CS3241 Computer Graphics

Lab 3: Let there be light!

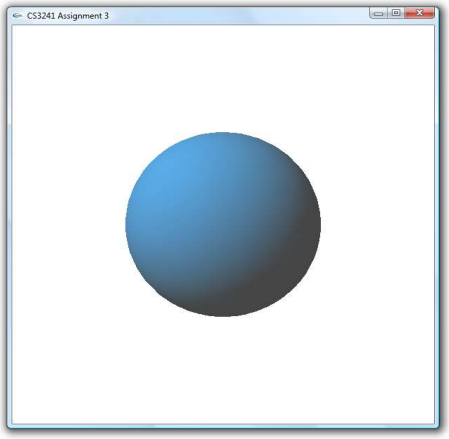
The objective of this assignment is to implement illumination and shading in OpenGL. Play around with the sample program "**`Lab 3 sample.exe`**" to get a feeling of what you are going to do. By pressing 1-4, you can switch among different scenes. By pressing 'S', smooth shading will be toggled between flat shading and smooth shading. And in the same manner, 'H' is for highlight (specular reflection).

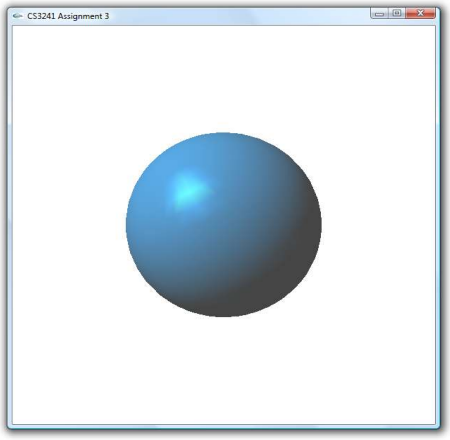
Instructions

Open the file "**`Lab 3.sln`**" for Windows or "**`Lab3.xcodeproj`**" for Mac. You are given a skeleton program that displays a ball with only flat shading.

Step 1: Add in the normals for smooth shading.

By pressing 'S', the program will switch to smooth shading (variable **`m_Smooth`** is set to true). Implement the correct normals for each vertex in the function **`drawSphere()`**. Add suitable comments to explain why and how your normal is computed. After adding the normal computation, your sphere should look like the figure on the left.





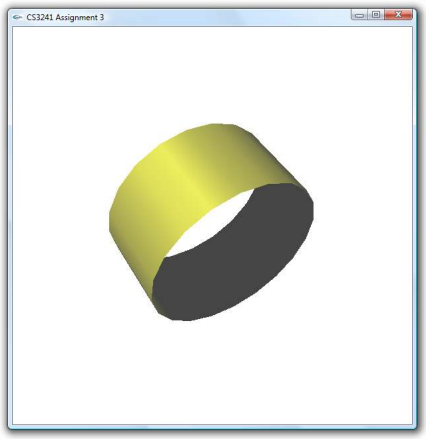
Step 2: Add in the highlight.

By pressing 'H', the specular term will be active and there should be some highlight on your object. You should put in the material property if the variable **`m_Highlight`** is on. So your sphere should look like the right figure above.

Step 3: Create your own object(s).

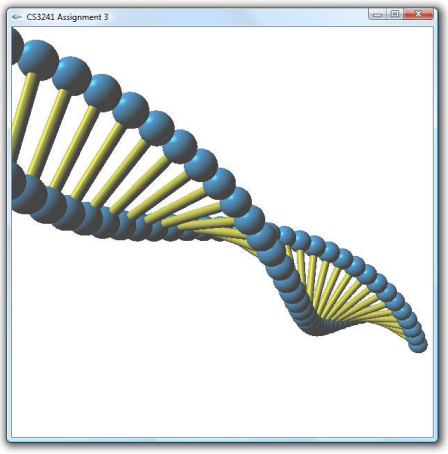
Now, it is your turn to show your creativity and make your own object. The main drawing routine is the function **`display()`** in the file **`main.cpp`**. Modify it if necessary and make your program display another primitive object by pressing '2'. You have the freedom to create any object that is not totally flat. Namely, your new object **must** have some curved surfaces, e.g. an ellipsoid, a cone, a paraboloid, etc.

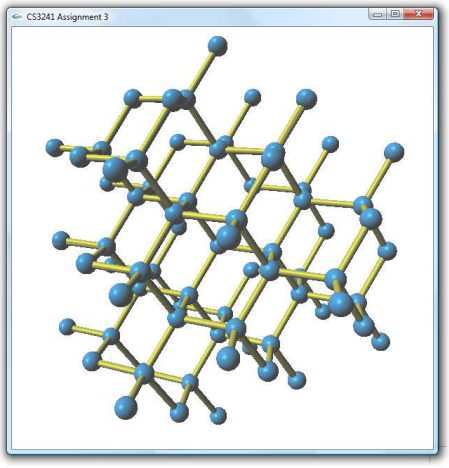
Do not create an object that only consists of flat surfaces such as a cube or a tetrahedron. Your new object must be able to be displayed in **all** flat shading, smooth shading and with or without highlight modes. You can create more than one primitive object if you want.



Step 4: Create your own composite objects.

A composite object is constructed by various transformations and copies of your primitive objects. Construct a function to display your composite objects and name your creation an artistic title. The sample program gives two such objects, the one on the left is called "Da Human Code" and the right is called "Diamond" in the figure below. In this assignment, construct two such composite objects (for '3' and '4').





We will grade how innovative and beautiful your objects are as well as how compact and clever your program is. E.g., the examples above are only about ten lines of C++ code. However, please do NOT create objects that are associated with racism, pornography or violence.

In this assignment, you cannot use any predetermined 3D object drawing function, e.g. **`drawSolidSphere()`** from the GLUT library, to help you create an object. You must specify all vertices and normals of the object by yourself.

Suggested Bonuses:

- Metallic material (Hint: Specular color is the same as material color)
- Shiny material (Hint: Specular highlight should be clearer)
- Animation

Submission

1. Write a **readme.txt** file including:
    - Your matric number
    - Primitives and transformations you have used
    - Any other things the TA should know?
    - Indicate the material for your objects

    What you are drawing

    Methods you have modified – only display()?

    What is the coolest thing(s) in your drawing
2. Zip your **main.cpp** and **readme.txt**, rename it to your student number + ".zip" and submit it up to IVLE.