CSE4204 | Section C | Computer Graphics Lab | Assignment – 3

[10 marks] Part A:

Create a 3D cube using index buffer. Color of a vertex is the absolute value of its coordinates, i.e., absolute values of X, Y and Z coordinates will be treated as the R, G and B values respectively of its color attribute. In addition to that, you must introduce a border for the object as shown in the diagrams below.

For pressing LEFT and RIGHT arrow keys, the cube will rotate (+ve) along the Y and X-axis respectively. And for pressing the UP and DOWN arrow keys, the border will increase and decrease respectively. [See the lower right image of the diagram below.]

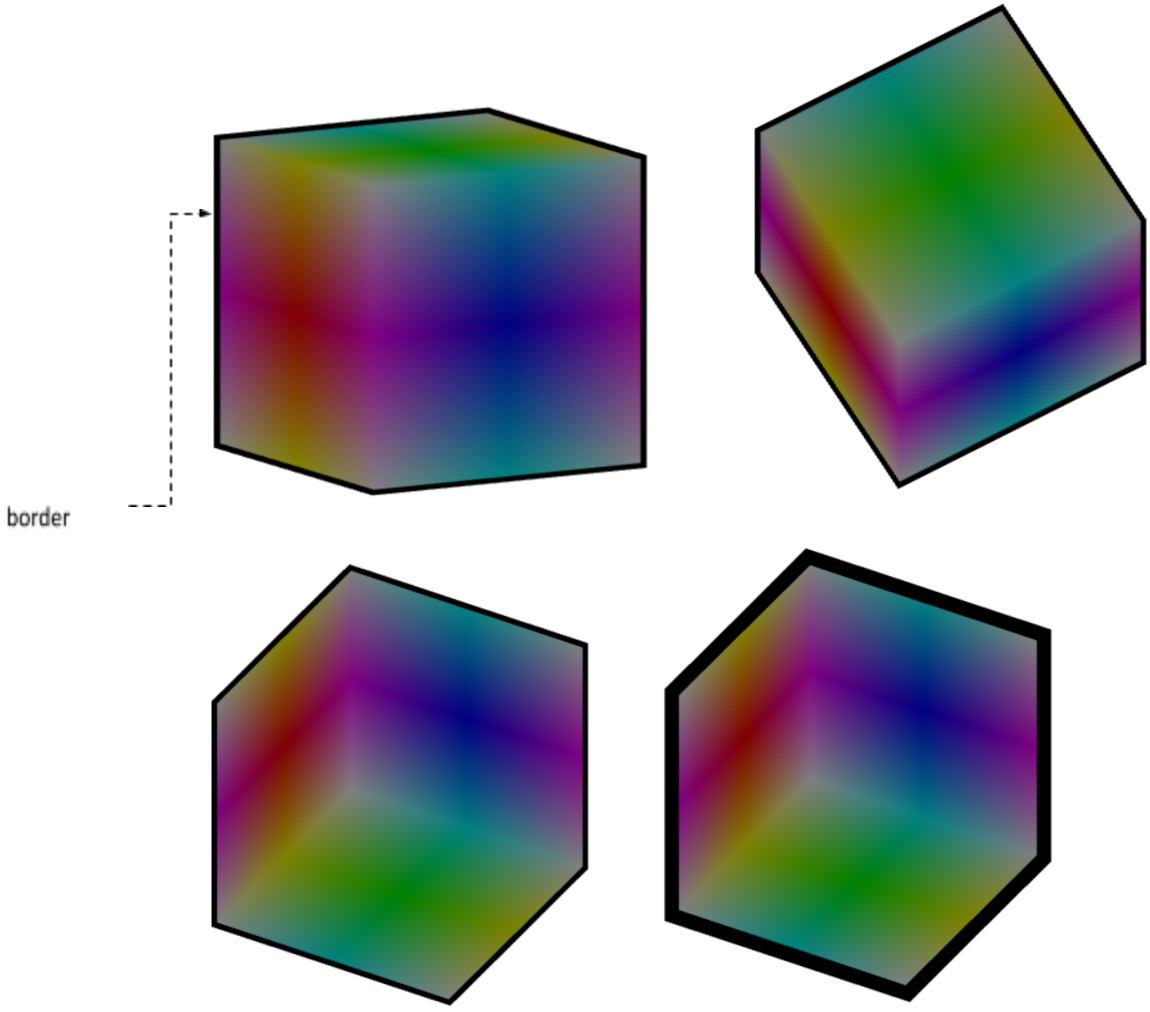

Figure: Different Example States of the 3D Cube. **Video Demo is Available Here:** shorturl.at/sQWY9

Hints:

- Do not define color information outside the shader specifically, rather reuse the coordinate information inside the shaders.
- To draw the border, you can call the draw functions two times. One will draw the cube with black color, but being slightly scaled up. For the next call, the cube will be drawn with faces' colors, but not being scaled. This difference between scaling factors will be appeared as a border in the canvas.
- To handle the colors for two different draw calls, you can use control statements inside shaders that will switch between different `gl_FragColor`.
- Be careful while using `gl.COLOR_BUFFER_BIT` for second draw call.

[10 marks] Part B:

Create two 3D Cubes and continuously rotate them along the positive X-axis. The cubes must be placed side by side and the colors of the cubes must be different. You have the freedom to choose the colors. The rotation speed of one cube must be greater than the other. Apply perspective projection on them. Also, you should be able to change the camera position by key pressing. For pressing up, down, left and right arrow keys the camera will move upwards, downwards, left and right respectively. For pressing A and S buttons the camera will move towards the viewing direction and away from the viewing direction respectively.

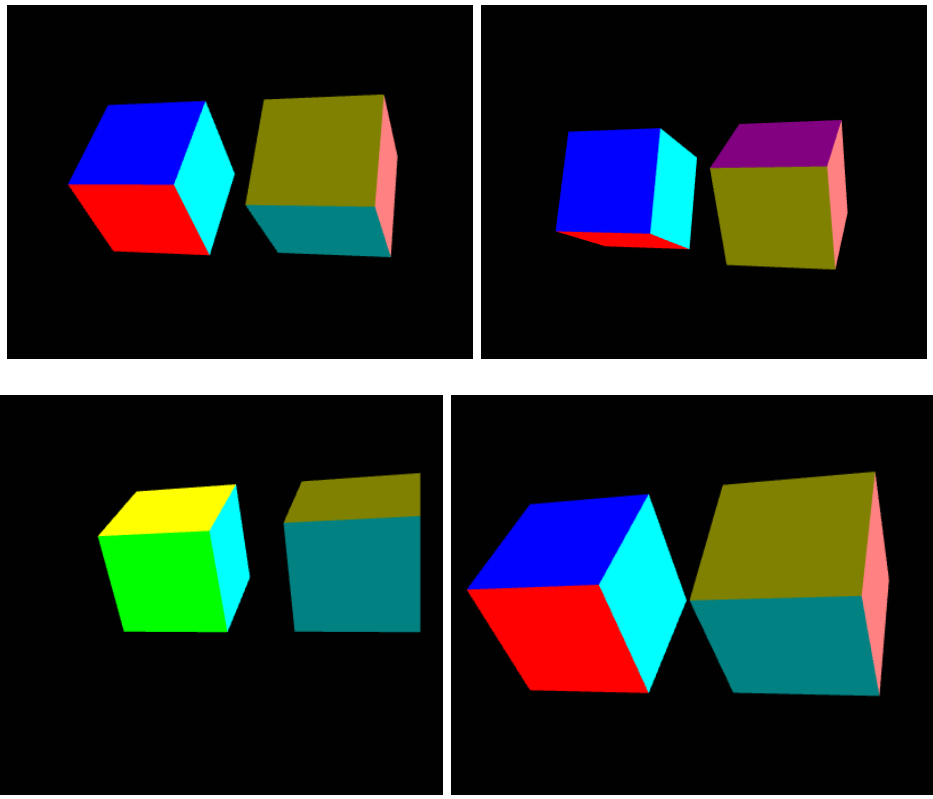

Figure: Different Example States of the 3D Rotating Cube for different camera positions. **Video Demo is Available Here:** shorturl.at/cwEO8

Hints:

- You can use window.requestAnimationFrame(callback) method to continuously rotate the cubes
- For drawing two cubes you can use multiple shaders
- For rotation, you only need to update the parameters of the rotation matrix

Evaluation: Coding + Viva