Lesson 1

João Fonseca -103154

Diogo Paiva - 103183

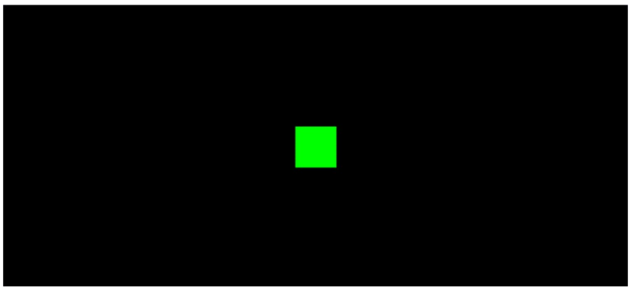
Information Visualization, 2023 (MSc Computer Science and Engineering, University of Aveiro)

Introduction

This report addresses the exercises done in Lesson 1 of Three.js.

Exercise 1

We were asked to develop a simple scene with a cube.

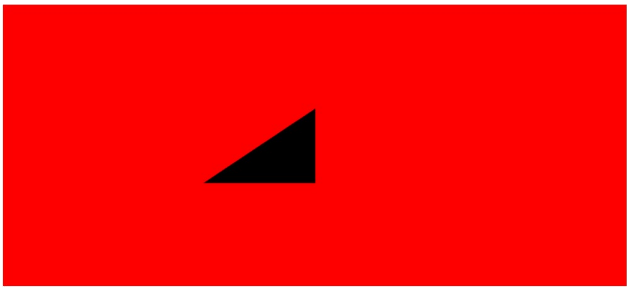


Exercise 2

Change previous example to visualize a black 2D triangle.

To modify the code to obtain the same scene without modifying the camera position we changed the position in the z axis to -5.

```
const vertices = new Float32Array( [
    -3.0, -1.0,  -5.0,
    0.0, -1.0,  -5.0,
    0.0,  1.0,  -5.0,
] );
```



Exercise 3

Allow the mapping of different colours in a mesh face and modify the previous scene to match the figure presented by the professor.

We had to use the "side" flag in the material of 2 of the triangles with the argument THREE.DoubleSide because only triangles with normal facing towards the camera are rendered.

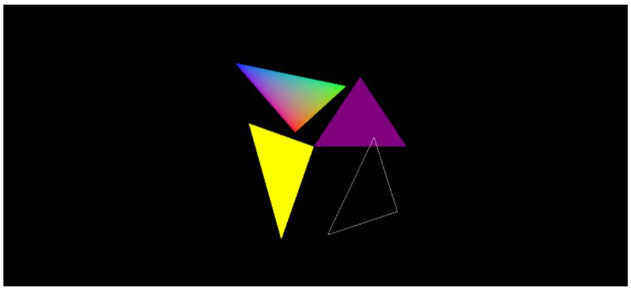
An alternative and more elegant way to solve the issue is:

Adjusting Normals: Ensure that the normals of the geometry are consistent and facing outward. In Three.js, you can recalculate normals using geometry.computeVertexNormals() to ensure they're facing the correct way.

```
material2.side = THREE.DoubleSide;
```

And for the last triangle we add to the property wireframe in its MeshBasicMaterial.

```
material4.wireframe = true;
```



Exercise 4

Update the viewport each time the window size is modified.

```
window.addEventListener('resize', () => {
    renderer.setSize( window.innerWidth,
window.innerHeight );
    camera.aspect = window.innerWidth /
window.innerHeight;
    camera.updateProjectionMatrix();
});
```

Exercise 5

Modify the first example to show the cube in wireframe and visualize 4 other geometries changing some of their default parameters.