

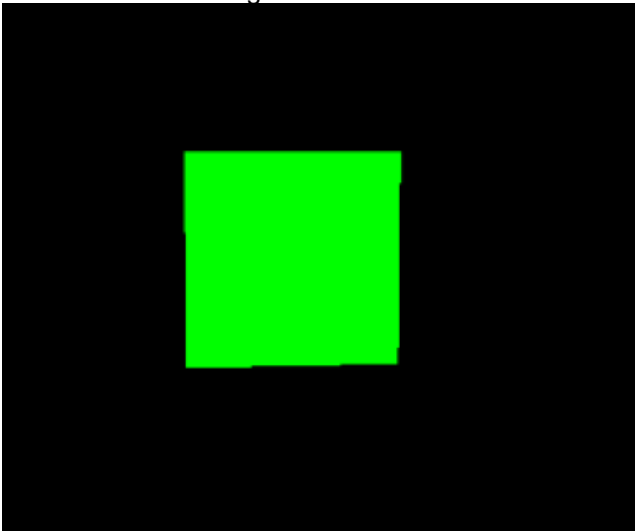
Three.js - Lesson 1: Introduction

João Farias, 98679 and Ricardo Rodriguez, 98388

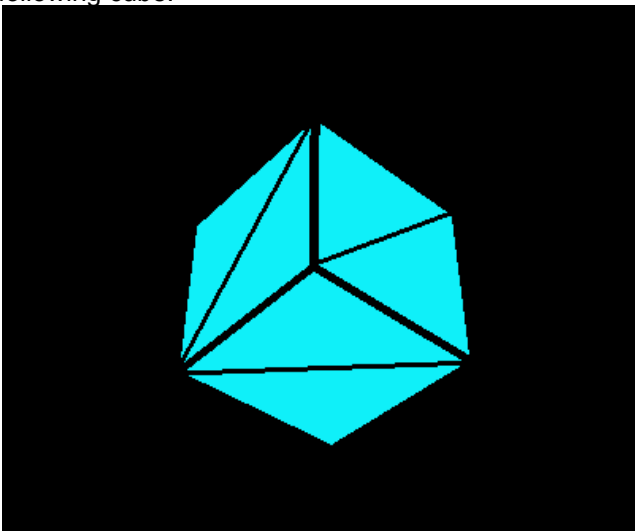
Information Visualization, 2022/2023 MSc Software Engineer, University of Aveiro

Exercise 1

The first exercise, after following the code available, we reach the following cube:

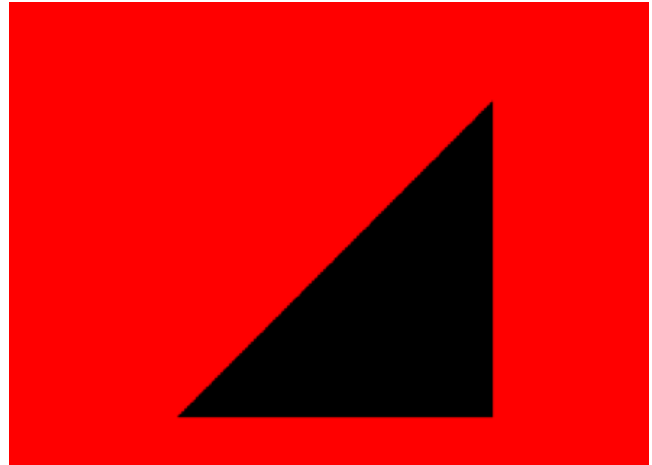


This uses a Box Geometry giving it a green material color and applying a simple rotation in the x and y axis. We decided to change the cube and reach the following cube:



This cube has the wireframe property, which is represented by the black lines.

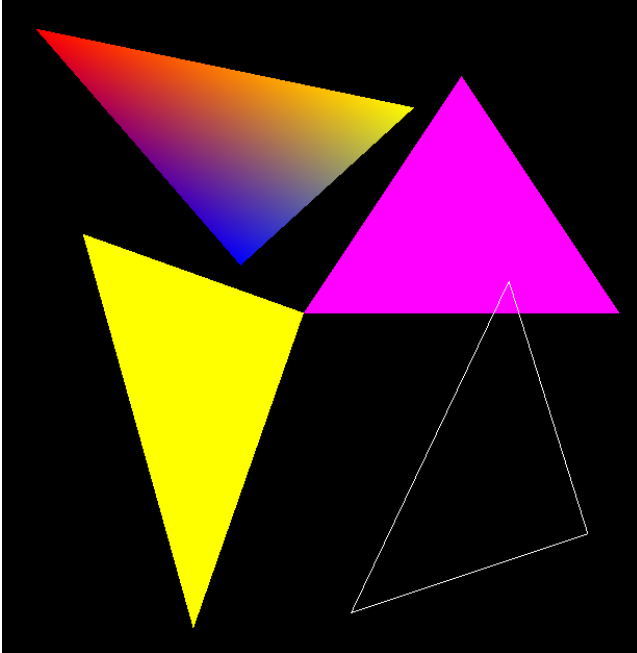
Exercise 2



To complete this exercise, instead of using BoxGeometry, we used BufferGeometry. We also defined the vertex coordinates through an array, applying them by setting the position attribute to our coordinate array.

About the color, the triangle was changed to black and the background was changed to red by simply changing the color's hex value.

Exercise 3



In this exercise, we define the top left triangle with `vertexColors` equals **True** which colors the triangle with this fade effect. The vertex has the following colors: red, yellow and blue. The top right triangle and the bottom left triangle wouldn't be shown if the property `side` was not set as `DoubleSide`. The last triangle's structure is a wireframe. This is done by changing the default value of `wireframe` to `true`.

Exercise 4

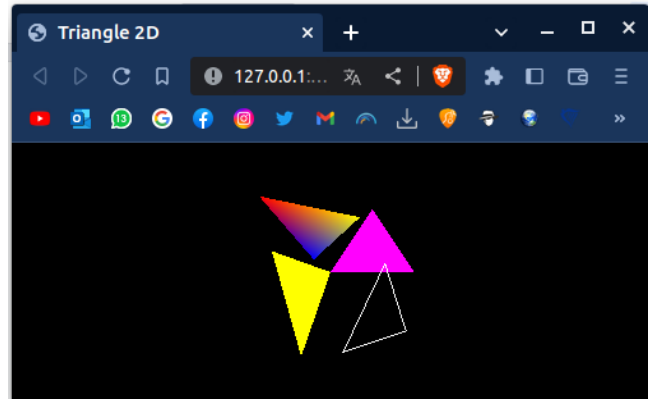
To update the Viewport we defined the following code:

```
window.addEventListener("resize", onResize, false);

function onResize() {
  camera.aspect = window.innerWidth / window.innerHeight;
  camera.updateProjectionMatrix();
  renderer.setSize(window.innerWidth, window.innerHeight);
}
```

This adds an Event Listener that will resize the window, whenever the window changes. If this code was not added, the triangles would

disappear if the browser window was resized.



As you can see, the triangles are smaller than the previous ones, once the window is smaller too.