

# Shader example

Alexandre Valle de Carvalho

# Initialization: an array of CGFShaders

```
this.testShaders=[  
  new CGFshader(this.gl, "shaders/flat.vert", "shaders/flat.frag"),  
  new CGFshader(this.gl, "shaders/uScale.vert", "shaders/uScale.frag"),  
  new CGFshader(this.gl, "shaders/varying.vert", "shaders/varying.frag"),  
  new CGFshader(this.gl, "shaders/texture1.vert", "shaders/texture1.frag"),  
  new CGFshader(this.gl, "shaders/texture2.vert", "shaders/texture2.frag"),  
  new CGFshader(this.gl, "shaders/texture3.vert", "shaders/texture3.frag"),  
  new CGFshader(this.gl, "shaders/texture3.vert", "shaders/sepia.frag"),  
  new CGFshader(this.gl, "shaders/texture3.vert", "shaders/convolution.frag")  
  new CGFshader(this.gl, "shaders/tvarying.vert", "shaders/tvarying.frag") ];
```

# Initialization: the selection of active shader

```
this.texture = null;
```

```
this.appearance = null;
```

```
this.selectedExampleShader=8;
```

```
this.wireframe=false;
```

```
this.scaleFactor=50.0;
```

## Initialization: uniforms

// texture will have to be bound to unit 1 later, when using the shader, with "this.texture2.bind(1);"

```
this.testShaders[4].setUniformsValues({uSampler2: 1});
```

```
this.testShaders[5].setUniformsValues({uSampler2: 1});
```

```
this.testShaders[8].setUniformsValues({selColor: [1.0, 0.0, 0.0, 1.0] });
```

```
this.texture2 = new CGFtexture(this, "textures/FEUP.jpg");
```

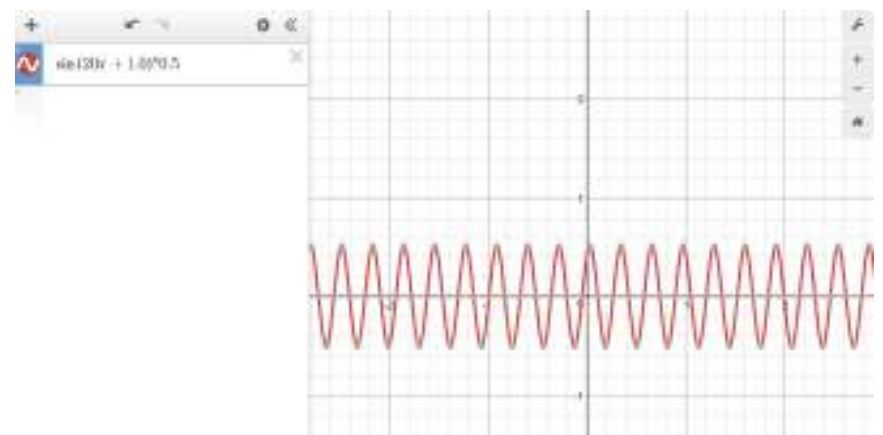
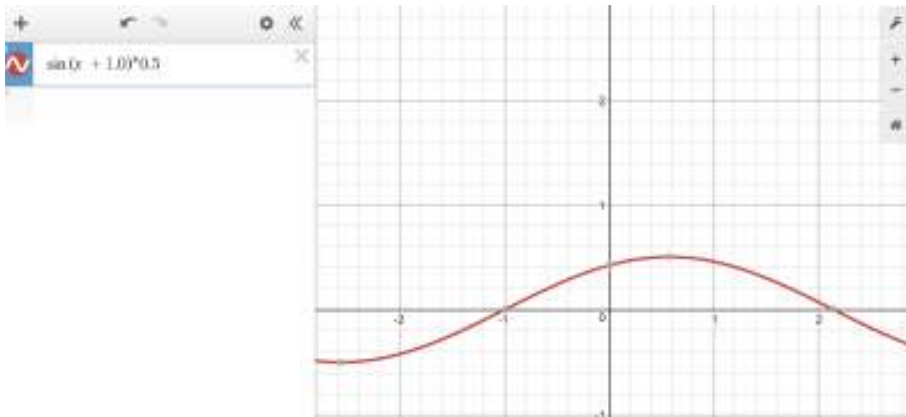
```
this.updateScaleFactor();
```

# Initialization: set shader uniform: scale

```
LightingScene.prototype.updateScaleFactor=function(v) {  
    this.testShaders[1].setUniformsValues({normScale: this.scaleFactor});  
    this.testShaders[2].setUniformsValues({normScale: this.scaleFactor});  
    this.testShaders[5].setUniformsValues({normScale: this.scaleFactor});  
    this.testShaders[8].setUniformsValues({normScale: this.scaleFactor});  
}
```

# Update function (per frame)

```
LightingScene.prototype.update = function (time) {  
    var factor = (Math.sin((time * 3.0) % 3141 * 0.002)+1.0)*.5;  
    this.testShaders[8].setUniformsValues({timeFactor: factor});  
}
```



<https://www.desmos.com/calculator>

# Display function (per frame): initialization

```
LightingScene.prototype.display = function () {  
    // Clear image and depth buffer every time we update the scene  
    this.gl.viewport(0, 0, this.gl.canvas.width, this.gl.canvas.height);  
    this.gl.clear(this.gl.COLOR_BUFFER_BIT | this.gl.DEPTH_BUFFER_BIT);  
    this.gl.clearColor(0.1, 0.1, 0.1, 1.0);  
    this.gl.enable(this.gl.DEPTH_TEST);  
}
```

# Display function (per frame): initialization

```
// Initialize Model-View matrix as identity (no transformation)
this.updateProjectionMatrix();
this.loadIdentity();
```

```
// Apply transformations corresponding to the camera position relative to the origin
this.applyViewMatrix();
```

```
// Update all lights used
this.lights[0].update();
```

```
// Draw axis
this.axis.display();
this.appearance.apply();
```



## Display function (per frame): initialization

```
this.setActiveShader(this.testShaders[this.selectedExampleShader]);  
// draw  
this.pushMatrix();  
this.texture2.bind(1);  
this.translate(0,-6,0);  
this.scale(0.5,0.5,0.5);  
this.rotate(-Math.PI/2, 1, 0, 0);  
this.teapot.display();  
this.popMatrix();  
this.setActiveShader(this.defaultShader);  
}
```

# Vertex file

```
#ifdef GL_ES
precision highp float;
#endif

attribute vec3 aVertexPosition;
attribute vec3 aVertexNormal;
attribute vec2 aTextureCoord;
uniform mat4 uMVMatrix;
uniform mat4 uPMatrix;
uniform mat4 uNMatrix;

uniform float timeFactor;
uniform float normScale;
varying vec4 coords;
varying vec4 normal;
```

# Vertex file

```
void main() {  
    // vertex pushed outwards according to the normal and scale  
    vec4 vertex=vec4(aVertexPosition+aVertexNormal*normScale*timeFactor*0.1,  
1.0);  
    // projected vertex  
    gl_Position = uPMatrix * uMVMMatrix * vertex;  
    // normal variable  
    normal = vec4(aVertexNormal, 1.0);  
    coords = vertex / 10.0;  
}
```

# Fragment file: example

```
#ifdef GL_ES
precision highp float;
#endif

varying vec4 coords;
varying vec4 normal;
uniform float timeFactor;
uniform vec4 selColor;
void main() {
    gl_FragColor = normal;
    gl_FragColor.rgb=mix(gl_FragColor.rgb, selColor.rgb, timeFactor);
}
```

# Fragment file: another exemple

```
#ifdef GL_ES
precision highp float;
#endif

varying vec4 coords;
varying vec4 normal;
uniform float timeFactor;
uniform vec4 selColor;

void main() {
    // half the object gets the normal color. The other half sums r+g+b
    if (coords.x > 0.0)
        gl_FragColor = normal;
    else {
        gl_FragColor.rgb = abs(coords.xyz)/3.0;
        gl_FragColor.a = 1.0;
    }
    gl_FragColor.rgb=mix(gl_FragColor.rgb, selColor.rgb, timeFactor);
}
```