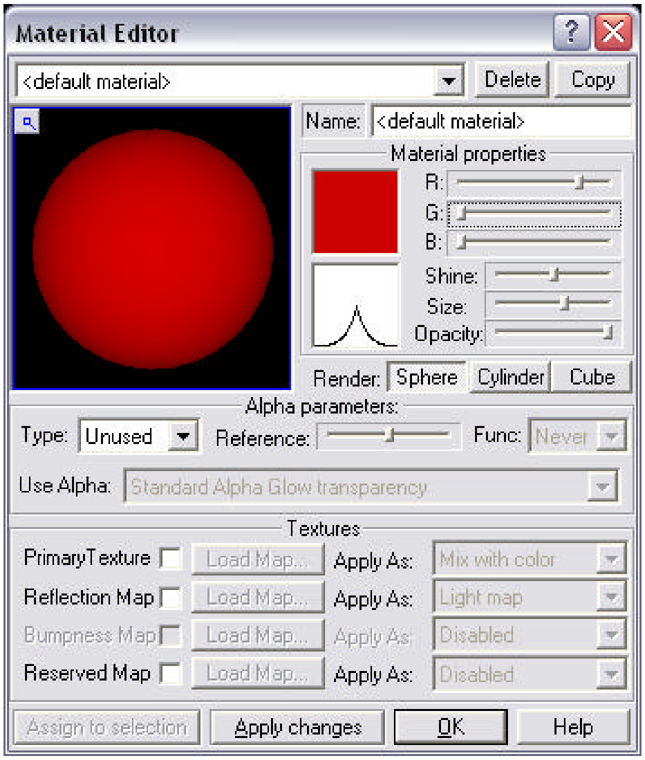
Think about a car (this is something I do almost constantly). Now, think about the different kinds of raw materials that make up that car. There is steel or aluminum for the body panels, some plastic and chrome for the trim, perhaps rubber on the bumper, and vinyl, cloth, or leather in the interior. All these different materials have different properties. For example, the steel is hard, shiny, and painted. On the other hand, the cloth is soft, flexible, and matte. In a similar way, you have materials in your model as well, but the differences are more subtle. For example, materials can have different textures, they can be reflective or matte, they can be darkened or bright, and they can be transparent or opaque. Therefore, you will end up grouping faces that have similar properties AND who share the same texture into the same materials.



Here comes an interesting concept. Faces are not directly assigned to textures. Faces are assigned to *materials*, and materials are assigned textures to display. So, to determine some of the properties of the faces you have selected, you will assign them to a certain material. To the right is a diagram of the pipeline for materials, textures and faces. Hopefully, it won't confuse you more than that last paragraph.

As you can see, the texture represents the entire image, which is then assigned to a material, where shading and reflection properties are added to it. Then, faces which belong to that material can display a portion of that texture. So, faces have to belong to the materials that are assigned the correct texture in order for the "skin" of your model to look correct.

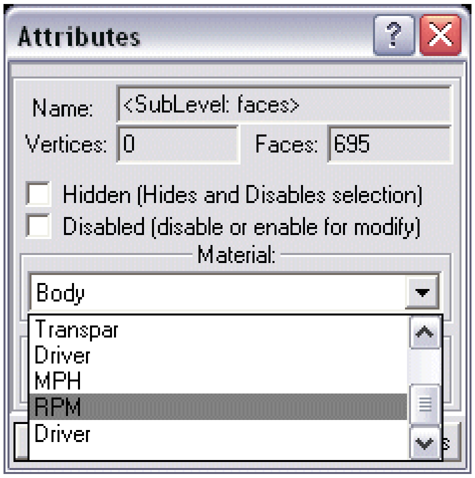
If you have Z-Modeler open, hit the "e" key now. This will display the Material Editor. This is where you can assign textures to materials, change the color of the material, and choose whether it is reflective, matte, opaque, transparent, et cetera. It should look something like this:



The controls of the material editor are fairly straightforward. Let's break them down in a table.

Section	Description
Name and Controls	The two buttons at the top, Delete and Copy, delete or copy the selected material. Copying is the only way to create new materials. You should type a name into the name box of each material you copy, that way you can tell them all apart.
Material Properties	The six sliders in this section control how your material displays. The R, G, and B sliders all affect how much Red, Green, and Blue tint is applied to your material. As you can see in the picture, I have no Green or Blue and a high Red value. This has given me a strong red material. The Shine and Size sliders affect the brightness and largeness of any specular highlights, if you have them turned on. This allows your materials to have less or more specularity. It is really only useful for in-Z-Mod renders and DOF Export The Opacity slider has to do with the transparency of the material with regard to a certain setting from the following section.
Alpha Parameters	The Alpha Parameters deal with transparency in certain materials. Depending on which game you export to, you will have to use different Alpha Parameters in order to make transparency work right in that game.
	Unused: Use this setting if you want a completely opaque material.
	Glowing: This is the most common setting for transparency. Use this, along with the sub-setting "Standard Alpha Glow Transparency" to allow the Primary texture's alpha channel dictate the transparency of the material. This way, if the alpha channel is a dark grey, it will be barely visible, but if it is light grey, it will be mostly visible. All the rest of the sub-settings in this setting seem to make the entire material semi-transparent, and glowing. In conjunction with these settings, you can set the transparency of the material with the Opacity slider above.
	Testing: This setting looks at the alpha channel of a material, and draws the texture as totally transparent if the alpha channel meets the conditions you set. You can set it to make everything less than or equal to a certain value transparent, and then you adjust the value with the "Reference" slider. Or you can have it be everything greater than or equal to, everything not equal to, everything equal to, or everything greater or less than a certain value.
	Color Key: This setting allows you to pick a color within the texture itself (nothing to do with the alpha channel) which should be transparent. You have probably seen games whose textures have bright pink areas which designate that it will be transparent. This is how you make an equivalent setting in Z-Modeler.
Textures	This section allows you to specify which texture images will be assigned to the current material. The Primary texture is the one which you will map to. It is *the* texture you use. The Reflection Texture allows you to load an environment, or reflection map, which will reflect onto your model. As you can see, in the future, there may be bump mapping in Z-Modeler, which will be quite cool, and when it is implemented, the "Bump Map" section will allow you to add it in. The Reserved Map is used for damage texture in some games (like Midtown Madness 2). Each texture can be loaded in one of three modes, Unknown, Mix with Color, or Light Map. Unknown is just like disabling the texture. Mix with color is the setting you use to make the image lay on top of the mesh. Light map is the setting for reflection.

Now, if you have materials set up, how does Z-Modeler know which faces are assigned to what materials? You can assign this using the Attributes box. You can only assign faces to materials, so switch to the faces mode of an object, select a group that you want in a certain material, and hit CTRL+A. In this box, select your material, as shown here.



After you have selected the material that you want those faces to belong to, click OK. That's it. Now, however, you have to tell Z-Modeler what portion of the texture that you want those faces to display. This is where many people get frustrated and give up in their modeling, and that's the art of UV Mapping. Let's talk about it in our final section for the chapter.

UV Mapping

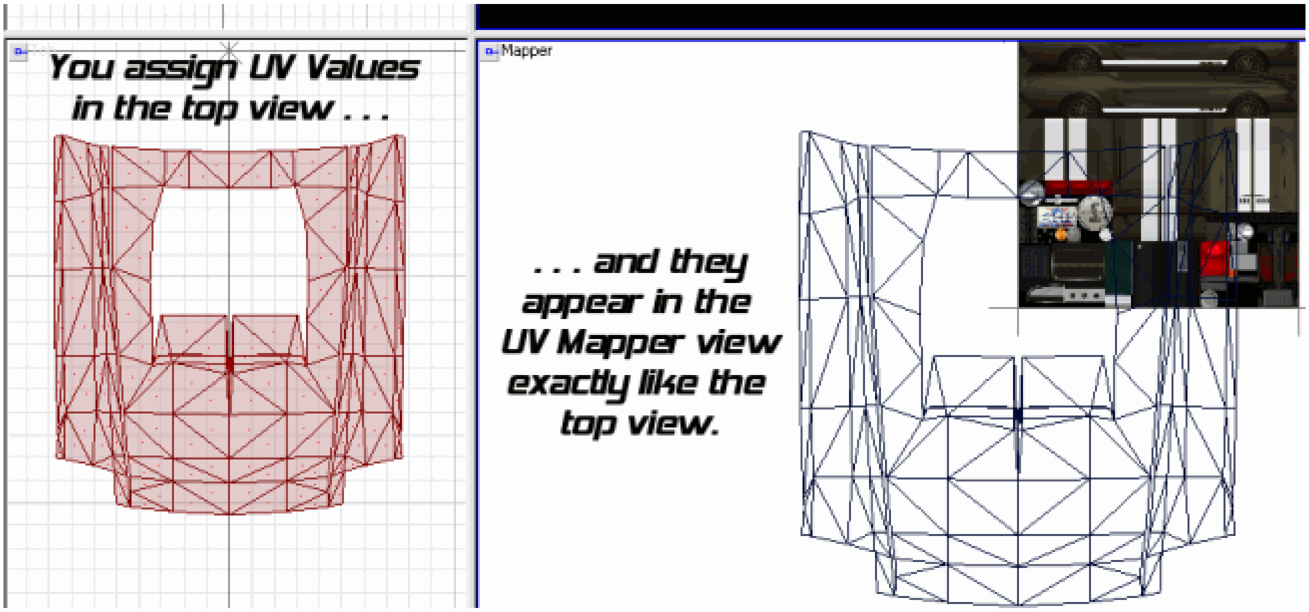
Let me start with explaining why they call it UV Mapping. Basically each vertex has 5 properties. It has its familiar X, Y, and Z values, which tells the computer where in its virtual 3d world this vertex exists, and it has a U and a V value. These values tell the computer where on the texture map the vertex exists. Then, when you place the vertices of a certain face on a texture, using U and V values, the face will display the texture that falls in between those UV coordinates. Do you have to know this? Absolutely not. It just helps. Z-Modeler makes this process easy. Let's look at how.

Assigning UV Values

Assuming that you've correctly assigned the faces that you want to map to a material, you can automatically assign U and V values to your verts. Now, to start with, your faces have no UV values, so they basically have no texture. To alter the UV values of your verts, Z-Modeler provides a special UV Mapper view. Any view in Z-Modeler can be a UV Mapper, and you can activate one by clicking on the menu at the top-left of any view, and selecting UV Mapper. But, before anything is displayed there, you have to Assign UV Values to verts. How does that work? Think about what you want to do with your verts. You want to spread them out on top of an image. So, to do this in the best possible way, we assign UV values based on the mesh's actual shape. The shape of the mesh is viewed in the normal views in Z-Modeler. Let me break it down step by step first, then I'll explain pictorially. Assigning works this way:

- Activate a UV Mapper in one view of Z-Modeler
- Select Faces which you want to map
- Switch to the view which shows the group of faces in the orientation which you want to map them (i.e. Left or Right view for polys facing the side, Top or Bottom view for polys facing up or down)
- Select the command Surfaces-->Mapping-->Assign UV.
- Hit spacebar (SEL Mode ON)
- Click anywhere in the view which you selected in step 3
- Switch to the UV Mapper view

As you can see, you assign UV Values based on their actual shape as seen from the view you assigned them from. Here's a picture which gives an example:



This allows you to map your faces to texture with little distortion of that texture. At this point, I should mention that the Surfaces-->Mapping-->Assign UV tool has one other function. If you have already assigned and mapped faces to UV Values, this tool just displays them in the UV Mapper view. In other words, previously mapped faces are not reassigned new UV's if you click on them with the Assign UV tool. They just display in the view with their current values. Now, we know how to Assign UV values, we have to know how to alter them.

Mapping Faces

Now, let us assume that you have assigned the UV Values for a section of your mesh. Z-Modeler places these faces on the texture in the same place with regard to the origin as they were in the regular view. So, if your face was, for example, 1.59648 units away from the origin in the regular view, it would be 1.59648 units away from the origin in the mapper view. So, you may have to move, rotate, and rescale your faces in the UV Mapper view to fit the particular area of the texture that you want. Remember that you were in the faces level in order to assign those UV values, and we can't move, scale or rotate faces in any view. So, switch to the objects level, and now you can move that particular object inside the UV Mapper view. Basically, you need to scale it down to fit the area of texture you've selected. Then you need to move it over that texture, and perhaps distort it slightly to fit. Next, you should consult the 3d View to make sure that the texture is where you want it on the mesh, and then you tweak accordingly. I recommend doing this on the vertex level. This gives you a higher level of control over the texture's display. Basically, your skills in moving, scaling, and rotating which you honed in making the 3d model in the first place come into play here. I've prepared a huge animation for you to enjoy. It shows me mapping that hood that I assigned in the last section. You can view it here. Notice that I changed the position of the axis for scaling purposes, that I mirrored along the Y axis, and distorted the image slightly to fit the texture. These are good mapping practices that will save you loads of time. These, and others will be discussed in our last subsection, Mapping tips.

Mapping Tips

To a point, the mechanics of mapping a mesh can be taught. However, the *art* of clever, precise mapping is something that comes with time. Some practices can help you speed this learning process up.

Scale to a Local Axis

As you could see from the large animation in the last section, I used the Period key to move the axis when I scaled my object in the Mapper view. This is very useful, because it allows you to scale any group of verts down to a specific area of the texture. It also allows you to scale accurately, since you avoid lining up the object, then rescaling it, then lining it up again. You can also use the Display-->Place Axis command to accomplish this, but it is better to use '.' instead. So, you start by lining up the texture around the area you want to map it to, then you find a good reference point to scale it to, like one corner of the area, or perhaps the center, if you can find that. Then, using Scale, you can scale the object down into the area.

Map at High Zoom

Another good technique to use is to zoom way in when mapping. This allows you to control down to the pixel where your verts are mapped. You must use high zoom in mapping to ensure accuracy. If you don't, you may wind up with little lines around the edges of your mapped areas where they overlap to the next area. You zoom in and out using the + and - keys on the numeric keypad.

Understand Stretching/Compressing

A big part of mapping is actually stretching or compressing the texture into the mesh. Because you probably distorted the shading contours of the texture when you made it, you need to counteract that effect in order to get good shading on the car. Basically put, when a small area of texture is mapped to a large area of the mesh, it's stretched, and when a large area of texture is mapped to a small area of mesh, it's compressed. Common sense tells us that compression looks better than stretching, and this is true. Stretched textures tend to look pixelated, and should be avoided. However, a combination of both techniques is essential. I think that one misconception that a lot of modelers have is that your mapping object has to look exactly like the actual object. This is not true, because changes made to the mapping object do not affect the shape of the real object. So, if you have to disproportionately scale your mapping object to fit your texture, don't be afraid to do so.

Tile Textures

When working with track textures, tiling should be your friend. Instead of using 2048x2048 textures to represent grass in a track or scene, you can use one 256x256 texture and tile it. If a texture can be repeated as a pattern with little or no obvious lines or repetition, then it can be tiled. Tiling is easy in Z-Modeler. You just have to scale it larger and larger until you have tiled it as much as you like it. However, a word of caution. This only works when the texture is the only one in the image! You cannot use this technique with an image that contains more than one texture.

Mirror/Rotate

You can use the Mirror and Rotate commands in Z-Modeler's mapping window to move the verts as well, so make good use of them. If you see that some text is mirrored, don't flip the texture, mirror the mapping object. However, a word of caution, make sure you line it back up well, and mind the modification axes you have active when you mirror. You don't want it to end up upside down as well. Rotation is another good way to move verts around. Often the orientation of the texture is different from the mesh. Therefore you have to rotate it to make it fit.

Avoid Messy Mapping

Finally, avoid messy mapping. Try and keep the mapping objects lined up with each other whenever they are adjacent. Try to also avoid mapping a whole mass of polys to a certain area with no regard for their shape, because this can cause shading difficulties. It will also make it easier when you go to tweak the mapping later.

That does it for Textures and Mapping. This is a very confusing subject for a lot of people, so if you understood everything the first time through, good for you! If you're having a problem, I suggest you play with some of the concepts discussed here with a real mesh, then come back and read it again. We first talked about textures, what they do, how to use them, and we talked about the best applications of them. Next, we delved into the nether regions of Alpha Channels. After that, we talked about Materials, 3d go-betweens for textures and meshes. Then I told you how to use them and manipulate them in Z-Modeler. Finally, we discussed mapping, the skill that, if you've made it this far, will either limit your progress or launch you to the next level. What's in that next level? For one thing, the art of Normals Theory. In my next chapter I will attempt to explain what normals do, and how we can manipulate them for our own benefit.