

j3d

bartek drozdz

j3d

j3d is an engine designed to address specific issues
I encountered when learning webgl
and working on the ROME project

webgl

PROBLEM A

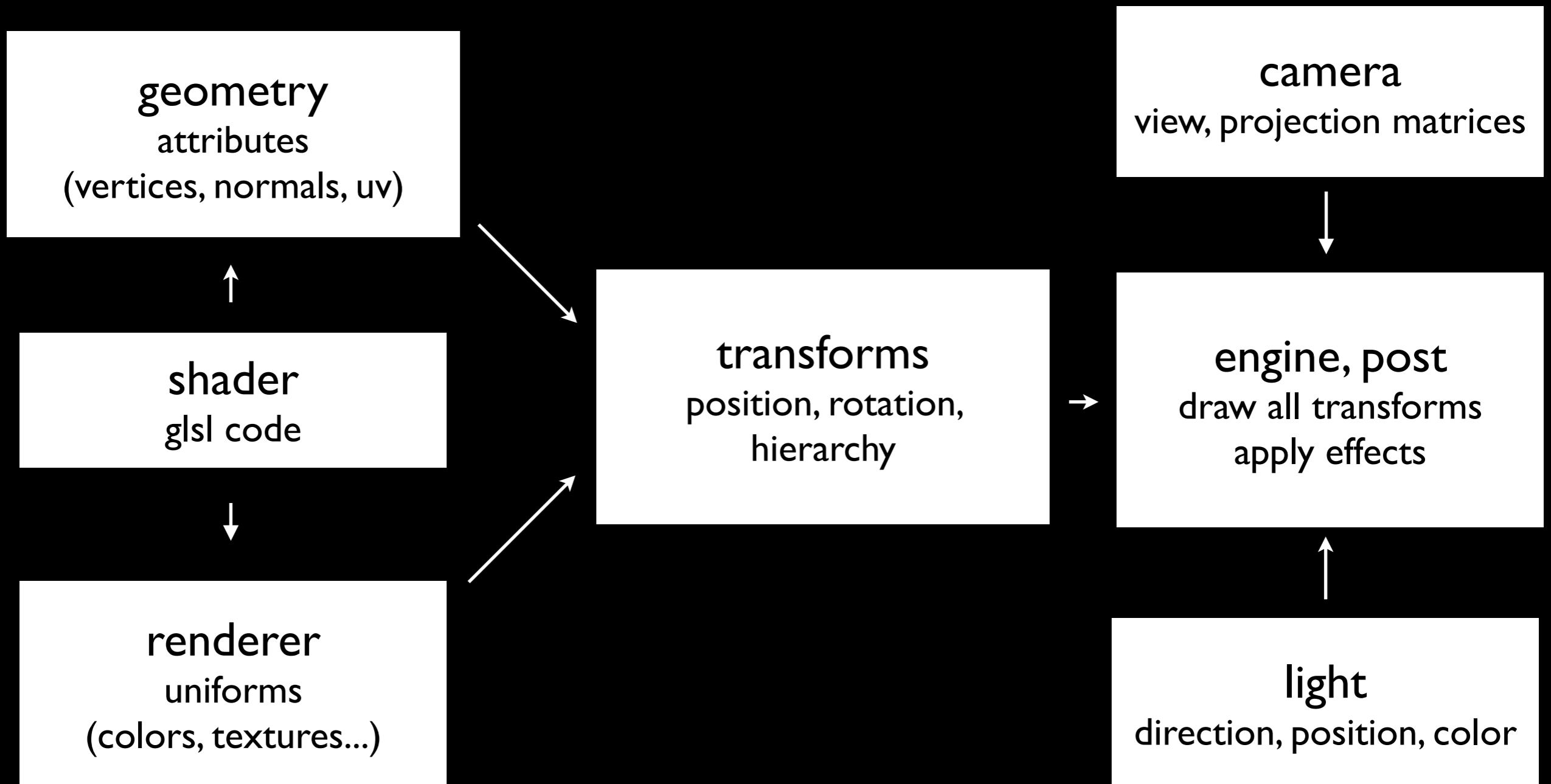
lots of boiler-plate code, same problems to solve each time

j3d

SOLUTION TO PROBLEM A

geometries, transforms and scene hierarchy
camera, light & post effects

j3d, architecture



webgl

PROBLEM B

no standard asset format, no scene editors

j3d

SOLUTION TO PROBLEM B

Unity3d - a great scene editor, works with modeling software

j3d, unity exporter



webgl

PROBLEM C

editing shaders is a pain

webgl, shaders

```
uniform float uSpec;  
uniform float uSh;  
  
varying vec3 vLight;  
varying vec2 vTextureCoord;  
  
void main(void) {  
    vec4 p = mMMatrix * vec4(aVertexPosition, 1.0);  
    gl_Position = pMatrix * vMatrix * p;  
    vTextureCoord = getTextureCoord(aTextureCoord);  
    vec3 n = normalize( nMatrix * aVertexNormal );  
    vLight = computeLights(p, n, uSpec, uSh);  
}
```

webgl, shaders

```
<head>
<script id="vs" type="x-shader">
  // vertex shader code
</script>
<script id="fs" type="x-shader">
  // fragment shader code
</script>
<script type="text/javascript">
  v = document.getElementById("vs").firstChild.nodeValue;
  f = document.getElementById("fs").firstChild.nodeValue;
  // v, f = shader code as string, compiled at runtime
</script>
</head>
```

webgl, shaders

can only be embedded in HTML, not as separate file
embedding in HTML only works for simple demos

webgl, shaders

```
VertexShader = [  
    "attribute vec2 aVertexPosition;",  
    "attribute vec2 aTextureCoord;",  
  
    "varying vec2 vTextureCoord;",  
  
    "void main(void) {",  
    "    gl_Position = vec4(aVertexPosition, 0.0, 1.0);",  
    "    vTextureCoord = aTextureCoord;",  
    "}"  
].join("\n");
```

webgl, shaders

~~can only be embedded in HTML, not as separate file,~~
code **INSANELY** hard to read and edit,
no syntax coloring...

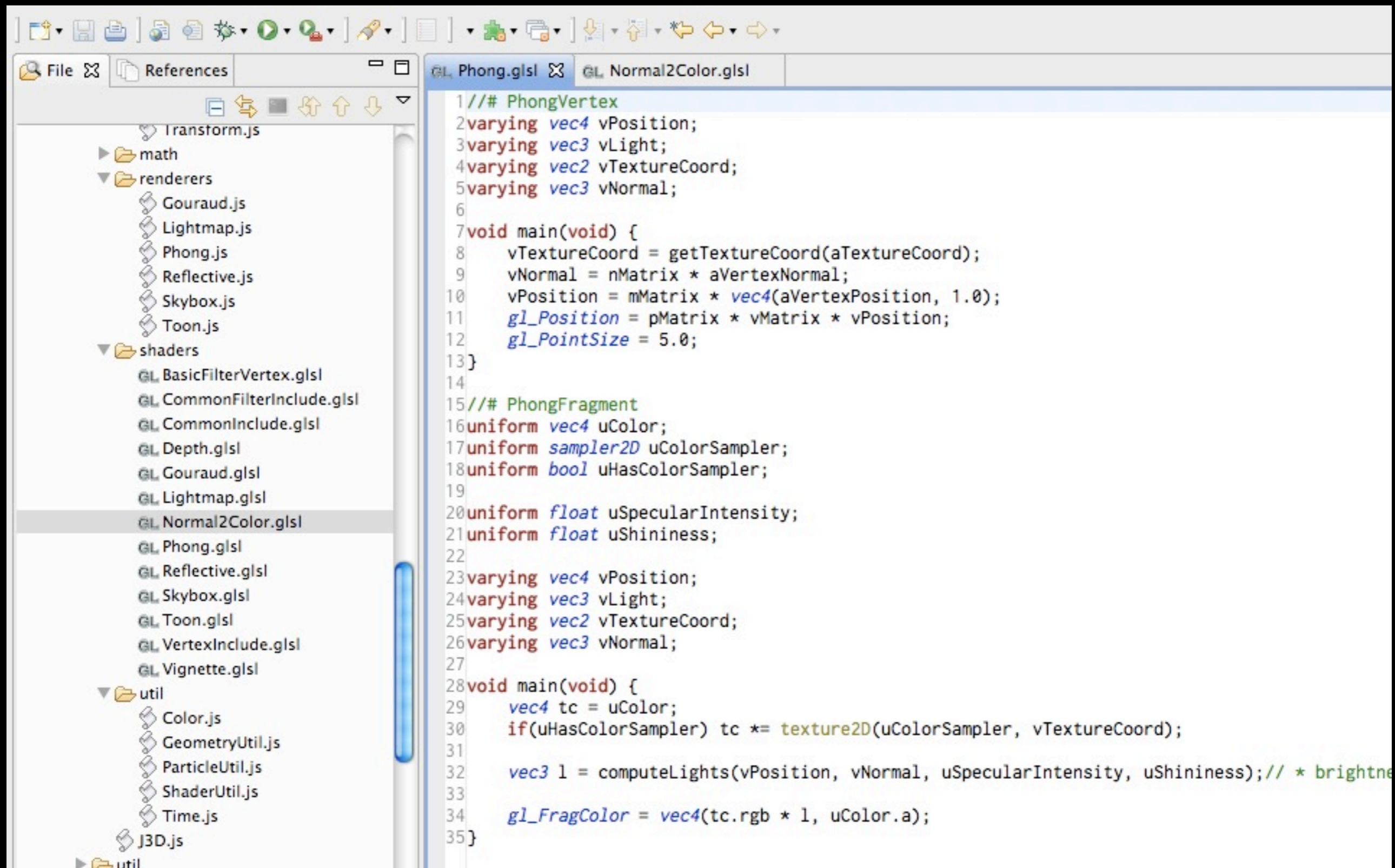
j3d, shaders

SOLUTION TO PROBLEM C

GLSL file loaded with XMLHttpRequest

special comment format to add some necessary metadata

j3d, shaders



```
1//# PhongVertex
2varying vec4 vPosition;
3varying vec3 vLight;
4varying vec2 vTextureCoord;
5varying vec3 vNormal;
6
7void main(void) {
8    vTextureCoord = getTextureCoord(aTextureCoord);
9    vNormal = nMatrix * aVertexNormal;
10   vPosition = mMatrix * vec4(aVertexPosition, 1.0);
11   gl_Position = pMatrix * vMatrix * vPosition;
12   gl_PointSize = 5.0;
13}
14
15//# PhongFragment
16uniform vec4 uColor;
17uniform sampler2D uColorSampler;
18uniform bool uHasColorSampler;
19
20uniform float uSpecularIntensity;
21uniform float uShininess;
22
23varying vec4 vPosition;
24varying vec3 vLight;
25varying vec2 vTextureCoord;
26varying vec3 vNormal;
27
28void main(void) {
29   vec4 tc = uColor;
30   if(uHasColorSampler) tc *= texture2D(uColorSampler, vTextureCoord);
31
32   vec3 l = computeLights(vPosition, vNormal, uSpecularIntensity, uShininess); // * brightness
33
34   gl_FragColor = vec4(tc.rgb * l, uColor.a);
35}
```

j3d, shaders

```
1 //# PhongVertex
2 varying vec4 vPosition;
3 varying vec3 vLight;
4 varying vec2 vTextureCoord;
5 varying vec3 vNormal;
6
7 void main(void) {
8     vTextureCoord = getTextureCoord(aTextureCoord);
9     vNormal = nMatrix * aVertexNormal;
10    vPosition = mMatrix * vec4(aVertexPosition, 1.0);
11    gl_Position = pMatrix * vMatrix * vPosition;
12    gl_PointSize = 5.0;
13}
14
15 //# PhongFragment
16 uniform vec4 uColor;
17 uniform sampler2D uColorSampler;
18 uniform bool uHasColorSampler;
19
20 uniform float uSpecularIntensity;
21 uniform float uShininess;
```

j3d tour

unity scenes, shaders, particles, lightmaps...



j3d tour

```
// 1. Basic setup
```

```
engine = new J3D.Engine();
```

```
sun = new J3D.Transform();
```

```
sun.light = new J3D.Light(J3D.DIRECT);
```

```
cube = new J3D.Transform();
```

```
cube.geometry = J3D.Primitive.Cube(1, 1, 1);
```

```
cube.renderer = new J3D.Phong();
```

```
eye = new J3D.Transform();
```

```
eye.camera = new J3D.Camera();
```

```
engine.camera = eye;
```

```
engine.scene.add(eye, cube, sun);
```

```
engine.render();
```

j3d tour

```
// 2. Loading a scene exported from Unity
```

```
e = new J3D.Engine();
```

```
J3D.Loader.loadJSON("meshes.json", function(m) {  
    J3D.Loader.loadJSON("scene.json", function(s) {  
        J3D.Loader.parseJSONScene(s, m, e);  
        e.render();  
    })  
});
```

j3d tour

```
// 3. Loading a shader
```

```
var goldhead; // Instance of J3D.Shader
```

```
engine = new J3D.Engine();
```

```
J3D.Loader.loadGLSL("../demo/shaders/GoldHead.glsl",  
    function(s) {  
        goldhead = s;  
        initScene();  
    }  
);
```

what's next?

interactivity and collision detection,
more features in unity exporter,
animation support,
better documentation,
optimization

and more demos!

demos and links:

everyday3d.com/

build and source code:

github.com/drojdjou/J3D

tutorials:

github.com/drojdjou/J3D/wiki

twitter:

[@bartekd](https://twitter.com/bartekd)

thank you!