INB381 – Shaders

Download Week4-workshop.zip

Extract the files and folders, as you will modify them to complete this week's workshop. We will discuss the code and our alterations.

Exercise 1

Modify the provided code to change the appearance from Figure 1 to Figure 2.

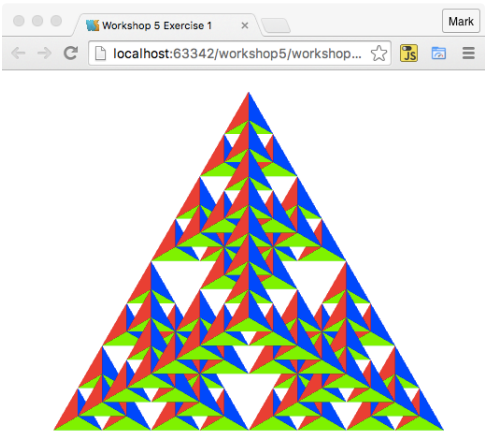


Figure 1.

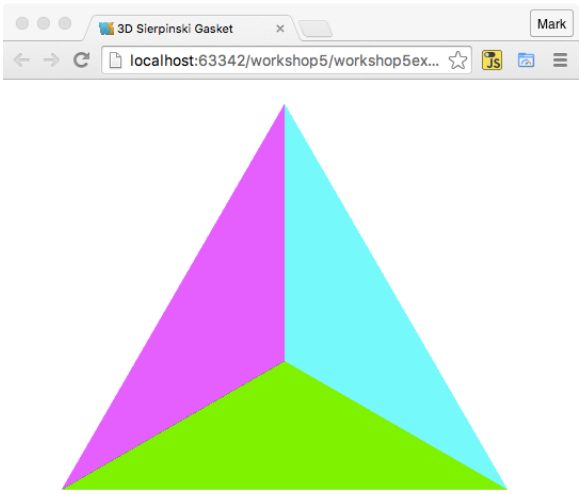
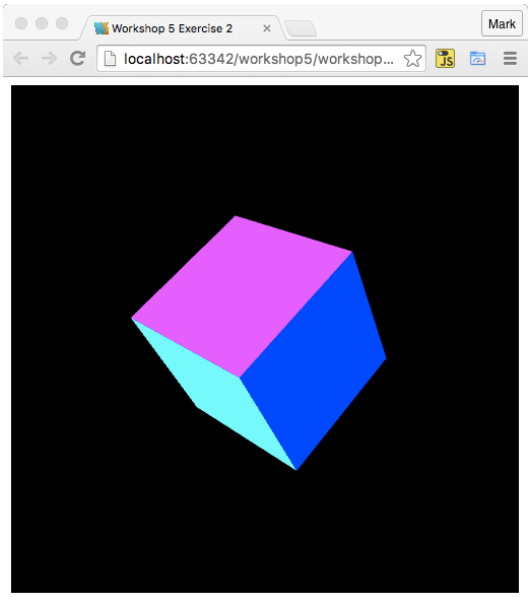


Figure 2

Exercise 2

1. Complete the vertex data for the missing faces of the cube
2. Now change the colour of the faces.



Exercise 3

Make sure you can access the central server, Fastapps04.qut.edu.au, which will be used for submitting your assignments. The instructions are listed in the 'Server Access' file.

Picking Review

Examine LightingPlusPicking.html and LightingPlusPicking.js files to review how to perform picking.

The basic theory is as follows:

1. Create a framebuffer that you can render to so that you don't affect the screen,
2. Render the scene to the framebuffer using a different colour for each object. (Don't forget to turn lighting effects off if you do this since they will affect the color),
3. Read the pixel colour at the position where the user's mouse was clicked,
4. Compare the colour to the colours of the rendered objects to determine which object was clicked;

WebGL doesn't support occlusion testing (although it might one day appear as an extension on some hardware). Because it's based on OpenGL ES it has none of the selection and feedback modes from regular OpenGL.

That leaves you with either doing it entirely yourself in software - or doing a very old-school trick of creating an FBO that's the same size as your selection area (maybe a very tiny FBO if you're just interested in where the mouse is clicked - larger if you want everything within a rectangular area) and drawing each distinct "object" using a super-simple shader in a different colour. When you've rendered everything, you read back the FBO into the CPU and read the colour of the pixel(s) to figure out which object was nearest from the camera. (See the last paragraph on p.418 of the prescribed text Interactive Computer Graphics).

What an "object" is depends on what you want to pick between. if you need to know down to the nearest triangle - then you'll be drawing each triangle in a unique colour. If you only need to know at the scale of an entire mesh - then draw each mesh in a different colour.

The number of objects/triangles you can pick between is limited by the number of distinct colours you can draw - which on very low end hardware might be as few as 65536. If you need to distinguish more than that then you'd have to render everything in batches of at most 65536 and read back the FBO after each batch.