Nova Design - Main
Tutorials - Main
Aladdin's Lamp - Main

Aladdin 4D Support Newsletter · 4th Quarter, 1991 · Issue #1

(Originally "4DPro Master")

The following Newsletters were originally published by Adspec Programming and are the property of Nova Design, Inc.

Logo illustration by Kermit Woodall with Lamp modeled by Bob Lanham
Data entry and revised by Bob Lanham
Images and Screen grabs by Bob Lanham
Project files converted to Aladdin4D ver. 5 by Bob Lanham
HTML coded by Wil Haslup

Contents

Welcome: Premier Issue!

by Greg Gorby

Welcome to Aladdin's Lamp! Since this is the premier issue, I'd like to take a little space and talk about the goals and contents of the newsletter. The goals are simple. Aladdin's Lamp has only one purpose, to help you be more creative and productive with Aladdin 4D.

All information, tutorials and articles presented in the newsletter will be directed toward this end. We will be illustrating both simple and advanced techniques of animation, texture mapping, rendering, editor tools, and general program use. There will usually be accompanying drawings on the companion disk.

Who writes the newsletter? The bulk of the writing at first will be by myself. I am the author of both the program and the manual, and the logical person to begin the support program. However, I would like to invite your input! My knowledge and experience with the program is important, but it also makes it difficult for me to know those questions that a new user wants to know! If you have a question, no matter how simple, please let me know.

We will consider publishing any articles that you submit if we agree they are of use to others, and space permits. Other things we would like to include are any drawing files you would like to share. Drawings that illustrate animation techniques are great, as a re unusual or difficult to draw objects. And of course, those users that are doing video titling with the software are always interested in new fonts, so if you would like to share we would love to include it! Any contributions that you make will be greatly appreciated by us and by everyone using the software. We can't reimburse for contributions, but you will be given credit in the newsletter, and who doesn't like to see their name in print!

We would also like to hear from those of you who are selling the output of the software in some way. We are sure that all users would be interested in this! Let me know how you approach your clients, the type of client, the type of work you sell and the equipment

necessary to complete the work, the area of the country that you sell in, and the approximate amount you charge for the work. Remember, any work or information you submit will be printed in the newsletter, so don't send it if you don't want others to know.

When you send articles, please send a signed cover letter stating that what you submit is your original work and giving us permission to publish it. We cannot publish it unless we have this permission.

Thank you for using Aladdin 4D,

Greg Gorby,

Adspec Programming

Contents

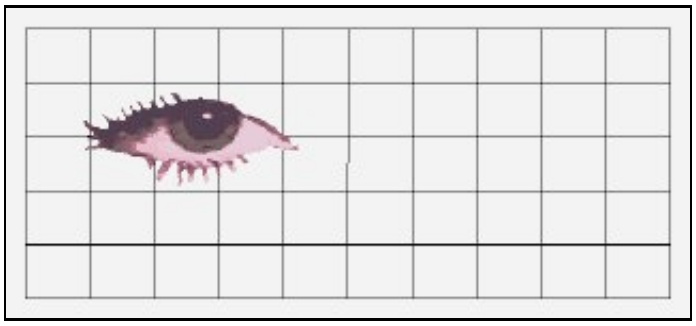
Precise Positioning of Bitmaps in Aladdin 4D Using Deluxe Paint

by Greg Gorby

(NOTE: In Aladdin's editor settings, set the units of measure to "units" before beginning this tutorial.)

The ability of Aladdin 4D to easily decal bitmaps along with its very free, independent poly interface opens up some new possibilities in bitmap application. You can selectively apply a small bitmap to only a few polys out of an object, and further, custom fit the bitmap to these polys! In this article I am going to take you through an example to illustrate the steps you would take to do this. For the example, we are going to map some simple images to several places on a simple gridded plane. We are going to precisely place the images using both Aladdin 4D and Deluxe Paint (Personal Paint won't work for some of these steps).

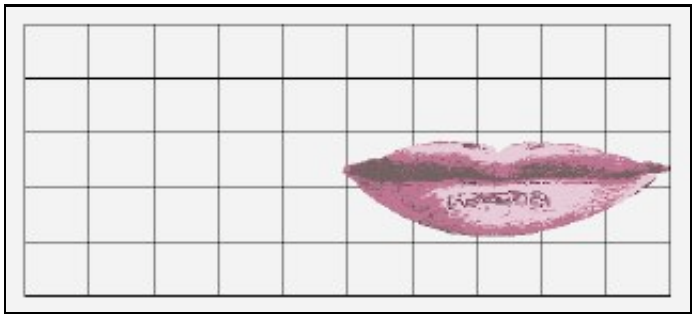
Start up Aladdin 4D. Use the "make rectangle" tool to make a rectangular grid with an x,y,z of 10000,0,5000 and x and z segments of 10 and 5. Make sure you have the grid option on. Flat view in the y axis. Zoom in until the grid is large, but not outside the editor screen. Go to the render settings and make sure that only the permission for "light" is checked, and set the screen mode for 640x400, 16 color (DBLNTSC: High Res No Flicker works). Render it, and you will see the grid as a wireframe. Save this picture as an IFF ILBM. It is only a temporary picture, so I usually save it to RAM:. Start Deluxe Paint and load the picture you just saved from Aladdin 4D. Now jump to the spare page of DPaint by pressing the j key. Load the picture called CustFit.pic. This picture has some images on it that we are going to put on the grid. Press the b key and pick up the eye image as a brush. Now press the j key. Line up the eye with the grid in the position shown. IMPORTANT: both pictures must have the same pixel resolution.



Now without moving the mouse, press the j key to jump back to the other picture, and put the image down. Have you got the idea yet? The grid is an exact map of the position of the polys in 3D space, and we are using this map to precisely position the eye on the polys we want it on. When Aladdin 4D maps the image (in the projection mode) it automatically scales the image to the polys selected, so we can select only a few of the polys, and make sure the image fits them perfectly! Let's do this.

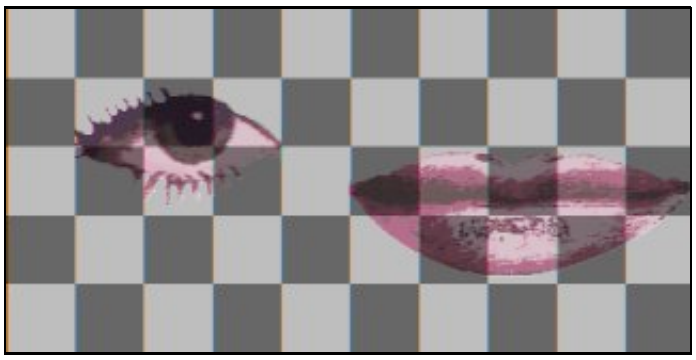
Press the j key to return to the grid. Press the b key to get ready to pick up a brush. Line up the reference lines with the top left of the polys we want to map the image to. Press the left button, and drag the cursor to the right bottom of the last poly we want to map the image to, but DON'T LET GO YET. Now while holding the left button, press the j key. The other picture comes to the front. Let go of the button. Guess what? The eye has been captured as a brush that will exactly fit the polys we want to map to! Save the brush as eye.bru. When we go back to Aladdin 4D, we can load this brush as a texture, select only these same polys, and map it. It will be a perfect fit.

While we are still here in the paint program, let's do the mouth. Pick up the mouth as a brush. Jump (j) to the grid picture, and position it as shown.



Jump back to the image picture, and you will notice that the brush will overlap the other images if we just put it down. No problem; hit the clear button to erase the picture. Now jump back to the grid, position the brush, jump back to the now clear screen, and put it down. Jump back to the grid, get the brush tool, and line up the reference lines with the top left of the polys we want to map the image to. Press the left button, and drag the cursor to the right bottom of the last poly we want to map the image to, but DON'T LET GO YET. Now while holding the left button, press the j key. The other picture comes to the front. Let go of the button. You now have the mouth as a brush that exactly fits these polys. Save it to the same directory as you saved the eye. We are done with Deluxe Paint, so quit the program and return to Aladdin. You should still have the same project loaded in Aladdin.

Using the multiple select tool, select the part of the gridded plane where the eye will be mapped, and hit "t" for texture. Now load the eye texture brush you just saved from DPaint. Set the texture to projection on the y axis, and decal mode. Accept it. Do the same for the mouth brush. Phong shade the grid-plane, because it is required for decal texturing. Now for a quick checkerboard effect, select the upper corner of the grid. You will see alternating squares on the grid select. Hit "a" for attributes, and change their reflectivity to an RGB of 100,100,100. OK the attributes. Go to the render settings, and set your permissions to solid, lights, phong and textures. Render it. There they are, just like we anticipated, in the exact position we wanted.



This method only works with the projection textures. Keep in mind that the brush you cut out in DPaint should always be rectangular. If the polygons you are decaling are irregular, just grab a brush that covers them exactly horizontally and vertically. Also note that if the polygons are at an angle other than flat in one of the main planes, you must rotate them so that they are flat before texture mapping them, then rotate them back into the desired position. There are many benefits to using this method of bitmap positioning. First, the screen ratio remains identical to that of the original brush, eliminating any possibility of vertical or horizontal stretching of the image to fit the polys. Second, you can effectively increase the "resolution" of the image by zooming in only the area that is to be mapped before saving out the wireframe. This then allows you to use a larger brush for the map in DPaint, which will in general improve the image, especially for text. You can also resize the wireframe in DPaint to fit the brush! As long as the aspect ratio of the resized wireframe is the same as the original, the fit will be good. And finally, it is always fun to actually paint on the wireframe in DPaint, filling in squares and embellishing with decorations, then capturing the finished work and using it. It is almost as though you can paint directly on the object in 3D space. This can be especially creative to do a title! Try writing a simple word, saving the wireframe, and drawing a little snake within each letter. If your artistic skills are good, you can imitate Middle Ages illumination of manuscripts in this way for some unusual and really spectacular titling. Or you can make a title relate to a specific business, like plumbing. Also note that you can "poke holes" precisely in polygons by drawing in color 0, with some other color as the background, and using the genlock attribute on the texture! Be creative, and if you come up with something you are proud of, don't forget to share it with us.

Contents

Building a Starship

by Devon A. Graham

Hello, I would like to take this time to introduce myself. My name is Devon Graham. I have owned an Amiga computer for almost seven years. I started with an Amiga 1000 and now own a 3000 with about 200 MB of hard drive space. I first delved into 3-D drawing on the Amiga with a program called Turbo Silver 2.0, then upgraded to 3.0SV and finally to Imagine. Those of you who have done this also know how much money I wasted. I met Greg Gorby about four years ago at a computer store where I work. I can remember him coming down to the store one day with a little program that he ran on our Amiga 500, It drew a 3-D house on the screen and he could rotate it to any angle by using the numeric keypad. Little did I know that I was probably looking at the very first version of Aladdin 4D. I was amazed at what I saw and Aladdin continues to amaze me each time I use it. But enough of the hype, you know what a good program Aladdin is!

From those first days of my 3-D drawing career, I have always wanted to design the Starship Enterprise and be able to move it around making my own animations with it. Greg will be able to tell you that it is hard to know which I love more: Star Trek or the Amiga. I have tried and tried but it wasn't until I got my first beta version of Draw4D-Pro (early name of Aladdin 4D) that I could really start on my cosmic venture. In creating my Enterprise, I have had to develop some techniques in reproducing objects in Draw4D-Pro and getting them to look like their "real world" counterparts. It is fairly easy to draw an imaginary object because no one has ever seen it before. However, I think you can all agree that when you try to replicate something as well known as the Starship Enterprise, everyone you show your work to will know exactly how it should really look. With this in mind I had a few obstacles to overcome in order to effectively reproduce the ship.

Even though we are looking to reproduce a 3-D object, we must think in terms of two dimensions and rely on the features of Aladdin 4D (extrudes, lathes, etc.) to give us the third dimension. That way we only have to think in two dimensions when we draw it on the screen. With this in mind I took a good look at the ship. Planning my angle of attack to figure out which parts should be extruded and which parts should be lathed. Being the devoted Trekkie that I am, I had complete blueprints of the ship including front, top, and side views. From these I could tell the exact dimensions for the ship. The only problem was how to convert them to an Aladdin format. For this I decided to use a coordinate system. I made a graph on transparency film which I could lay over my blueprints and then proceed to take measurements and come up with points to enter into Aladdin. I made the transparency by using Pagestream and laying out dotted lines every eighth of an inch on the page. Every fifth line was solid so it would be easier to count them. I then printed this using my laser printer to a transparency. The same approach could be done by taking a piece of graph paper and going to a printer and asking them to copy it onto a transparency for you. From here the only task is to overlay the transparency on the blueprints and write down all of the (x,y) coordinates that need to be plotted. For example, to do the saucer section of the Enterprise, I lines up the x-axis of my graph to cut the saucer in half: top and bottom. I lined up the y-axis to bisect the saucer left and right passing through the bridge on top and the navigational dome on the bottom. From here I wrote down all of the coordinates for the

outline if the saucer section. I called each line on my graph a distance of 5 units which would mean that I could estimate down to the nearest 1 unit to come up with the coordinate of a point. For instance, if a point on the saucer was at coordinates (23,31), I would then scale these coordinates by multiplying them by 100 and enter the corresponding point into Aladdin 4D as (2300,3100). By changing the factor of 100, we can size the object to any size we need in Aladdin. The only thing you have to remember is that once you pick a number to scale by, you cannot change it, otherwise you will lose the proportion of the object you want to create. Please note that even though I called my points (x,y) from my graph, they were actually entered into Aladdin as (x,z) so that my saucer outline would spin around the z axis when lathed. The next problem you will encounter is to decide how many points to use when entering the outline. This can only be decided by you and how detailed you want to make your object. The only thing that you need to remember is: The more polygons, the longer the rendering time. But as you will see next, you can compensate for this by using textures effectively.

Greg is to be congratulated on the degree of control you have in the placement of textures in Aladdin. It is fantastic! I can think of no other program on the market that even comes close. You can completely and accurately control the placement of pictures and details on individual polygons. Because of this, most of the minute details of an object can be entered in this manner. For instance, on the Enterprise we have the docking ports that are shown throughout the ship. You may remember in the first movie, when Spock pilots his shuttle to the docking port located behind the bridge. Well it would be very difficult to build these types of details in 3-D. With the flexibility of the textures that Aladdin offers, you can put a picture of the feature, in this case the docking port, on the respective polygons and thereby giving the final look that you want. In my case I just went into Deluxe Paint III and hand drew a bitmap image that looked like a docking port. To give it dimension, I used a light grey color to resemble the recessed parts in the bitmap. It gives a realistic look without the overhead of the greater number of polygons that would normally be used to create these details. It was this flexibility of the textures which made me bug Greg enough to put in the decal mode of Aladdin textures. Just like on a plastic model, you can add the realism of details by creating "decals" and apply them to their respective polygons.

Well, I hope that my insights into the Enterprise will help you when you're creating objects in Aladdin 4D.

Contents

The Hard Object

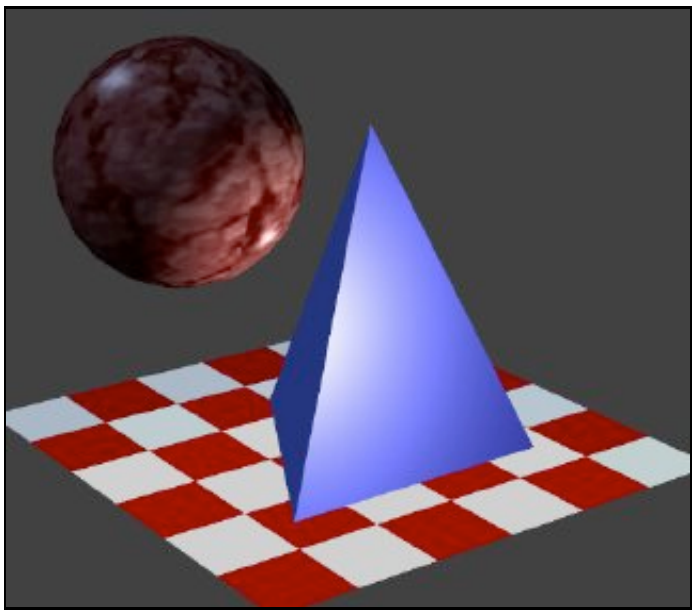
by Greg Gorby

Believe it or not, I've received a few calls telling me "I can't get my object hard!" After a little counseling, the problem is always solved. In this article we're going to examine what controls the "hardness" of objects in Aladdin 4D, and in the process, you will find out how

to create those nice hard looking objects easily and quickly.

Think about the texture of a rounded stone that you might find in a stream. When you pull the wet stone from the stream, it looks like a highly polished, very hard object, but when it dries, it just looks like a round stone. The actual hardness of the stone hasn't changed, but the visual clues that tell your mind that it is hard have. What clues are these? Basically, there are two. First, the stone is a darker color when wet. Second, there are bright specular highlights on the surface of the wet stone. You may think the most important of these is the highlights, and you would be right, but the highlights are enhanced by the darker color of the stone! Both are important. The more contrast between the stone and the highlights, the harder the stone looks.

Now think about the image you want to create on the monitor. The brightness that the phosphors of the monitor can reach, unlike reality, have a fixed limit. The brightest pixels on the screen can only be "white", never brighter! If these white pixels are to represent the highlights of your hard object,then the object itself must be darker than this, or the highlights will not be visible. So to attain a hard object, you can simply decrease those factors that affect the color so that it will be darkened, and increase those that affect the highlight, so that it will be brighter than the base color of the object.



The factors affecting the base color are the reflectivity of the object, its color (rgb) settings, and the amount and angle of the light striking it. The factors affecting the highlight are the reflectivity of the object, the color of the light, and the hardness of the object. (Notice that the object must be Phong shaded, or the highlight is not calculated.) The base color of the object is set in the Edit Attributes window, and is shown in a small color preview box. You should darken the object's color by changing the RGB sliders, not by lowering the reflectivity slider. A general setting for the RGB might be 100,100,100. Lower settings increase the contrast between the base color and the highlight, making the object look even harder. (Higher settings lessen the hardness of the object be decreasing the contrast.) Set the reflectivity to its highest level, 255. The other factor affecting the highlight is the light angle and intensity. The intensity is easy. Just boost the light(s) to an intensity of 100%, when

using the global lights. If you've added your own local lights, make sure "create highlights" is checked, and edit the attributes of the lights to the color and intensity you desire. The light angle is also easy for round objects. As long as the light is on the same side of the object as the viewer, a highlight is usually visible. Flat objects are more difficult to highlight, but can be quite effective, as shown in the example. The pyramid shape uses Phong shading with the assign adjacents off. This drawing is called HardObject.4D.

Note that the highlight on flat objects or surfaces will "flash" the whole surface when using global lights, but will hold and move a contained highlight when using a local light (one you added yourself). The reason being the light from the globals is always parallel, while the angle of the light from the locals varies on the surface of the object. Also, when you edit the attributes of the object, you will see two sliders named "hardness" and "highlight size." Be sure to experiment with these until you get the effect you want.

So now you know how easy it is to make hard looking objects. Just keep the base color low, the reflectivity and hardness high, and, of course, Phong shaded. One note of interest here; if the object is texture mapped, the base color is derived at least in part from the bitmap texture. Lowering the strength of the texture on a dark but highly reflective poly base, or darkening the texture in a suitable program before loading helps these objects look hard even in their textured state.

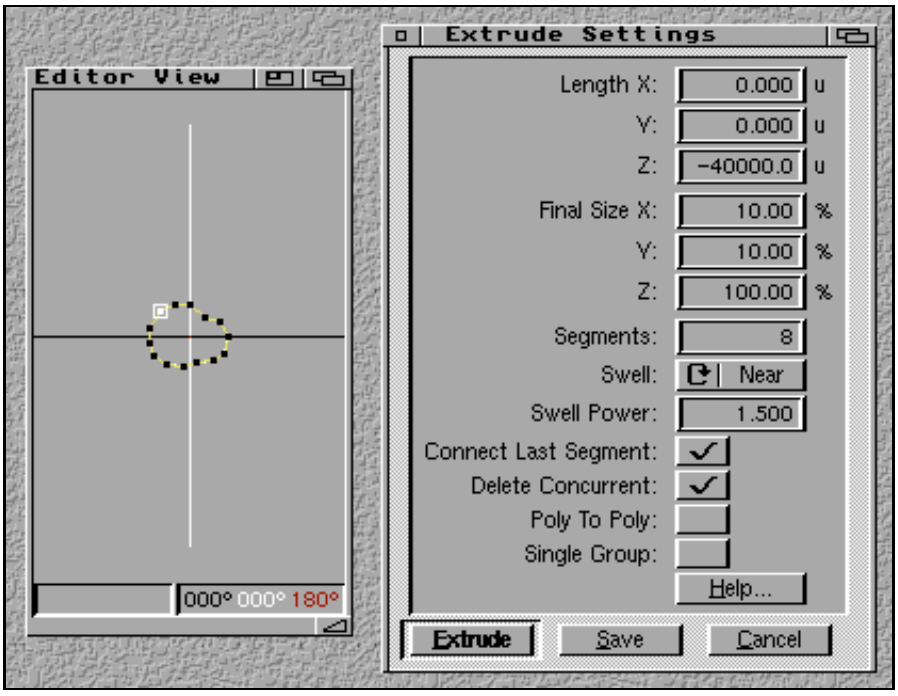
Contents

Creating Realistic Trees

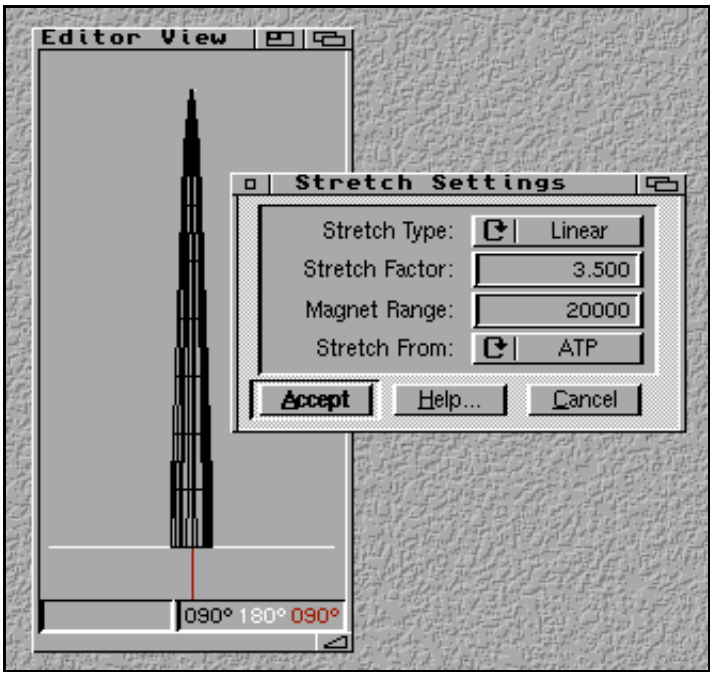
by Greg Gorby

(NOTE: In Aladdin's editor settings, set the units of measure to "units" before beginning this tutorial.)

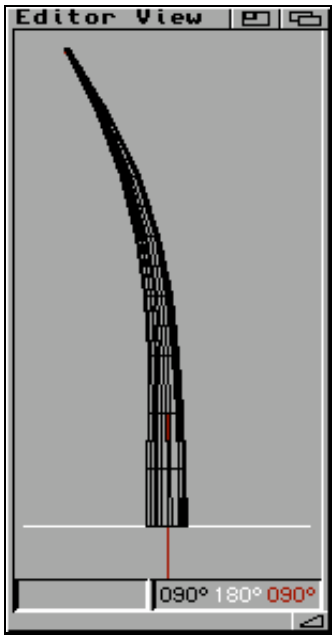
I'm going to suggest a method of making realistic trees in Aladdin 4D. It uses "genlock" textures. With this texture type selected, Aladdin will remove any part of a polygon that is mapped with the first color in the picture (color 0). If we map a texture that consists of just "dots" of color on a background of color 0, the only thing that will appear in the rendering are these dots. Let's use this to make the trees.



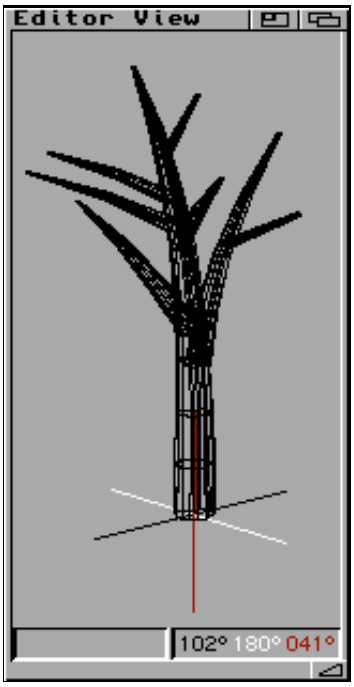
Flat view in the Z active axis. Next, make a template poly similar to the one shown here. Make it with a diameter of about 3000 units and 12 or 14 points. This is meant to be the cross section of the tree trunk that will be extruded and can be a plain circle if desired. Center the poly to the origin, and set the attachpoint to the origin. Right click the extrude tool. Set the extrude settings to -40000 z, 0 x, and 0 y, 8 segments, a swell power of 1.5, swell near, and connect last on. Use x and y final sizes of 10 percent, z at 100 percent, and extrude the template. You will have the shape shown. At this time delete the 2 polygons which are the 2 endcaps. Flat view in the Y active axis and move the screen down so that you can see all of the tree.



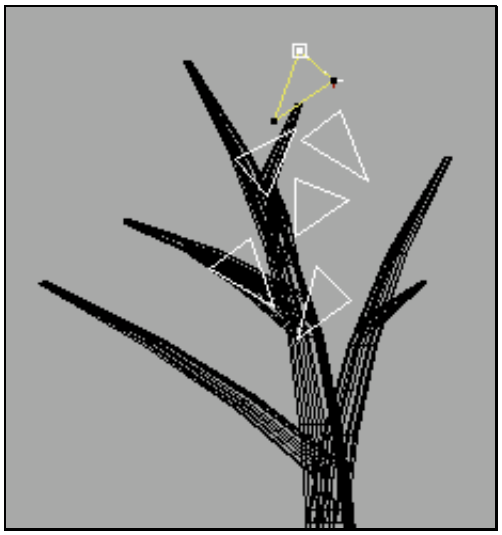
With no polys selected, use the right mouse button on the stretch tool to get to the stretch settings. Set stretch type to linear, stretch factor to 3.5, magnet range to 20000, stretch from ATP, and accept these settings. Now set the attachpoint at the bottom of the trunk. Select the top point of the trunk, left click the stretch tool, and pull it to the left a little. This will bend the trunk to a suitable shape. This is the main trunk.



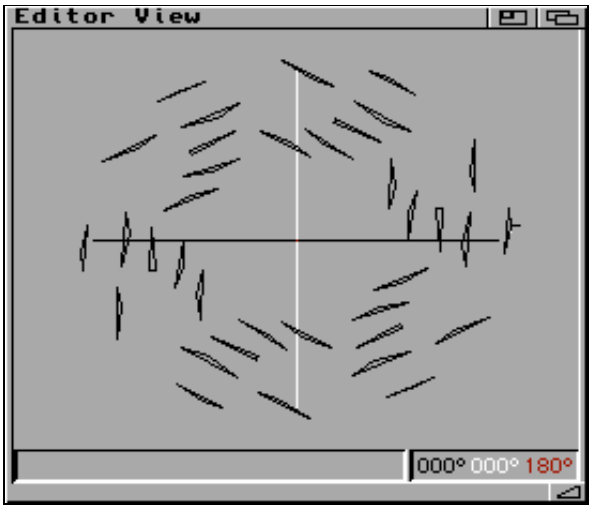
Let's use the top part of it for the secondary branch structure. Use the multiple select tool to select the top 4 segments of the trunk, and clone them offset at about 5000 units on the x. Now mirror them on the x, and move them to the approximate positions shown. Use the same method to select the top 2 or 3 segments of the original extrude and clone them 2 or 3 times. Now go to the z viewpoint, and rotate the new branches a little to make the tree fill out the 3D space, and position them at random as shown. There is no need to overdo the branch structure as we want most of the tree to be leaves. When you are finished, you will have around 200 polys in the trunk and branches. Now press and hold the 1 or 3 key on the number pad, and watch as the tree rotates, checking for any misalignment of the branches. When you are satisfied, select all the polys and group them.



Now let's do the leaves. Flat view in the y axis. Draw a simple triangle about 4000 units in size near the branches. Select the triangle, and selecting the x, y and z active axes alternately, rotate the triangle to random angles. Now clone the triangle 5 or 6 times, and rotate the new ones also at random angles on all axes. Keep the triangles relatively flat to the direction that the texture will wrap, so that each triangle faces the center of the tree, and keep them on one side of the tree.

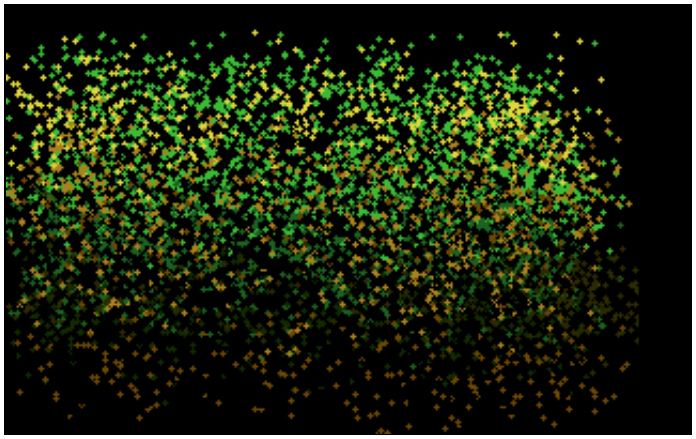


Flat view in the z axis, and move the triangles into a random pattern. Now set the attachpoint to the origin. Select all the triangles, and lathe them with solid off, 6 sweep points, 360 degrees, 100 percent size, and 0 offsets. You will now have a pattern of 36 or so rather random triangles. Select all 36 triangles and group them into a single group.



Right click on the clone tool, and change the clone settings to -8000 units on the z (0 x and y). Save these settings. Select the group of 36 triangles and clone them twice. Select the upper group and with the z axis active, hold the ALT key and resize them a little smaller than the other two groups. Show the branches, and reposition the triangles so they cover the branches. Group all 108 triangles into 1 group. Now set the clone defaults to 0,0,0 and select the new group and clone it once, and set it. Flat view on the z axis. One at a time, select each of the two groups, and with the z axis active, rotate them to random angles. You now have 216 triangles organized freely in space around the tree branches. You may want to select a few of the triangles and clone them, moving them to areas that in your opinion need them. The more triangles you make the fuller the final tree will be, Try to remember that the leaf patterns of a tree are not perfect. Purposely allow some areas to remain sparse and others relatively full. Rotate the view to inspect the distribution of the triangles. Group all 216 triangles into 1 group.

If you render the tree as it is, you will of course see just a bunch of triangles around the branch structure. Let's turn the triangles into leaves. Start up your favorite paint program. I use DPaint. Set the screen size to 320x200 and 32 colors. Make the palette colors an assorted imitation of fall colors, dark green, light green, orange, reddish orange, yellow and brown. Now resize your airbrush tool to about 2 inches in diameter. Select the second size brush, the one just slightly larger than a single point. Toggle off the menu and toolbox, and using each color alternately, airbrush the entire screen. Don't make the pattern even, because a little variety will make the tree more interesting and unpredictable. Make sure that you leave about half of the picture as color 0. Save this picture in your textures drawer.



Return to Aladdin. Select all the triangles, and edit their attributes to set their transparency to a value of 1, color to white, and reflectivity to the max. Now set their texture to the one you just painted. Edit the texture settings, setting the strength and color level to the maximums, map type to cylindrical projection, axis to z, and set the texture method to genlock. As a final step, select the branch structure, and set its color to a medium brown and shade the branches using Gouraud.

You're ready to look at the first render. Zoom in until the tree fills the screen top to bottom. In the render settings, turn on the fill, lights, Gouraud, and transparency permissions, and render frame 1.

You may want to adjust the light's position or color. Note the leaf patterns. If the triangle nature of the groups is very obvious, you do not have enough color 0 in the texture. If the

leaf pattern is too sparse, you can either increase the amount of color in the bitmap, or increase the number of triangles, or both. Smaller triangles result in sparser leaves, but less obvious triangles. Larger ones make the leaves denser, but also make the triangles more obvious. Experiment to find the results that most please you. If you see some horizontal streaking in the leaf patterns, those triangles are too "edge on" to the texture wrap. You can rotate them a little so that they face the center of the tree more, then reapply the texture. Also try using the texture antialiasing. It will make the leaves more dense. Remember that the darker areas are caused by the triangles actually showing their unlit sides. A local light placed in the tree or beside it can create some really wonderful effects. Notice also how you can see the branch structure of the tree periodically through the leaves, just as you can in a real tree.

You can add to the realism of the tree by adding a texture to the trunk and branches. A rope swing, or a kite stuck in the tree can also be very nice. You can change the appearance of the leaves by setting their reflectivity higher, or setting their color to maybe a pale green, and setting the texture strength lower so that it blends with this base color. You could also load a picture of a landscape as a background (F8 key) to give the tree a place to rest. Alternatively, you can create a ground grid and texture it with a grass texture.

After you have the tree made, try an animation rotating it around the z axis. During the animation the tree will show its true nature as a full 3D object. You could even add to the realism of the animation by using the deform modes of Aladdin and the stretch tool to create the illusion of wind lightly swaying the tree! And for a final effect, you could have a few of the leaves fall from the tree during the animation.

I hope this ability to create truly realistic 3D trees will help with your landscape projects. The same technique can be used to create wonderful hedges and shrubs to place in your architectural illustration and animations. Of course, once you have a tree that you like, you can save it, then use it in any of your drawings. The same tree can be cloned and altered slightly by resizing, stretching, changing color, etc. to make enough varied trees for any drawing. All of your trees and hedges should be place along with the buildings on a ground grid that has been textured with a grass texture.

Contents

Tips

by Greg Gorby

Extruded Fonts and Bridges

The fonts used in Aladdin 4D are of two types. They are differentiated by the .4df and .4dff appendage. The .4df stands for 4d font, and .4dff stands for 4d fillable font. The .4d fonts are meant to be used in those drawings that are going to be used in desktop publishing. After making the ProDraw clip they can be imported into ProDraw, turned into complex

objects, and used with no "bridge" between the inner and outer polys. This is important when the outline dolor of the font differs from the fill color, or when the font is outline only. The .4dff fonts are meant to be used for video work. They are actually only one polygon, regardless of the complexity of the letter. This is accomplished by using the join poly tool to connect the inner part of the letter with the outer part of the letter, and allows the letter to fill properly in the fill modes.

Even Spheres

When making your spheres, make your 360 degree lathe with exactly double the segments that are present in your 180 degree arc that serves as the template polygon. In this way the sphere will appear even from all directions! This is also true if you make a sphere using the "quadratic primitive" tool. Using the ellipsoid type, you would make the "segment 1" value twice that of the "segment 2" value.

Global Lights are Fast

Your images that include phong shaded objects will render much faster when using global lights only. The global lights have no actual position in space, just direction. You can think of them as the sun, or a source that is infinitely far away. Because of this, when the phong shaded object is rendered, only the calculations for the angle of the incident light need to be calculated. Local lights, on the other hand, do have a position in space. This means that not only the angle has to be calculated, but also the distance. This must be done for every pixel on the screen that is occupied by a phong shaded object. This can easily double or triple your render times! Of course, when you need local lights, use them, but be prepared for longer render times.

Interrupted Animations

As your proficiency with the program grows, you will find yourself wanting to make some long animations. Actually, we very seldom make an animation of less than 120 frames, but let's say that we want to render it in 360 frames (12 seconds on video at 30 frames per second) . If each frame is taking 5 or so minutes, this means that the full animation will take 30 hours. During this time, you will not be able to use the computer for any other work, unless you multitask, which will slow down the generation of the animation. This is not acceptable to us. We start the animation at quitting time. Then the next day, the first thing we do is hit the escape key and stop the animation! Aladdin will close the animation file, leaving all frames intact. The last frame is generally uncompleted. We then note the number of frames that were finished (usually by loading it into Deluxe Paint) or using a public domain utility. Then the partial animation is named according to how many frames it contains, for example, MyAnimation1_122.anim. Now the computer is free to complete the day's work. At quitting time we reload the drawing, and start the anim render with the next frame, in this case 123. This is continued until the anim is complete. When all parts of the anim are finished, we load each into Deluxe Paint, delete the incomplete frame, then append the next segment. When all segments are loaded, we save the then complete anim. The only thing you need to remember is that the anim you have must fit into your memory, along with the code from Deluxe Paint. With 5 megs of memory, you should be able to handle anims of at least 3 to 4 meg.

Changing Lights and Anims

This one comes from Mark Corbeiser of Modesto, CA. He points out that if you have the local lights change color during an animation, you should expect to have a much larger and slower playing animation than you will have if the lights stay the same during the anim. The reason? When the frame changes, if the lights have changed color, every pixel that is occupied on the screen by a poly that the light hits will change! This means much larger delta factors (changes between frames) resulting in larger and slower anims. One test anim I tried was 200,000 bytes without changing the lights, and played at 30 frames per second. When the lights changed, the anim jumped to over 2 megs and would not play faster that 7 FPS. Thanks Mark. This is definitely a very useful tip!

Contents

Matching Points

by Greg Gorby

The match points tool is one that is often overlooked. It has a common use that should be emphasized.

The world of Aladdin 4D is mixed between integers and floats. If you are not a programmer, this is certainly a nonsense statement, so let me explain. Numbers are held in the computer in two basic ways. One type of number is called an integer, and one is called a floating point number. An integer is a number without a fractional part. Examples would be 0, 1, 14, 14902857294, etc. A floating point number has a fractional part. Examples would be 0.0000, 1.01010, 14.7, 14902857294.4759849, etc. The advantage to integers is they are handled by the computer much more efficiently than floats. This can result in an operation happening up to 10 times faster than if floats are used. Even with a math coprocessor present, the integer calculations are still more efficient.

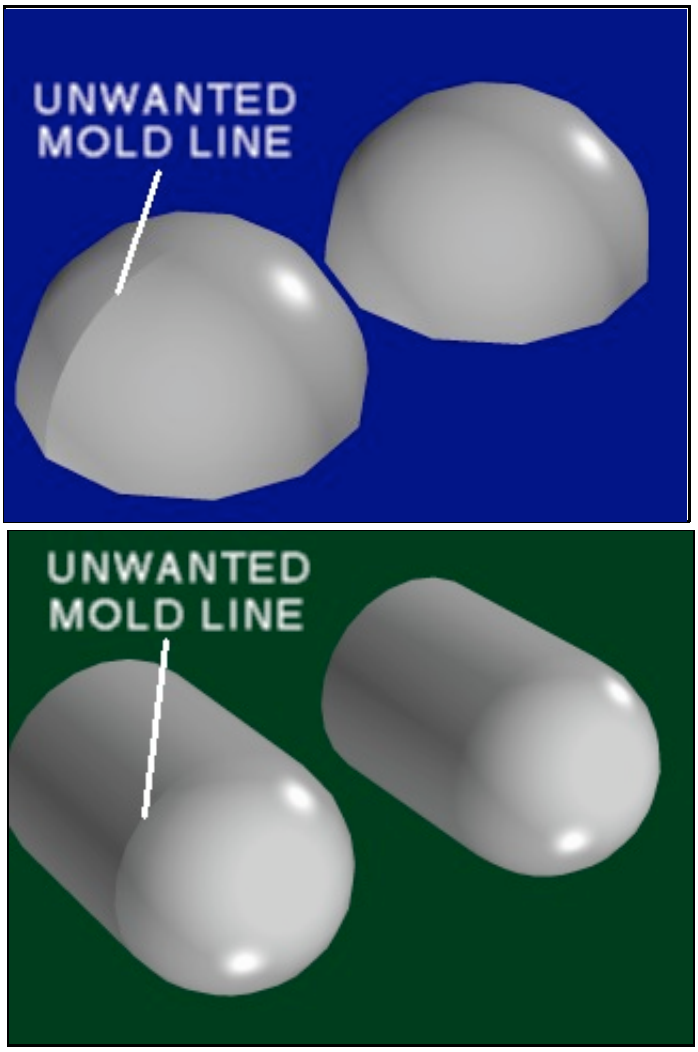
When you draw in the editor of Aladdin, you draw into an integer grid. You can place a point at 10000,10000,10000, but you cannot place it at 10000.5,10000.5,10000.5. However, when you perform an operation that requires a resizing or rotation of these points, they are first converted to floats. The points then undergo the transform needed, and the result is rounded down to the next lower integer for the final placement of the point. This means that a point that would actually end up with an x of say, -0.000000001, will find a final resting point of -1.

There is only one place that this rounding will affect your work in any significant way. This is when you use the lathe tool to create a solid. In some instances, the last polygon created by the lathe will not find a final place that is adjacent to the original template. The original template may have a y coordinate of 0, and the corresponding point of the last poly created may have a y coordinate of 1 or -1. Visually in the editor, you will never be able to see the

difference, but if you shade this lathed object, the adjacents will not be assigned across this area. The result looks as though it is a "mold line" in a molded plastic form.

To correct this situation, select the polys that are created around the template poly, and then use the match points tool on them. This tool will find those points that are closer than the minimum (set in the defaults menu item) and it will perform a snap on them. Notice that this is basically a sort algorithm, and can take some time for a large number of polys. After using the match points, free any shading , and then reapply it.

You do not usually have to use the match points on your lathes, but for those occasions when your placement and lathe angles cause this problem, it is a real time saver! Load the drawing Match1.4d and render it. You will see one form with the problem, and the second after using match points and reapplying the shading.



The second use of the match points is more obvious. Load and render the drawing Match2.4d. Here you see two "broom handles". As you can see, there is a problem with the shading on one of them. A circle was created and extruded to form the cylindrical part of the handle. A 180 degree arc was then made, and lathed 180 degrees to form the end of the handle. The end of the handle was snapped to the end of the cylinder. The formation of the

end involves floats that are rounded for final placement, as described above, so although the points look identical in the editor, they are a point or two apart. The polys at the junction of the cylinder and the end of the handle were selected, and the match points tool used to snap them together, Then the shading was applied. You will find that you don't often need this simple tool, but when you do, it can save you hours of time!

Contents

New Fonts

by Greg Gorby

There are three new fonts included with this newsletter. Of these, we have developed the Brush and Jas fonts. The third, deVille, is provided by Shane Earl of Salem, Ohio. (Font names are fun, aren't they? I loaded this font and realized it is derived from the Cooper family. Get it? Coop deVille!)

The Brush and deVille fonts extrude and take textures and shading wonderfully. The Jas font is a low memory type font, basically derived from 45 degree angles. It is very useful, because it really needs no shading, looking great in its faceted appearance. It also, of course extrudes and takes textures very well.

The Jas font is also very nice if you are doing deforms on a texture mapped title. Remember, after extruding a font and before texturing and deforming it, it is a good idea to use the "breakup" tool on the polygons. This will break those complex polys down into simple triangles.

Do you have a font you have created and would like to share? If so, please send it to us. (Note: At this time you would upload it to the Aladdin FTP uploads directory on Nova Design's web site.)

Contents

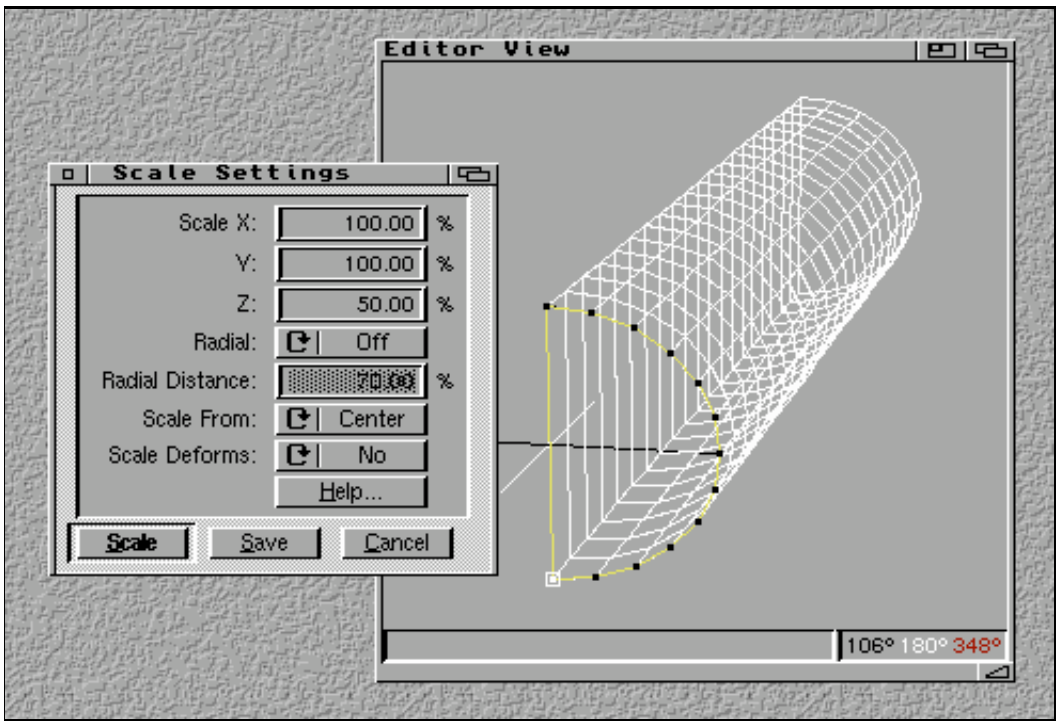
The Radial Option

by Greg Gorby

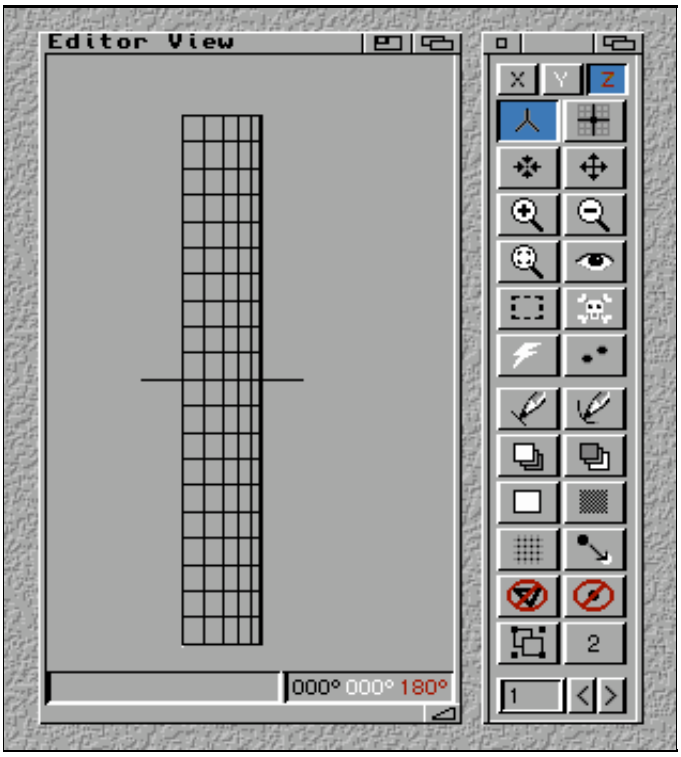
(NOTE: In Aladdin's editor settings, set the units of measure to "units" before beginning this tutorial.)

The scale tool has an option that is very exciting, and that you have probably never used. Let's do an example based on this tool, and in the process, gain some experience with it. This example is very simple to do.

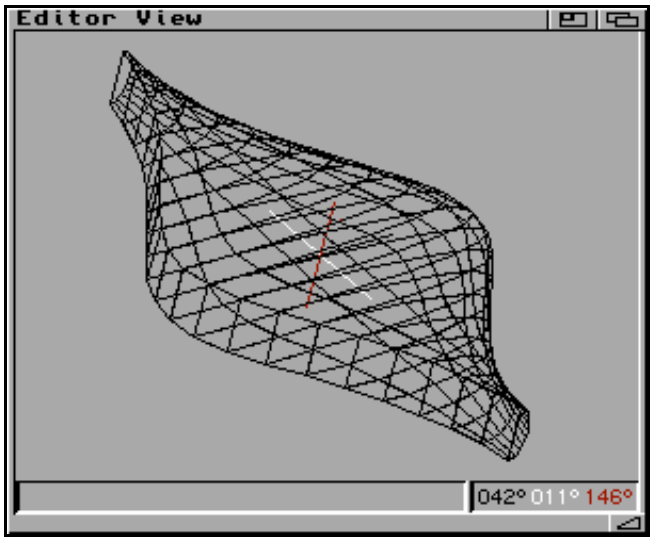
Get a new project and flat view in the Y axis. Use the "make arc" tool to make an arc of 180 degrees and 12 segments, with a single radius of 10000. Get your extrude defaults by right clicking on the extrude tool. Set the extrude to minus 80000 on the y, and 20 segments, connect last segment on, swell power 0, and 100% scaling. Save the new defaults. Select the arc, and extrude it.



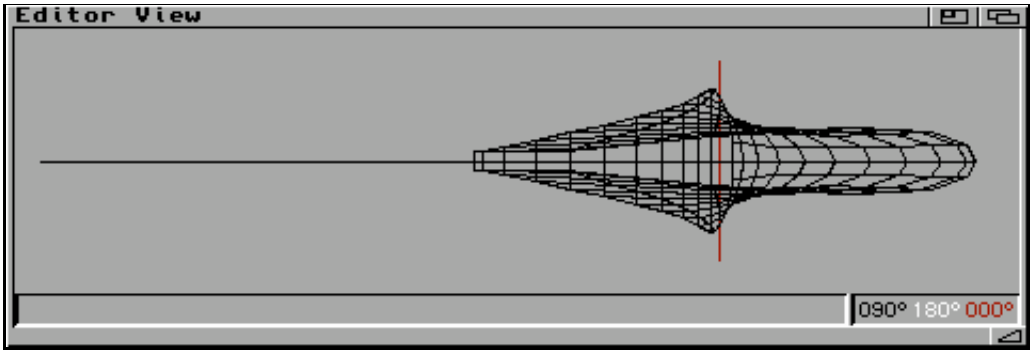
Select all polys and group them. Select the group and use the right mouse button to get the scale tool settings. Change the values to x,y,z of 100,100,50 and scale. The object will decrease by half along the z dimension. Now flat view in the Z axis. Place the attach point on the origin (right button click on the page move gadget). Select all, and center them to the attach point with a right mouse button click over the page center gadget. Now zoom out using the cursor up key until you can see the entire form on the screen. And finally, press the F7 key to enter parallel projection. Now save the drawing in its present state so you can reload it if you make a mistake in the following steps.



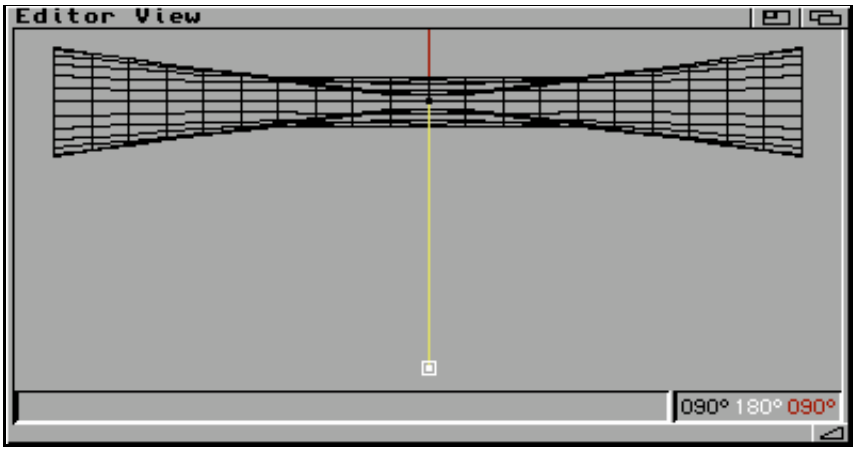
The half pipe that you have made is the clay that we will use to make the spaceship, so let's do it. Make sure nothing is selected. Go to the scale settings, turn on the radial option and save it. The numbers are of no consequence since we will use the tool interactively. Select the object. Now select the scale tool with a left click. Move the pointer to the bottom center of the editor window. Press and hold the left button and drag the pointer to the left. You will see the middle of the object expand and the ends of the object contract. Move the pointer until the ends are about half as wide as they were originally, and SET the object. Now select the object again, and repeat this step, using the scale tool until the ends are half of their present size. Then do it a third time. The reason you should do it in three separate steps is because the radial effect makes its changes in the point's position based on the point's original position. If you tried to make the final shape in one step, by the time the middle was "round" enough, the ends would have inverted. By doing it in separate steps, you can make the center swell as much as desired without having the ends invert. If you have made the ends invert, reload the original and go through this paragraph again until you have it as illustrated. Now save the drawing in this state. Rotate the view around until you can see and understand what shape you have.



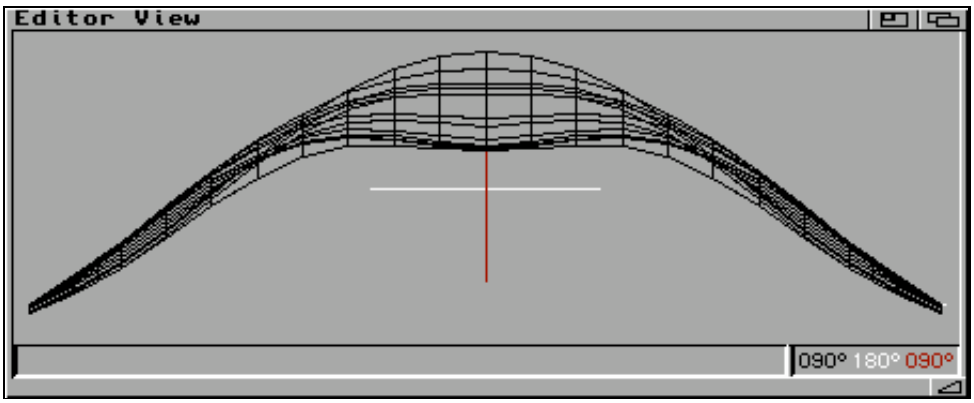
Flat view in the Y axis. Select the object, then use a right button click over the page move gadget. (This sets the attach point to the center of all the selected polygons, the geometric center of the ship.) Now get the freehand tool and using the bottom edge as a straightedge, draw a line to the left as shown and set it. This line is an "extender" and changes the sphere of influence of the radial effect as discussed a little later. Now select all (including extender) and get the scale tool. Move the pointer to the middle of the left screen edge, press and hold the left mouse button, and drag the mouse downwards until you have the shape shown. Remember, do not let any part of the shape invert on itself. If need be, you can do the procedure in two steps. In this case, you should be able to do it in one. Set everything and rotate the view to see what has occurred.



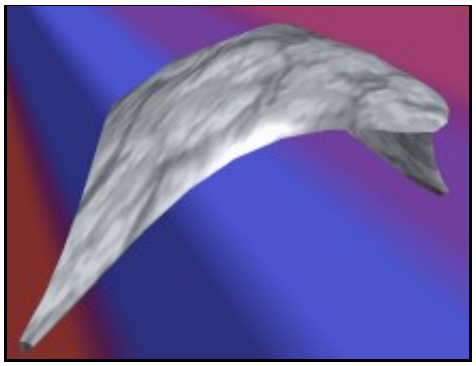
Delete the extender polygon, then flat view in the X. Set the attach point to the center of the ship again. Get the freehand tool and draw an extender down as shown.



Select all and get the scale tool. Using the left edge of the window as a straightedge, create the shape shown, this time moving the mouse upwards. This one will require two steps (apply the tool and set again) to get the final desired shape. Finally, delete the extender and rotate the object around to see the completed ship's form. If you wanted to you could create some simple lathes and add rocket engine ports to the rear, and maybe gun turrets on the "wings".



To render the ship for the best effect, you should first use the "to triangles" menu item, which will convert all these non-planar rectangular polys to triangles. Then phong shade it. Enable "smooth adjacent" with a maximum angle of 45. After shading, select all and set up the attributes (hit "a" key) with r,g,b of 80,100,120, with reflectivity of 255, maximum hardness, and highlight size of 240. Assign it a bitmap texture (hit "t" key) by loading the "marred.ham" texture. Set it to planar projection along the z axis, with strength and color of about 50 percent, and antialiasing on. Go back to perspective view (F7) and zoom in and rotate the view to get a better composition. It is now ready to render. The picture below also has a background (F8) made with the Helix procedural texture.



Now that you have experienced the radial option, let me talk a little about its operation. When you have selected polys and activate the radial option, it first finds the geometric center of the selected polys. Then it finds the farthest point from this center in the selected polys. It then describes a virtual sphere of influence at these coordinates in space with this radius. The center of the selected polys is the center off the sphere, and the distance from this center to the farthest point is the radius of the sphere. Now as you drag the pointer, the tool moves the points based on their relative nearness to the surface of this sphere, Those points that are nearest to the surface of the sphere are least affected, and those points that are farthest are most affected. The tool is excellent for creating swells and warps in existing 3D forms as we have done above. Try creating a gridded cube with segments of 6,6,6 and applying the tool to it. You should be able to actually turn the cube into a sphere, as well as making some sides concave, convex, etc. Try it on the cube using various extenders for some more freeform effects.

Note that when using the tool, you can move the pointer freely up and down, and left and right on the screen, affecting the two directions of the active view. You can also hold the ALT key, forcing the movement to be evenly distributed in these two directions or hold down the SHIFT key forcing the movement to be evenly distributed in all three dimensions. After some experience (and experimentation) with the tool, you will be able to intuitively know that this is quite often the tool of choice to modify your objects. As with any freeform tool, this tool changes the points of the polys individually, warping the polys themselves. If you see odd places in your renders, as dark areas or bright areas where they shouldn't be, you may need to break up some areas into triangles.

Contents

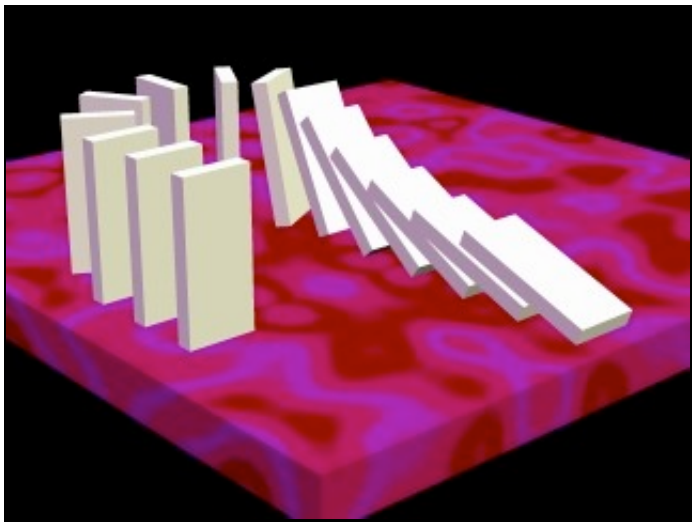
On The Disk

The files that originally accompanied these tutorials on disk have been converted to Aladdin 4D version 5 format and are available for download by clicking here.

You have already dealt with some of the drawings that are on this issue's disk. I would like to tell you a few important things about the ones you haven't seen yet.

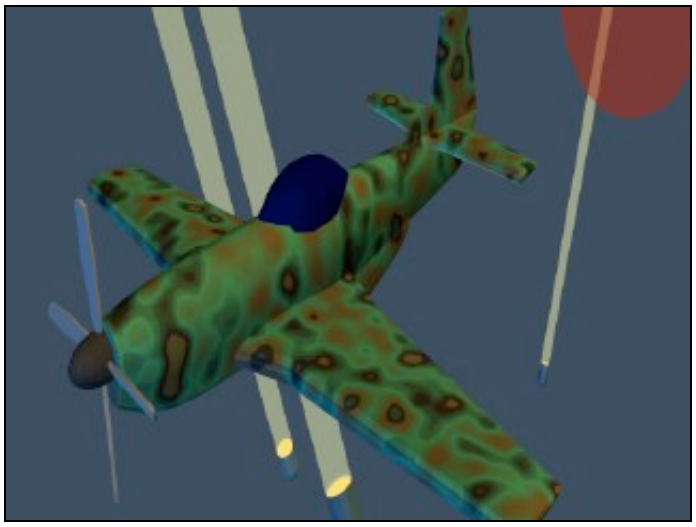
"Comehome.4d" is an NBC-like illustration. In the drawing, a square was extruded decreasing in size. The resulting shape was then cloned to create a grid pattern, and the unnecessary back faces deleted. A circle was created, and the stretch tool used to pull it into the teardrop shape. This was then used as a path to PathExtrude a smaller shape around. The teardrop shape itself was also extruded to form the "light" tubes. Finally this entire single extruded teardrop-shaped section was lathed 180 degrees with solid off. This formed the complete logo with 6 sections. Notice how the extruded tubes, when made partially transparent and set to a high glow attribute make very effective visible light shafts.

"Dominoes.4d" is a little animation I did on a whim (the most enjoyable kind). It has a few dominoes that begin to fall, and hit each other. As they fall, they make a circular pattern and come back at you. The paths' timings were quite critical, but manageable.

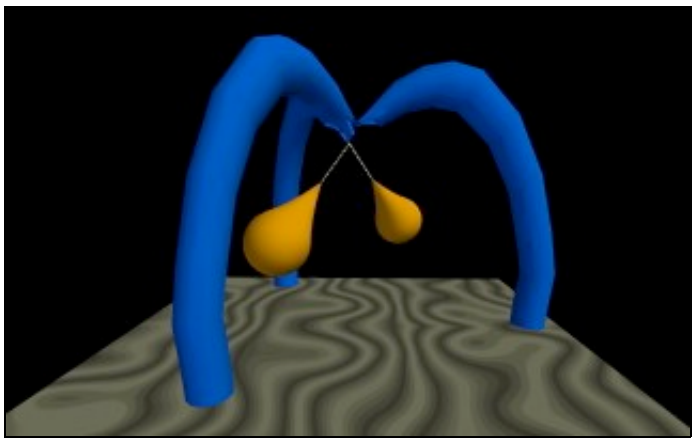


"Mustang.4d" is a more complex drawing. It is a rather simple representation of a mustang aircraft doing a fly-by. Do a preview anim save of this one and take a look at it. There are spotlights searching the skies, and bombs bursting. The spotlights are simple extruded circles. The endcaps of the extrudes are set to 0 transparency, but the pipes of the extrudes are set to high transparency. This not only gives you a shaft of light, but also a circular "source" at its base. The bomb blasts are really interesting. A rather random path was

drawn. A sphere was created, centered to the first point of the path, and assigned to the path. A deform level was started for the path, and the level was ended without changing the sphere. Now the sphere was selected and resized to be very small. During the animation, the sphere would grow to its original position in the deform. The path was altered to have movement with the point lock on ("p lock" in the edit path panel). It also changes the transparency of the sphere from 0 to 255. The sphere was set to change from white to bright orange to a dark red. And finally the number of cycles for these changes was set to match the number of points in the path. The net result of all this is what you see. Little bursts of light at the center of an expanding sphere that fades out as it gets larger and changes to a darker red.



"Plumb.4d" is a a very nice drawing of two teardrop forms suspended by cords from an interesting set of legs. The teardrops were made with a simple lathe. The cords were made using a small triangle, and lathing it by a large number of degrees (around 2100) with an offset on the Z. A simplified cord, one third the number of polys, can be made doing the same thing with a straight line with the last segment off. Notice that there are actually two teardrop forms, and paths take care of their separation during the animation. Having them in the same space in the editor makes conceptualizing their relative motion simpler.



"SchoolHouse.4d" is a rather simple drawing of a one room schoolhouse I saw in Vermont. It illustrates one way of putting windows, doors, etc. in place. All the basic "cube" shapes

of the building were built from grids where the basic unit squares of the grid are the same size. This makes it easy to use a smaller brick texture, and tile it on. Not only can you use a small texture, but if the unit squares are the same size, the brick texture will be even. The roof takes a more direct approach, projecting a single large texture onto its surface. The windows and shutters are actually separate polygons placed a hundred or so units away from the outside of the house. The illusion is quite convincing, with a minimum of modeling.

Of course, it is also possible to actually digitize the entire side view of an existing building, with windows, boards, bricks, etc. in place, then project the bitmap into place on a model constructed just to receive it. These are often unbelievable in appearance, but require rather large bitmaps. The approach is to take photos of the building from all four sides, avoiding perspective as much as possible, and choosing an overcast day so the lighting is relatively even on all sides. These photos are then digitized and used as guides in drawing the house. The digitized images are then mapped to the house in projection mode.

Included are several examples of titling effects that are relatively easy to achieve: "shake.4d", "stretch.4d", "ooooz.4d", and "thump.4d" use deform levels to get the effect and are straight forward. "Spin.4d" is a simple rotation, and is effective when doing multiple titles. You can just set up each title in a separate space, then render them as separate anims. These can then be genlocked to tape one after the other for a consistent, animated title presentation.

"WhyWait.4d" is another title that is quite common. It demonstrates how to move a large 3D extruded title from off screen, rotating into place in the center of the screen. First, you make the word and extrude it. You then draw a straight (or segmented) path from its center out into space while flat viewed in the Z axis. The first point of this path is at the center of the word, so the word will move from this point. This is not what is wanted, so change the first point of the path to the other end! Now select the word and center it to the new first point. During the animation, the word will start at this position, and move back to its original position.

Contents

Thanks!

I would like to take a moment to thank you all for spreading the word (and not the disk!) about Draw4D-Pro (Aladdin 4D's Precursor). As of the date of this writing, the program is only about 6 weeks old and has had almost no advertising, yet the success of the program has surpassed our expectations! This is mostly due to you, the users of the program, letting your friends know just how good it is. We look forward to continuing to polish and improve the software.

Thank you!

***

Links:

- Main
- Aladdin's Lamp issue 2 newsletter
- Aladdin's Lamp issue 3 newsletter
- Aladdin's Lamp issue 4 newsletter
- Aladdin's Lamp issue 5 newsletter
- Aladdin's Lamp issue 6 newsletter
- Aladdin's Lamp issue 7 newsletter
- Aladdin's Lamp issue 8 newsletter
- Aladdin's Lamp issue 9 newsletter
- Aladdin's Lamp issue 10 newsletter
- Aladdin's Lamp issue 11 newsletter
- Aladdin's Lamp issue 12 newsletter
- Aladdin's Lamp issue 13 newsletter
- Aladdin's Lamp issue 14 newsletter
- Aladdin's Lamp issue 15 newsletter
- Aladdin's Lamp issue 16 newsletter
- Aladdin's Lamp issue 17 newsletter
- Aladdin's Lamp issue 18 newsletter
