



# Sniper's Paradise

## *Who Loves You Baby!!!*

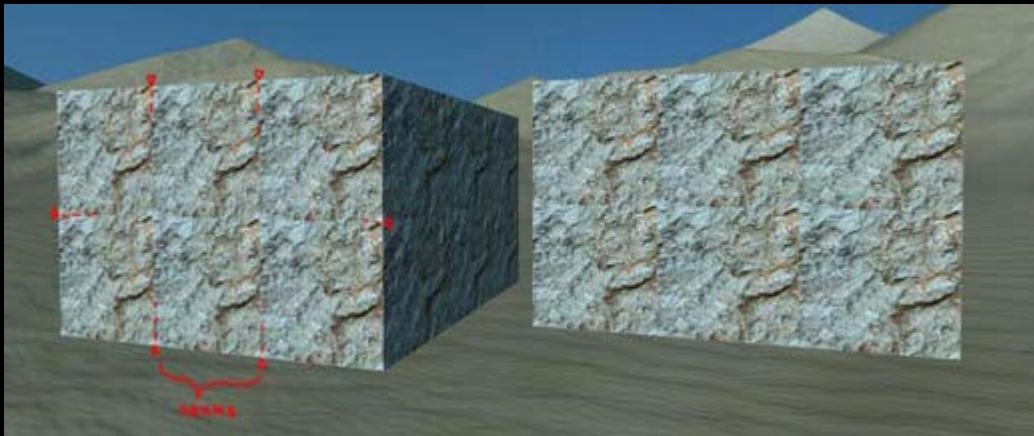
### Sniper's Paradise!

---

#### How to make a seamless tiling texture

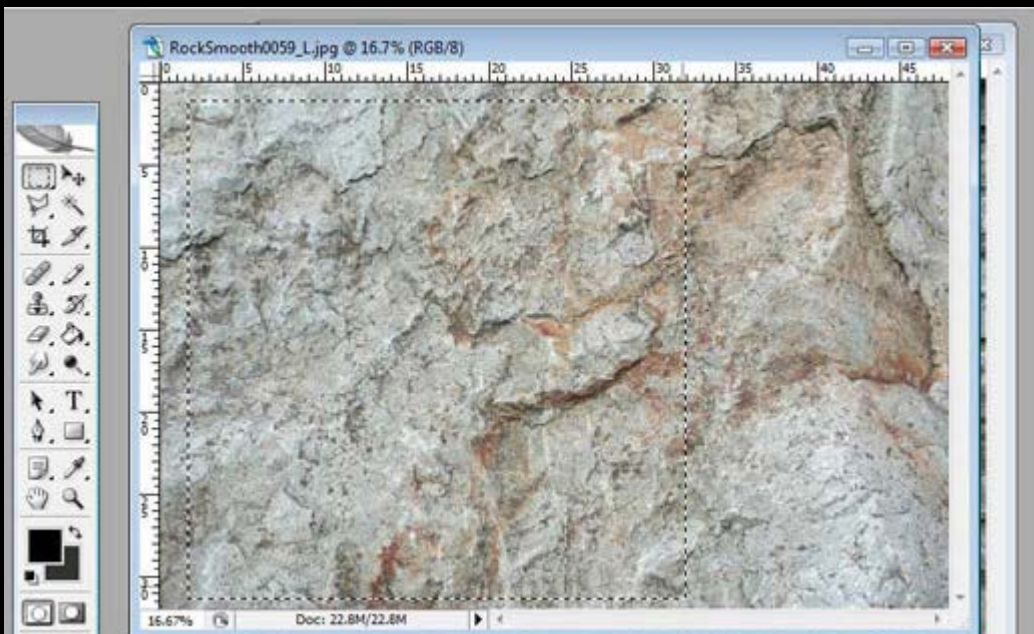
##### What are seams?

Tiling textures are much more efficient than unique textures, which is especially important for large objects. The main cost of this efficiency is that tiling textures can look unrealistic when used incorrectly -- real rocks are not made of repeating tiles. The tiling is most obvious when there are distinct details that repeat, and the worst kind of distinct detail is an edge seam -- a sharp discontinuity between the tiles. Here is an example of a tiling rock texture with seams (highlighted on the left, but still present on the right).

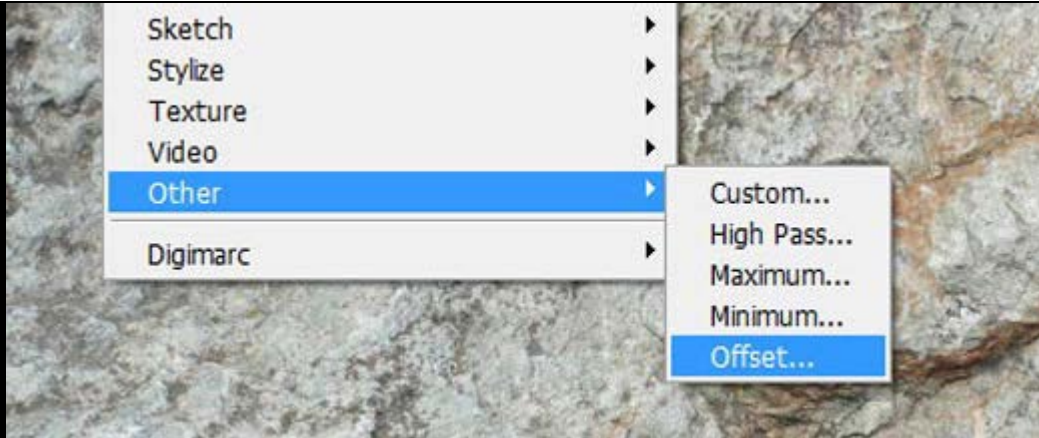


##### How do we remove seams?

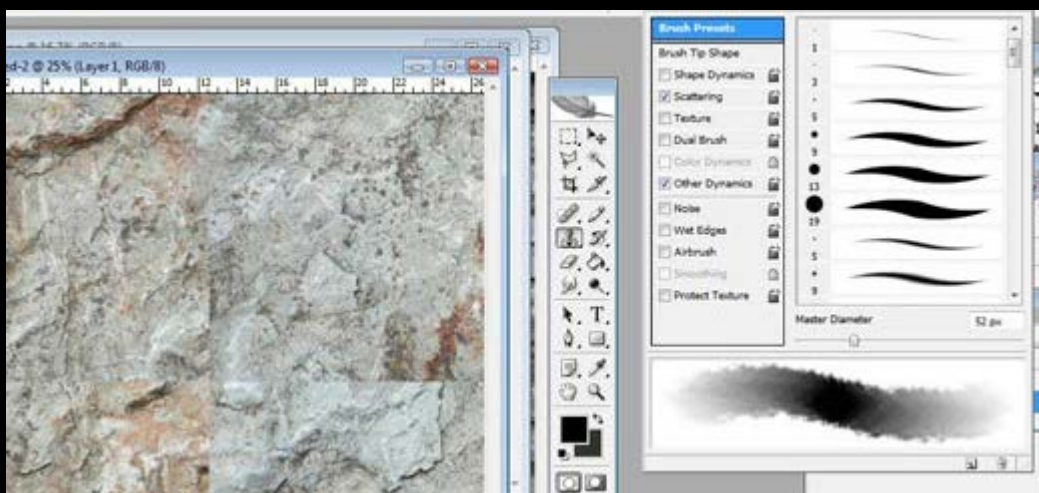
First, let's get our initial texture (with seams). For photosourced textures, let's start by getting a photo from CGTextures.com, and cutting out a square. Pick an area that looks fairly uniform.



Paste that into a new file, and then use the 'offset' filter to expose the seams. The offset filter lets you shift the image over in a way that 'wraps around', so that the pixels that go off the right edge show up on the left.



Now that we've offset the image by half its size in each direction, the seam shows up as a cross through the middle of the image. Now pick a soft brush, and paint out the seams using the clone (rubber stamp) tool. The rubber stamp tool lets you paint pixels from one part of the image onto another part.

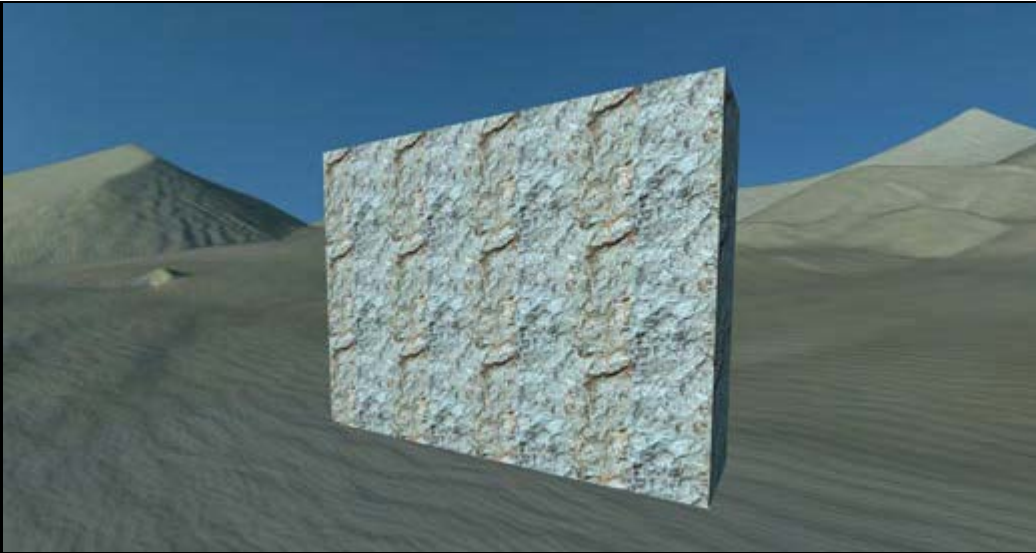


Here is the texture after painting out the seams.



That's really all there is to basic seam removal. Here is the final seamless tiling material in the engine:

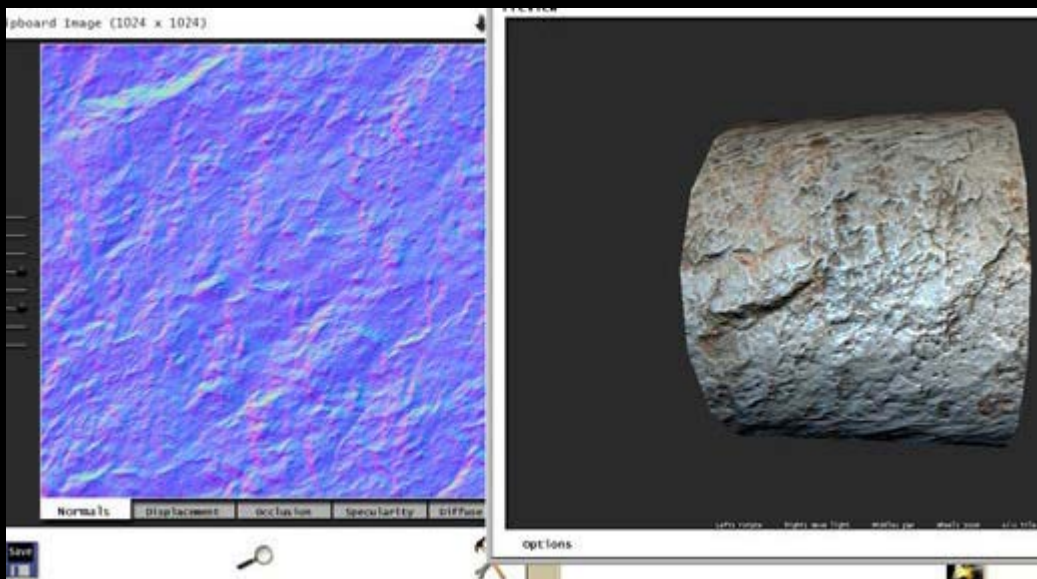




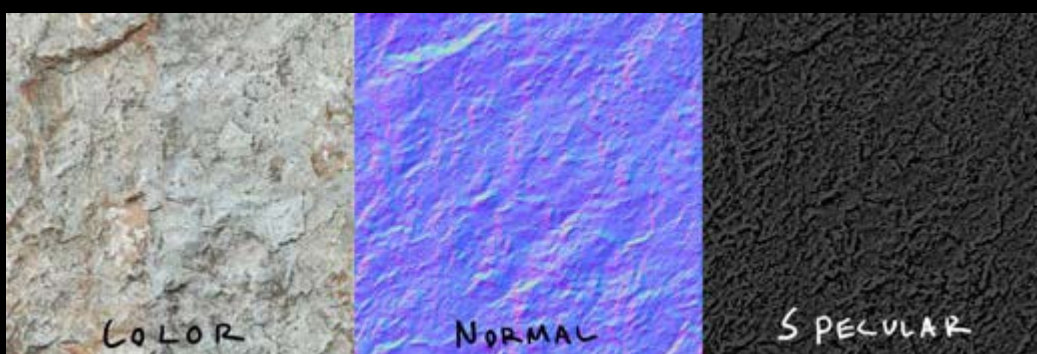
This could still use a lot of work though. Even without seams, the tiling is obvious because of the features and colors in the texture. For example, there is one dark shadow underneath a rock which is obviously repeated over and over -- if we end up using this particular texture in the game I will have to fix that! When making tiling textures, we have to walk a fine line between making it tile invisibly, and making it visually interesting.

### How do we test tiling textures in the Phoenix engine?

There are a few more details you'll have to know to make tiling texture blocks like this in Phoenix (in the Overgrowth alphas). First you will need to add a normal map and specular map to take full advantage of the lighting system. If you want, you can just use the default 'diffusebump.tga' texture and skip making a normal map, but I'll explain how to use normal maps for anyone who is interested. You can make a decent normal map from a color map very easily using CrazyBump. If you don't have CrazyBump there is a free Nvidia plugin that works in Photoshop.

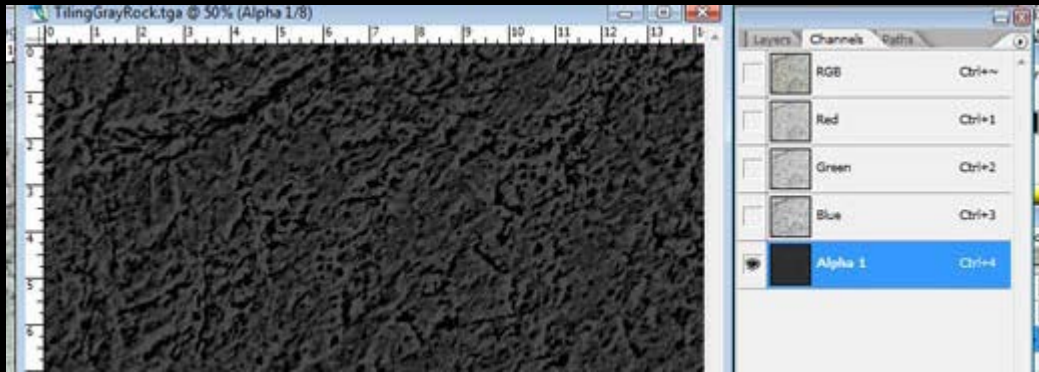


You will also probably want a specular map (showing how shiny the texture is, from black [matte] to white [chrome]). If you don't want your texture to be shiny at all, leave the alpha channel black. To make my specular map I just grabbed the one calculated by CrazyBump, but you can also just use a darker desaturated version of your color map. Here are the color, normal and specular maps for this texture.



To use fewer textures, Phoenix uses the alpha channel of the color map to store the specular

map. To do this, you have to make sure your color map has an alpha channel, and then copy your specular map into it. Here is the 'channels' panel for the color map in photoshop, with the specular in the alpha channel.

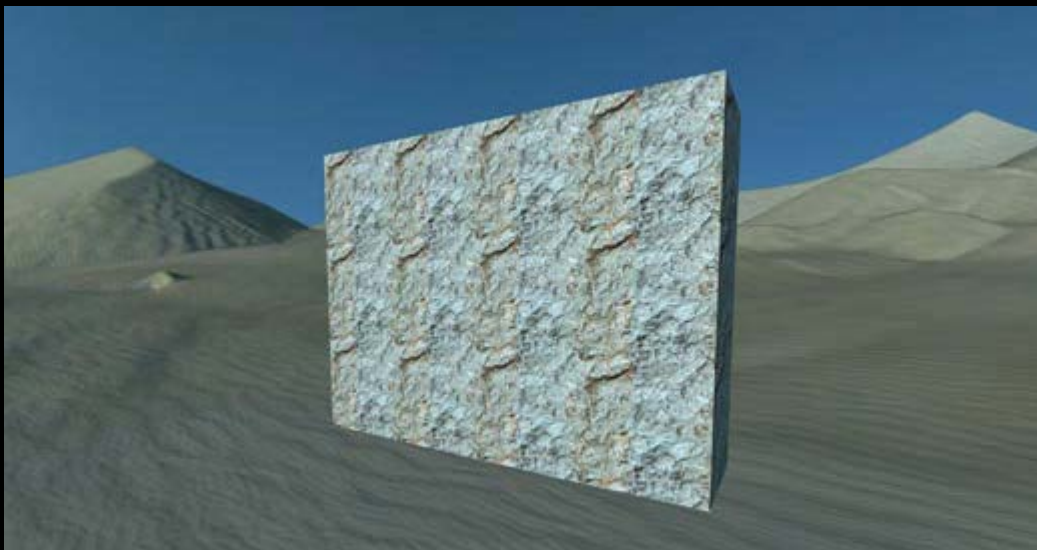


Now you can export both the color/specular map and normal map as .tga files, and create the .xml object file to load them in the game. The easiest way to make this file is to copy an existing block object .xml file and just edit the paths to the textures. Here I copied "blackCatRoad.xml" file and edited the and paths to point to my GrayRock images:

Now I can just load my block in the engine and make walls or floors or whatever I want! My first attempt came out too bright because the specular map was not dark enough -- a bright color map with a bright specular map will just get washed out like this.



I fixed it by just darkening the maps a bit.



Although it seems like a lot of steps, adding materials to the engine is really not too hard.



[Back To Top](#)