



UNIVERSITY OF
LINCOLN

Advanced Graphics (CGP9018M)

Workshop #4: Local illumination with GLSL

Intro

For this week's workshop content, you have two options. You can either work on your assignment or previous workshops, or you can complete the tasks set out in this workshop brief. These tasks mostly involve using an online GLSL editor to write shaders which perform local illumination on a 3D mesh.

Please note that it is recommended that you finish all previous workshops before continuing with this one. This week's workshop requires the completion of workshops 1 to 3. If however, you still need any help, please ask one of the demonstrators for assistance!

Task #1: Writing flat shaders

Your first task is to load up <http://shdr.bkcore.com/>. Once you have navigated to this website, try to complete the following tasks.

- Create a flat shader that renders every pixel of the object with a red colour.
- Can you extend this bit of code to use a variable colour value? For example, can you easily change it to a blue colour? What data type would you use here?
- Using the **time** variable, can you modify the colour of the object over time?
- Can you change the colour of the object depending on the current screen-space coordinates (**gl_Position**)?

Task #2: Adding in Lambertian reflectance

Using the lecture slides as a guide, complete the following tasks.

- Write a simple diffuse shader which uses $\mathbf{N} \cdot \mathbf{L}$. Your light direction can be any direction you wish to choose.
- Can you control the intensity of the diffuse light by multiplying it by a scalar **float** value?
- Add in an ambient light term, with an ambient light intensity.

Task #3: Adding in Specular highlights

Using the lecture slides as a guide, complete the following tasks.

- Using your diffuse light completed in the previous task, add specular highlights to the final pixel value. You should use Phong's original reflection model which uses a reflected ray \mathbf{R} .
- Can you convert this to use the Blinn-Phong model? (Using a halfway vector \mathbf{H} instead of \mathbf{R})
- Can you control the intensity of the specular light with a variable?

Task #4: Modifying the vertex shader

Open up the vertex shader using the interface. Once you have done this, try to complete the following tasks.

- Modify the vertex shader so that each vertex is 'extruded' by a certain distance. (Hint: Think about how you could use the *normal* to do this)
- Can you modify this to extrude inwards instead of outwards?
- Using the surface normal and vertex position, along with `sin(...)`, can you distort the position of vertices?

Additional Task: Other tasks

Once you have completed these tasks, can you:

- Can you use the surface normal as the colour of currently rendered pixel?
- Create a chequerboard pattern on the object in screen-space?
- Create a chequerboard pattern on the object in world-space?
- Can you split the Phong–Blinn shading model into a separate function with parameters?
- Implement rim lighting with no cutoff value? Can you modify this so that a cutoff value can be supplied which controls the width of the highlight?
- Create an outline around the object using the rim equation?
- Create glossy highlights using the modified rim equation?
- Combine diffuse, specular and glossy highlights with a controllable diffuse colour?