Computer Graphics (MIEIC)

Topic 3

Lighting and Materials

Objectives

Manipulate the components involved in lighting, specifically lights, normals, and the reflection components of the materials.

Preparation of the Work Environment

For this work, you must use the base code that is provided for this class in Moodle. You will be asked in one of the exercises to include in the base code the objects that you created in the second practical class (***ex2***), namely the ***MyTangram*** and ***MyUnitCube*** (it is assumed that you used the nomenclature defined in the worksheet for the second practical class).

Practical work

The following points describe the topics covered during this practical class, as well as the tasks to be performed.

Some of the tasks are noted with the icon 📷 (image capture). At these points, you should capture and save an image of the application in the browser (e.g., using Alt-PrtScr on Windows or Cmd-Shift-3 on Mac OS X to capture to the clipboard and then save to file in an image management application of your choice). At the end of each class, students must rename the images to the format **"ex3-t<class> g<group>-n.png"**, where **class** corresponds to the class number (e.g., 2MIEIC01 is class '**01'**), and **group** corresponds to student group number (defined in the TP group file in Moodle), and **n** corresponds to the number provided in the exercise (e.g., **"ex3-t1g01-1.png"**).

In tasks marked with the icon JS (code), students must create a **.zip file from the folder that contains your code (typically in the 'ex1' folder, if you have code being used in other folders, include it too)**, and name it **"ex3-t<class>g<group>-n.zip"**, (with class, group and provided number identified as described above **"ex3-t1g01-1.zip"**).
At the end (or throughout the work), one of the elements must submit the files via Moodle, through the link provided for that purpose. Only one member of the group is required to submit the work.

Experiences

The scene created in the base code contains a plane (***MyPlane***), barely visible and with a reddish color, and two lights off. In the graphical interface in the upper right corner, you will find a series of controls for geometry, materials, and lights, which you must use for the following points.

Ambient

1. Although the lights are off, the plane is visible due to the **ambient component** of the material and the **global ambient lighting** in the scene. Switch between objects in the *'Selected Object'* dropdown to check the absence of edge definition, since the same color is assigned to the surfaces regardless of their orientation.
2. Create a control on the interface that allows you to vary the intensity of the global ambient lighting of the scene (defined in the ***initLights()*** function), using a *slider*, and check the differences in the scene.
   Note: Check the documentation for ***CGFscene.setGlobalAmbientLight()***.

Diffuse

3. Restart the scene and change the applied material to *'Red Diffuse'*, making the plane invisible (this material has no ambient component). Activate only light 0, by enabling the *'Enabled'* checkbox. The axis and the lights become more visible, but the plane remains invisible. Check the visibility of the other available objects in the *'Selected Object'* dropdown under these conditions.
4. With the plane selected, vary the Z position of light 0, using the sliders in *'Light 0 / Position'*, so that it passes to the front of the plane, illuminating it.
5. Set the position of light 0 to [2, 2, 1] and rotate the camera, checking that the color gradient in the objects with only a diffuse component does not vary with the position of the observer.

Specular

6. Restarting the scene to return to the initial settings, activate light 1, and change the material in the dropdown to *'Red Specular'*, which will create a red gradient in the plane. Slightly rotate the camera to see the color variation in the gradient, which is dependent on the **specular components** of the applied material and the active light, as well as the position of the light and the camera relative to the surfaces. You can change the material to *'Red Diffuse'* for comparison.
7. Change the material to *'Custom'* and change the colors of the ambient and diffuse components to black (#000000) and the specular component to yellow (#ffff00). You should see the plane with a little yellow reflection. Rotate the camera so that the center of the **reflection** is approximately in the center of the square.
8. Vary the *'Shininess'* value of the *'Custom'* material, and check for differences in **intensity** and amplitude of specular reflection.
9. Vary the complexity of the plane, and check the difference in the appearance of the specular reflection.

Combination of lighting components

10. Restart the scene, change the material applied in the material dropdown to *'Custom'*, and check the colors of the components of this material in the 'Custom Material' group. The plane should have a faint dark blue color.
11. Set the position of light 0 to [1, 1, 1], and activate light 0, which will create a red to blue gradient in the plane.
12. Vary the Z position of light 0, using the slider in *'Light 0 / Position'* to check the variation in the plane gradient.
13. Place the light 0 at [0, 0, 0.2], and vary the complexity of the object, bringing the camera closer to better observe the details in the plane.

Attenuation

14. Restart the scene, reapply the *'Custom'* material, and activate light 1, placing it at position [0, 0, 0.2]. Change the Z value to move the light away from the plane. The plane should appear more illuminated, although the light is moving further away.
15. Reduce the **constant attenuation** component of light 1 to 0.5, for a more consistent variation in intensity of reflection with distance.
16. Experiment with different combinations of the three attenuation components and observe the differences in the object's illumination as the distance varies.

Exercises

Add to the base code the files containing the ***MyTangram*** (and the classes needed to create the tangram) and ***MyUnitCube*** classes, as created in the previous class (practical class 2), placing the *Javascript* files in the respective folder for this class, and adding a reference to them in the *main.js* file list.

1. Create an instance of **MyTangram** and another of **MyUnitCube** in the ***init()*** function of the scene, and add them to the list of available objects presented in the GUI (check how it is done for existing objects).
2. Apply some of the previous lighting experiments with these new objects, and note that there are probably inconsistencies with what would be expected, due to the fact that the **normals** for these objects were not declared.
3. Declare the normals for the different objects in the ***initBuffers()*** function in their classes, starting with the cube. You may have to repeat vertices that are shared by faces with different orientations, and therefore will have different normals depending on the face on which they are to be used (specifically in **MyUnitCube**, and any double-sided pieces of the tangram).
4. Create a material with similar color to wood, with low specular component, and add it to the available list of materials (follow the example of *'Red Diffuse*', in the ***initMaterials()*** function of the scene). Test with the cube. (1 📷)
5. Within the **MyTangram** class, create a material with a high specular component for each piece, colorized according to the provided image (applying it to the respective piece).

6. Apply the *'Custom'* material to the Tangram piece created with ***MyDiamond***, controllable with the interface. (2 📷) (1 JS)

Checklist

Until the end of the work you should save and later submit the following images and versions of the code via Moodle, **strictly respecting the naming rule**:

- 📷 **Images** (2): 1,2 **(named as "ex3-t<class>g<group>-n.png"**)
- JS **Code in zip file** (1): 1 (**named as "ex3-t<class>g<group>-n.zip"**)