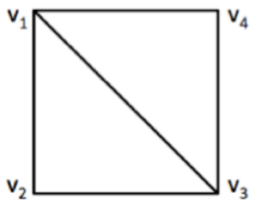
Worksheet VIII

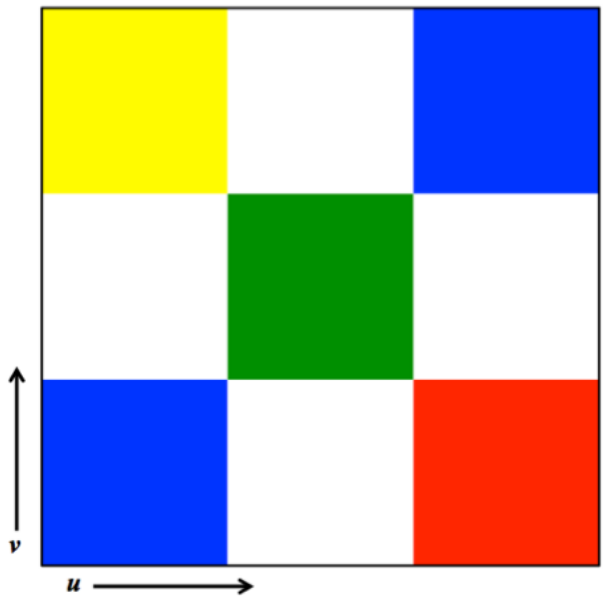
Answer all the problems completely on a separate sheet of paper. Read all the problems closely, and ask if you have any questions on what a problem means. This worksheet is due at the start of class on Mon, Nov 17.

Problem 1 (7 pts)

Below on the left are the contents of an .obj file that defines a ***textured quad***. Remember that `v` represents a vertex-position, `vt` represents a texture-coordinate, and `f` lists the vertex-position / texture-coordinate of each vertex in a triangular face. On the right is a simple, multi-colored texture.

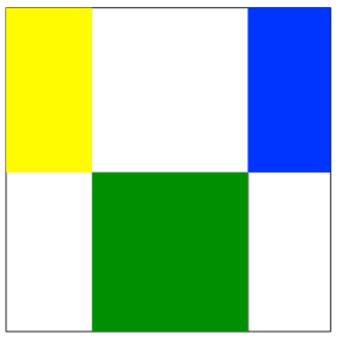
```
v  0.0  1.0  0.0
v  0.0  0.0  0.0
v  1.0  0.0  0.0
v  1.0  1.0  0.0

vt  0.0  0.5
vt  0.0  0.0
vt  1.0  0.0
vt  1.0  0.5

f  1/1  2/2  3/3
f  1/1  3/3  4/4
```





(a) If this texture were mapped to this quad, what color would the point at (0.5, 0.5, 0) in object space—the center of the quad—be? Explain your answer in a sentence or so.

(b) List a replacement set of texture-coordinates for this same .obj file so that the rendered quad has approximately the appearance shown below:



Problem 2 (4 pts)

What is **mipmapping**? Why is it useful?

Problem 3 (6 pts)

Explain how to use **environmental mapping** to render model a chrome teapot in WebGL.

Problem 4 (4 pts)

Consider the following simple **fragment shader**. This shader applies a diffuse lighting model to the given fragment; note the interpolated variables passed in from the vertex shader.

```
precision mediump float;
varying vec4 vColor; //vertex color
varying vec3 vNormal; //vertex normal
varying vec4 vPosition; //vertex position
varying vec2 vTexCoord; //vertex texture coordinate

uniform vec4 uLightDirection;
uniform vec3 uLightColor;
uniform vec3 uAmbientColor;
uniform sampler2D uTexture;

void main() {


    vec3 normal = normalize(vNormal);


    vec3 lightDir = normalize(uLightDirection);


    float diffuse = max(dot(normal, lightDir), 0.0);


    gl_FragColor = vec4(diffuse*uLightColor*vColor.rgb, 1.0);


}
```

Detail (in code!) the modifications to the shader so that it performs **bump mapping** using the given uniform texture.