

## Assignment 3: Shading via Lighting and Colors

Loyola Marymount University, CSMI-371: Computer Graphics

Professor Alex Wong

**Due Friday April 20th, 2018 11:59 PM**

### Overview

Given a scene, our goal is to generate objects using hierarchical modeling (based on assignment 2) and add shading (colors) to these objects by defining a light source (a 3-dimensional vector), computing the surface normals, and generating the observed colors using Gouraud shading.

Note that we define the illumination (e.g. the observed colors) as follows:

$$I_c = C_c(amb_c + diff_c(\vec{n} \cdot \vec{h})) + spec_c C_c(\vec{n} \cdot \vec{h}) \text{ for a color channel } c$$

The skeleton code is provided in a separate `assignment3.cpp` file

Note: you may reuse the objects you defined in assignment 2, but you also **MUST** define new objects (e.g. if you have already defined a chair and a table, then perhaps a lamp or a cup)

Once you have defined an object, I would suggest keeping the operations that generate the object as a standalone function (e.g. `build_chair(...)` for generating a chair) such that you may call these functions to quickly prototype a scene.

We will be working with an `ObjectModel` class which contains 4 `vector` objects, holding information about the points defining each plane, their respective base colors, their normals (each point on the plane should have the same normal) and the actual observed colors.

I have provided you with an overloaded function `init_base_color`, which you can use to define the base colors for each plane (to color a cube, for example, you will need to call this 6 times).

There are 3 portions for this assignment: 1) generating surface normals, 2) applying Gouraud shading, 3) rendering objects with colors based on camera and light source positions.

The next section of the specification will detail the road map for these portions.

## Surface Normals

In order to know how light will interact with an object, we must know the object's surface normal, which can be obtained using the cross product of two vectors on the object surface (which is a plane where normals are well-defined). You will implement the `cross_product` method and use it to implement the `generate_normals` method.

## Gouraud Shading

We will implement our shading equation which models how an object will be illuminated. You will implement the `dot_product` method in order to measure the strength of the light rays reflecting off an object. We will use the `dot_product` to implement the `apply_shading` method.

Note: for this assignment we are implementing Gouraud shading, which includes the interpolation of the colors between each point defining the surface. Luckily the interpolation is already taken care of by OpenGL so all we will need to do is to define the observed colors for each point via the shading equation.

## Rendering Objects with Colors

In the previous assignment, we simply let the colors be defined as a static base color value (grays were suggested). We now know how light behaves and how we should model its interaction with objects. Hence, we will need to populate each member of `ObjectModel`. The scene will be constructed using the defined `points` (converted to an array via `vector2array` and instantiates the scene in the `init_scene` function). The colors will be rendered via `colors` (also converted to an array using `vector2array` and instantiates the scene colors in the `init_colors` function).

#### Submission:

You will submit the following to Bright Space as a zip file named `assignment3.zip`

The zip file should contain the following:

- 1) `assignment3.cpp`
- 2) A short video panning over regions of your scene
- 3) A photo in the form of JPEG, JPG or PNG (`scene.{jpeg, jpg, png}`) taken of the scene with which you are basing your model on

#### Grading:

I will be compiling the assignment using the following command:

```
g++ -o assignment3 assignment3.cpp -lglut -lGLU -lGL
```

Your code must compile for me to assign points!

Your assignment will be graded on:

- 1) the correctness of your implementation of the surface normals
- 2) the correctness of your implementation of your shading model
- 3) effort placed recreating the real world scene

#### Late Policy:

For each day the assignment is late, 50% of its worth will be deducted, e.g. 100% on time, 50% 1 day late, 25% 2 days late, etc.