

GAMECAVE EFFECTS ENGINE 3.X
ALPHA MAPPING TUTORIAL
WRITTEN BY RHYS ANDREWS



TABLE OF CONTENTS (CLICK TO VIEW CHAPTER)

Introductory 2

An Example..... 3

The Functions 4

 Sprite Alpha Masking 4

 Background Alpha Masking 4

Alpha Masking with Adobe Photoshop..... 5

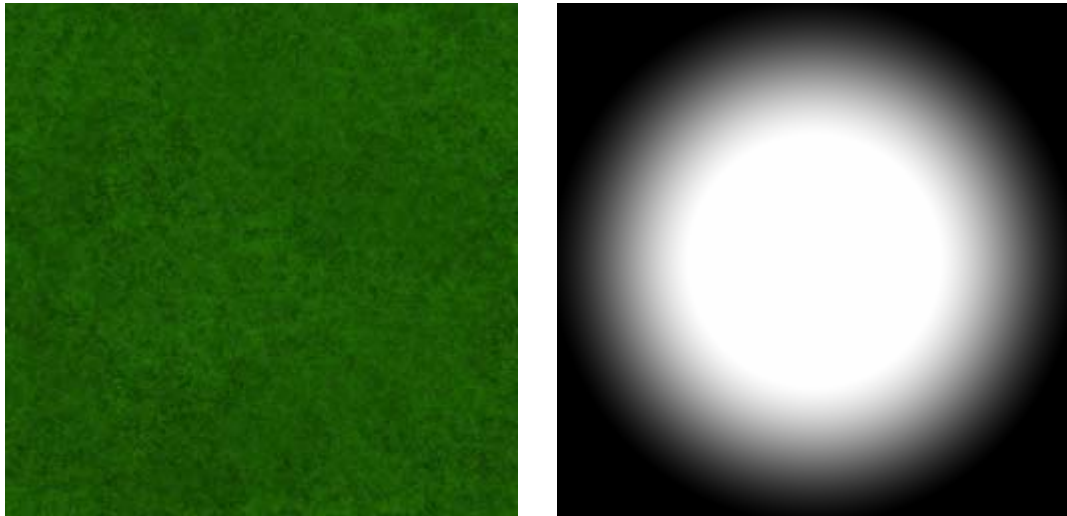
Final Revision..... 9

INTRODUCTORY

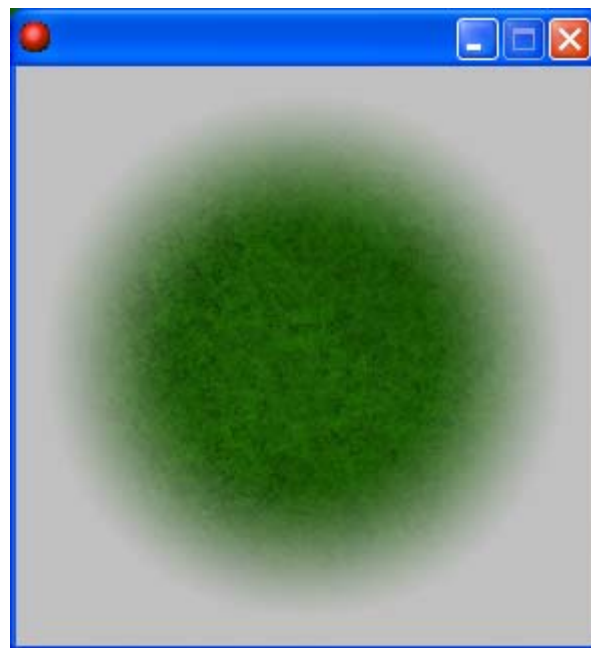
Alpha Mapping is a method of applying alpha masks to sprites; giving them accurately defined soft-edges (for instance), and is commonly used in all commercial software. This tutorial, though short, will teach you the 2 important functions needed to apply “alpha masks” to sprites and backgrounds, and how to make / understand alpha masks.

AN EXAMPLE

The best way to explain what *Alpha Mapping* actually is, is providing an example. Here, we have the **sprite**, and the **alpha mask**:



The **sprite** is a simple 256X256 image of grass. The **alpha mask** is a “radial gradient”. Now, let’s see what happens when we set the alpha mask for the *sprite* to this alpha mask on the right:



Quite interestingly, all the white in the alpha mask has become textured with the grass. Or, more literally, the transparency of each pixel of the **sprite** has been determined by the brightness of the same pixel in the **alpha mask**. Like the GameMaker Manual states, “The alpha (transparency) values in the sprite uses the hue values in the alpha mask”.

THE FUNCTIONS

So, now that you understand how alpha mapping works, let's understand the functions you need to use to set alpha masks to sprites.

SPRITE ALPHA MASKING

```
sprite_set_alpha_from_sprite(ind,spr);
```

This function sets a sprite to an alpha mask. Alpha masks are also sprites, so we're setting a "sprite to a sprite". The **sprite** is known as `ind` (argument0), and the **alpha mask** is known as `spr` (argument1).

Open up your copy of GameMaker 6.1 (registered), and draw a 100X100 sprite (whatever you like, just make it 100X100). Then, create *another* sprite at 100X100, and draw something else, this time only using greyscale values (black, greys, white). Finally, add a new object, set its sprite to the first sprite you created, and in its create event, input this code:

```
sprite_set_alpha_from_sprite(sprite0,sprite1);
```

Stick the object in a blank room, and play the game. There. You've set the alpha mask of `sprite0` to the hue values in `sprite1`. Let's lay out a few rules for this function. The function only needs to be used *once*, and until the end of the game, that sprite uses the determined alpha mask. So make sure that you do not stick the function in a place where it's going to be executed over and over. And also make sure that the sprite you're setting the alpha mask for, is only ever used when it needs that alpha mask. For instance, you might want that grass to have an alpha mask in one level, but be pure opaque (fully visible) in another. In such a case, you'll need two copies of the sprite. Saying that, remember that an alpha mask can be used for many sprites.

Secondly, it's a good idea to name your alpha masks like this:

Sprite name: `sprBoy`

Alpha mask name: `sprBoy_alpha`

This means, you know that the alpha mask for `sprBoy` is `sprBoy_alpha`. If, however, you have many sprites that all use one alpha mask, you could name the alpha mask a category that fits all the sprites using it... for instance:

Sprite name 1: `sprGrass`

Sprite name 2: `sprSand`

Sprite name 3: `sprSnow`

Alpha mask name: `sprTerrain_alpha`

BACKGROUND ALPHA MASKING

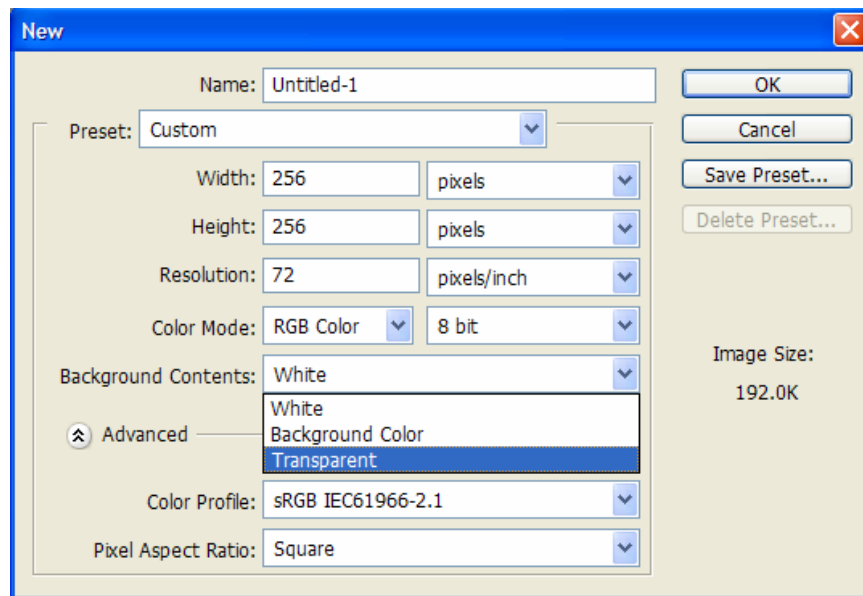
```
background_set_alpha_from_background(ind,back);
```

This function works just like the previous function (including all rules and arguments), however, the function is looking for **backgrounds**, not **sprites**. In other words, you need to have an alpha-mask as a background, and a background to apply the alpha mask to.

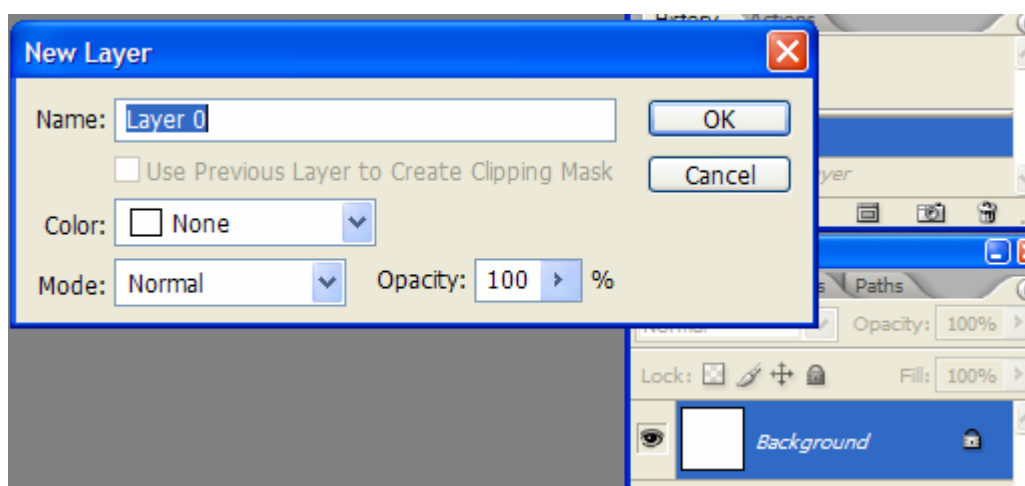
ALPHA MASKING WITH ADOBE PHOTOSHOP

Adobe Photoshop is a great tool, if not the greatest, for graphic design and sprite-work. Fortunately, with Photoshop, there's a very easy method for creating an alpha mask for your photo-shop document. This chapter is based on teaching you how to create an alpha mask for your work with Adobe Photoshop 7.0 or CS2.

Firstly, when you start your new sprite (file > new); make sure you set your background contents to *Transparent*.



This is to ensure that all the empty space around the sprite is “black” in the alpha mask, meaning, invisible. If you’ve already created your work without checking *Transparent* for your background contents, you can, alternatively, double click on the **background** layer, which converts it to an unlocked layer. Then, simply delete the layer (right-click on the highlighted layer on the layer list, then click ‘delete layer’), or if it’s the only layer in the layer-list (meaning, you cannot delete it), click the “eye” next to the layer to make it invisible. You can then delete it later.



Now you have a transparent canvas. If you were to create an alpha mask of this transparent canvas, it would simply be pure black. So now, you can start working on your artwork. It is according to the transparency of each pixel in your artwork that the alpha mask will be generated to.

When you have finished your work, save the project as a PSD (photoshop document) to prevent data loss. Before we mask the sprite, let me show you how *my* work is going to turn out as an alpha mask:



To do this, we need to first “*rasterize*” all the text-layers. This means you won’t be able to edit those text-layers again as text; so make sure all the text is right. Then, right click on the layer, and select “**Rasterize Type**”, or “**Rasterize Layer**”. Now, go “*Image > Calculations...*” If Calculations is blocked out, it’s clear that you have not rasterized all layers, or possibly your background is not transparent. When you finally get into the calculations window, pick each option as follows:

(Source 1)

Layer: Merged

Channel: Transparency

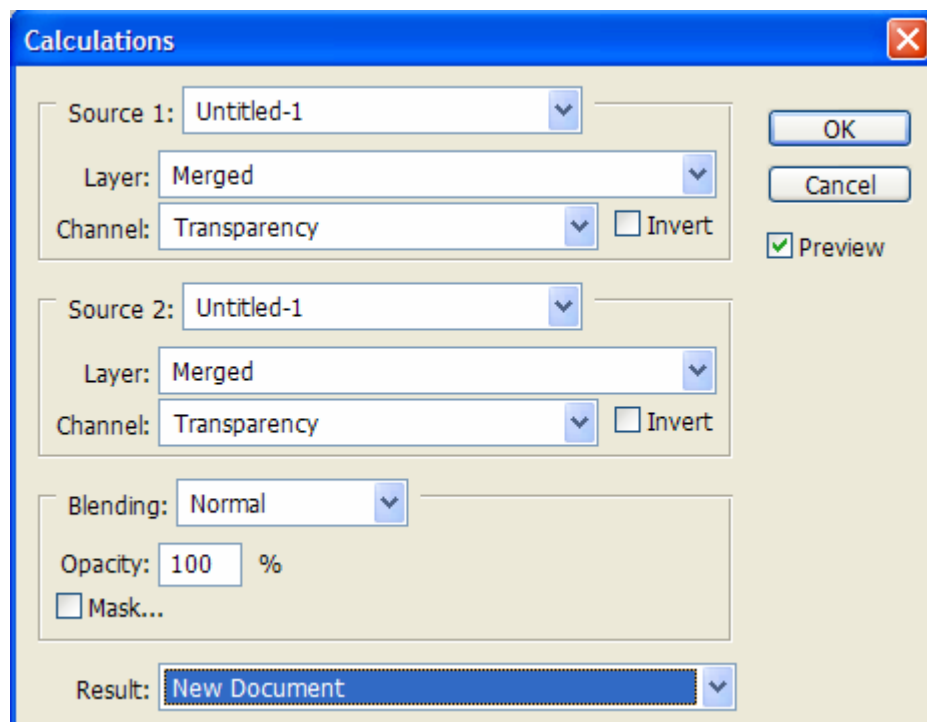
(Source 2)

Layer: Merged

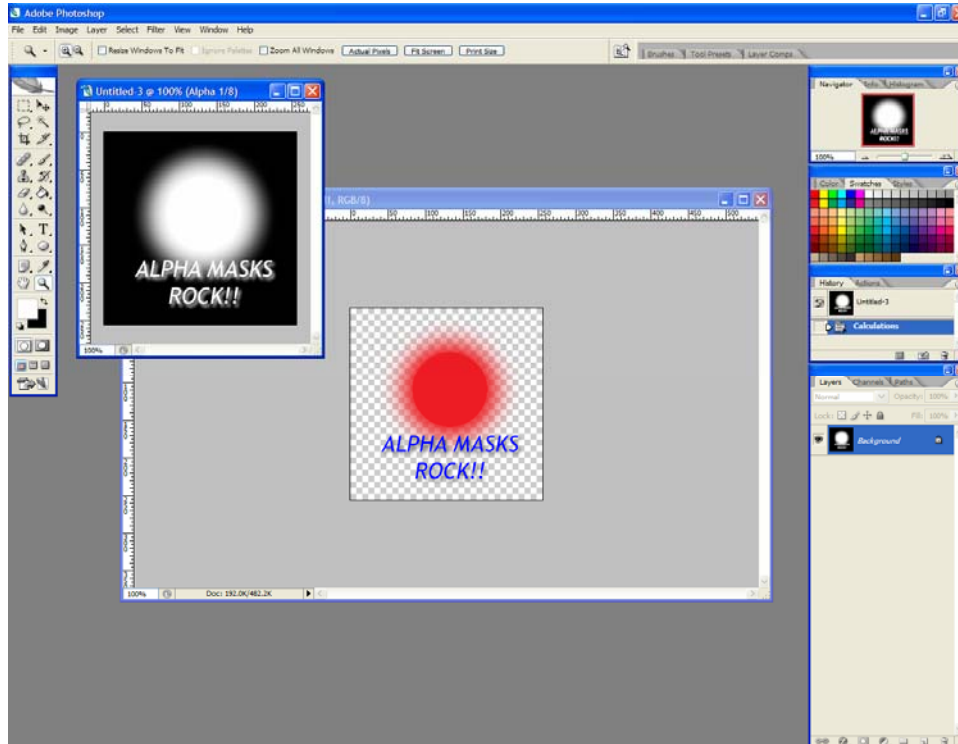
Channel: Transparency

Blending: Normal

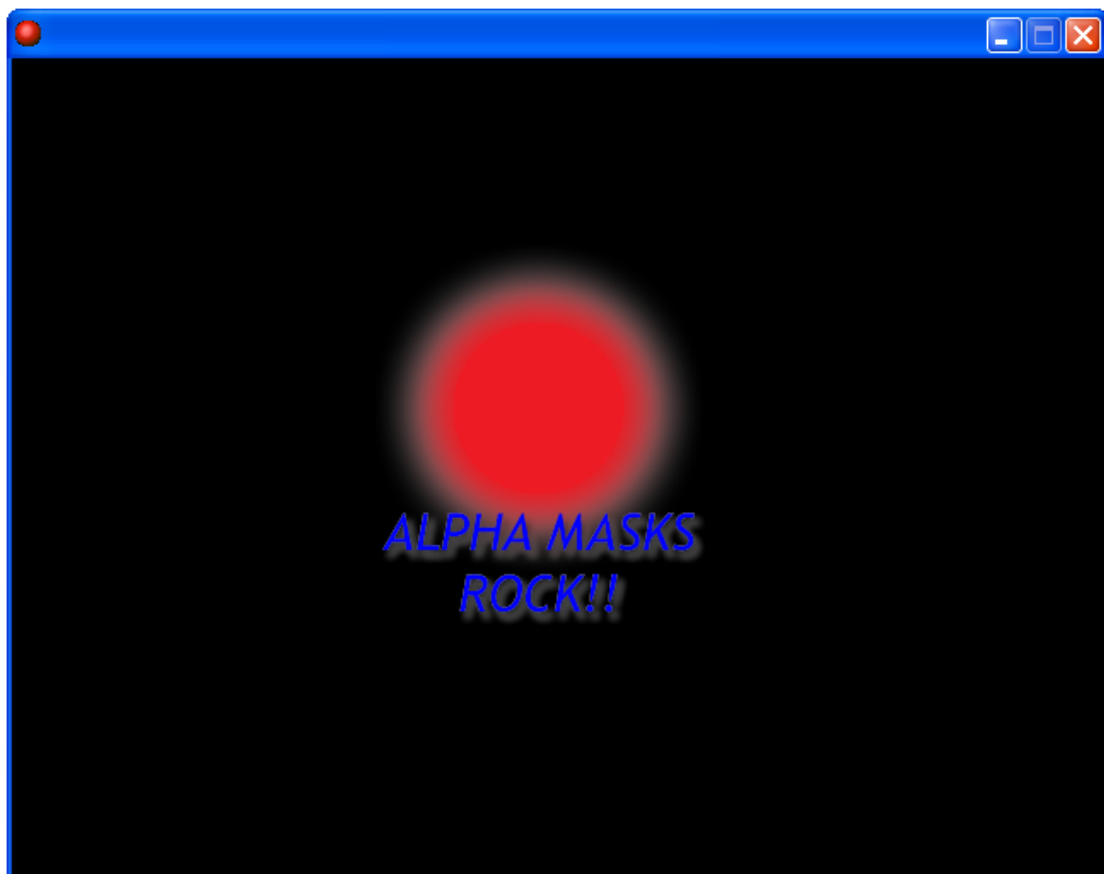
Result: New Document



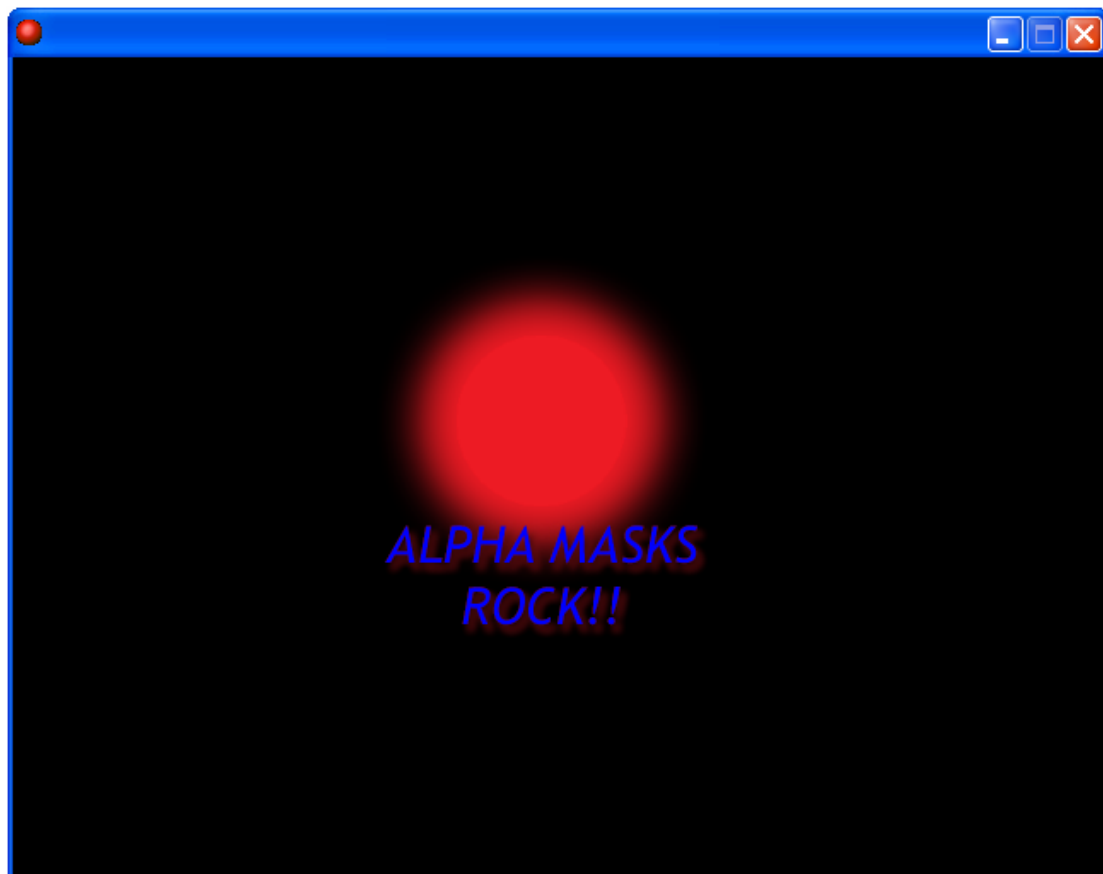
And click “Ok”. A new document will open, with your alpha mask. Select all (ctrl a) and copy (ctrl c). Then, paste in your GameMaker Sprite Editor window, and enjoy!



As a final tip, sometimes it's good to restore a background colour to your original work after the alpha mask is created. Because GameMaker takes transparency from a PNG or GIF as white (and if you save your original sprite as a JPG, photoshop replaces transparent pixels with white anyway), and so, you may end up with something such as this in GameMaker:



As you can see, this has a bit of white in the half-faded-out areas of the text's shadow and the gradient edges of the circle. Where as, if you placed in a red background into your document after the alpha mask was created, you would get this:



FINAL REVISION

By the end of this tutorial, you should have hopefully learnt how alpha masks work, and, if you're a mad Photoshop user like myself, learnt how to create alpha masks with your Photoshop art-works; for GameMaker or otherwise. Though this can be a fun and easy feature to play around with in GameMaker, it can sometimes have some complicated thoughts needed to be put into it, in order to make some sprites look right with their alpha masks. So go ahead and practice; Alpha masks are great for

- Soft Edges on Text and Images
- Putting Gradients on Sprites
- Blending two types of terrains (such as snow and grass) together (or tiles?)
- Much more.

Enjoy!

- Rhys Andrews