

## Assignment 4: Shading via Illumination and Colors

Loyola Marymount University, CSMI-371: Computer Graphics

Professor Alex Wong

Given a scene, our goal is to generate objects using hierarchical modeling (based on assignment 3) 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:

$$I_r = C_r(a_r + d_r(\bar{n} \cdot \bar{l}) + s_r(\bar{n} \cdot \bar{h})^m)$$

$$I_g = C_g(a_g + d_g(\bar{n} \cdot \bar{l}) + s_g(\bar{n} \cdot \bar{h})^m)$$

$$I_b = C_b(a_b + d_b(\bar{n} \cdot \bar{l}) + s_b(\bar{n} \cdot \bar{h})^m)$$

Note: you may reuse the objects you defined in assignment 3, but you also **MUST** make sure that you have sufficient objects in the scene to make it look interesting (at least a few colors)

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 (randomly generated). 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`. The set of base colors of the scene will be stored in `base_colors`.

The `apply_shading` method will be used to generate the actual observed colors using the surface normals, light source, and camera. The observed colors are stored as the `color` member of `ObjectModel`.

#### Submission:

You will submit the following to Bright Space:

- 1) assignment4.cpp
- 2) Either 6 different viewpoints taken from different angles of your scene submitted as JPEG, JPG or PNG with the names: view{1,2,3,4,5,6}.{jpeg, jpg, png} or 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 assignment4 assignment4.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 the illumination equation
- 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.