

## **ITextures**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> IITextures	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		January 19, 2023
<i>SIGNATURE</i>		

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>IITextures</b>	<b>1</b>
1.1	Ian's Imagine Textures . . . . .	1
1.2	Author Information . . . . .	1
1.3	Contents of Archive . . . . .	2
1.4	Example.iob . . . . .	2
1.5	Disclaimer, ick. . . . .	2
1.6	Other Imagine Programs . . . . .	2
1.7	Requirements . . . . .	3
1.8	AttMath.itx . . . . .	3
1.9	ColorCube.itx . . . . .	4
1.10	ColorSpiral.itx . . . . .	5
1.11	EdgeFill.itx . . . . .	6
1.12	LedCount.itx . . . . .	7
1.13	ShowCount . . . . .	7
1.14	RadialColor.itx . . . . .	8
1.15	Roughness.itx . . . . .	8
1.16	SetChild.itx . . . . .	10
1.17	Ugly.itx . . . . .	10

---

## Chapter 1

# IITextures

### 1.1 Ian's Imagine Textures

Textures for Imagine 3.x (V1.1)  
by @{"Ian Smith" link Author }

Information:

@{ " Requirements " link Requirements } @{" Other Programs " link Other } ←  
@{" Contents " link Contents} @{" Disclaimer " link Disclaimer}

Textures:

@{" AttMath.itx " link AttMath } Perform math on various ←  
paramaters.  
@{" ColorCube.itx " link ColorCube } Fill a volume with all 16 ←  
million colors.  
@{" ColorSpiral.itx " link ColorSpiral } Spiral 2-5 user defined colors ←  
outward.  
@{" EdgeFill.itx " link EdgeFill } Draw lines around the edges of ←  
triangles.  
@{" LedCount.itx " link LedCount } Flash the led when a ray hits ←  
this texture.  
@{" RadialColor.itx " link RadialColor } Spread colors out radially.  
@{" Roughness.itx " link Roughness } Roughness with user defined size ←  
.  
@{" SetChild.itx " link SetChild } Apply attribuites to children.  
@{" Ugly.itx " link Ugly } Really mess up an objects color.

### 1.2 Author Information

I can be reached via the following...

Snail Mail:

Ian M. Smith  
27 Gibson Street  
North East, PA 16427

E-Mail:

IanSmith@moose.erie.net

### 1.3 Contents of Archive

```

AttrMath.itx                2580 ----rwed 03-Nov-94 02:28:14
ColorCube.itx              2484 ----rwed 15-Oct-94 20:58:44
ColorSpiral.itx            2316 ----rwed 03-Nov-94 22:31:36
EdgeFill.itx               1988 ----rwed 29-Oct-94 16:51:02
@{ "Example.iob" link Example }                2798 ----rw-d 03-Nov-94 21:28:20
IITextures.Guide           19699 ----rw-d 12-Apr-95 17:37:51
LedCount.itx               916 ----rwed 29-Oct-94 03:40:40
RadialColor.itx            2856 ----rwed 03-Nov-94 19:35:36
Roughness.itx              2032 ----rwed 03-Nov-94 19:45:46
SetChild.itx               2116 ----rwed 28-Oct-94 01:35:32
ShowCount                  6524 ----rwed 29-Oct-94 03:47:18
Ugly.itx                   1028 ----rwed 12-Apr-95 17:29:12

```

### 1.4 Example.iob

There is an "Example.iob" included that you can render to get a look at some of the the textures. To get Imagine to find the textures on the example objects, copy the IITextures directory to RAM: first. On my system I have been making a directory for everyone that has written textures, to keep them separate from Imagines. That way Impulse doesn't get a phone call asking why their LedCount texture gives enforcer hits by accident when someone forgets that they added a new texture or two.

### 1.5 Disclaimer, ick.

Disclaimer and other stuff:

This software is in the public domain. I take no responsibility if it replaces all your large, detailed Imagine projects with a sphere hovering majestically above a checkered ground.

Send me a new car if you like them, or an e-mail message if you can't find a big enough envelope.

Also send any comments, bug reports, or interesting ideas or uses for the textures here.

### 1.6 Other Imagine Programs

Ian's Imagine Utilities

-----

IIUtilities.lha on AMinet in gfx/3d.

Version 1.3 as of Feb-03-95.

These 8 utilities will perform usefull (I hope!) operations on Imagine 3.x staging or object files. All files are ARexx scripts and have the instructions in the headers except Stars and Jitter which are binaries.

- Jitter - Randomly disturb an objects points. Like Lightwave. Works on all objects in a group and states.
- StageFrames - Increase or decrease the length of an animation and automagicly scale the actor bars. Very useful for playback at faster rates or motion blurring.
- ReduceFrames - Used with StageFrames to determine how to reduce.
- StageScale - Increase or decrease every object's size in a scene and reposition them. Used to reduce the rendering time for raytracing.
- StageShift - Put X blank frames at the beginning of an animation so you can merge two animations. (Needs ISL to merge)
- StageLengthen - Add frames to an animation and/or lengthen actor bars to the new size.
- ImTotal - Read the IMRT hunk from Imagine's IFF files and display rendering time. Will do wildcards and show total and average rendering time.
- ShowStage - Parse a staging file and list every object and brush loaded for the entire scene or a single frame. Shows totals for points, lines, faces and bitmap pixels.
- Stars - Generate starfields quickly and easily. It creates a spherical, cubical or torus shaped shell of stars with a variable thickness. Stars can be randomly shaped colored and sized and made from triangles, cubes or disks.

## 1.7 Requirements

All textures require an Amiga system with at least a 68020 and an FPU.

Integer versions of Imagine are nor supported, sorry.

## 1.8 AttMath.itx

AttMath.itx

Type: Color, Reflect, Filter.

Overview:

=====

This Imagine 3.x texture modifies the color, reflect or filter values of your object. Negative, multiplication and max/min clipping are supported.

Usage:

=====

All the parameters can be used and are applied in the order that they are listed in the texture requester.

---

Parameters:

=====

Negate Color, Negate Reflection, Negate Transparency

-----

These simply negate the color values for each type. Set to 1.0 to enable each.

\* Color, \* Reflection, \* Transparency

-----

The values here are multiplied to the current values. Values above 255 and below 0 are not allowed via the Max/Min parameters below.

Max Color, Max Reflection, Max Transparency

Min Color, Min Reflection, Min Transparency

-----

These values set the maximum and minimum colors allowed on the object. It is not recommended to use values less than zero, and using colors over 255 may work strangely.

Notes:

=====

Not much to say, a pretty simple texture. Useful for special effects or to tweak an existing texture.

## 1.9 ColorCube.itx

ColorCube.itx : A space filling texture.

Type: Color, Reflect, Filter.

Overview:

=====

This Imagine 3.x texture creates a three-dimensional color cube that surrounds your object. Using this texture on a cube results in half a million colors flowing smoothly over the surface. Repeatable color, reflect and filter mapping are supported.

Usage:

=====

The default parameters provide a three dimensional non-repeating color-mapped space that contains all 16 million colors. The coloring is based on the texture axis. Red runs along the X axis, green follows Y and blue is mapped to Z. At 0,0,0 (based on the texture axis) you get black, and at the end of the positive side is white. You must make sure that you place the axis properly when not using repeat. Your object should be completely enclosed within the positive side of the texture axis, just like when applying a brush map. This texture is similar to the Rainbow.itx that comes with Imagine 3.0 with the main difference being the ability to modify the objects colors instead of replace them.

Parameters:

=====

ColorBlend, ReflectionBlend, and TransparencyBlend.

-----

These three values determine how strongly the texture is mixed with the existing values of your objects surface. A 0.0 would leave the surface untouched, a 0.5 would result in an even mix between the old and the new, and a 1.0 (default for color) leaves nothing of the original attributes.

MaxRed, MaxGreen, and MaxBlue.

-----

This is the color used at the far side of the texture axis. The default is set to white. Lowering these values will darken the entire object, raising them too much can wash it out. Try 300-350 for bright colors. Setting any to -1 will turn off that color axis.

ColorRepeat, ReflectionRepeat, and TransparencyRepeat.

-----

This turns on the repeat for each of the different mapping techniques. Normally the texture is only applied inside the bounding box. This option repeats the texture smoothly for infinity in all directions. Set to one to enable repeat, any other value to disable it.

Notes:

=====

A rotating cube standing on one edge can be a good example animation for 24 bit boards, or ones like DCTV or the new HAM-8 modes.

In trace mode, transparency can give great shadows and lighting effects. Try morphing an object out of camera view, using the filtered rays to light your scene. Interesting.

Rotate the axis in one or more directions and apply it repeatedly to the ground for a good effect.

## 1.10 ColorSpiral.itx

ColorSpiral.itx : 2D texture.  
Type: Color, Reflect, Filter.

Overview:

=====

This Imagine 3.x texture draws a flow of colors around the texture axis much like a radar screen sweeps around. You can have 2, 3, 4, or 5 colors in the sweep and have the spiral as well.

Usage:

=====

Apply this texture to a default plane for an example of what it does. The texture is centered at the textures axis, and draws it in the X any Y directions. The Z axis has no effect on this texture.

Parameters:

=====

Spiral Every X Units

---



-----  
 This tells the texture how much to spiral the colors. A negative number will not spiral the texture. A positive number will give a spiral effect. The default is 100, which means the colors spiral 360 degrees every 100 Imagine units. Smaller numbers give more spirals.

Color 1 Red, Color 1 Green, Color 1 Blue  
 Color 2 Red, Color 2 Green, Color 2 Blue  
 Color 3 Red, Color 3 Green, Color 3 Blue  
 Color 4 Red, Color 4 Green, Color 4 Blue  
 Color 5 Red, Color 5 Green, Color 5 Blue  
 -----

These are the colors that flow over an object. You can use from 2 to 5 colors. To use less than 5 colors, put a -1 in the Red value of the colors you don't want. You MUST disable the last colors first or you will get unpredictable results. So for a 4 color spiral, use a -1 in the Color 5 Red parameter.

#### Notes:

=====

Using 5 shades of blue and white with a lot of spiral can give a great whirlpool effect, especially when animated. You can animate it by rotating the texture axis along the Z axis, or by changing the Spiral amount and morphing the object.

When used on the ground you have to rotate the texture axis for the proper effect, the default axis is set up to be used on a plane.

## 1.11 EdgeFill.itx

EdgeFill.itx

Type: Color, Reflect, Filter.

#### Overview:

=====

This Imagine 3.x texture can simulate a wireframe object or the Latticeize function by coloring the edges of an objects faces.

#### Usage:

=====

You can set the color, reflect and filter values separately. Using the color value can give a wireframe look while the filter values can give a simulation of the latticeize function.

#### Parameters:

=====

Color Edge Size, Reflect Edge Size, and Filter Edge Size  
 -----

This is a fractional value between 0.0 and 1.0 that tells the texture how much of the faces to fill. To get an effect similar to wireframe mode use a small value. This doesn't give a perfect wireframe rendering however. If you have very long and stretched triangles, the lines will vary in thickness. Experiment to get the effect you want. A value of 0.2 is good for imitating latticeize.

Color Blend, Reflect Blend, and Filter Blend

-----

This sets the mix between the texture and the objects original color. A 1.0 gives the texture its full effect, a lower number will blend the texture with the original attributes.

Color Red, Color Green, Color Blue,  
Reflect Red, Reflect Green, Reflect Blue,  
Filter Red, Filter Green, and Filter Blue

-----

This is where you give the values that will color the edges of the faces. If you want to try out the latticize effect, set the objects filter value to 255,255,255 and the filter values in the texture requester to 0,0,0. This will not work right if the object is set specular since the empty holes will still give a specular highlight.

## 1.12 LedCount.itx

LedCount.itx  
Type: None

Overview:  
=====

This Imagine 3.x texture can optionally blink the LED when a ray strikes it and/or keep track of the total number of hits.

Usage:  
=====

This is a simple texture to apply. There are only two settings, one to control the LED blink, and another to keep track of the ray hits. The included program @{ "ShowCount" link ShowCount } will display the number of ray hits after ↔  
hits after  
or during a render. The ray counting is a bit of a hack. It uses location \$0 to keep track of ray hits. Location \$0 is cleared when the texture is called before rendering, and then updated each time the texture is called. If you have an MMU and use Enforcer, you will get a large number of errors if you use this function. You have been warned.

Future:  
=====

Is there one? I could do this in a system-friendly manner I suppose, but is anyone interested? As it stands, I really can't think of a real use for this texture other than "Gee, neat flashing light." so improving it is not a high priority. I'd be interested if anyone does find this useful. :-)

## 1.13 ShowCount

Simply run from the shell, no paramaters.

---

## 1.14 RadialColor.itx

RadialColor.itx

Type: Color, Reflect, Filter.

Example Object: Plane, Ground.

Overview:

=====

Produce non-random circular patterns of color on an object.

Usage:

=====

This is a simple texture to implement. All three types of mapping can be used at once.

Parameters:

=====

Color Red Length, Color Green Length, Color Blue Length  
 Reflect Red Length, Reflect Green Length, Reflect Blue Length  
 Filter Red Length, Filter Green Length, Filter Blue Length

-----  
 RadialColor applies its colors based on the distance from the textures axis. Set these values to tell the texture how to color your object. The length values tell how many units it takes for a color to range from zero to its maximum value of 255. Setting all three RGB values produces grays, and setting each to a different value gives a large range of colors. A -1 tells the texture to leave that value unchanged.

Smooth Color Repeat

-----  
 Normally the colors start over at 0 from 255. Setting smooth causes it to eliminate the sharp color change and replace it with a smooth gradation. The default is on. Set it to 0 to turn this feature off.

Color Mixing, Filter Mixing, Reflect Mixing

-----  
 This sets how much to mix the texture color with the objects colors. A 1.0 overrides the objects colors, lower starts to blend the texture in instead of just replacing the colors.

Notes:

=====

Normally a sphere is a dull object to use with this texture. However, try a default sphere and use very small values for the color length. Example: R=0.1 G=0.2 B=0.3  
 This works because the triangles on a sphere are flat and not curved, so the distance from the axis changes very slightly along the surface. Phong shading has no effect on this because it only modifies the lighting, and does not change the actual surface of the object.

## 1.15 Roughness.itx

Roughness.itx : A space filling texture.

Type: Color

Type: Color

Overview:

=====

This Imagine 3.x texture creates an effect similar to the roughness setting in the attributes requester. This texture can be used with animations without the problems the roughness attribute has.

Usage:

=====

Applying the texture to an object will give the appearance of a rough surface. This is not altitude mapping however, but a color change. The texture works by varying the surface color from its original values to either black or white. You control how far the color is allowed to wander. By default, the texture has an infinite resolution and can't be rotated without giving an ugly static pattern. You can pixelize the random pattern to eliminate that problem.

Parameters:

=====

Roughness

-----

This is equivalent to the roughness slider in the attributes requester. A value of 1.0 will vary the color all the way to the Min and Max values that are described below. A value of 0.1 will only slightly vary the surface color.

Graininess

-----

A value of zero here turns off the pixelization function. The texture will always show a fine random pattern no matter how far you zoom in or out. This will not animate if you move the camera or object. You can include objects like this as long as they do not move. If you increase this value it will begin to pixelize the surface. A good rule of thumb is to increase this value until you can start to see the pixelization effect, and then use a slightly smaller number. When the object is panned or rotated, the roughness pattern will remain fairly steady, although it will still have problems around the edges.

Low Red, Low Green, Low Blue and High Red, High Green, High Blue

-----

These values are what the surface of the object will be modified to. The defaults are full black and full white. This gives the effect of lightening and darkening the color. Setting a value to -1 will leave that color untouched. So changing the blue and green values to -1 would only change the red component of the objects surface.

Notes:

=====

It's not a perfect replacement for the roughness attribute, but for background objects that don't move it works perfectly. Fiddling with the graininess factor can give objects that look good when when animated.

The randomness factor isn't perfect yet.. sometimes a 'seam' shows down the textures axis. Moving it off of the object will solve that problem.

## 1.16 SetChild.itx

SetChild.itx

Type: Color, Reflect, Filter, Fog.

Overview:

=====

This Imagine 3.x texture lets you set the color, reflect, filter and fog settings for a group of objects all at once.

Usage:

=====

You can override the default values of an object by setting the appropriate color values or fog length. The blend is how much the new values take effect. A 1.0 will leave nothing of the objects color, where a 0.1 will only slightly color the object. A -1 in any of the color values will turn off that setting.

Parameters:

=====

Color Red, Color Green, Color Blue  
Reflect Red, Reflect Green, Reflect Blue  
Filter Red, Filter Green, Filter Blue and Fog Length

-----  
These modify the objects attributes. Set them to -1 to disable.

Color Blend, Reflect Blend, Filter Blend, and Fog Blend

-----  
Set the blend to 1.0 to replace the original attribute, lower to blend the new settings in.

Notes:

=====

The fog setting is not as useful as it could be. Fog is only enabled by setting the fog length to something other than 0.0 in the attributes requester. This texture can modify the fog setting, but it has no effect if fog has not been set in the attributes. The opposite is also true. If fog is turned on, setting the textures fog attribute to 0.0 will not turn it off.

## 1.17 Ugly.itx

AttMath.itx

Type: Color.

Overview:

=====

---

This Imagine 3.x texture scrambles the color of an object in a very ugly and probably useless manner. :-)

Usage:

=====

This texture creates a buffer from 1 to 1000 cells in size, and then fills it with the color of the object when a ray hits it. When the buffer is full, it begins to empty it all over the object. This texture is global, and does not know what object it is on. Therefore colors will 'spill over' onto other nearby objects. It is also dependant on the path of the ray, so moving the object, camera, or even other objects will cause the mess to change.

Parameters:

=====

Image Shift (1-1000)

-----

This is the only parameter. This gives the size of the buffer. Very small numbers will have little effect, and large ones mess the object up pretty bad. Don't use numbers over 1000 or you will most likely crash your machine. No error checking is done for speed.

Notes:

=====

You're not going to want to use this on anything critical, as it really is fairly unpredictable what it does. Might be good for some cyberspace scene when you need to show an object being deleted or infected or something. This does not add any new colors to an object, only scrambles the existing ones somewhat. So it will not do anything to a solid object. Try applying a brushmap first, then the Ugly texture.

---