

j3d - adventures in webgl

j3d - adventures in webgl

bartek drozdz

stockholm, sweden

how it all started?

I was a happy flash developer until a few months ago...

interactive music video

but there's a trick...

interactive music video

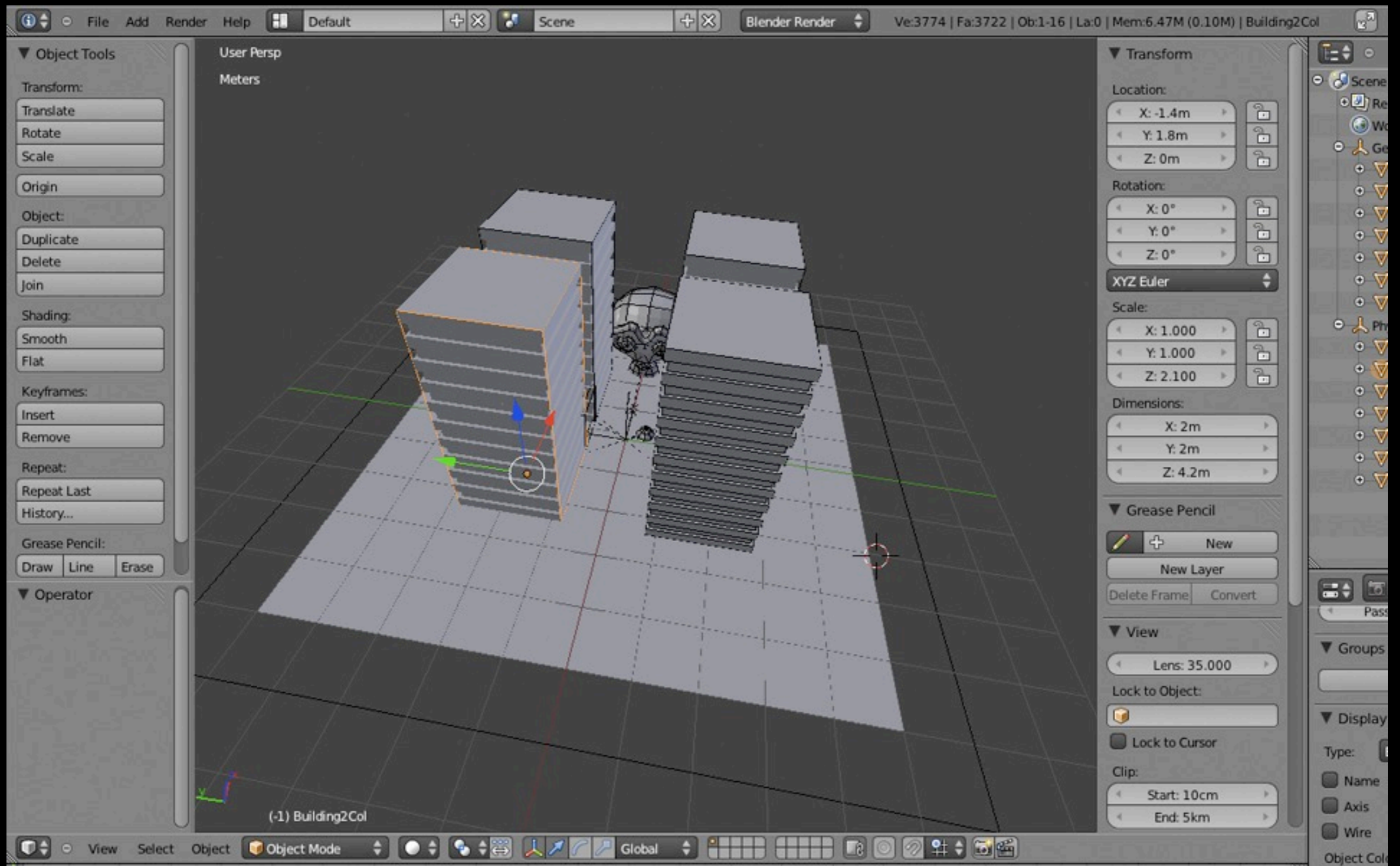
but there's a trick...

it needs to be webgl

prototyped with unity3d



scenes built with blender



assets exported to JSON

```
005_cube.html 009_3d.html 011_animation.html 006_texture.html Animal.js centipede.js ✕
?id /*/ for f() if try
18 var model = {
19
20   "version" : 2,
21
22   "scale" : 10.000000,
23
24   "materials": [ {
25     "DbgColor" : 15658734,
26     "DbgIndex" : 0,
27     "DbgName" : "comp_centipede_rig:blinn8SG"
28   }],
29
30   "vertices": [331,287,-696,226,281,-573,232,388,-640,578,-22,-627,223,433,-844,295,398,-799,359,403,-872,297,462,-990,428,468,-1
31
32   "morphTargets": [
33     { "name": "centipede_walkA_001", "vertices": [331,287,-696,226,281,-573,232,388,-640,578,-22,-627,223,433,-844,295,398,-799,359
34     { "name": "centipede_walkA_002", "vertices": [157,369,-659,34,286,-560,36,388,-629,431,302,-520,33,432,-833,103,397,-786,168,40
35     { "name": "centipede_walkA_003", "vertices": [-35,466,-614,-145,289,-537,-150,386,-606,196,622,-425,-156,431,-810,-86,396,-764,
36     { "name": "centipede_walkA_004", "vertices": [-161,505,-620,-253,286,-557,-263,387,-625,35,711,-403,-287,432,-828,-214,397,-789
37     { "name": "centipede_walkA_005", "vertices": [-210,455,-652,-320,280,-595,-326,388,-661,4,537,-428,-379,433,-858,-301,399,-830,
38     { "name": "centipede_walkA_006", "vertices": [-236,352,-706,-357,294,-636,-357,389,-705,-7,272,-497,-451,435,-886,-369,400,-875
39     { "name": "centipede_walkA_007", "vertices": [-252,266,-792,-335,314,-684,-349,391,-755,4,40,-663,-483,436,-909,-401,401,-917,-
40     { "name": "centipede_walkA_008", "vertices": [-223,267,-876,-252,309,-734,-291,393,-796,54,12,-882,-456,437,-917,-377,402,-943,
41     { "name": "centipede_walkA_009", "vertices": [-165,274,-897,-134,295,-746,-189,391,-791,65,-18,-1026,-375,436,-876,-302,401,-91
42     { "name": "centipede_walkA_010", "vertices": [-58,286,-866,22,281,-726,-37,388,-754,89,-23,-1076,-233,432,-813,-167,398,-865,-2
43     { "name": "centipede_walkA_011", "vertices": [152,368,-865,218,285,-722,151,387,-740,353,301,-1097,-46,431,-787,16,396,-844,-34
44     { "name": "centipede_walkA_012", "vertices": [344,467,-859,388,290,-731,319,387,-746,591,623,-1029,122,431,-798,186,397,-853,13
45     { "name": "centipede_walkA_013", "vertices": [483,507,-850,495,288,-739,430,389,-762,767,713,-923,240,433,-835,310,399,-884,269
46     { "name": "centipede_walkA_014", "vertices": [621,457,-843,581,282,-726,531,390,-769,931,539,-829,356,435,-875,434,401,-911,408
47     { "name": "centipede_walkA_015", "vertices": [717,352,-789,625,295,-683,603,390,-748,1002,273,-665,456,435,-889,539,401,-908,53
48     { "name": "centipede_walkA_016", "vertices": [692,265,-761,595,313,-666,590,391,-738,928,40,-598,479,436,-909,565,401,-908,584,
49     { "name": "centipede_walkA_017", "vertices": [616,266,-751,513,307,-649,514,391,-722,851,11,-604,443,436,-914,527,402,-893,568,
50     { "name": "centipede_walkA_018", "vertices": [499,274,-741,393,295,-629,398,391,-700,738,-18,-628,360,436,-900,439,401,-866,493
51   ],
52
53   "morphColors": [
```

built with three.js



ro.me

directed by Chris Milk

Google Creative Labs, Aaron Koblin & mrdoob

North Kingdom

Mirada

<http://ro.me>

j3d, another webgl engine

WHY?

j3d, another webgl engine

WHY?

first: learning experience

j3d, another webgl engine

WHY?

first: learning experience

second: focus on the fun part, not the boring part

webgl

PROBLEM A

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

webgl

PROBLEM B

no standard asset format, no scene editors

webgl

PROBLEM C

editing shaders is a pain

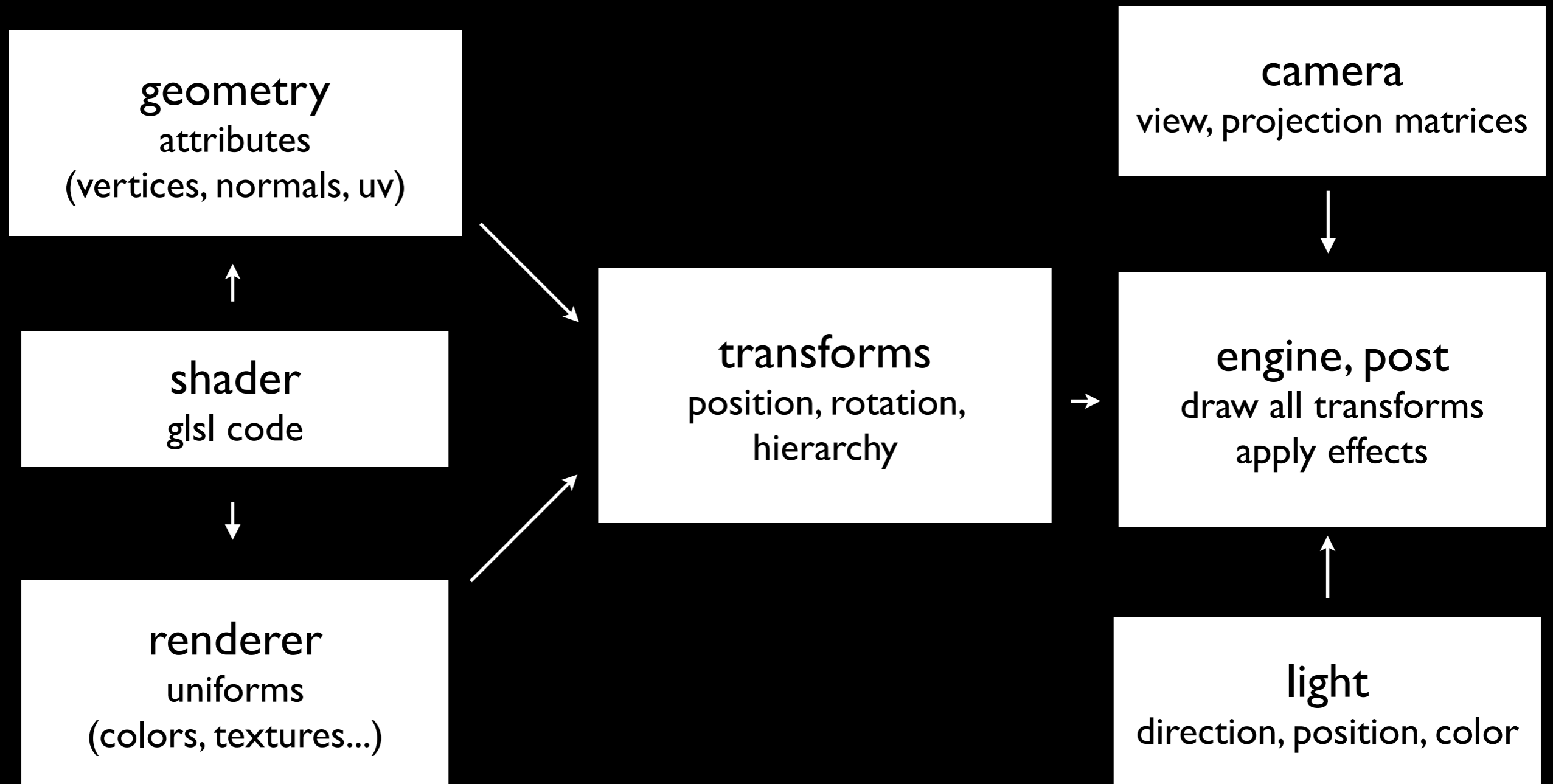
j3d

SOLUTION TO PROBLEM A

geometries, transforms and scene hierarchy

camera, light & post effects

j3d architecture

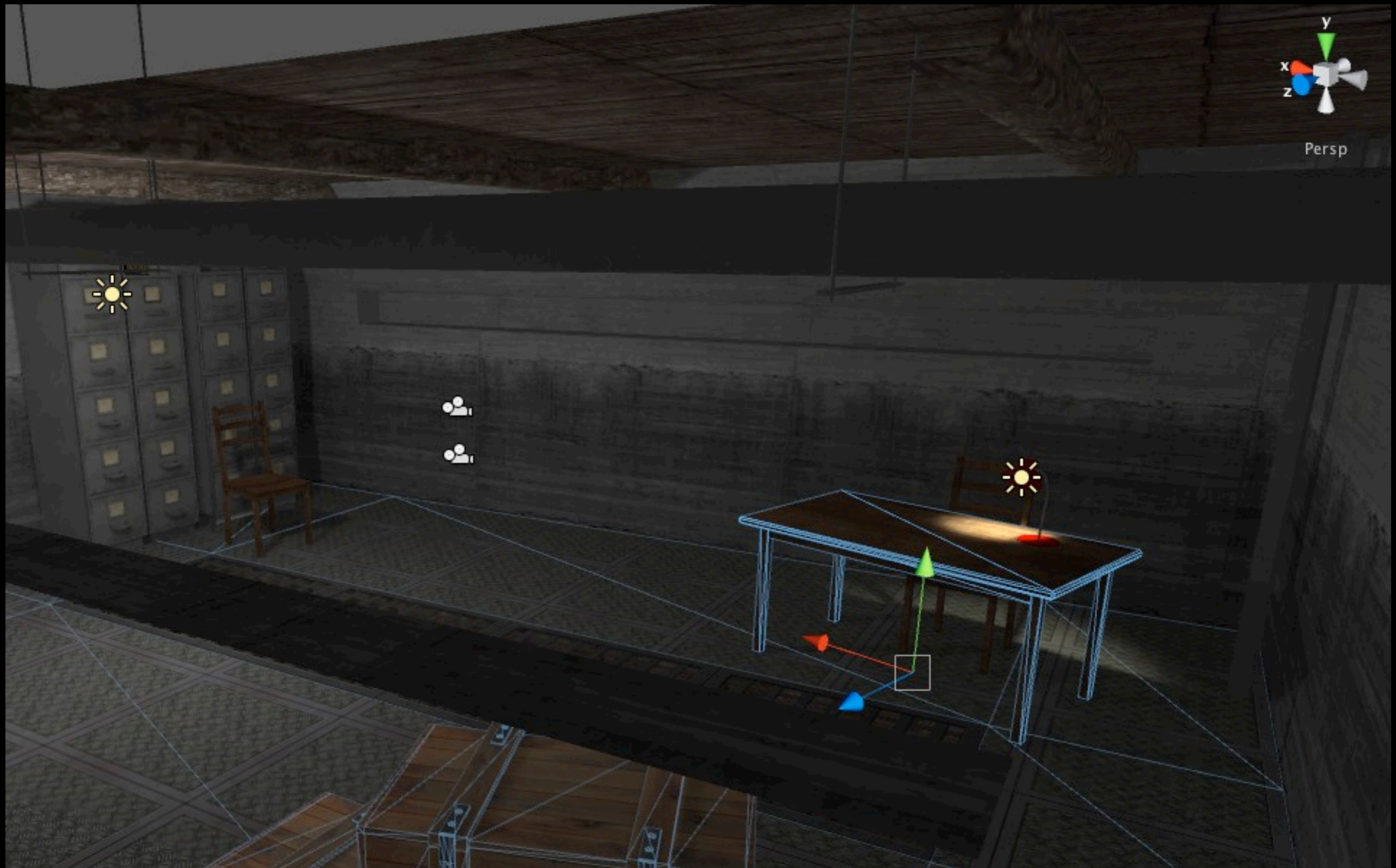


unity

SOLUTION TO PROBLEM B

Unity3d - a great scene editor, works with modeling software

unity exporter



shaders

PROBLEM C

editing shaders is a pain

shaders

what is a shader?

shaders

a shader is a small program that runs on the GPU and is responsible for rendering

shaders

shaders are at the heart of webgl

shaders

shaders are not written with Javascript, but GLSL instead
GLSL has a C-like syntax

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);  
}
```

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>
```

shaders

can only be embedded in HTML, not as separate file

shaders

embedding in HTML only works for tutorials

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");
```

shaders

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

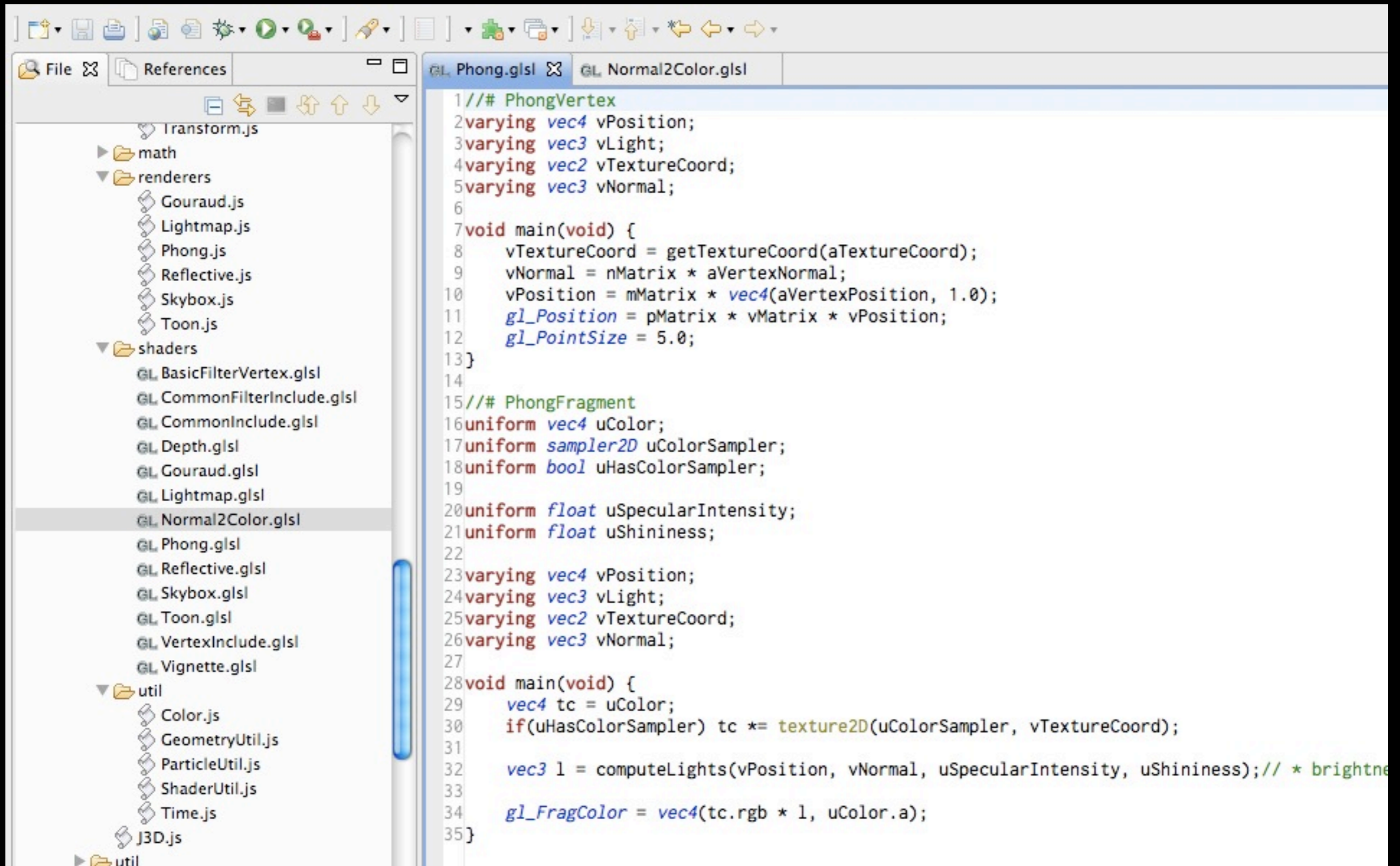
shaders

SOLUTION TO PROBLEM C

GLSL file loaded with XMLHttpRequest

special comment format to add some necessary metadata

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}
```

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;
22
```

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);
```

```
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. Creating a particle system
```

```
g = new J3D.Geometry();
```

```
v = [];
```

```
numParticle = 12000;
```

```
while(numParticle-- > 0) {}
```

```
    v.push(Math.random() * 100);
```

```
}
```

```
g.addArray("aVertexPosition", v, 3);
```

j3d tour

```
// 4. 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();  
    }  
);
```

j3d tour

```
// 5. Creating a post effect
```

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

```
post = new J3D.Postprocess(engine);
```

```
post.filter = filter // also instance of J3D.Shader
```

```
post.render();
```

what's next?

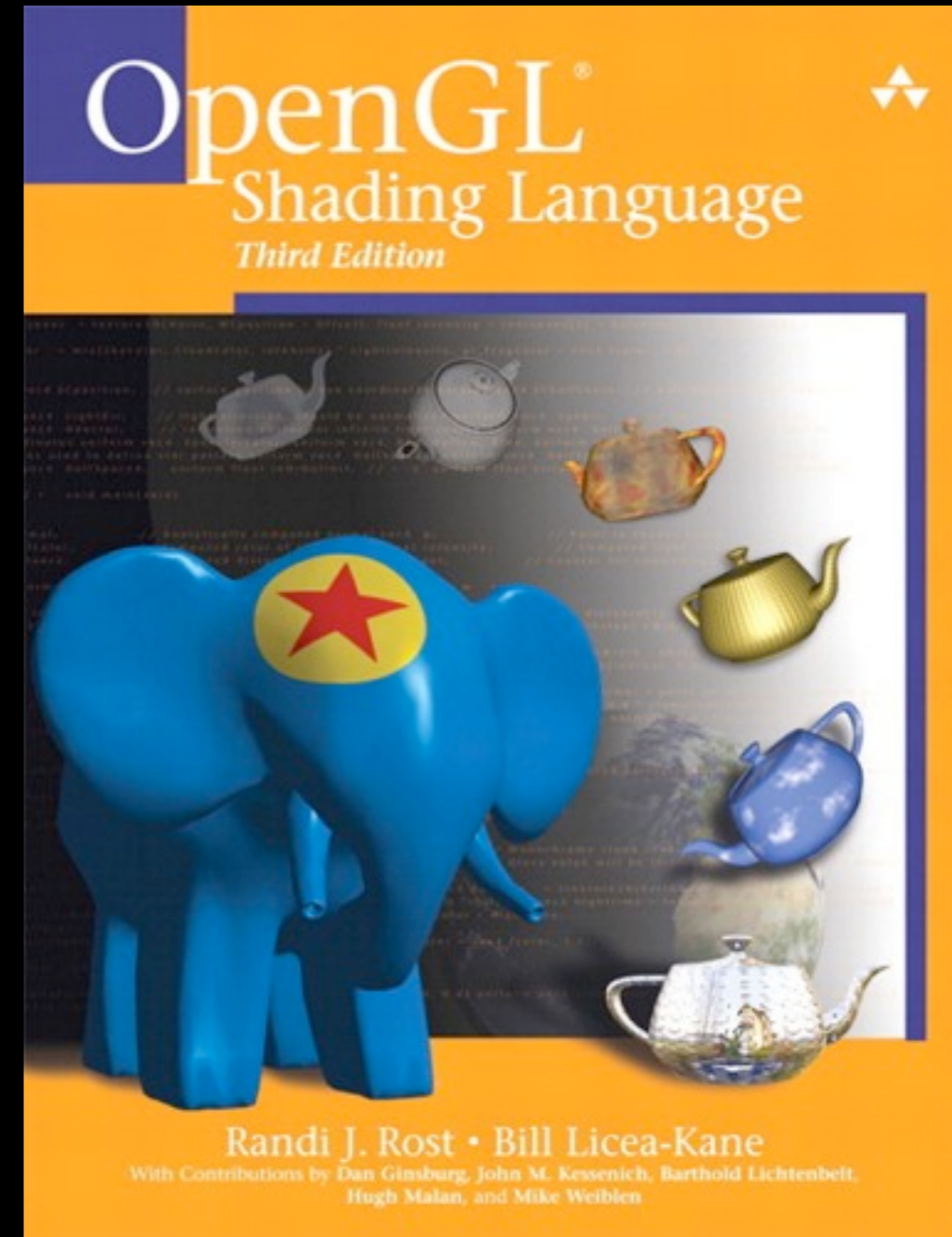
interactivity and collision detection,
more features in unity exporter,
animation support,
better documentation,
optimization

what's next?

and more demos!

recommended reading

OpenGL
Shading Language



demos and links:

everyday3d.com/

build and source code:

github.com/drojdjou/J3D

tutorials:

github.com/drojdjou/J3D/wiki

this presentation:

everyday3d.com/ogs11/j3d-webgl.pdf

my comments:

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

thank you!