POLITECNICO
MILANO 1863

Exercises 03

Computer Graphics 2021

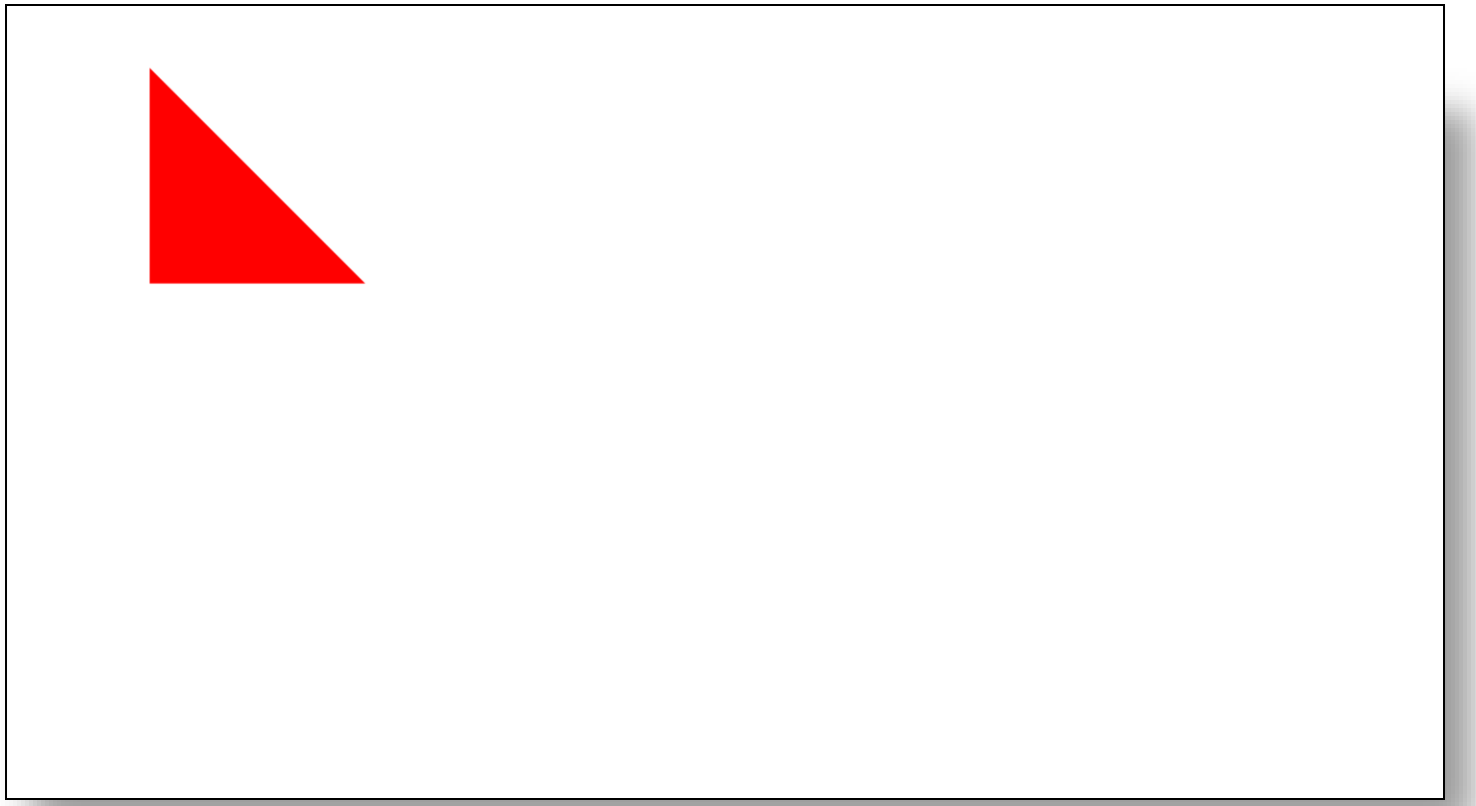
Erica Stella (erica.stella@polimi.it)

DISCLAIMER

- These exercises are purely for learning purposes, they will NOT be asked during the exams
- Don't worry if you don't finish them today, solutions will be posted, but **please review them before the next lesson** because we will build upon them
- If you have questions, you can use the forum or ask me next time ☺
- Before plunging into the code, please open the file for a quick look ☺

Ex 1

- Translate the triangle -0.8 on the x axis and 0.3 on the y axis, and pass the translation matrix to the **`matrix`** GLSL uniform
- Hint: use the **`MakeTranslateMatrix()`** function in *utils.js*
- Final look:



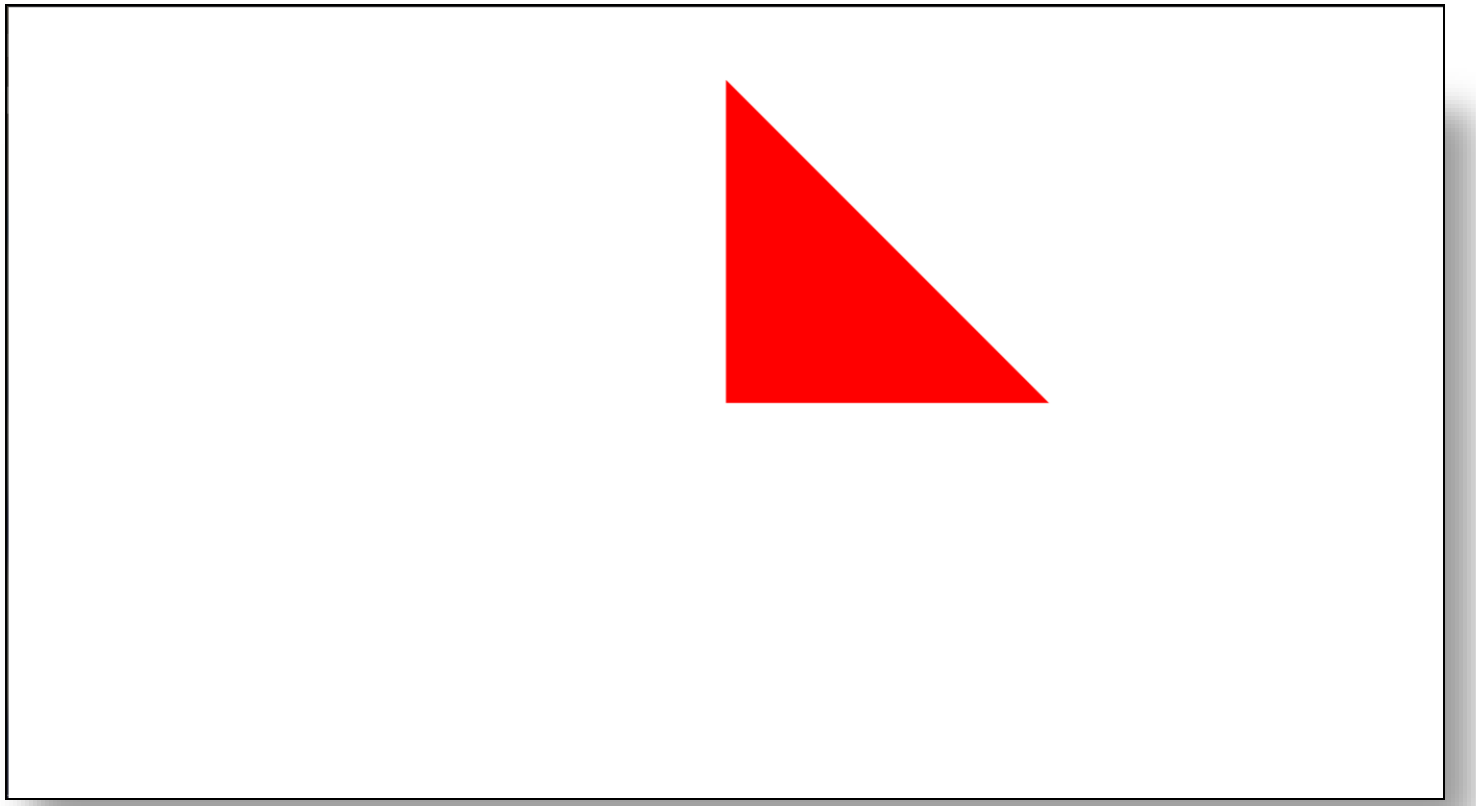
Solution Ex 1

script.js

```
function main() {
  [..]
  var matrixLocation = gl.getUniformLocation(program, "matrix");

  [..VBO setup for vertices and indices..]
  gl.useProgram(program);

  var matrix = utils.MakeTranslateMatrix(-0.8,0.3,0.0);

  //Pay attention! This line must be after "useProgram" otherwise
  //webgl is not able to find the matrixLocation, and then to set its value
  gl.uniformMatrix4fv(matrixLocation, gl.FALSE, utils.transposeMatrix(matrix));

  //Just to be sure this is the currently bound IBO
  gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);
  gl.drawElements(gl.TRIANGLES, indices.length, gl.UNSIGNED_SHORT, 0 );
}
```

Ex 2

- Scale the triangle 1.5 times and pass the scaling matrix to the **`matrix`** GLSL uniform
- Hint: use the **`MakeScaleMatrix()`** function in *utils.js*
- Final look:



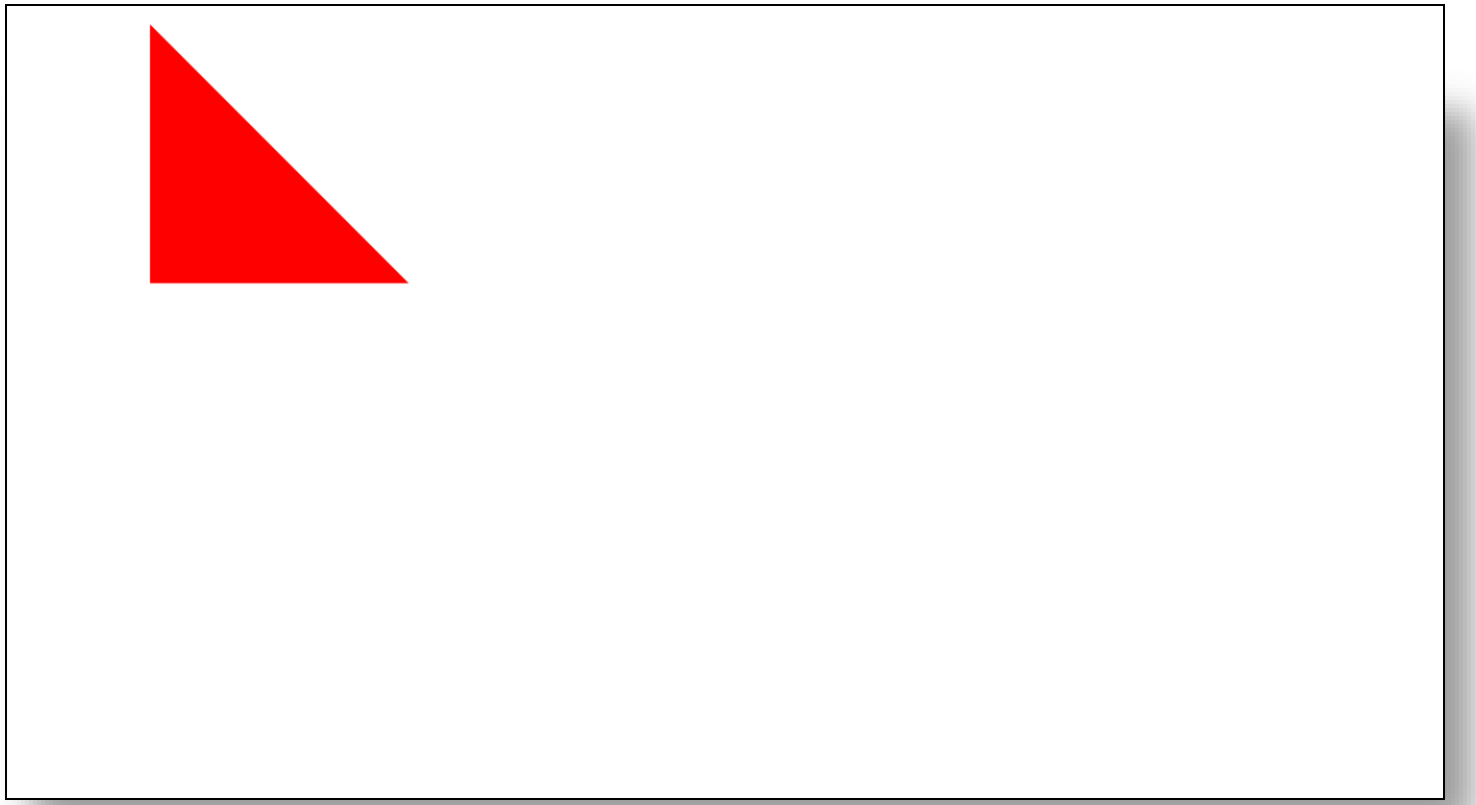
Solution Ex 2

script.js

```
function main() {
  [..]
  var matrixLocation = gl.getUniformLocation(program, "matrix");

  [..VBO setup for vertices and indices..]
  gl.useProgram(program);

   var matrix = utils.MakeScaleMatrix(1.5);

  //Pay attention! This line must be after "useProgram" otherwise
  //webgl is not able to find the matrixLocation, and then to set its value
  gl.uniformMatrix4fv(matrixLocation, gl.FALSE, utils.transposeMatrix(matrix));

  //Just to be sure this is the currently bound IBO
  gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);
  gl.drawElements(gl.TRIANGLES, indices.length, gl.UNSIGNED_SHORT, 0 );
}
```

Ex 3

- Scale the triangle 1.2 times, apply this translation (x:-0.8, y:0.3), and pass the resulting matrix to the **`matrix`** GLSL uniform
- Hint: you can use the **`multiplyMatrices()`** function in utils.js but it is not the only way ☺
- Final look:



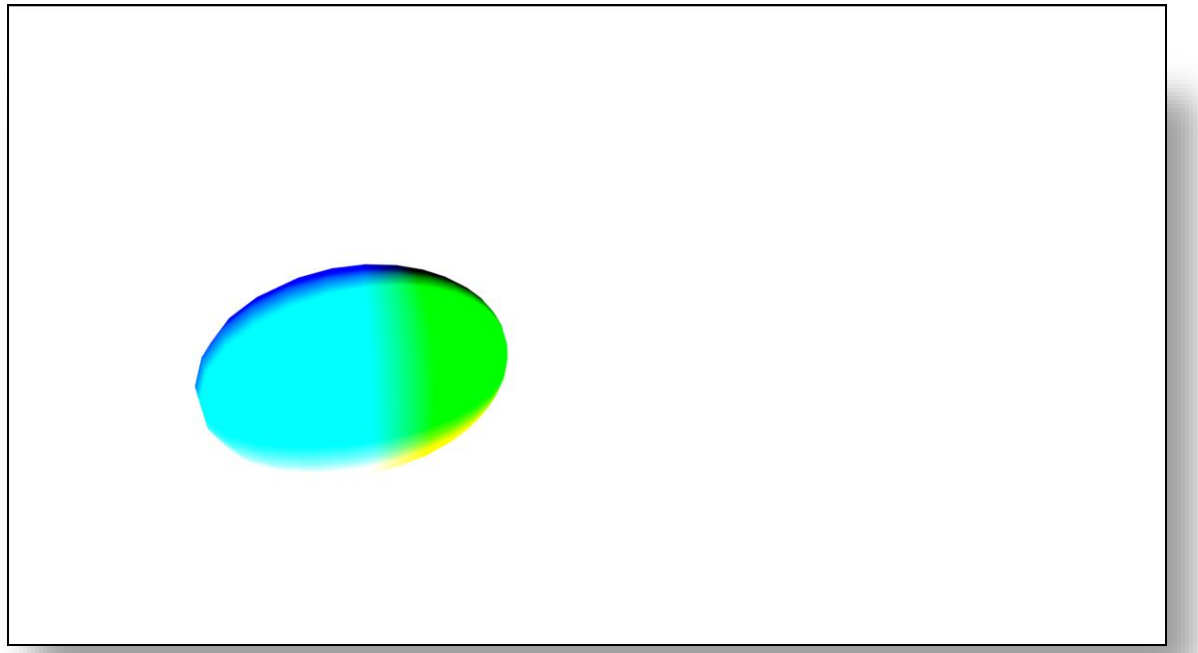
Solution Ex 3

script.js

```
function main() {
  [..]
  var matrixLocation = gl.getUniformLocation(program, "matrix");

  [..VBO setup for vertices and indices..]
  gl.useProgram(program);

  //Can be done either with these 2 matrices multiplied together or..
  //var scaleMatrix = utils.MakeScaleMatrix(1.2);//*****NEW*****//
  //var translationMatrix = utils.MakeTranslateMatrix(-0.8,0.3,0.0);//*****NEW*****//
  //var matrix = utils.multiplyMatrices(translationMatrix, scaleMatrix);//*****NEW*****//

  //Just with the MakeWorld function which implicitly multiplies the scaling and translatio
n matrices
  var matrix = utils.MakeWorld(-0.8, 0.3, 0.0, 0.0, 0.0, 0.0, 1.2); //*****NEW*****//

  gl.uniformMatrix4fv(matrixLocation, gl.FALSE, utils.transposeMatrix(matrix));
  gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);
  gl.drawElements(gl.TRIANGLES, indices.length, gl.UNSIGNED_SHORT, 0 );
}
```

Ex 4

- Make a World-View-Projection matrix for the sphere and pass the resulting matrix to the **`matrix`** GLSL uniform
  - World Matrix: Translation (x:-10, y:3, z:-5), Rotation (x:20, y:47, z:110), Scale 2.0
  - View Matrix: Camera position (x:0, y:20, z:10) Elev:15, Angle:10
  - Perspective Matrix: Fovy 120 Near 0.1 Far 100
- Final look:



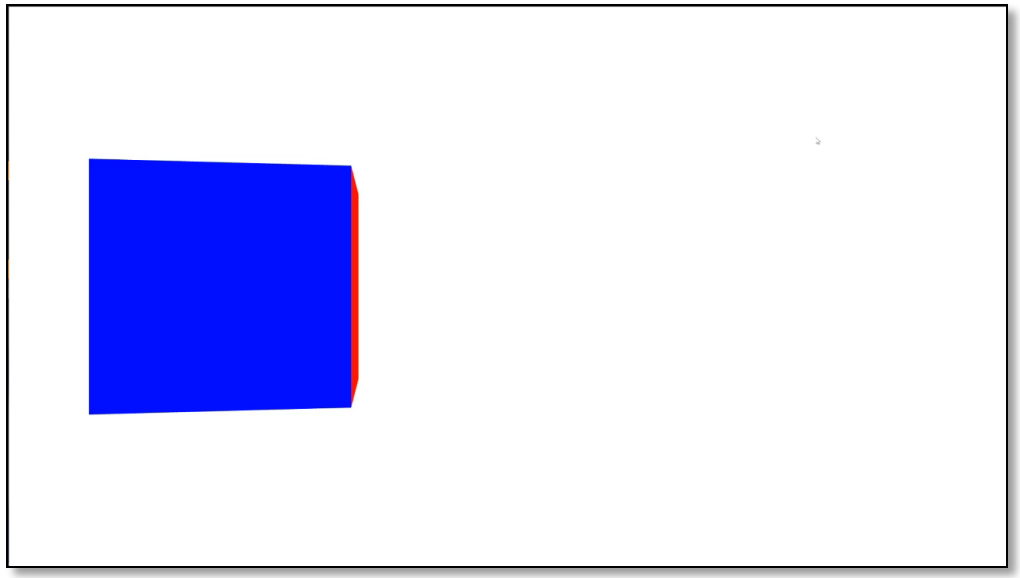
Solution Ex 4

script.js

```
function main() {
  [..]
  var matrixLocation = gl.getUniformLocation(program, "matrix");

  [..VBO setup for vertices and indices..]
  gl.useProgram(program);

  var worldMatrix = utils.MakeWorld(-10.0, 3.0, -5.0, 20.0, 47.0, 110.0, 2.0);
  var viewMatrix = utils.MakeView(0, 2.0, 10.0, 15.0, 10.0);
  var perspectiveMatrix = utils.MakePerspective(120, gl.canvas.width/gl.canvas.height, 0.1,
     100.0);
  var viewWorldMatrix = utils.multiplyMatrices(viewMatrix, worldMatrix);
  var projectionMatrix = utils.multiplyMatrices(perspectiveMatrix, viewWorldMatrix);

  gl.uniformMatrix4fv(matrixLocation, gl.FALSE, utils.transposeMatrix(matrix));
  gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);
  gl.drawElements(gl.TRIANGLES, indices.length, gl.UNSIGNED_SHORT, 0 );
}
```

Ex 5

- Write the **`keyFunction()`** function so that:
  - The 6 button makes the cube rotate around the z axis of -5.0 degrees
  - The 7 button makes the cube rotate around the z axis of +5.0 degrees
  - The 1 button makes the cube decrease its scale of 0.1
  - The 2 button makes the cube increase its scale of 0.1
- Hint: https://keycode.info/
- Final look:



Solution Ex 5

script.js

```
var S = 1.0;
var Rz = 0.0;

function drawScene() {
  [..]

  worldMatrix = utils.MakeWorld(0,0,0,0,0,Rz,S);

  [..]
}
```

```
function keyFunction(e){
    if (e.keyCode == 54) {  // 6
      Rz-=5.0;
    }
    if (e.keyCode == 55) {  // 7
      Rz+=5.0;
    }
    if (e.keyCode == 49) {  // 1
      S-=0.1;
    }
    if (e.keyCode == 50) {  // 2
      S+=0.1;
    }
    //Remember to place this here
    //otherwise you won't see the
    //changes ☺
    window.requestAnimationFrame(drawScene);
}
```

Ex 6

- Use VAOs instead of VBOs for drawing the cube. The base code is the same as the one in the solution of the previous exercise.
- Final look: same as Exercise 5

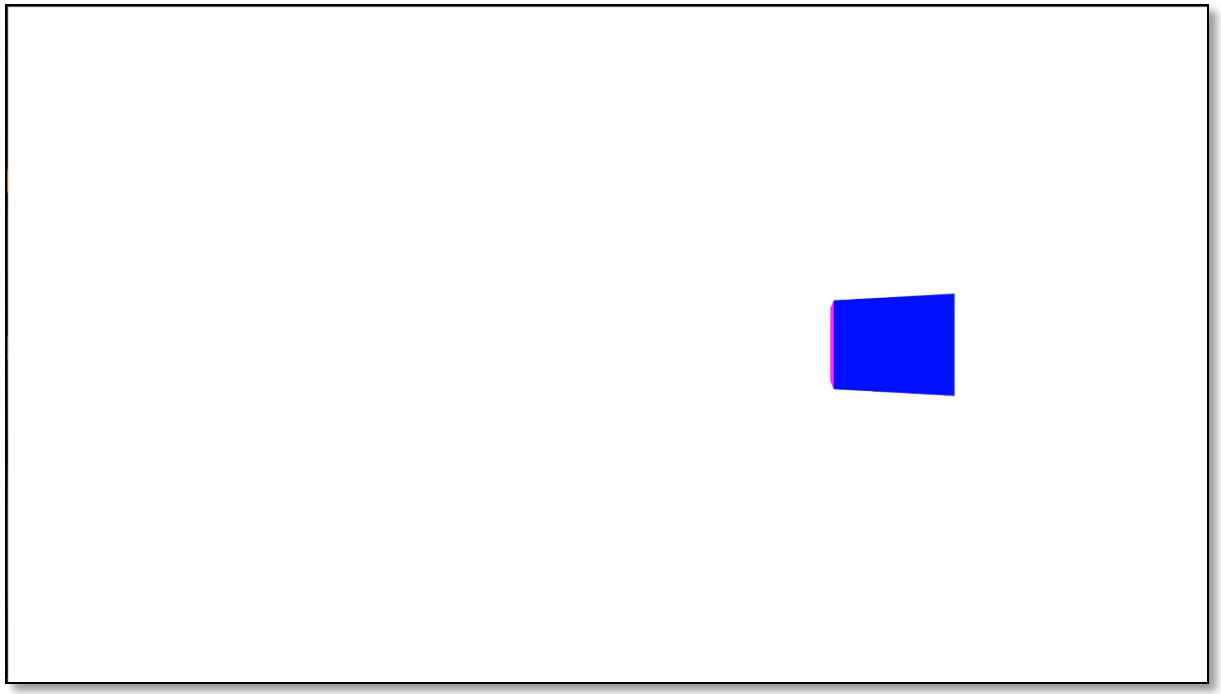
Solution Ex 6

script.js

```
function main() {
  [..]
  vao = gl.createVertexArray();
  gl.bindVertexArray(vao);

  [Set up all VBOs for vertices, colours, and indices]

  drawScene();
}
function drawScene() {
  gl.bindVertexArray(vao);
  [Set up all matrices and do the draw call]
}
```

Ex 7

- Modify the **`animation()`** function to make the cube translate on the x axis between -1.5 and 1.5 with a **`deltaC`** of 0.003 * time_between_frames
- Final look:



Solution Ex 7

script.js

```
function animate() {
  var currentTime = (new Date).getTime();
  if (lastUpdateTime) {
    //currentTime - lastUpdateTime is the time passed between frames
    var deltaC = (3 * (currentTime - lastUpdateTime)) / 1000.0;

    if (flag == 0) cubeTx += deltaC;
    else cubeTx -= deltaC;

    if (cubeTx >= 1.5) flag = 1;
    else if (cubeTx <= -1.5) flag = 0;
  }
  worldMatrix = utils.MakeWorld(cubeTx, cubeTy, cubeTz, cubeRx, cubeRy, cubeRz, cubeS);
  lastUpdateTime = currentTime; //Need to update it for the next frame
}
```