

# CPSC 314

## Assignment 3: Shades of Armadillo

Due 11:59pm, March 13, 2014

### 1 Introduction

The goal of this assignment is to explore lighting and shading of 3D models. You will be simulating four different styles of shading: Gouraud, Phong, Blinn-Phong, and Toon (cartoon-like). In this assignment you will be *adding* new pairs of shaders (both vertex and fragment shaders) to carry out these tasks. You will be writing a bit more code, but you can modify the shaders from previous assignments for this purpose.

### 2 Work to be done (100 pts)

The template code is found in the main assignment directory. It is very similar to the previous templates. In Part 1 you will create four different types of shaders and apply them to the unshaded spheres included in the template, and also to the armadillo. You will need to edit the code and create some new shaders for this assignment.

#### Part 1 : Exercises (70 points)

1. (15 pts) **Gouraud Shading.** Compute the shade (i.e., the color) of the object at each given vertex. Use the Phong reflection model for this purpose. The vertex color must then be interpolated across each face, using varying variables. Implement this with a pair of shaders (gouraud.vs.glsl and gouraud.fs.glsl).

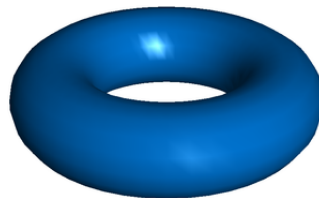


Figure 1: Gouraud shaded torus with Phong reflection model (Source: Wikipedia).

2. **(15 pts) Phong Shading.** In this part, compute the shading for each fragment, once again using the Phong reflection model. You will need the interpolated (i.e., varying) values of the fragment's position and normal. This will produce nicer looking shading. The following image, taken from the Wikipedia article on the Phong reflection model, shows how the different components look individually and summed together:

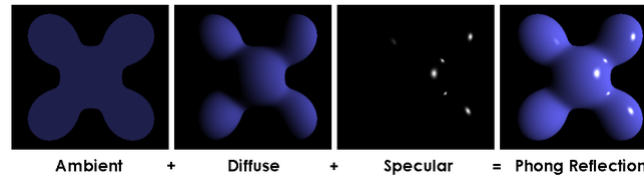


Figure 2: Phong Reflection Model

The main calculations should all go in the fragment shader (in file `phong.fs.glsl`). Note that you will still need a vertex shader (in `phong.vs.glsl`), to pass the appropriate information to your fragment shader.

3. **(15 pts) Blinn-Phong Shading.** In this technique, the specular component is computed differently. Instead of computing the dot product between the Bounce vector and view vector, we compute the dot product between the “halfway vector” (between light and viewing directions) and the surface normal. Implement this with a new pair of shaders.
4. **(15 pts) Toon Shading.** This is an example of a “non-photo realistic” (or NPR) shader. It emulates the way cartoons use very few colors for shading, and the color changes abruptly, while still providing a sense of 3D for the model. This can be implemented using a quantized version of the diffuse reflection term. Instead of making the intensity smoothly vary with  $\cos \theta$  (where  $\theta$  is the angle of incidence between the light vector and normal vector), you quantize this variation into a small number of steps.
5. **(10 pts) Apply the shaders on the armadillo.** Use keys 1-4 to apply Gouraud, Phong, Blinn-Phong and Toon shaders respectively on the armadillo. Figure 3 shows the output might look like.

If you need some inspiration the following movies and video games were rendered with toon shading (also called cel shading) techniques:

[http://en.wikipedia.org/wiki/List\\_of\\_cel-shaded\\_video\\_games](http://en.wikipedia.org/wiki/List_of_cel-shaded_video_games).

## Part 2 : Creative License (30 pts)

This part is up to you. You should create a new shader or modify one of the above shaders to implement some interesting lighting effects similar to the complexity of the ones above. Some suggestions are:

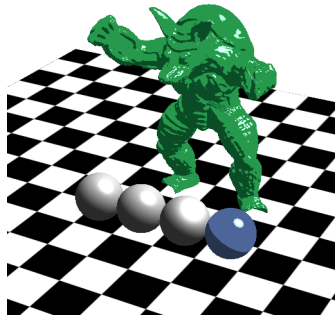


Figure 3: The spheres are shaded using Gouraud, Phong, Blinn-Phong and Toon shaders from left to right respectively. The armadillo is using a toon shader in this figure

- Implement anisotropic material, such as brushed metal. See Section 14.4 of textbook.
- Add spot lights and directional lights, perhaps animate them.
- Explore other non-photo realistic shaders.
- Simulate fog.
- Implement subsurface scattering for translucent materials like skin (advanced).

You might come across a lot of really interesting shaders that require texture mapping or use of textures in some way (shadow mapping, sky boxes, etc.). Please hold off on these until the next assignment since that will deal specifically with texture mapping.

Bonus marks may be given at the discretion of the marker for particularly noteworthy explorations.

**Hand-in Instructions:** You do not have to hand in any printed code. Create a README.txt file that includes your name, student number, and login ID, and any information you would like to pass on to the marker. Create a folder called “a3” under your “cs314” directory. Within this directory have two subdirectories named “part1,” and “part2”, and put all the source files, your makefile, and your README.txt file for each part in the respective folder. Do not use further sub-directories. The assignment should be handed in with the exact command:

```
handin cs314 a3
```