



CCWC 2023

Aspects of VR WebXR Exercises

Phillip G. Bradford
University of Connecticut

Some OpenGL

```
g++ triangle.cpp -o triangle -lglut -lGLU -lGL
```

Update the triangle to be a different shape

```
g++ rectangle.cpp -o rectangle -lglut -lGLU -lGL
```

Show changes to the rectangle based on modifying the matrix

Some WebGL

Modify canvas-gettransform-settransform

To get other objects on the screen with different translations

<canvas></canvas> Tags: Rakesh Baruah

Drawing 2D and 3D graphics – render state and behavior

```
<body>
  <h3>WebGL</h3>
  <canvas id='myCanvas'></canvas>
</body>
<script type="text/javascript">
  const g1 = myCanvas.getContext('webgl');
  if (!g1) {
    alert('WebGL not available');
    console.log('WebGL not available');
  } else {
    alert('WebGL good');
    console.log('WebGL good');
  }
</script>
```

canvas-gettransform-settransform

```
<canvas></canvas>
```

```
<canvas></canvas>
```

```
<script>
```

```
const canvases = document.querySelectorAll('canvas');
```

```
const ctx1 = canvases[0].getContext('2d');
```

```
const ctx2 = canvases[1].getContext('2d');
```

```
ctx1.setTransform(1, .2, .8, 1, 0, 0); // h-scale, v-skew, h-skew, v-scale, h-trans, v-trans
```

```
ctx1.fillRect(25, 25, 50, 50); // fillRect(x, y, width, height)
```

```
let storedTransform = ctx1.getTransform();
```

```
console.log(storedTransform);
```

```
ctx2.setTransform(storedTransform);
```

```
ctx2.beginPath();
```

```
ctx2.arc(50, 50, 50, 0, 2 * Math.PI); // arc(x, y, radius, startAngle, endAngle)
```

```
ctx2.fill();
```

```
</script>
```

OpenGL - http://www.dgp.toronto.edu/~ah/csc418/fall_1999/tut/square

```
#include <stdio.h>
#include <GL/glut.h>
void display(void)
{
    glClear( GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.0, 0.0);
    glBegin(GL_POLYGON);
        glVertex3f(2.0, 4.0, 0.0);
        glVertex3f(8.0, 4.0, 0.0);
        glVertex3f(8.0, 6.0, 0.0);
        glVertex3f(2.0, 6.0, 0.0);
    glEnd();
    glFlush();
}
```

OpenGL -

http://www.dgp.toronto.edu/~ah/csc418/fall_1999/tut/square

```
int main(int argc, char **argv) {
    printf("hello world\n");
    glutInit(&argc, argv);
    glutInitDisplayMode( GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);

    glutInitWindowPosition(100,100);
    glutInitWindowSize(300,300);
    glutCreateWindow("square");

    glClearColor(1.0, 1.0, 1.0, 0.0);    // black background
    glMatrixMode(GL_PROJECTION);          // setup viewing projection
    glLoadIdentity();                     // start with identity matrix
    glOrtho(0.0, 10.0, 0.0, 10.0, -1.0, 1.0); // setup a 10x10x2 viewing world

    glutDisplayFunc(display);
    glutMainLoop();

    return 0;
}
```