NOTE: The phong shading here is optional. You don’t need to have a set color for the object you are shading, the important thing is the day vec4.

FRAG SHADER:

varying vec3 N;

varying vec3 v;

uniform float num; //the hour of day

void main (void){

vec3 L = normalize(gl\_LightSource[0].position.xyz - v);

vec3 V = normalize(-v);

vec3 R = normalize(-reflect(L,N));

//vectors

vec4 Ia = gl\_FrontLightProduct[0].ambient;

vec4 Id = gl\_FrontLightProduct[0].diffuse \* max(dot(N,L), 0.0);

vec4 Is = gl\_FrontLightProduct[0].specular \* pow(max(dot(R,V),0.0), 1.0);

//colors!!!

vec4 Ma = vec4(1.0, 0.0, 0.0, 1); //any color

vec4 Ms = vec4(1.0, 1.0, 1.0, 1);

vec4 day;

if(num > 0.0 && num <= 6.){

day = vec4(0.0, 0.0, 0.2, 0.25);

}else if(num > 6. && num <= 12.){

day = vec4(0.7, 0.0, 0.3, 0.25);

}else if(num > 12. && num <= 18.){

day = vec4(1.0, 1.0, 0.0, 0.55);

}else{

day = vec4(0.0, 0.3, 0.2, 0.25);

}

float minus = 1. - day.w;

vec4 Md = vec4((Ma.x \* minus + day.x \* day.w) ,(Ma.y \* minus + day.y \* day.w), (Ma.z \* minus + day.z \* day.w), 1);

gl\_FragColor = Ia \* Ma + Id \* Md + Is \* Ms;

}

VERT SHADER:

varying vec3 N;

uniform float num;

varying vec3 v;

void main(void){

v = vec3(gl\_ModelViewMatrix \* gl\_Vertex);

N = normalize(gl\_NormalMatrix \* gl\_Normal);

gl\_Position = gl\_ModelViewProjectionMatrix \* gl\_Vertex;

}

PLACE THIS UNIFORM VALUE SOMEWHERE IN MAIN:

GLuint program = glCreateProgram();

int loc = glGetUniformLocation(program, "num");

int hours = ((time (0)/60/60) - 7) % 24;

glUniform1f(loc, hours);