

In-Depth Paintshop Pro Object Creation Tutorial

The goal of this tutorial is to show you the basics of making an object using version 7 of Jasc Software's Paint Shop Pro. Paint Shop Pro is a great cost alternative to Photoshop. This tutorial is not intended to show you all that you need to know about object making. There are many different tools, and just as many techniques to making objects for The Sims. My intention is to give you a good base so that you can use this as a launch pad to making your own objects.

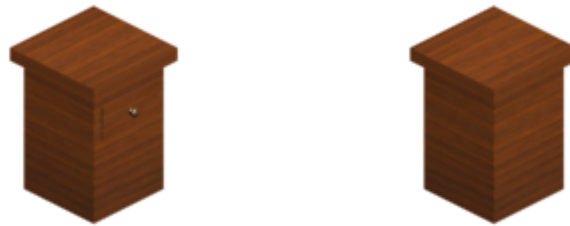
In my opinion, Paint Shop Pro is a much better program for object making than Photoshop. Why? One word. Shortcuts. There are many vital and crucial shortcuts in PSP that makes object making much less of a hassle and headache. In this tutorial, I will point out helpful shortcuts by bolding them and enclosing them in brackets. **[Like this]**.

Step 1.

First off, we'll need to "prep" our object for operation. Let us imagine for a moment, that through your amazing 3d or Paint Shop Pro skills, you were able to produce this snazzy little end table.



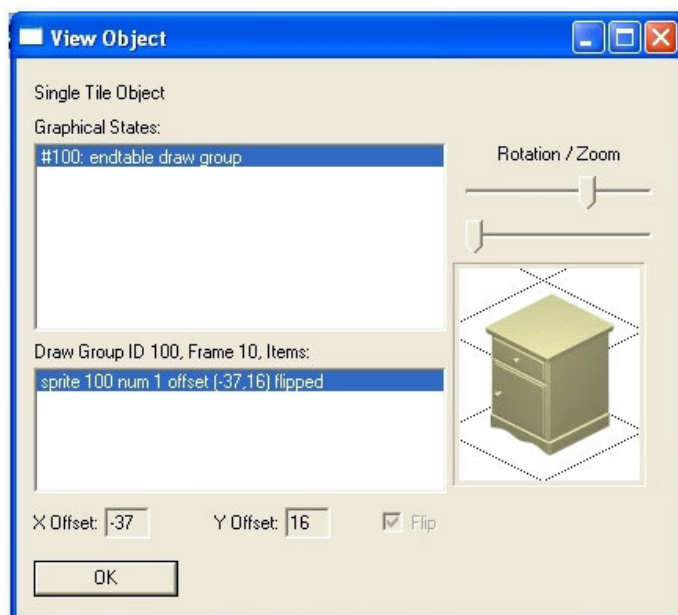
Step 1 Cont.



If you'd like to follow along with this tutorial, you can click on these pictures and save them to your harddrive. They are the front and back views of our table.

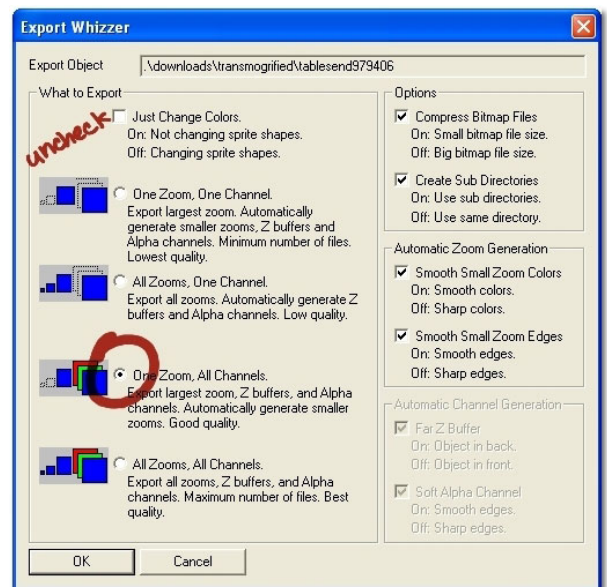
For this technique, it is important that you render your object using anti-aliasing if you're using a 3d program, or make your object anti-aliased, should you make it in PSP. The anti-aliased version of our end table will ultimately become our alpha channel. I find it is easier to paint in the jaggies than to make an alpha channel from scratch.

It's always a good idea to find a pre-existing Sims object that is similar in shape and size (and function) to base our object on. For this tutorial, we are going to use the cheap Pinegulcher end table.



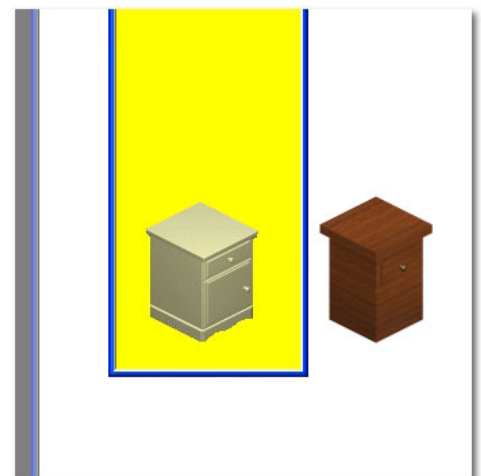
Step 2.

Now we are going to clone the end table using Tmog. In the Export Whizzer, uncheck "Just Change Colors", and choose "One Zoom, All Channels" from the menu and send it to your desired folder.



Open the "sprite 100" folder and open "endtable-
graphic_large_front_p.bmp" in PSP and place your object side-
by-side with it. You'll have to eyeball it to see whether your
object is similar height and width. You may have to resize it a bit
before you're comfortable with it.

Your object will almost always be a little off in size, so don't
stress yourself if they aren't absolutely identical. (tip: If you DO
resize it, always be aware of your picture quality. If it gets too
blurry, you may have to sharpen it up.)



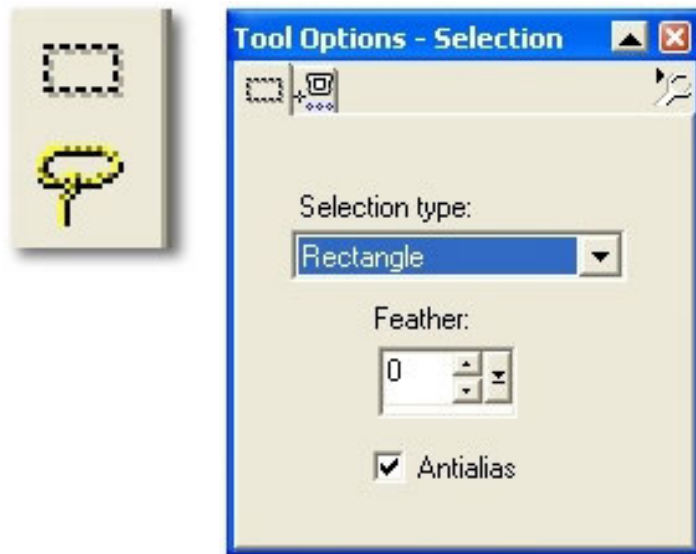
Step 3.

Before we go on, it is important to isolate our table. If you've made your object in PSP using layers, you'll need to merge any layers you may have, EXCEPT the background, and skip to the next step.

If you've rendered your object from a 3d program, or made it in PSP WITHOUT Layers (or if you messed up and used the background for part of your table/object, then you'll need to make sure your image is flattened with no other layers before proceeding), then we'll need to isolate it by selecting it out.

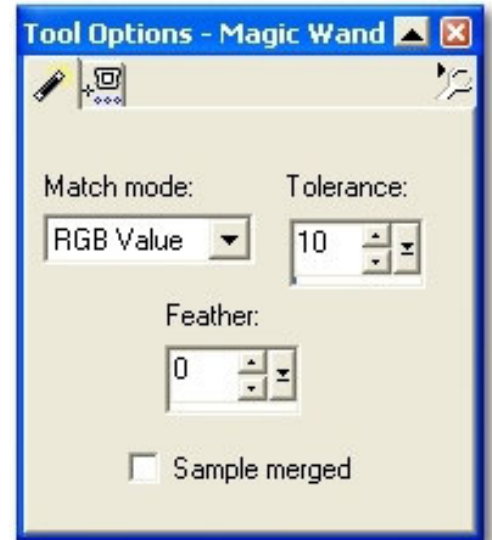
Right now, it only LOOKS like it's sitting alone, but we're really seeing is our table surrounded by a fill of white, and we don't want that.

By choosing either the "Selection" or "Freehand" tools (Pic Below), you can view the control palette for selections.



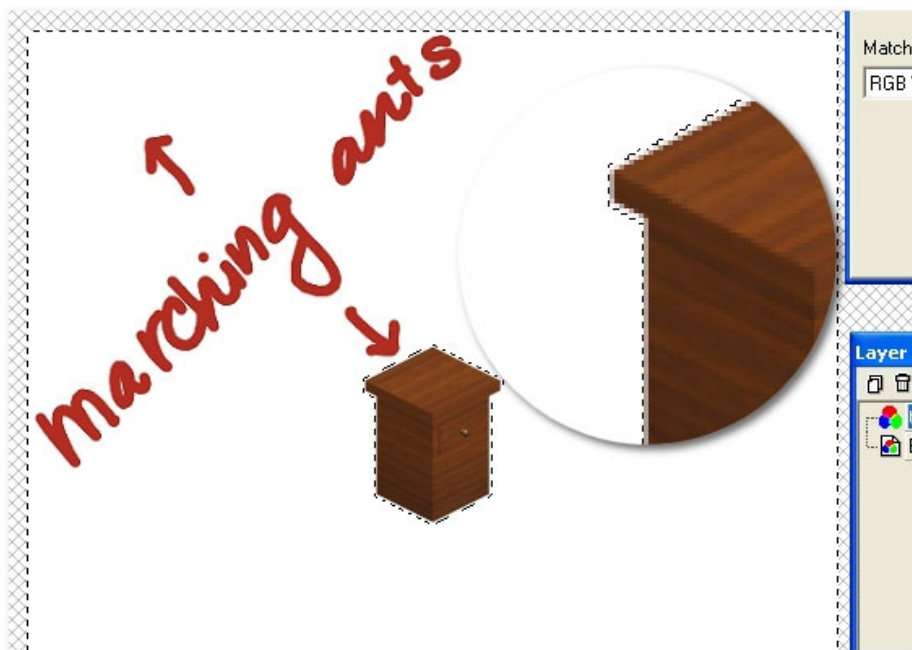
Step 4.

Check the Antialiased box and set Feather to 0. Next, select the Magic Wand tool, . Set the tolerance to 10 and set Feather to 0 on the control panel.



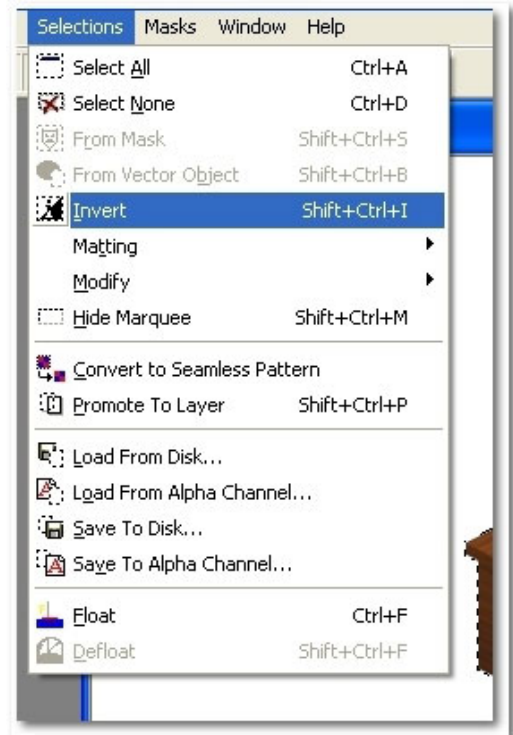
Select the white area outside the object. By doing this, the Magic Wand Selects those pixels that are PURELY white. That is, having a color value of R: 255 G: 255 B: 255. It will not pick up the pixels that blend from white to another color. In the table's case, brown.

By checking the Anti-aliased box earlier, we've also made sure that the magic wand will preserve the transparency of all pixels, so that we'll have a nice clean alpha channel later on. With the white selected out, you're table should look something like this:

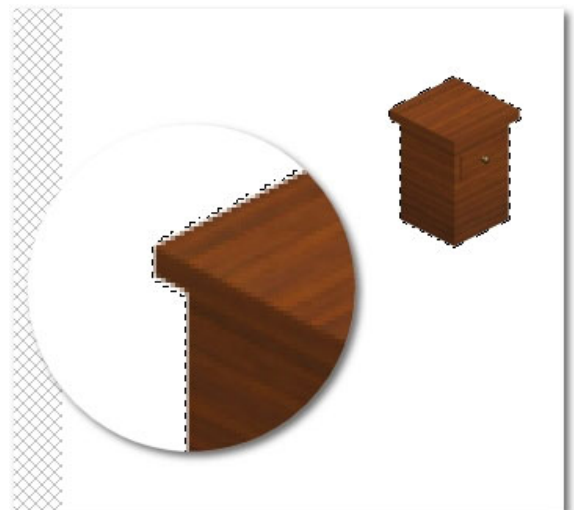


Step 5.

With the marching ants, or marquee running along the outside of the table. Now, we'll need to invert our selection, so that the table will be selected, instead of the white. In the File menu, go to Selection>Invert [**Ctrl+Shift+I**].



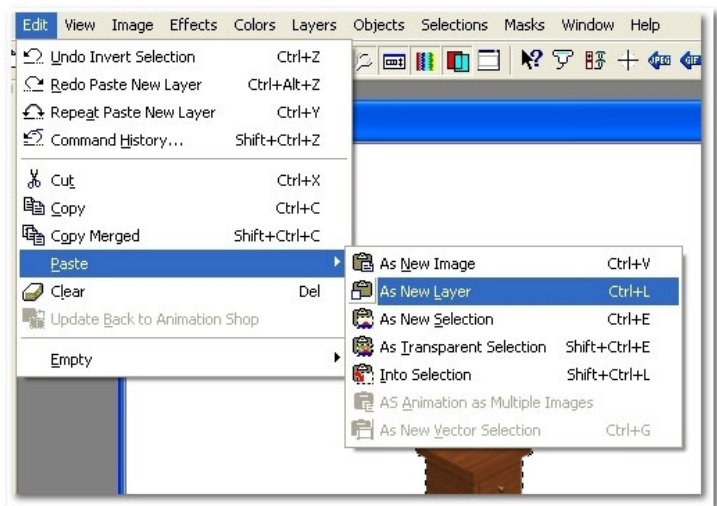
Once the selection is inverted, it will look like this:



Step 6.

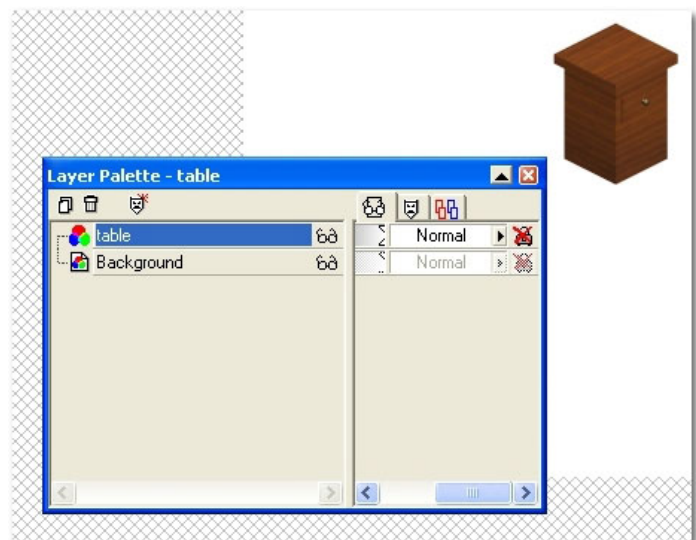
Once you've accomplished that, create a new file 500 pixels large by 500 pixels wide (you may have to make it larger depending on the size of your object) and save it as `sprite_endtablefront.psp` (it must be a .psp file!!). This file will be the workspace for our little table.

Copy your table/object **[Ctrl+C]** and paste it as a new layer in the new file `"sprite_endtablefront.psp"`. Edit>Paste>As a new Layer **[Ctrl+L]**.



Step 7.

Double click the new layer and name it "table".

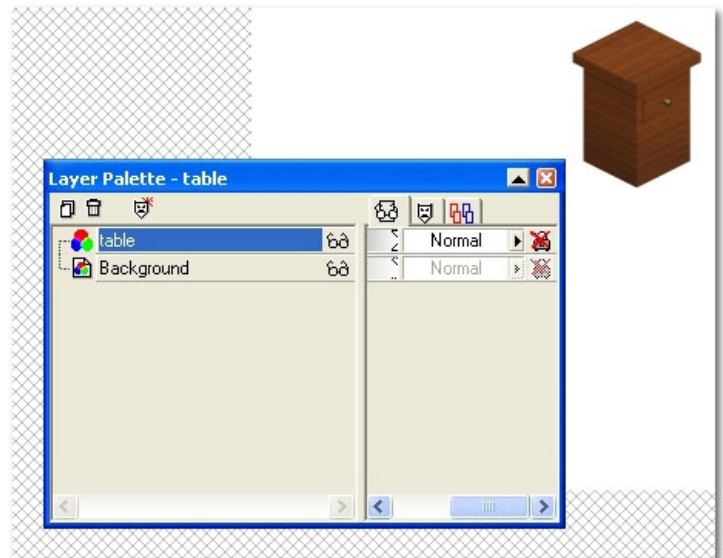


Now we're able to move onto making our P sprite and alpha channel!

Step 8.

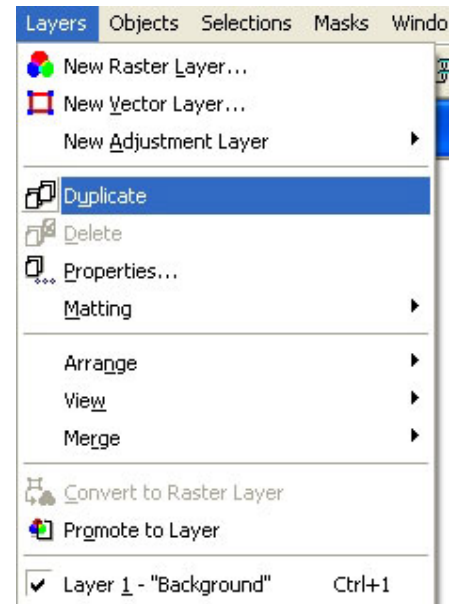
The p sprite is what contains all the color information for an object. Right now, our table still exists as a copy of the Pinegulcher table. So now, we've got fix that.

As we stand, we have a document titled "sprite_endtablefront.psp" open in PSP containing only two layers. One of those layers should be the default "Background" layer, while the other holds our precious little table.



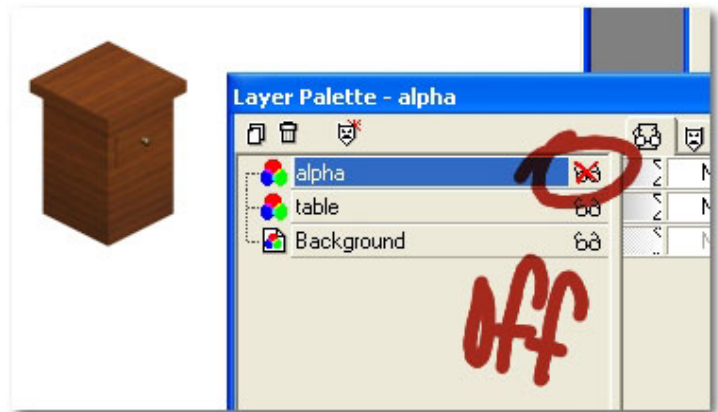
Step 9.

Duplicate the layer named "table" so that we'll now have TWO tables. In the File menu go to Layers>Duplicate.



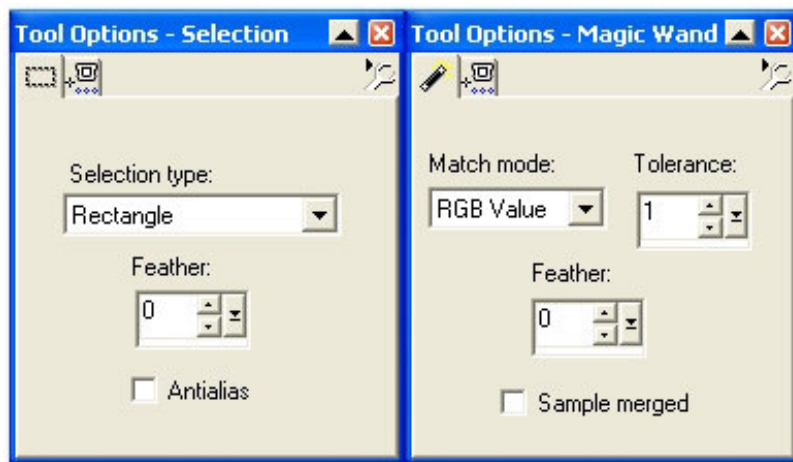
Step 9 Cont.

Rename your new table layer from “Copy of table” to “alpha” and turn off the layer.



Step 10.

Choose the selection or freehand tool and UNCHECK antialiasing in the control panel. Next, choose the magic wand tool and turn it's tolerance level down to 1.



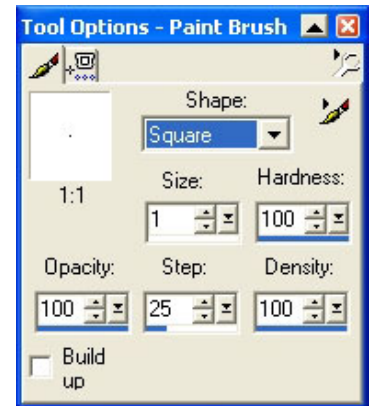
Using the magic wand, return to the “table” layer and select the area outside the table. Invert the selection so that now the table is selected. By turning off anti-aliasing and lowering the tolerance, we are telling the magic wand to select all pixels within that layer, regardless of opacity.

For our P sprite, we don't have to worry about opacity, and our handy magic wand has just made all our pixels within the selection opaque! Now all we have to do is “paint” paint away the whitish pixels surrounding our table.

Step 10 Cont.

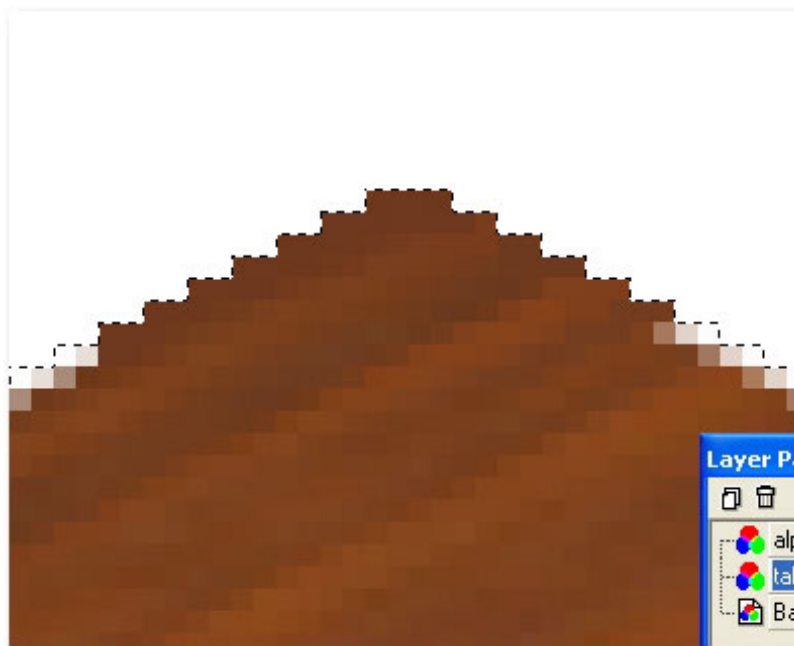
MAKE SURE YOUR MARQUEE IS ALWAYS ON DURING PAINTING. Without the marquee (or selection) the pixels retain their transparency until they have been painted over. You may end up trying to paint over pixels that still have partial transparency, and those ugly yellow bits will creep into your object when you go to put the yellow background in.

Make sure that the layer named "table" is now selected, and choose the paintbrush tool from the tool palette. You'll need to put the brush to these settings:



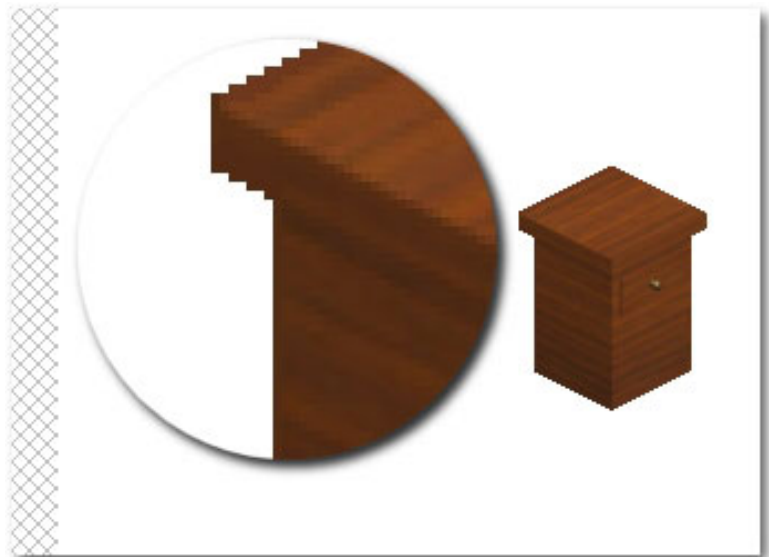
Step 11.

With our selection on, we can begin painting in our pixels. By holding down the Ctrl key, you can convert the paintbrush to an eyedropper and sample colors in the same area. This way, we can color according to the grain of the wood.



Step 11 Cont.

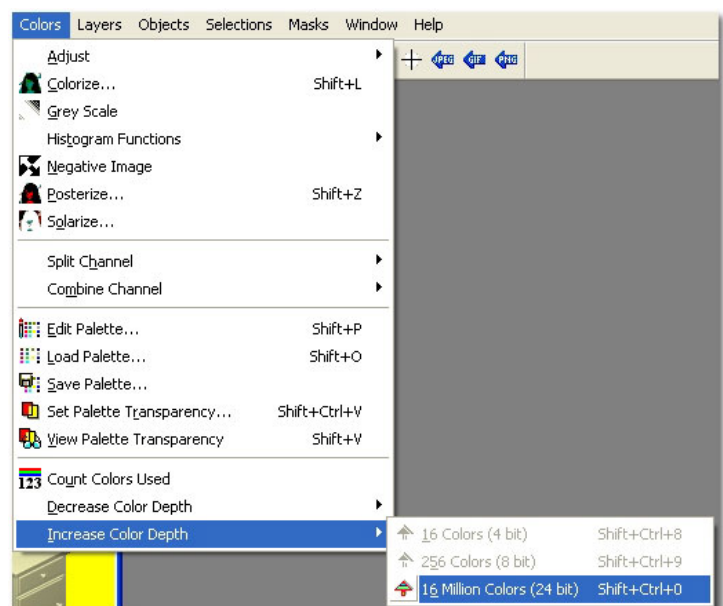
To make things easier, you may also increase your brush width to 2 or 3 to cover a greater area. Zoom in and out as need to complete your painting. Once you've finished painting your table, it should look something like this:



Go to the folder containing the Pinegulcher table. Open the folder named "sprite 100" and open "endtable-graphic_large_front_p.bmp" in PSP. Now we're ready to replace the maxis table with our very own.

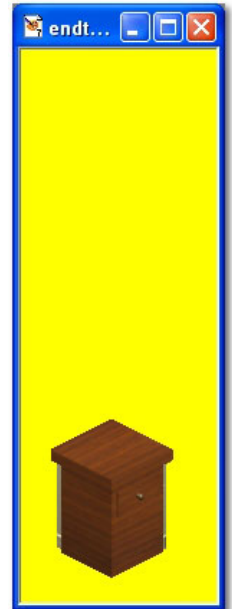
Step 12.

The first thing to do is to turn the maxis p sprite into a 24 bit color image. As it is, it's only 256 color, or 8 bit. To do this, go to the Color menu>Increase color depth>16 million colors **[Ctrl+Shift+0]**.



Step 12 Cont.

Copy the pixelated table from the “sprite_endtablefront.psp” file and paste it as a new layer **[Ctrl+L]** in the “endtable-graphic_large_front_p.bmp” file. Move the table around a bit so that it is roughly in the same area as the original table.



Step 13.

Now we’re going to get rid of that ugly yellow table behind our image. Right click the background color in the color palette. A dialog box titled “Recent Colors” should pop up. The bottom two rows hold a record of the last ten colors you used (they should all be brown from coloring in that table! =^P).

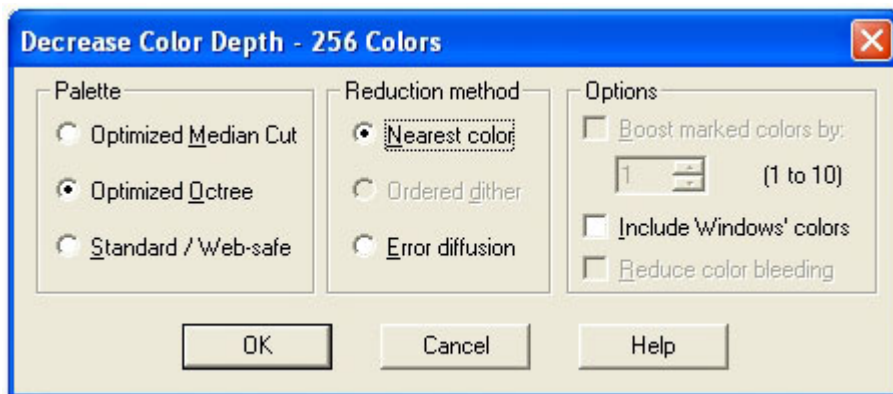


Step 13 Cont.

But what we're interested in is the top rows. They contain several stock, or base colors. If you'll notice, one of the boxes in that top row contains that horrendous mustard yellow color found in our p sprite. Luckily, Mr. Hopkins chose a color so nauseating that no one would ever dream of using it in their object. The color for background of a P sprite is R: 255 G: 255 B: 0. Basically, it's the yellowest, yellow possible on your screen. Choose the nauseating yellow as your new background color.

Select the layer titled Background. This is what holds the maxis table. We don't need it so just hit delete!

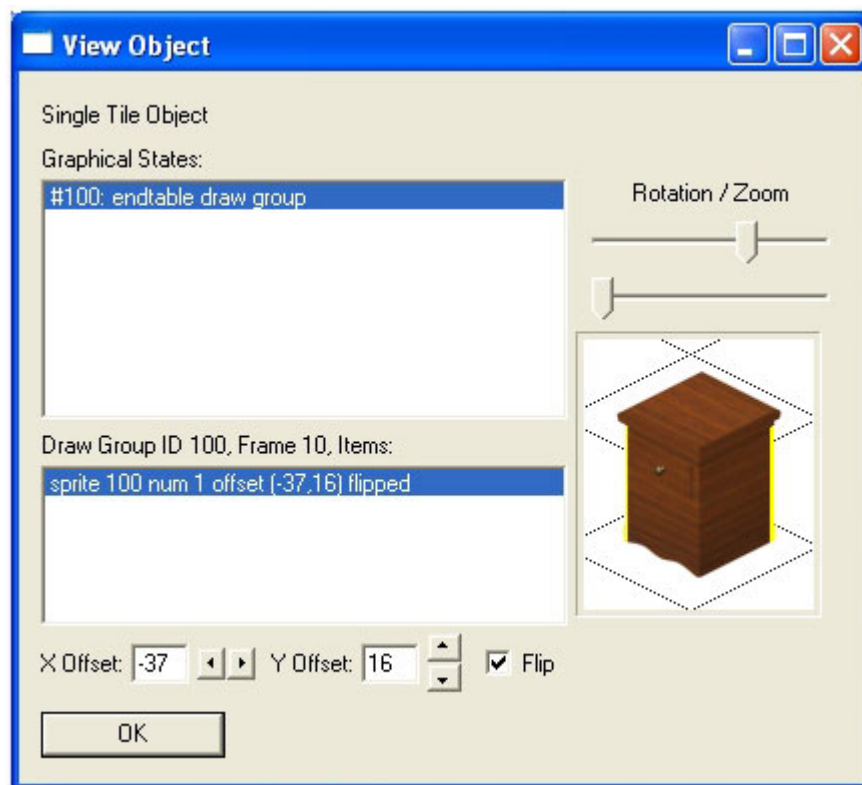
Now our snazzy little table has replaced that old ugly pinegulcher table. Decrease the color value to 256 color. Colors>Decrease color depth>256 color **[Ctrl+Shift+3]**. Use these settings:



Step 14.

Voila! We now have a snazzy little table as our p sprite! Save **[Ctrl+S]** the new p sprite and import using Tmog.

Whoops! What's this?



For some reason, our table is there, but it ain't shaped like our table! Well, that's where the alphas and z buffers come in....

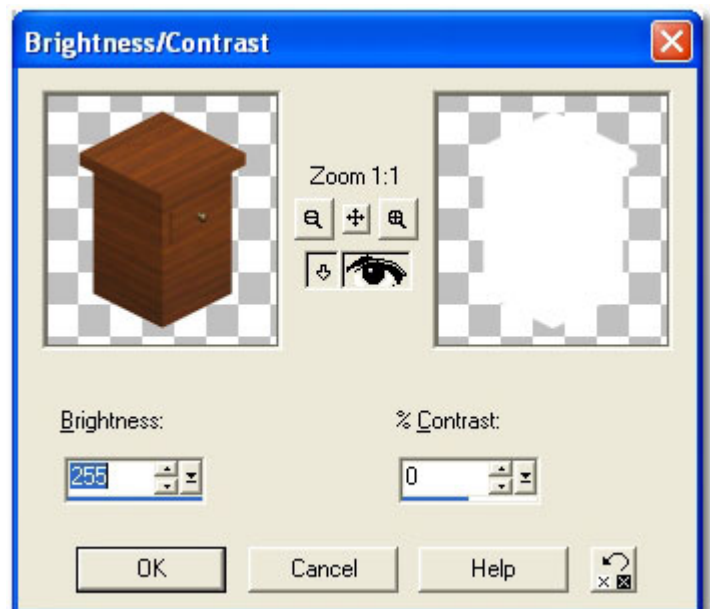
Now to conquer those alphas!

Step 15.

The alpha channel is what contains the blending information for a sims object. And lucky for us, we've already got a nice alpha whiting--I mean, waiting for us. Go back to your "sprite_endtablefront.psp" file, and click the layer you named "alpha". Since an alpha channel is nothing more than a white image blending into a black background, all we have to do is turn this layer's brightness WAY up, until it's completely white. But rather than working on this layer, we're going to copy it. Duplicate the layer named alpha, so that you now have a layer named "Copy of alpha". This will be our actual alpha channel. Why? Because, if you think about it, the original alpha layer is the only anti-aliased, isolated copy we have of our table. So, we'll work on the copy in case we ever need a copy of our original table for any reason.

Make sure that "Copy of alpha" is selected and go to Adjust> Brightness/Contrast in the Colors menu **[Shift+B]**.

This will open a dialog box. Set Contrast to 0, and Brightness to 255. You should now have a completely white copy of the table.



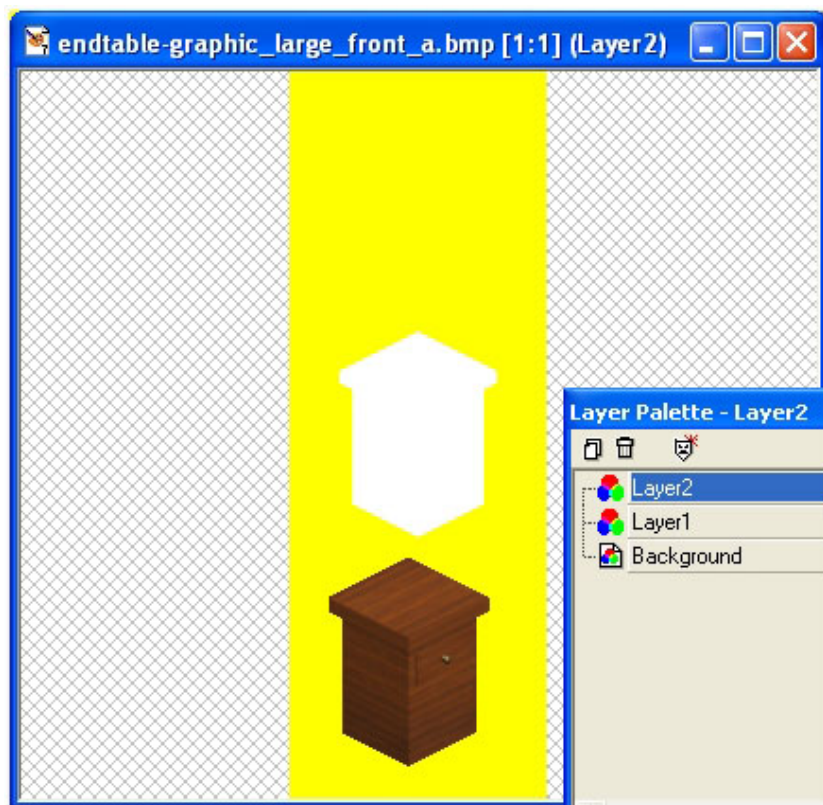
Step 16.

Return to the folder containing the Tmog files for your table and open the "endtable-graphic_large_front_a.bmp" from the "sprite 100" folder. If "endtable-graphic_large_front_p.bmp" is not currently open in PSP, open that one too. The placement of our alpha-table, depends on how we placed our table in the p sprite. If an alpha channel does not line up properly with it's corresponding p sprite, it will not show correctly in Tmog.

If you look at the color palette for the "endtable-graphic_large_front_a.bmp" file, you'll notice that the rainbow of colors has been replaced by a gradient of grey values going from black at the very top, to white at the very bottom. That's because an alpha is read as a **greyscale** image in PSP. For now, we'll need it to be a 24-bit color image.

Using the Color menu, or **[Ctrl+Shift+0]** shortcut, turn "endtable-graphic_large_front_a.bmp" into a 24-bit color image. Select the p sprite for our table and hit **[Ctrl+C]** to copy it. Paste it as a new layer into the "endtable-graphic_large_front_a.bmp" file.

Now go back to "sprite_endtablefront.psp" and select the layer titled "Copy of alpha". Copy that layer and paste it into the "endtable-graphic_large_front_a.bmp" file as a new layer. Your results should look something like this:



Step 17.

Now we can dispose of the layer containing the p sprite. You can do this either by hitting the delete button while the layer is selected, or clicking the trashcan icon in the "Layers" palette.

Select the color black as your background (you can do this manually, or by using the 'Recent Colors' palette), and choose the Background layer of "endtable-graphic_large_front_a.bmp". Hit delete to clear the Pinegulcher alpha away. Return the alpha file to a greyscale image by choosing "Greyscale" in the Colors menu. The last step is to **flatten** the alpha channel by choosing Layers>Merge>Merge All (Flatten).

Save, and now you've got a completed alpha! No sweat right? All that's left is, Those dreaded z buffers!



Step 18.

Z buffer stink. There's no two ways about it.

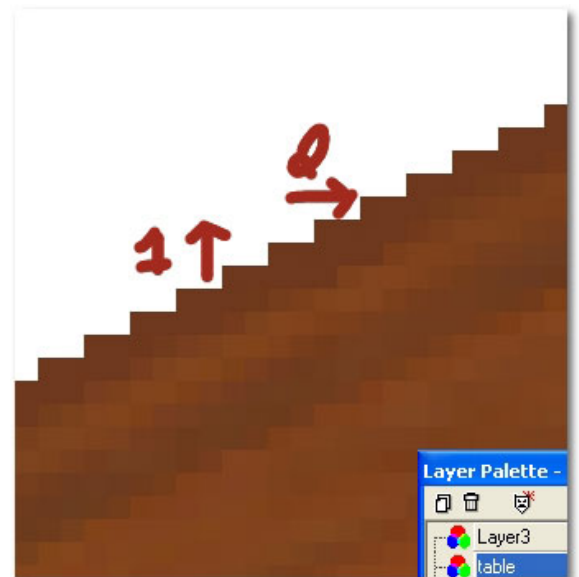
Actually, they aren't so bad once you've gotten the hang of them. Trust me on this one. Luckily for us, we have an original maxis object to base our endtable's z-buffers on. So basically, we're going to steal them.

Open the file "endtable-graphic_large_front_z.bmp". Using the same magic wand settings we used to select our table to paint in the jaggies, select the z-buffer from Pinegulcher file and paste it into your workspace ("sprite_endtablefront.psp" remember?). We'll make our z buffer in the workspace.

Making your own Z-buffers can be a daunting process depending on the shape and size of the object. Every object presents it's own challenges. A good resource for beginning is the [Simfreaks](#) z-buffer tutorial. It gives some valuable information (especially for our buddy the endtable).


First we'll need to excise the top of the end table from the base. We'll use the "Freehand" selection tool. Before we continue, it's important that you understand something about the pixel representation of sims objects.

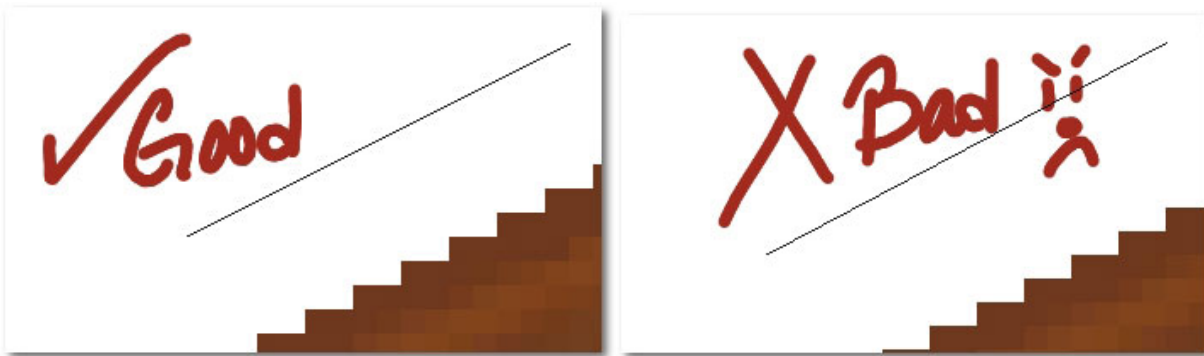
Imagine in your head, if our endtable was real. The top would have straight edges rather than diagonal ones. In the Sims, straight lines for height are represented correctly, as straight lines from top to bottom. But straight lines lengthwise or widthwise are represented as a staircase with a rise of 1 pixel and a run of two pixels.



Step 19.

When we remove the top of the Pinegulcher z-buffer, we'll need to make sure that OUR selections also follow this pattern. The way to do this is, again, to eyeball it. By using the "Freehand"

selection tool  draw out a line until it looks smooth like the "good" example.



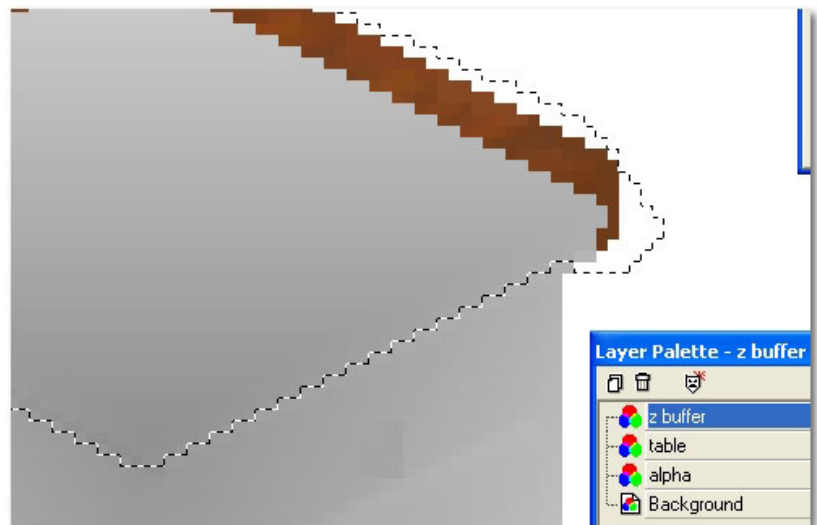
Step 20.

Notice that the good example isn't *completely* straight like a 45 degree selection, but it *is* completely smooth and even. The bad example has jumps in its line and is therefore unacceptable and unusable. Practice it a few times until you can make a perfectly stairsteped selection.

Now back to our table.

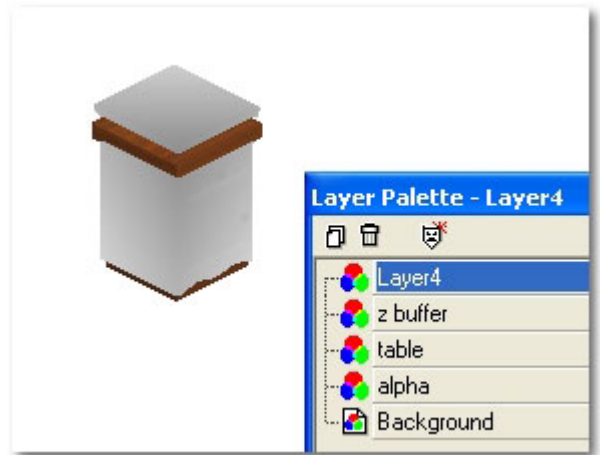
Using the new technique, use the "Freehand" tool to select the surface of endtable. Make sure that the anti-aliasing box is unchecked in the control panel.

When selecting, be sure to pay attention to color variations between the base of the endtable and the top. When finished, it should look more or less something like this:




Step 21.

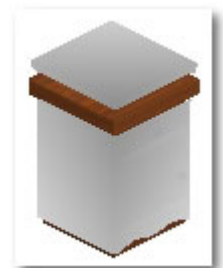
Cut the selected top **[Ctrl+X]** away from the z-buffer and paste it to it's own layer within "sprite_endtablefront.psp" file. Deselect your selection. Now it should look like this:




Step 22.

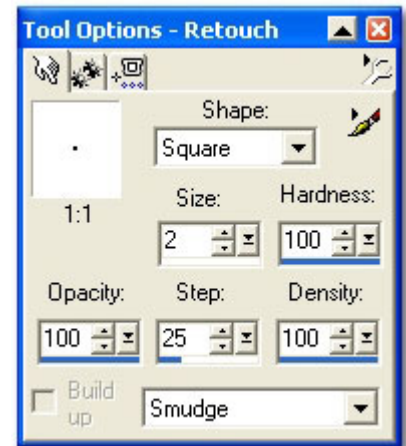
First we'll create the base of the table since that will be easiest. We're going to use a smoothing technique to make the base z-buffer of the pinegulcher table fit our endtable.

Using the "Move" tool , move the base of the pinegulcher z-base so the topmost edges fit against to top edges of our endtable base. Be sure to line up their corners as well. It should look something like this:

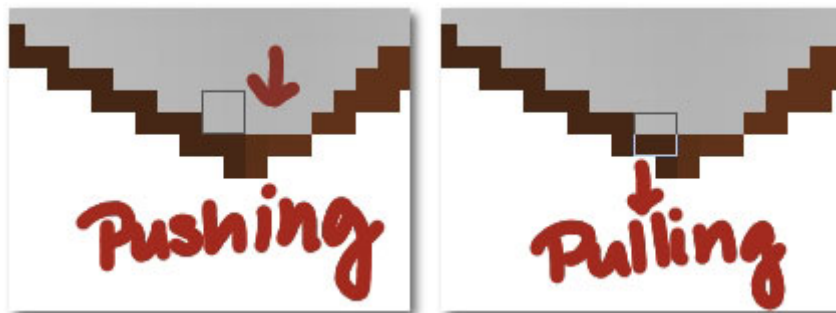


Step 23.

With the "Retouch" tool , we're going to pull the pixels down so that they cover the bits of our table that are sticking out beneath. Set the "Retouch" tool to these settings in the control panel:

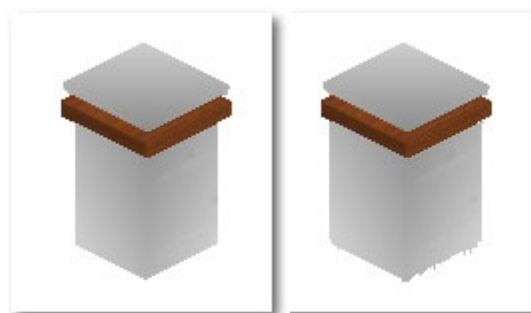


What we will do is *pull* the pixels down, not *push* them down.



Step 24.

Pull the pixels down so that all the bottom edges of the end table are covered. Once you've finished, reselct the layer named "table". Using the magic wand at it's current settings, select outside the table. Reactivate the layer where the base z-buffer of our table is. Click delete and that will clean up any pixels outside of the table (don't worry about deleting away the top since it's on its own layer ;^)).

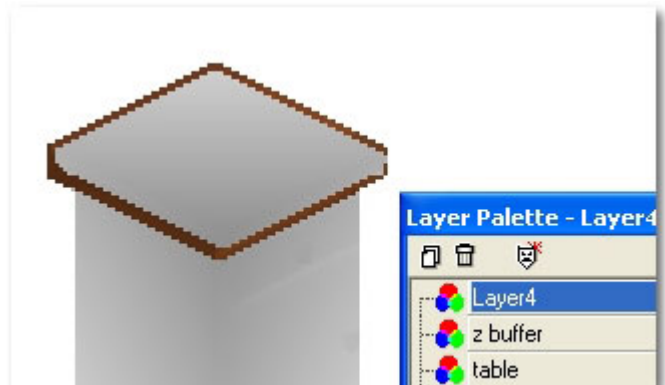


Step 25.

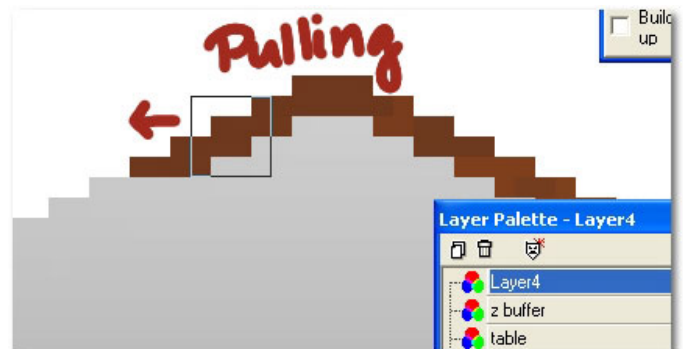
The top of the table will be a bit more complicated. We were lucky that the base of the Pinegulcher table was slightly larger than the base of our endtable. Not so with the top. The top is a bit larger than the Pinegulcher table, so two things will be different this time. First, we'll have to center the z-buffer top in the top of our endtable. Second, we'll be smoothing things differently, because we are working on a surface.

Surfaces are important in the Sims games because they are where simmies place things like lamps, vases, and food. They have to be dark enough to be placed next to things like couches and curtains, but light enough so that we can see the things placed ontop of them. Quite the connundrum.

To finish the top, move the seperated top of the z-buffer into the middle of our endtable's top. Sort of like this:



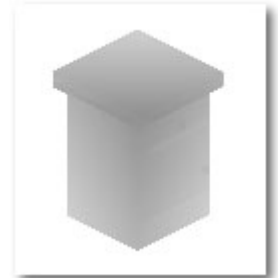
We'll be smoothing from *side-to-side* rather than down for our surface. However, you will need to smooth upward and downward for those areas where side-to-side smoothing doesn't work.



Step 26.

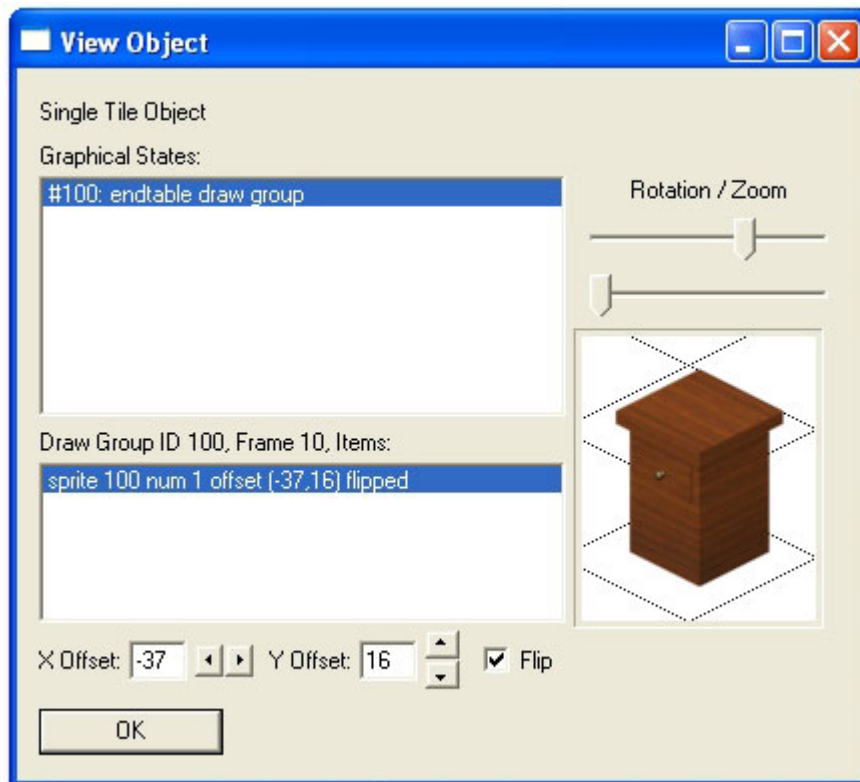
Once you've finished smoothing, clean up any wayward pixels using the same method that we used to clean away the base pixels.

Now you should have something similar to this:



Finally, merge the two z-buffer layers by turning off every layer except the two containing your base and top z-buffers (the Background layer as well). Go to Layers>Merge>Merge visible (If merge visible is greyed out, that means that you have a turned-off layer selected in your "Layers" palette). Rename the layer entitled "Merged" to "z buffer." All that's left is to paste your new z-buffer into the "endtable-graphic_large_front_z.bmp" file the same way we did the alpha channel. Make sure that your z sprite is lined up properly with the p sprite.

Now you can import into Tmog, and there you have it! A new endtable.



Step 27.

Fortunately, the back view is the exact same shape and size as the front view, so we can just use the front view's alphas and z-buffers for the back, while using the back view for the p sprite.

Play with the x, y coordinates in Tmog to get your table placement just where you want it. Start up your game to check whether everything is working fine. At this point, the only things that should be giving you problems are the x, y positioning or z-buffers. Test it against lamps and phones to make sure the table's placement is correct. Once you're sure about placement, THEN you can work on tweaking any z-buffer issues. There is a great tutorial on livin-it-up.net to help you out with that.



Good luck, and happy object making!