**Name:\_\_\_Weston Buck\_\_\_**

**Computer Graphics, Lab 10**

**Due Wed, 10/25**

**(10 pts)**

1. Change the code to use the blueflower.jpg image in /resources.
2. Examine the blueflower.jpg image in /resources. What size is it (in pixels)?

**256x256**

1. What is the approximate pixel size of the rectangle (on screen) that the texture is being mapped to?

**400x400**

1. Given your answers to the above, which is being used, minification or magnification?

**Magnification**

1. Change the code to make the canvas larger (say, 800x800); keep everything else the same. *In terms of what is happening in the fragment shader*, explain the effect on the displayed image.

**The image is getting stretched to fit a bigger size in pixels.**

1. Leaving the canvas at 800x800, set the TEXTURE\_MAG\_FILTER to NEAREST and re-test. Which looks better? Explain the difference in the algorithm (used by the graphics card) between the default value and the NEAREST value.

**Nearest finds the closest pixel to where we are trying to get and uses that color. Linear averages the 4 pixels it hits between and uses that color.**

1. Return your canvas to size 400x400. Modify the code to make the square rotate around the z axis (see RotatingTriangle code in chapter 4). Clearly, the image rotates without being corrupted; are the interpolated numbers passed to texture2D in the shader being rotated? Explain how this is working.

**The points of the texture are still being mapped onto the quad base on the location of the pixel on the quad. So the same texel will always be put on the same pixel on the quad. So the texture itself is not rotating, just the quad.**

1. Modify the code to texture a *triangle* as follows (using the “middle” of the texture).



Turn in: Hardcopy with answers to questions, and demo #8 on Wed.