

COMP 370 assignment #3: Simple Interactive Animation

Thomas Williamson

id: 588206

2021/10/20

The goal of this project is to render a basic interactable animation of a directional object going back and forth across the plane. Implement a button for each, stop animation, start animation, and change direction. Also implement a menu to change the colour of the object to red, blue, green, and yellow.

```
1  <!--
2  COMP 370 assignment #3: SimpleInteractiveAnimation
3  Thomas Williamson
4  id: 588206
5  2021/10/20
6  -->
7  <html>
8
9  <script id="vertex-shader" type="x-shader/x-vertex">
10 #version 300 es
11
12 in vec4 aPosition;
13 in vec4 aColor;
14
15 uniform float xDelta;
16 uniform float uTheta;
17
18 out vec4 vColor;//vertex
19
20 void main()
21 {
22     float s = sin(uTheta);
23     float c = cos(uTheta);
24
25     gl_Position.x = (-s*aPosition.y + c*aPosition.x) + xDelta;
26     gl_Position.y = s*aPosition.x + c*aPosition.y;
27     gl_Position.z = 0.0;
28     gl_Position.w = 1.0;
29     vColor = aColor;
30 }
31 </script>
32
33 <script id="fragment-shader" type="x-shader/x-fragment">
34 #version 300 es
35
36 precision mediump float;
37
38 in vec4 vColor;
39 out vec4 fColor;
40
41 void main()
42 {
43     //fColor = vec4( 1.0, 0.0, 0.0, 1.0 );
44     //fColor = vec4( 0.0, 1.0, 0.0, 1.0 );
45     //fColor = vec4( 0.0, 0.0, 1.0, 1.0 );
46     //fColor = vec4( 1.0, 1.0, 1.0, 1.0 );
47     //fColor = vec4( 1.0, 1.0, 0.0, 1.0 );
48     fColor = vColor;
49 }
50 </script>
```

```
51
52 <script type="text/javascript" src="../../Common/initShaders.js"></script>
53 <script type="text/javascript" src="../../Common/MV.js"></script>
54 <script type="text/javascript" src="SimpleInteractiveAnimation.js"></script>
55
56
57 <body>
58 <div style = "position:absolute; left:20px; top:10px;">
59   <button id="directionButton">Change Direction</button>
60   <button id="startButton">Start</button>
61   <button id="stopButton">Stop</button>
62 </div>
63 <div style="position:absolute; left:20px;top:35px ">
64   <select id="colourmenu" size="4">
65     <option value="0">Red</option>
66     <option value="1">Blue</option>
67     <option value="2">Green</option>
68     <option value="3">Yellow</option>
69   </select>
70   <div>
71     <canvas id="gl-canvas" width="512" height="512"> </canvas>
72   </div>
73 </div>
74
75 </html>
76
```

```

1  //<!--
2  // COMP 370 assignment #3: SimpleInteractiveAnimation
3  // Thomas Williamson
4  // id: 588206
5  // 2021/10/20
6  // -->
7  "use strict";
8
9  var program
10 var bufferId
11 var positionLoc
12 var gl;
13 var arrow;
14 var xDeltaLoc;
15 var thetaLoc;
16 var directionRight = true;
17 var xChange = 0.0;
18 var delta = 0.05;
19 var theta = 0.0;
20 var speed = 100;
21 var ColorLoc
22 var cBuffer
23 ▼ var colorA = [ 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0,
24                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
25                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
26                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
27                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
28                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
29                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
30                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
31                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
32                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
33                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
34                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
35                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
36                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
37                  1,0,0, 1,0,0, 1,0,0, 1,0,0,
38                  1,0,0, 1,0,0, 1,0,0]
39
40 ▼ window.onload = function init(){
41     var canvas = document.getElementById("gl-canvas");
42
43     gl = canvas.getContext('webgl2');
44     if (!gl) alert("WebGL 2.0 isn't available");
45
46 ▼     arrow = new Float32Array([
47         0,0,
48         0.15, 0.2,
49         0.4, 0,

```

50	0.15, -0.2,
51	0,0,
52	-0.0, 0.2,
53	-0.008, 0.2,
54	-0.016, 0.199,
55	-0.023, 0.198,
56	-0.031, 0.196,
57	-0.039, 0.193,
58	-0.046, 0.19,
59	-0.054, 0.187,
60	-0.061, 0.183,
61	-0.068, 0.178,
62	-0.075, 0.173,
63	-0.082, 0.168,
64	-0.088, 0.162,
65	-0.094, 0.155,
66	-0.1, 0.149,
67	-0.106, 0.141,
68	-0.111, 0.134,
69	-0.117, 0.126,
70	-0.121, 0.118,
71	-0.126, 0.109,
72	-0.13, 0.1,
73	-0.134, 0.091,
74	-0.137, 0.081,
75	-0.14, 0.072,
76	-0.143, 0.062,
77	-0.145, 0.052,
78	-0.147, 0.042,
79	-0.148, 0.031,
80	-0.149, 0.021,
81	-0.15, 0.01,
82	-0.15, -0.0,
83	-0.15, -0.01,
84	-0.149, -0.021,
85	-0.148, -0.031,
86	-0.147, -0.042,
87	-0.145, -0.052,
88	-0.143, -0.062,
89	-0.14, -0.072,
90	-0.137, -0.081,
91	-0.134, -0.091,
92	-0.13, -0.1,
93	-0.126, -0.109,
94	-0.121, -0.118,
95	-0.117, -0.126,
96	-0.111, -0.134,
97	-0.106, -0.141,
98	-0.1, -0.149,

```

99         -0.094, -0.155,
100         -0.088, -0.162,
101         -0.082, -0.168,
102         -0.075, -0.173,
103         -0.068, -0.178,
104         -0.061, -0.183,
105         -0.054, -0.187,
106         -0.046, -0.19,
107         -0.039, -0.193,
108         -0.031, -0.196,
109         -0.023, -0.198,
110         -0.016, -0.199,
111         -0.008, -0.2
112     });
113
114     gl.viewport(0, 0, canvas.width, canvas.height);
115     gl.clearColor(1.0, 1.0, 1.0, 1.0);
116     gl.clear(gl.COLOR_BUFFER_BIT);
117
118     program = initShaders(gl, "vertex-shader", "fragment-shader");
119
120     bufferId = gl.createBuffer();
121     gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
122     gl.bufferData(gl.ARRAY_BUFFER, arrow, gl.STATIC_DRAW);
123
124     positionLoc = gl.getAttribLocation( program, "aPosition" );
125     gl.vertexAttribPointer( positionLoc, 2, gl.FLOAT, false, 0, 0 );
126
127     xDeltaLoc = gl.getUniformLocation(program, "xDelta");
128     thetaLoc = gl.getUniformLocation(program, "uTheta");
129
130
131     gl.useProgram( program );
132     gl.enableVertexAttribArray(positionLoc);
133
134     colourBuff();
135
136
137     document.getElementById("directionButton").onclick = function (event) {
138         directionRight = !directionRight;
139         if(!directionRight){
140             theta = 3.14159;
141             gl.uniform1f(thetaLoc, theta);
142         }else{
143             theta = 0;
144             gl.uniform1f(thetaLoc, theta);
145         }
146     };
147     document.getElementById("startButton").onclick = function (event) {

```



```

148     delta = 0.05;
149 };
150
151 document.getElementById("stopButton").onclick = function (event) {
152     delta = 0.0;
153 };
154 var m = document.getElementById("colourmenu")
155 m.onclick = function(event){
156     console.log(m.selectedIndex);
157     switch (m.selectedIndex) {
158         case 0:
159             colorA = [1,0,0, 1,0,0, 1,0,0, 1,0,0,
160                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
161                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
162                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
163                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
164                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
165                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
166                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
167                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
168                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
169                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
170                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
171                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
172                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
173                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
174                     1,0,0, 1,0,0, 1,0,0, 1,0,0,
175                     1,0,0, 1,0,0, 1,0,0, 1,0,0];
176             colourBuff();
177         break;
178         case 1:
179             colorA = [0,0,1, 0,0,1, 0,0,1, 0,0,1,
180                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
181                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
182                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
183                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
184                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
185                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
186                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
187                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
188                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
189                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
190                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
191                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
192                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
193                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
194                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
195                     0,0,1, 0,0,1, 0,0,1, 0,0,1,
196                     0,0,1, 0,0,1, 0,0,1, 0,0,1];

```

```

197         colourBuff();
198     break;
199     case 2:
200         colorA = [0,1,0, 0,1,0, 0,1,0, 0,1,0,
201                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
202                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
203                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
204                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
205                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
206                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
207                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
208                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
209                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
210                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
211                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
212                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
213                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
214                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
215                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
216                 0,1,0, 0,1,0, 0,1,0, 0,1,0,
217                 0,1,0, 0,1,0, 0,1,0, 0,1,0];
218         colourBuff();
219     break;
220     case 3:
221         colorA = [1,1,0, 1,1,0, 1,1,0, 1,1,0,
222                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
223                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
224                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
225                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
226                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
227                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
228                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
229                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
230                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
231                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
232                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
233                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
234                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
235                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
236                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
237                 1,1,0, 1,1,0, 1,1,0, 1,1,0,
238                 1,1,0, 1,1,0, 1,1,0, 1,1,0,];
239         colourBuff();
240
241     }
242 }
243
244 render();
245

```



```

246 }
247 function colourBuff(){
248     cBuffer = gl.createBuffer();
249     gl.bindBuffer(gl.ARRAY_BUFFER, cBuffer);
250     gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(colorA), gl.STATIC_DRAW );
251
252     ColorLoc = gl.getAttribLocation( program, "aColor");
253     gl.vertexAttribPointer(ColorLoc, 3, gl.FLOAT, false, 0, 0);
254
255     gl.enableVertexAttribArray(ColorLoc);
256
257 }
258
259 function render()
260 {
261     gl.clear( gl.COLOR_BUFFER_BIT );
262
263     if(xChange>0.55){
264         directionRight = false;
265         theta = 3.14159;
266         gl.uniform1f(thetaLoc, theta);
267     }else if(xChange<-0.55){
268         directionRight = true;
269         theta = 0;
270         gl.uniform1f(thetaLoc, theta);
271     }
272     if(directionRight==true){
273         xChange = xChange + delta;
274     }else{
275         xChange = xChange - delta;
276     }
277     gl.uniform1f(xDeltaLoc, xChange);
278
279     gl.drawArrays(gl.TRIANGLE_FAN, 0, 65);
280
281     setTimeout(
282         function () {requestAnimationFrame(render);},
283         speed
284     );
285 }

```

Change Direction Start Stop

Red
Blue
Green
Yellow



Change Direction

Start

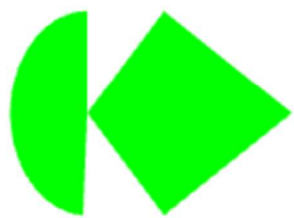
Stop

Red

Blue

Green

Yellow



Change Direction

Start

Stop

Red
Blue
Green
Yellow



Change Direction

Start

Stop

Red
Blue
Green
Yellow

