

One of my students (Tomas Alejandro Mondragon) pointed out the following changes were necessary for some of the code to run on Firefox. I made all these changes to the code for the class and I am pretty confident that all the code now runs on Chrome, Firefox and Safari.

Line 55 of

<http://cs.unm.edu/~angel/WebGL/7E/03/rotatingSquare2.js> should read

```
switch( event.target.index ) {
```

Simuarly, line 59 of

<http://cs.unm.edu/~angel/WebGL/7E/03/rotatingSquare3.js> should read

```
switch( event.target.index ) {
```

and lines 51 through 53 should read

```
document.getElementById("slider").onchange =  
function(event) {  
    speed = 100 - event.target.value;  
};
```

Line 28 of

<http://cs.unm.edu/~angel/WebGL/7E/03/square.js> should read

```
canvas.addEventListener("mousedown",  
function(event) {
```

Line 54 of <http://cs.unm.edu/~angel/WebGL/7E/03/triangle.js> should read

```
canvas.addEventListener("click", function(event)  
{
```

Line 66 of <http://cs.unm.edu/~angel/WebGL/7E/03/cad1.js> should read

```
canvas.addEventListener("mousedown",
```

```
function(event) {
```

Line 46 of

<http://cs.unm.edu/~angel/WebGL/7E/03/cad2.js> should read

```
canvas.addEventListener("mousedown",  
function(event) {
```

Lines 171 to 185 in

<http://cs.unm.edu/~angel/WebGL/7E/07/textureSquare.js> should read

```
document.getElementById("zFarSlider").onchange =  
function(event) {  
    far = event.target.value;  
};
```

```
document.getElementById("zNearSlider").onchange  
= function(event) {  
    near = event.target.value;  
};
```

```
document.getElementById("aspectSlider").onchange  
= function(event) {  
    aspect = event.target.value;  
};
```

```
document.getElementById("fovSlider").onchange =  
function(event) {  
    fovy = event.target.value;  
};
```

```
document.getElementById("Texture  
Style").onclick = function( event) {  
    switch(event.target.index) {
```

Line 147 of

<http://cs.unm.edu/~angel/WebGL/7E/07/pickCube.js> should read

```
canvas.addEventListener("mousedown",  
function(event) {
```

Line 183 of

<http://cs.unm.edu/~angel/WebGL/7E/07/pickCube2.js> should read

```
canvas.addEventListener("mousedown",  
function(event) {
```

Line 217 of

<http://cs.unm.edu/~angel/WebGL/7E/07/pickCube3.js> should read

```
canvas.addEventListener("mousedown",  
function(event) {
```

Line 187 of

<http://cs.unm.edu/~angel/WebGL/7E/07/pickCube4.js> should read

```
canvas.addEventListener("mousedown",  
function(event) {
```

Lines 146 though 154 of

<http://cs.unm.edu/~angel/WebGL/7E/09/robotArm.js> should read

```
document.getElementById("slider1").onchange  
= function(event) {  
    theta[0] = event.target.value;  
};  
document.getElementById("slider2").onchange  
= function(event) {  
    theta[1] = event.target.value;  
};  
document.getElementById("slider3").onchange  
= function(event) {  
    theta[2] = event.target.value;  
};
```

Lines 330 to 375 of <http://cs.unm.edu/~angel/WebGL/7E/09/figure.js>
should read

```

document.getElementById("slider0").onchange =
function(event) {
    theta[torsoId ] = event.target.value;
    initNodes(torsoId);
};

document.getElementById("slider1").onchange =
function(event) {
    theta[head1Id] = event.target.value;
    initNodes(head1Id);
};

    document.getElementById("slider2").onchange
= function(event) {
        theta[leftUpperArmId] =
event.target.value;
        initNodes(leftUpperArmId);
    };
    document.getElementById("slider3").onchange
= function(event) {
        theta[leftLowerArmId] =
event.target.value;
        initNodes(leftLowerArmId);
    };

document.getElementById("slider4").onchange =
function(event) {
    theta[rightUpperArmId] =
event.target.value;
    initNodes(rightUpperArmId);
};
    document.getElementById("slider5").onchange
= function(event) {
        theta[rightLowerArmId] =

```

```

event.target.value;
    initNodes(rightLowerArmId);
};

document.getElementById("slider6").onchange =
function(event) {
    theta[leftUpperLegId] =
event.target.value;
    initNodes(leftUpperLegId);
};
document.getElementById("slider7").onchange
= function(event) {
    theta[leftLowerLegId] =
event.target.value;
    initNodes(leftLowerLegId);
};
document.getElementById("slider8").onchange
= function(event) {
    theta[rightUpperLegId] =
event.target.value;
    initNodes(rightUpperLegId);
};

document.getElementById("slider9").onchange =
function(event) {
    theta[rightLowerLegId] =
event.target.value;
    initNodes(rightLowerLegId);
};
document.getElementById("slider10").onchange
= function(event) {
    theta[head2Id] = event.target.value;
    initNodes(head2Id);
};

```

Lines 65 through 76 of <http://cs.unm.edu/~angel/WebGL/7E/10/>

[mandelbrot2.js](#) should read

```
document.getElementById("Center X").onchange
= function(event) {
    cx = event.target.value;

gl.uniform1f( gl.getUniformLocation(program,
"cx"), cx);
};
document.getElementById("Center Y").onchange
= function(event) {
    cy = event.target.value;

gl.uniform1f( gl.getUniformLocation(program,
"cy"), cy);
};
document.getElementById("Size").onchange =
function(event) {
    scale = 1.0/event.target.value;

gl.uniform1f( gl.getUniformLocation(program,
"scale"), scale);
};
```

Change to ortho.js in MV.js (10/5/2014)

The entries in the last column of of the matrix need minus signs in front

Chapter 6 examples (11/29/2014)

November 29, 1014

All the shaded sphere examples have been updated. There were errors

in the JS html files due to not forcing the w component of normals to be 0.0. Also one of the transformations in the html files needed to be changed.

Note that in these examples, the light source position is not changed when theta and phi are moved. Thus the viewer is fixed and the object is rotated. The light source position should be interpreted as the position of the light source in eye coordinates.

Chapter 11 teapot examples 4 and 5
Dec 1, 2014

Made consistent with lighting code in Chapter 6

Chapters 7 and 11

The reflection examples in Chapter 7 and the teapot examples in Chapter 11 have been updated to reflect the changes in Chapter 6.

Dec 3, 2014

Lines 516 and 608 in MV.js: make result a var

swap (1-s) and s in mix function to consistent with GLSL

March 1, 2015

Added new functionality to MV.js including matrix inverses and determinants for 2 x 2, 3 x 3 and 4 x 4 matrices.

Added a normal matrix function

Added a matrix print function for debugging

Changed the name of the matrix scale function to `scalem` since there were two functions named `scale` in the original MV.js.

Adding new examples to Chapters 3, 4, 6, 7 and 11. See individual README files

May 3, 2015

The rotating square examples for Chapter 3 were all updated to correct two errors that cancelled each other out. The first error was the order of vertices. As in the original code, the first triangle in the triangle strip was defined in clockwise manner which in WebGL defines a back face, which leads to all the triangles in the strip being back facing. This would cause a problem if culled is turned on. Second, the rotation had the x and y reversed making it a concatenation of a reflection and a rotation, which cancelled out the first error. All three examples have been fixed. Thanks to Ken Lodge (Charles Sturt University, Australia) for pointing out these errors.

May 15, 2015

Fixed all particle diffusion examples in Chapters 7 and 10 so they now run on Macs, Windows and Linux

May 22, 2015

Updated all slider code in interactive examples to use `event.target` instead of `event.srcElement`.

July 17, 2015

Cleaned up some html tags

July 18, 2015