# Introduction

I did a thing, pray it works.

# Body

## Requirement 1: Draw a simple cube

This requirement served as an introduction to using three.js, the premise of it was simple: make a cube, put it in the centre, add it to the scene. This was done as follows:

# 

With the result being visible in figure 1:

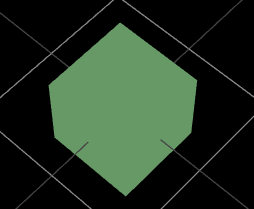
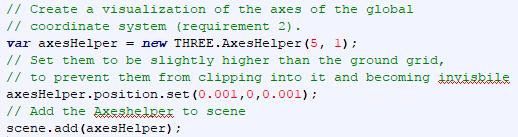
This is a simple introduction to using WebGL and three.js. Later it is covered why MeshBasicMaterial might not be the best choice due to shading issues and similar. The cube is centred at origin (0,0,0) with opposite corner points (-1,-1,-1) & (1,1,1). Faces are orthogonal to the x-, y-, and z- axes. No real difficulties were encountered here.

Figure 1x1x1 Cube

## Requirement 2: Draw coordinate system axes

The objective here is to visualise a three-dimensional axes system using three orthogonal lines in red, green and blue, to make it easier to see X, Y and Z further on. This was achieved as follows:



With the result being in figure 2:

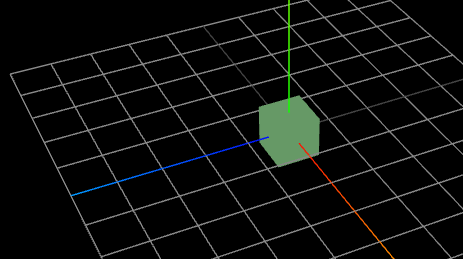
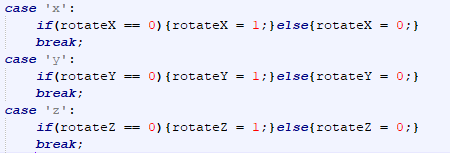
In figure 2 it is clear we achieved the 3 orthogonal lines, stemming out of the centre and going to the end of the grid. The only difficulty encountered with this was the fact that the grid tended to overlap the axes – this was fixed by offsetting the axes position by a negligible amount, so that the system knows that they are to be displayed on top, without significantly altering their position as to cause it to be in an erroneous place in the plane.

Figure XYZ indicator axes

## Requirement 3: Rotate the cube

The premise of this requirement is to be able to press a button and have the cube rotate. Initially I had opted to hold the button down and have the rotation increment each time, but instead I put a variable that allowed it to repeat infinitely until the button was next pressed. Achieved as follows:  
  


So when x, y or z is pressed, a variable (rotateXYZ) will flip from 0 to 1 or 1 to 0. In the animation part the following is put:  

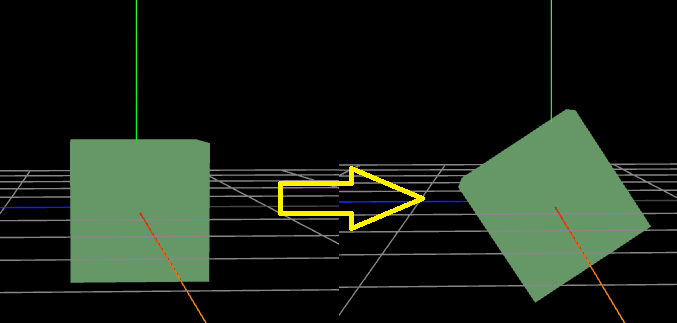

In figure 3 there is an example of rotating around the X-Axis. The program allows you to rotate in all directions, as well as reset it by pressing R, which simply resets all cube.rotation.xyz back to 0.

Figure 3 Rotating the cube around the X axis

## Requirement 4: Different Render Modes

# Conclusion 3

# 