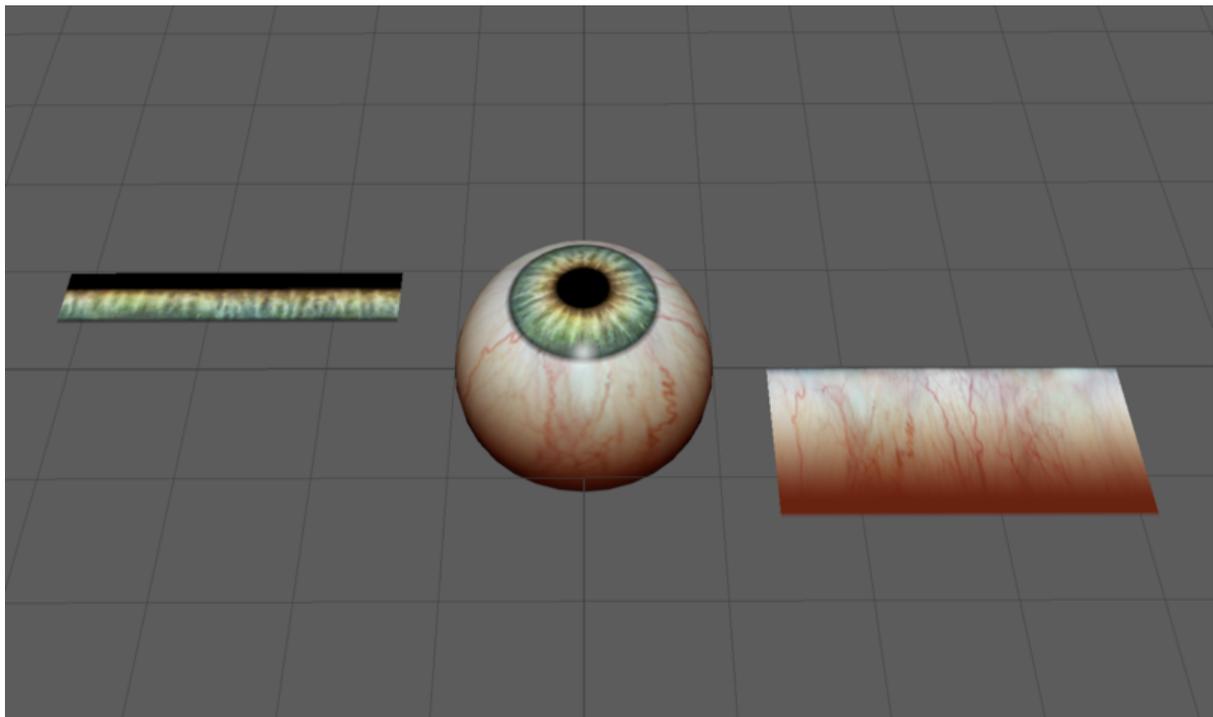




CustomEyesation: DIY Monster M4SK Graphics

Created by John Park



<https://learn.adafruit.com/customeyesation-diy-monster-m4sk-graphics>

Last updated on 2024-06-03 02:52:52 PM EDT

Table of Contents

Overview	3
<hr/>	
• Parts	
Texture Mapping Fundamentals	5
<hr/>	
• Quickstart	
• Customization Basics	
• Preparing Graphics	
• Editing Texture Maps	
• Editing Texture Maps	
• Sclera De-grossing	
• Texture Mapping	
• Projecting Texture Space to 3D Space	
• Two Maps	
• Texture Distortion	
• From 3D to 2D	
• From 2D to 3D	
Create Texture Maps	15
<hr/>	
• Practical Pre-distortion	
• Spike Eyes Example	

Overview



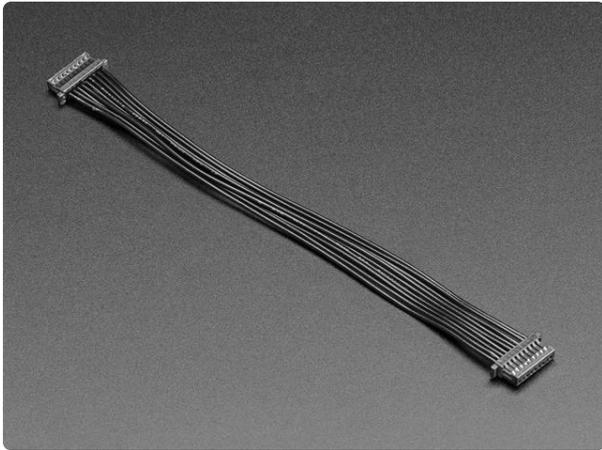
The Adafruit MONSTER M4SK is our most user friendly eyes project to date! This guide offers practical tips for creating your own custom texture map graphics to use on the iris and sclera for your bespoke eyeballs!

Parts



[Adafruit MONSTER M4SK - DIY Electronic Eyes Mask](https://www.adafruit.com/product/4343)

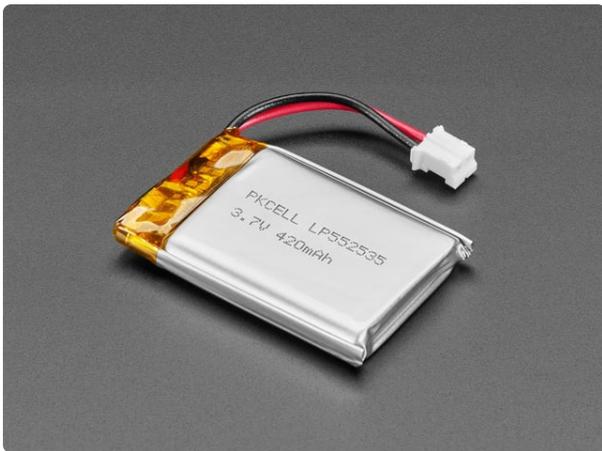
Peep dis! Have you always wanted to have another pair of eyes on the back of your head? Or outfit your costume with big beautiful orbs? The MONSTER M4SK <https://www.adafruit.com/product/4343>



JST SH 9-Pin Cable - 100mm long

This 9-pin cable is just about 100mm / 4" long and fitted with JST-SH female connectors on both ends. Compared with chunkier JST-PH these are 1mm pitch instead of 2mm,...

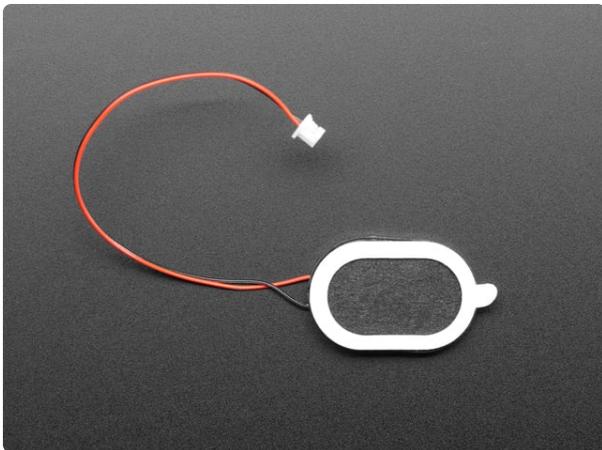
<https://www.adafruit.com/product/4350>



Lithium Ion Polymer Battery with Short Cable - 3.7V 420mAh

Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...

<https://www.adafruit.com/product/4236>



Mini Oval Speaker - 8 Ohm 1 Watt

Hear the good news! This wee speaker is a great addition to any audio project where you need 8 ohm impedance and 1W or less of power. We particularly like...

<https://www.adafruit.com/product/3923>

Either 2 of the glass lenses



Convex Glass Lens with Edge - 40mm Diameter

The eyes have it! Add this little lens to make a big expression with our Spooky Eyes demo for microcontrollers or Raspberry Pi. These are gorgeous glass (not plastic/acrylic!)

<https://www.adafruit.com/product/3853>

Or plastic lenses



Convex Plastic Lens with Edge

The eyes have it! Add this little lens to make a big expression with our Spooky Eyes demo for microcontrollers or Raspberry Pi. These are plastic lenses, with brilliant clarity and a...

<https://www.adafruit.com/product/3917>

Texture Mapping Fundamentals

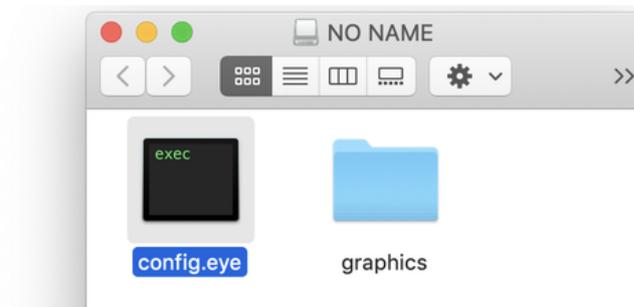
Quickstart

You should be familiar with the basic setup of the M4 Eyes project based on the most excellent [Adafruit MONSTER M4SK guide Quickstart \(https://adafru.it/FDD\)](https://adafru.it/FDD). Before you proceed with making your own texture maps, be sure you've got the [M4Eyes.UF2 \(https://adafru.it/FDD\)](https://adafru.it/FDD) working properly using the default [Hazel eyes graphics \(https://adafru.it/FDD\)](https://adafru.it/FDD).



Customization Basics

You will also need to be familiar with the how the eye system works -- check out [this Customization Basics page \(https://adafru.it/FDE\)](https://adafru.it/FDE) next. Make sure you're comfortable editing and saving the **config.eye** JSON file onto your M4SK and resetting the board to invoke your changes.



Preparing Graphics

Another important section to absorb from the MONSTER M4SK guide (are you noticing a trend here? It's a very important guide!) is on [preparing graphics \(https://adafru.it/FDF\)](https://adafru.it/FDF). This is important so that we are all on the same page with regard to eye anatomy and terminology.

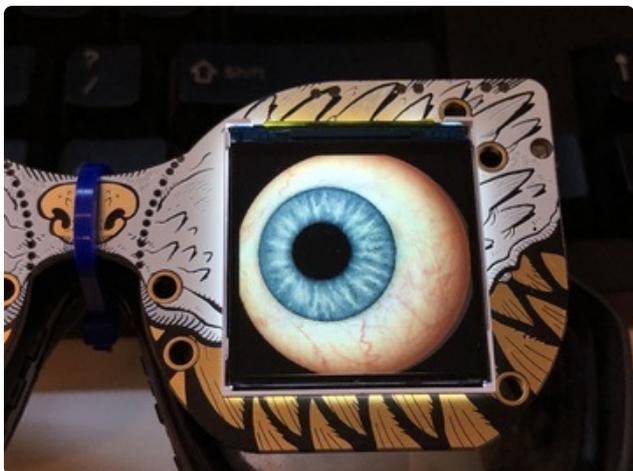
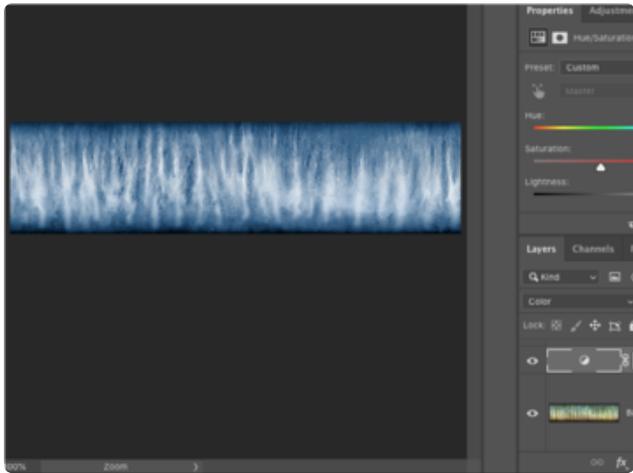
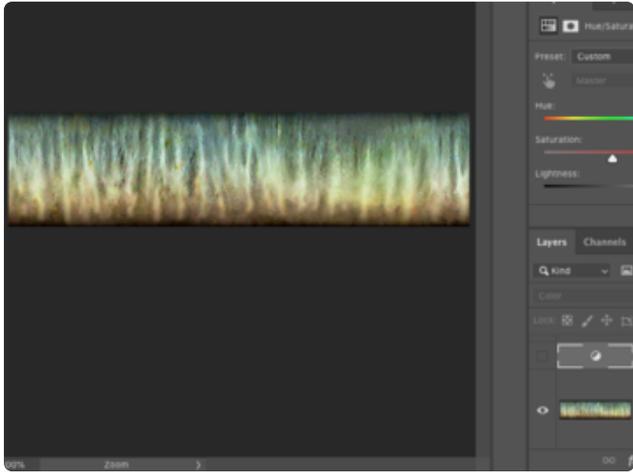
The most critical bit being the two texture maps we'll be using, how they work, and the image file formats/sizes to use.



Editing Texture Maps

Perhaps the simplest way to begin customizing the eye is to edit the colors of the example maps from the Hazel eyes graphics folder.

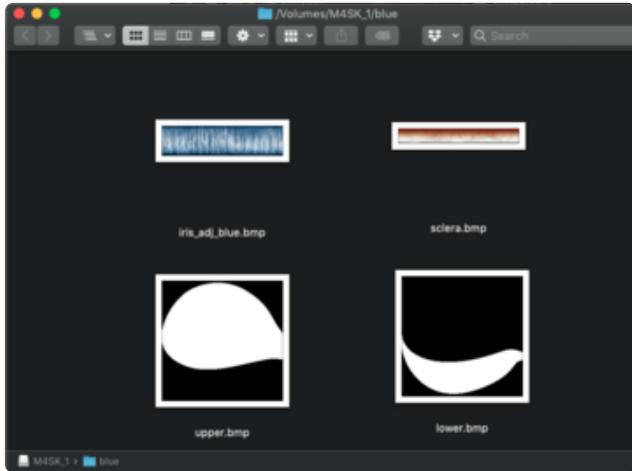
This way, you don't need to worry about image sizes and ratios, or the seam where the rectangular map ends meet when they're warped into the eyeball's spherical shape. You can simply open the maps into your image editor of choice, such as Photoshop, Preview.app, or GIMP, and then use the hue adjustment or colorization tools.



Editing Texture Maps

Perhaps the simplest way to begin customizing the eye is to edit the colors of the example maps from the Hazel eyes graphics folder.

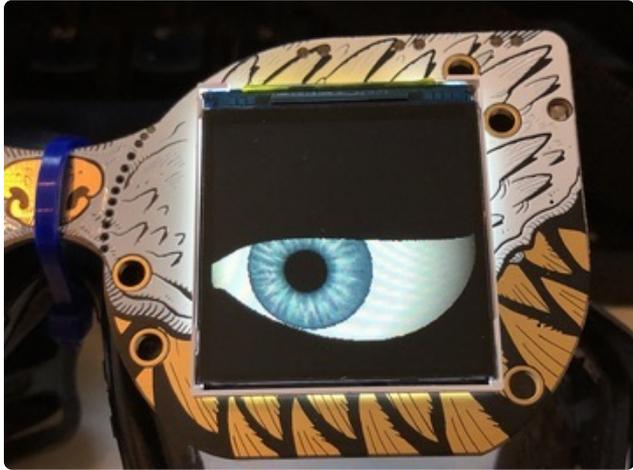
This way, you don't need to worry about image sizes and ratios, or the seam where the rectangular map ends meet when they're warped into the eyeball's spherical shape. You can simply open the maps into your image editor of choice, such as Photoshop, Preview.app, or GIMP, and then use the hue adjustment or colorization tools.



Above you'll find the color adjusted iris map I created. One nicely organized way to use it is to create a new folder on the MONSTER M4SK called **blue** that contains all of the same maps as the **hazel** folder. Then, download the **iris_adj_blue.bmp** map into that folder and remove the **hazel iris.bmp**.

You'll then edit the texture map paths in **config.eye** to match, like this:

```
{
  "boopThreshold" : 17500, // lower = more sensitive
  "eyeRadius"    : 125,
  "eyelidIndex"  : "0x00", // From table: learn.adafruit.com/assets/61921
  "pupilColor"   : [ 0, 0, 0 ],
  "backColor"    : [ 140, 40, 20 ],
  "irisTexture"  : "blue/iris_adj_blue.bmp",
  "scleraTexture": "blue/sclera.bmp",
  "upperEyelid"  : "blue/upper.bmp",
  "lowerEyelid" : "blue/lower.bmp",
  "left" : {
  },
  "right" : {
  }
}
```



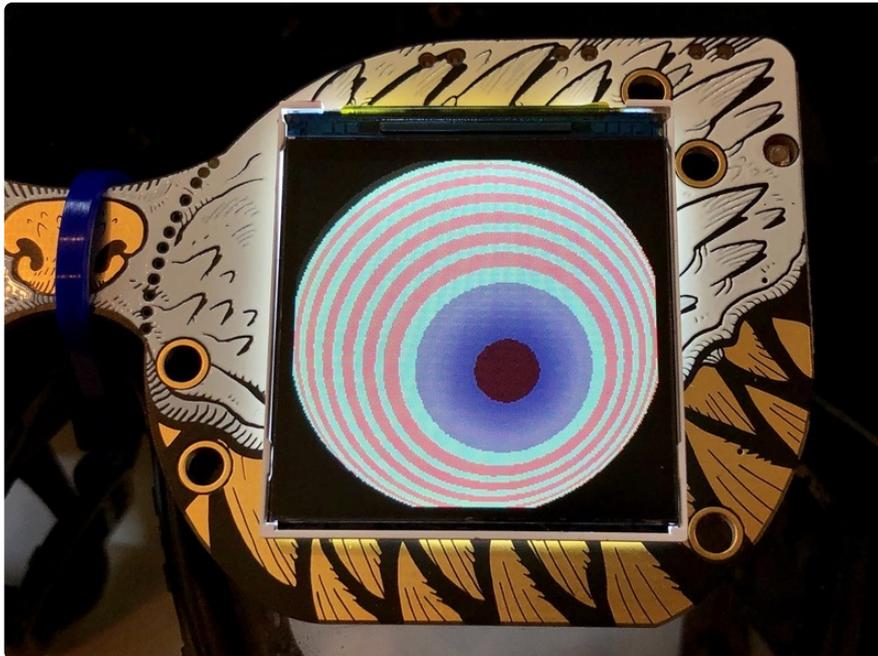
Sclera De-grossing

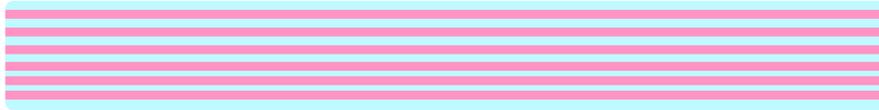
The default sclera map is fantastic -- it's got just the right amount of squiggly veins and reddish cast! If you're going for a more stylized look, you may want to tone down the veins, and maybe even add in a gradient from white to gray as the orb recedes into the socket, as a sort of pronounced fake shadowing or ambient occlusion.

Again, you can download the map below, add it to your MONSTER M4SK's **blue** folder, and then update the **config.eye** to use the **sclera_adj_white.bmp** file.



The sky's the limit when it comes to adjusting the iris and sclera maps. A nice way to start creating original maps is to keep them simple -- either solid colors or gradients. Here's a cartoonish, bonkers look, with a non-black pupil color dialed into the **config.eye** setting for good measure!





Fun Fact: the two maps shown above can actually be saved as only one pixel wide! Since the only variance is vertical, the width doesn't matter! Keep this in mind later when creating maps, it's a great way to save memory.

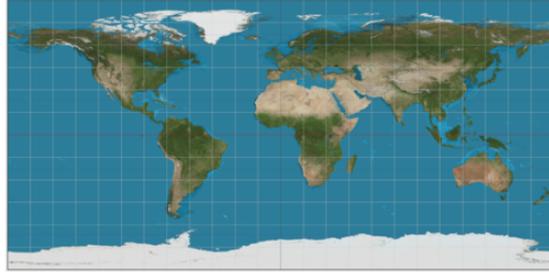


Texture Mapping

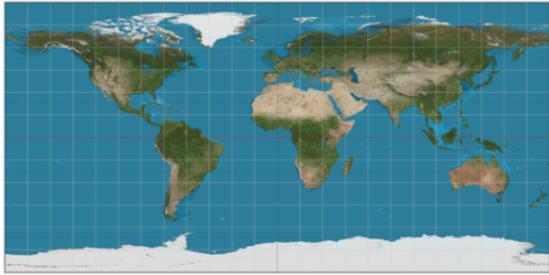
So far we've colorized existing maps, and made simple gradients. But what about more complicated designs? Before we create those, we need to learn about 2D and 3D texture space.

Projecting Texture Space to 3D Space

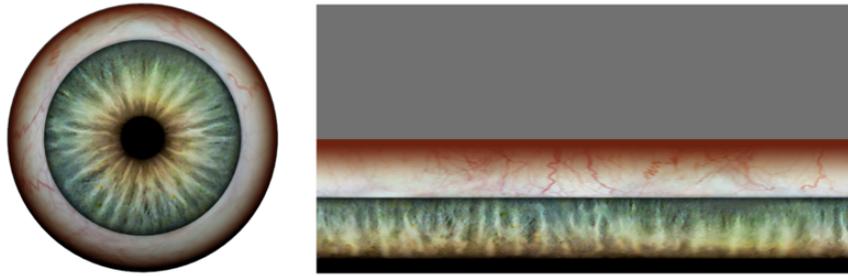
You'll use flat (2D), rectangular images called texture maps to be displayed in on the spherical eyes in 3D space. You can think of it like the familiar topic of unwrapping the spherical image of the Earth -- a 3D globe -- to create a flat map or poster, which is akin to a 2D texture map.



In fact, if we look at the south pole of the globe, we can see how the long, flat white section of the map projection wraps around the sphere's pole.



To take this a step further, here's what happens if we replace the Earth texture map with a single texture map of the pupil, iris, and sclera.

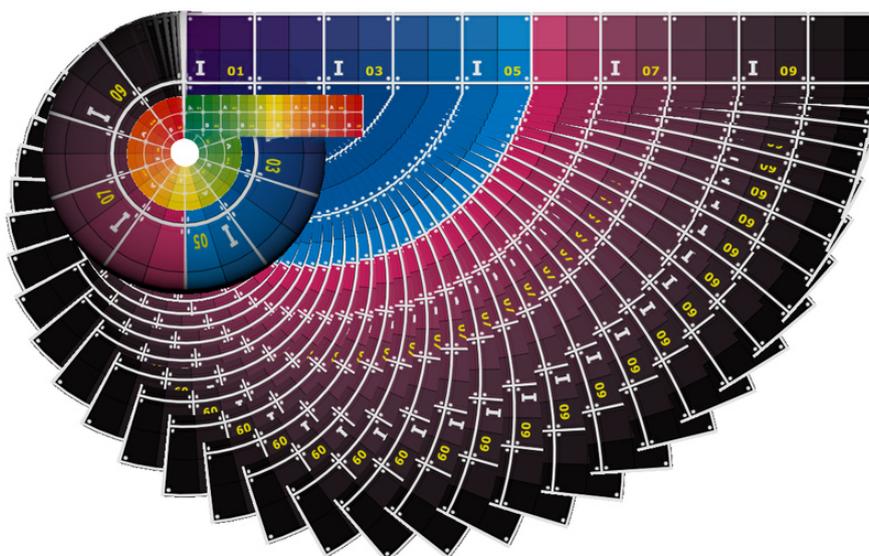


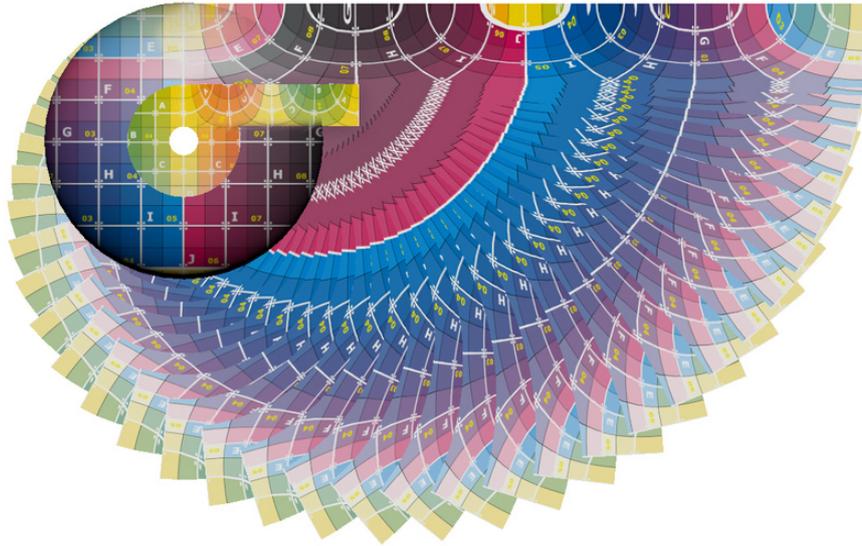
Two Maps

Rather than a single sphere, we can think of the M4 Eye as a set of four shells -- pupil, iris, sclera, and outer edge. The pupil and edge shells don't use texture maps, they're just solid colors, so there are really two maps we need to create for the iris and sclera.

Texture Distortion

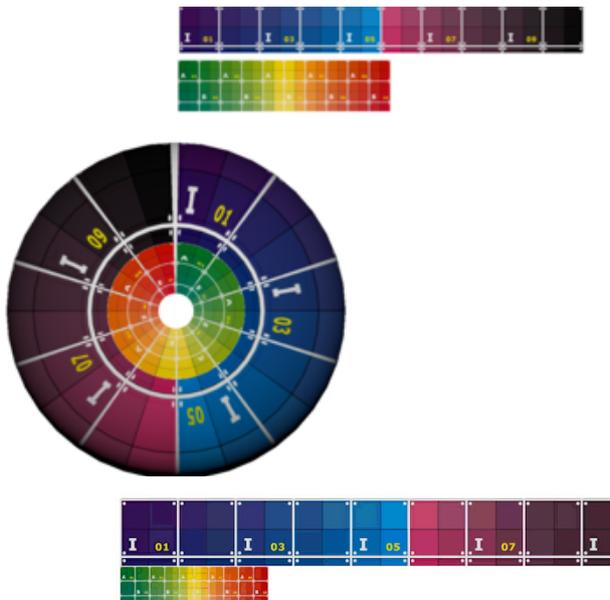
As you may remember from school, whenever we create a map projection from 3D space to 2D space, or vice versa, there will be some amount of distortion to the image. This is what makes tiny little things such as Greenland and Antarctica so huge on a Mercator projection map of the Earth!





From 3D to 2D

Just as in the Earth example, if we begin with an image that looks correct wrapped onto a 3D object, it will be distorted when unwrapped into 2D.



From 2D to 3D

Here, we can see the distortion in effect as we wrap a pair of grid textures from 2D to 3D.

Note, these texture maps were created to look correct in 2D space, so they appear distorted in 3D space.

Create Texture Maps

Practical Pre-distortion

The question you may now be asking is, how difficult is this? Honestly, it's not too bad! First of all, there are plenty of images that will look just fine if you create them in flat, 2D space and then let them get warped around the sphere of the eye. In fact, the realistic human iris can be created with even vertical striations in the 2D map and it looks just fine when it becomes distorted by the 3D re-mapping. Since it is an organic, semi-squiggly pattern, we're pretty forgiving of its minor distortions when looking at it.

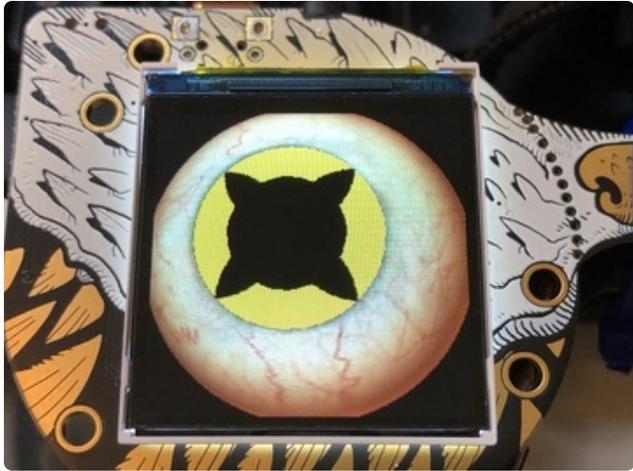
The greater difficulty comes when trying to do more mechanical, man-made patterns, such as mechanical cyborg eyes, alien writing patterns, sci-fi glyphs, camera-style aperture blades and so on. If we create those patterns in 2D, we won't get the results we're looking for after the transform to 3D.

Spike Eyes Example

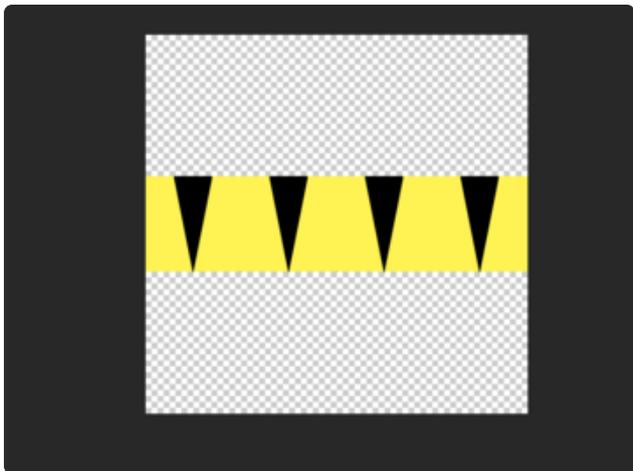
Here's an example -- lets say we want to create four triangular spikes in the iris.
Here's a pattern that makes sense when created in a flat, rectangular texture map:



However, when the map is warped to the 3D space of the eye, they become rounded spikes. They look really cool, but what if we are really intent on having straight sided iris spikes?



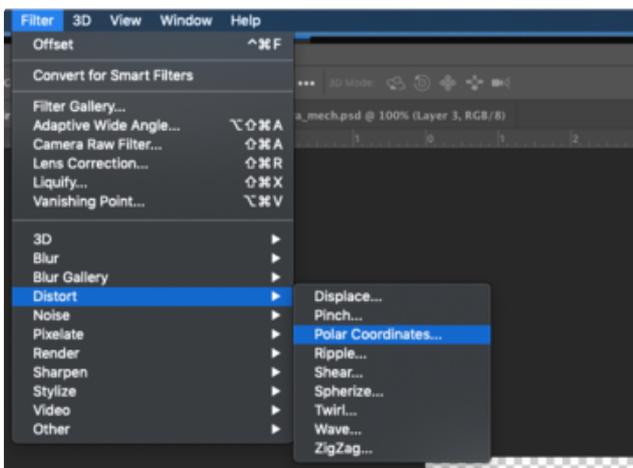
One solution is warp the texture map into a circular shape, fix the triangles to once again have straight sides, and then un-warp it back into a rectangle. Let's go over it step-by-step using Photoshop. (You should be able to replicate these steps in the free and open source GIMP software as well).



Step 1. Rotate the rectangular texture map 180 degrees. This is necessary due to the direction in which Photoshop's polar coordinates distortion works.

Step 2. Set the canvas height equal to the texture map's width. In this case, 512 x 512 pixels. This will give the distortion room to warp the rectangle into a circle.

Step 3. Use the **Filter > Distort > Polar Coordinates...** menu item to bring up the **Polar Coordinates** window.



Step 4. In the Polar Coordinates window, set the option to **Rectangular to Polar**, then hit **OK**.

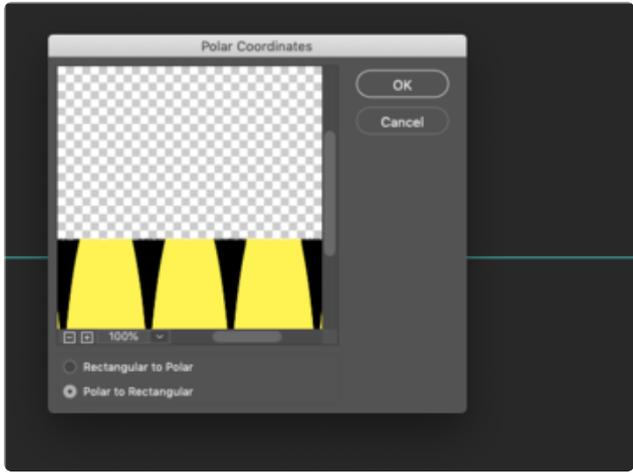
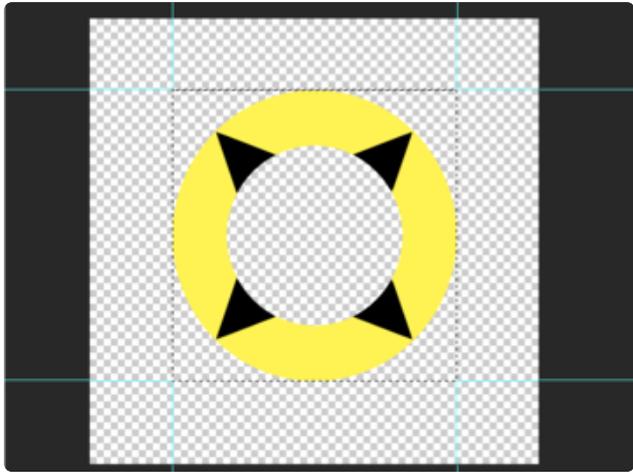
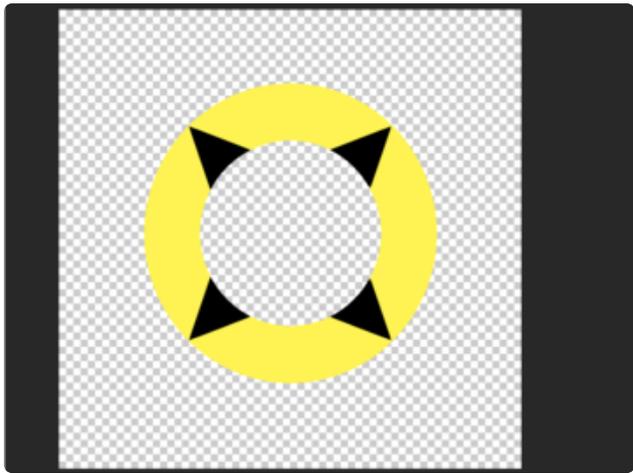
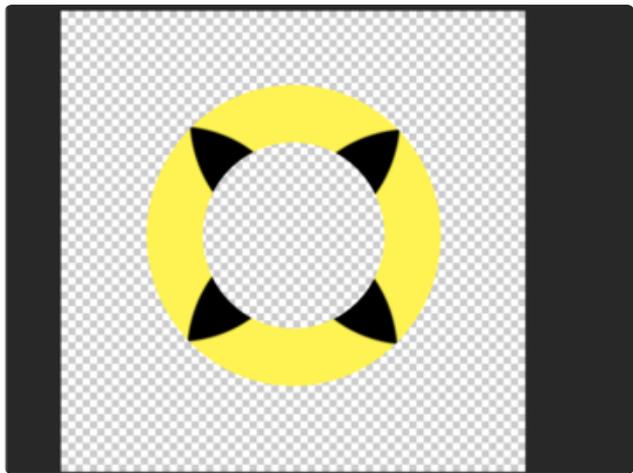
Step 5. The result of the distortion is the same bulging effect we saw on the M4 Eyes. Let's fix that! You can use the polygonal lasso selection tool to mask the nice, straight sided spikes and then paint out the bulging sections with yellow to match the background.

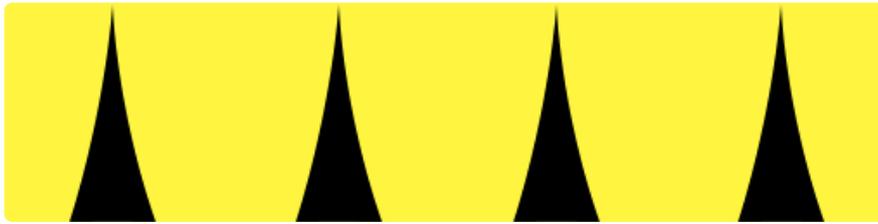
Step 6. In order to get a good result when un-warping the circle, make a rectangular marquee selection around it, keeping the

selection tight to the texture. You can see here that I used guide lines to assist.

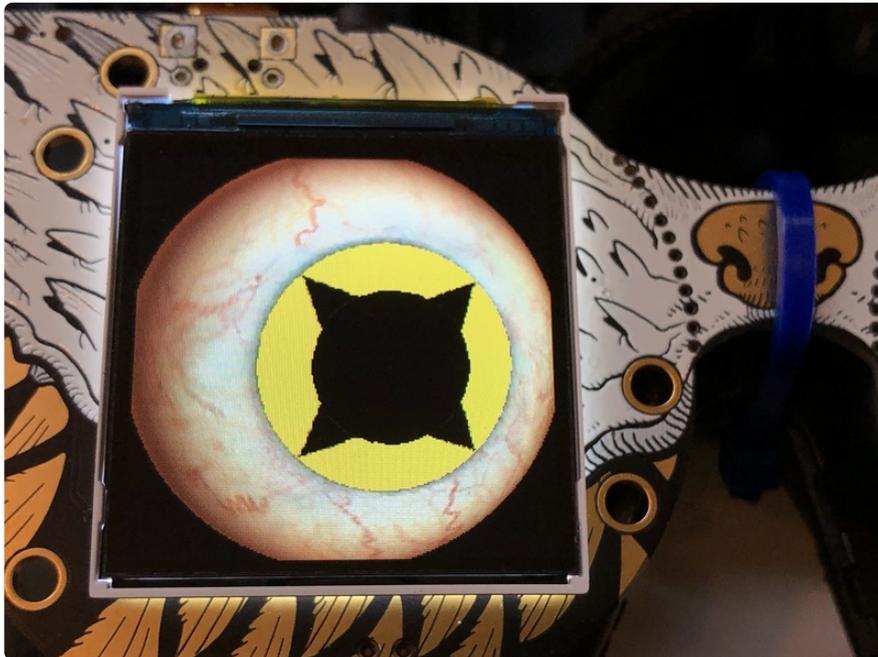
Step 7. Use the **Filter > Distort > Polar Coordinates...** menu to again bring up the window, but this time select the **Polar to Rectangular** option. Hit **OK** to turn the circle back into a rectangle.

Step 8. Now, you can rotate the image 180 degrees to get it back to its starting point, and then crop the canvas back to the original 512 x 128 pixels.





Notice how the spikes all now curve inward which will make them appear straight-sided when we apply it to the 3D eyes!



With these techniques, we can now create highly detailed texture maps that aren't distorted when applied to the M4 Eyes!

Here's a totally METAL set to try out. Have fun making your own weird and wonderful peepers!

