

# ***Computer Graphics***

***by Ruen-Rone Lee***  
**ICL/ITRI**



# ***Assignment #3***



Per Vertex Lighting



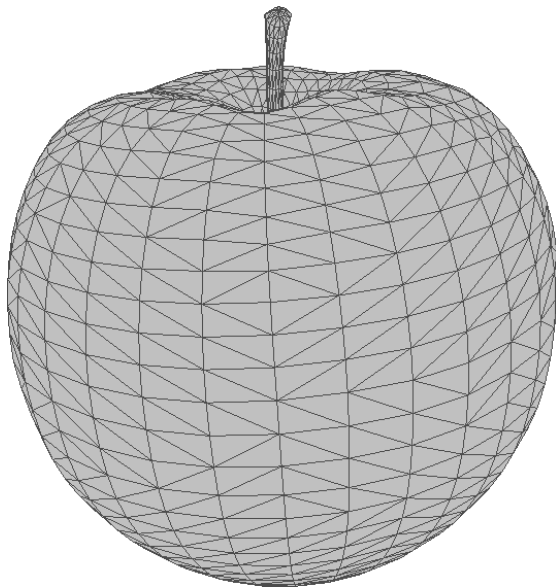
Per Pixel Lighting

***Lighting***

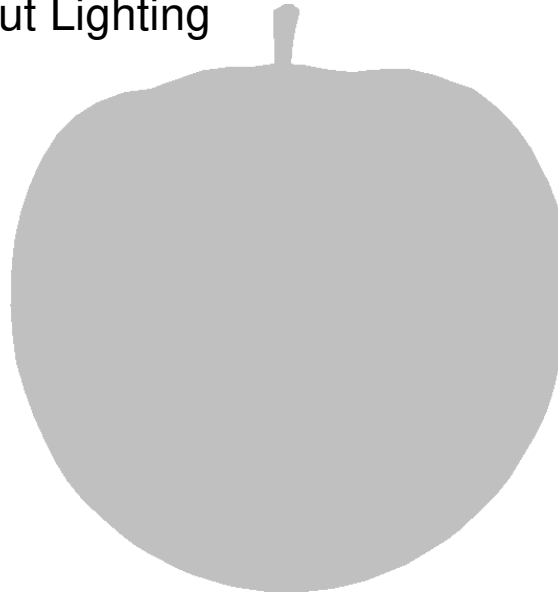


# *Lighting 3D Models*

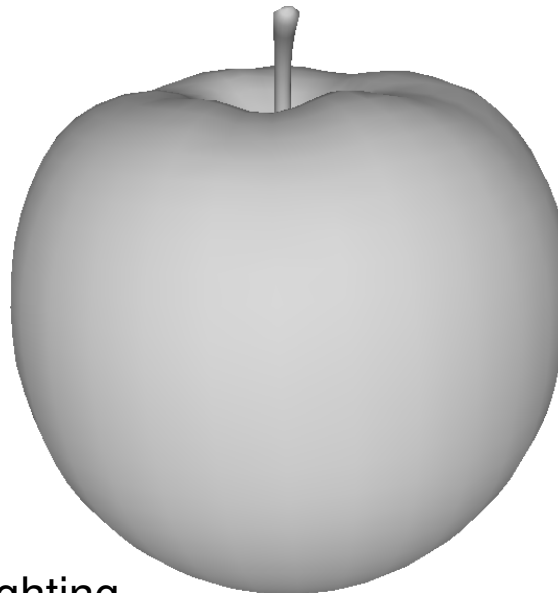
Original 3D Model



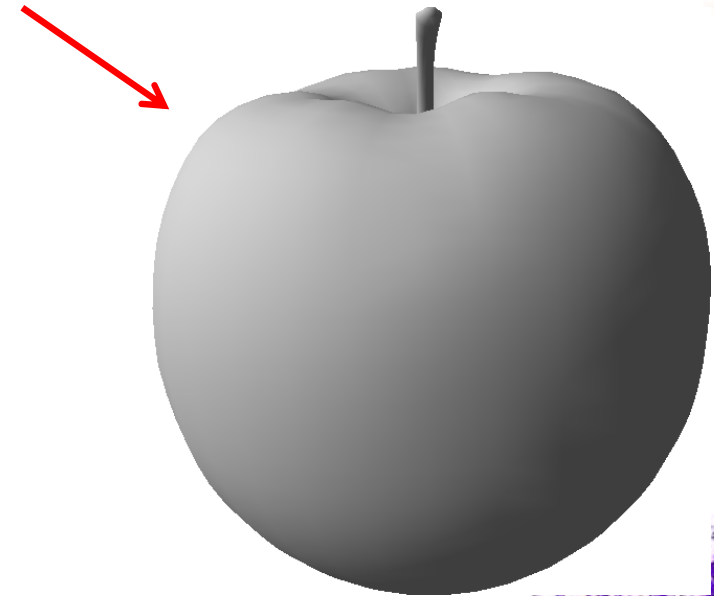
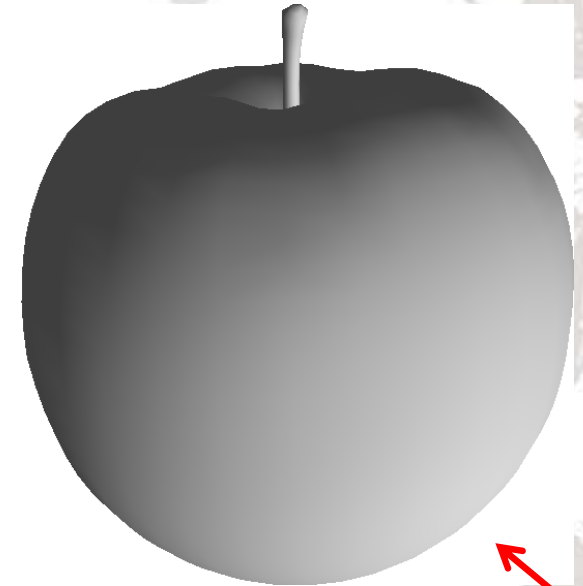
Without Lighting



With Lighting



Changing light source position



# *Requirement*

- ◆ You are required to write a program that can accept 3D test models as in assignment #1 & 2 and render the 3D models with smooth shading
- ◆ The models should be rendered with given light sources
- ◆ Three different light sources should be implemented, a directional light, a positional light, and a spot light.



# *Requirement*

- ◆ **Light source position should be able to change manually**
  - E.g., rotating around the 3D model
- ◆ **Vertex lighting is required**
  - All the lighting calculations are performed in vertex shader.
- ◆ **Per pixel lighting is also required**
  - All the lighting calculations are preformed in fragment shader.



# ***Requirement***

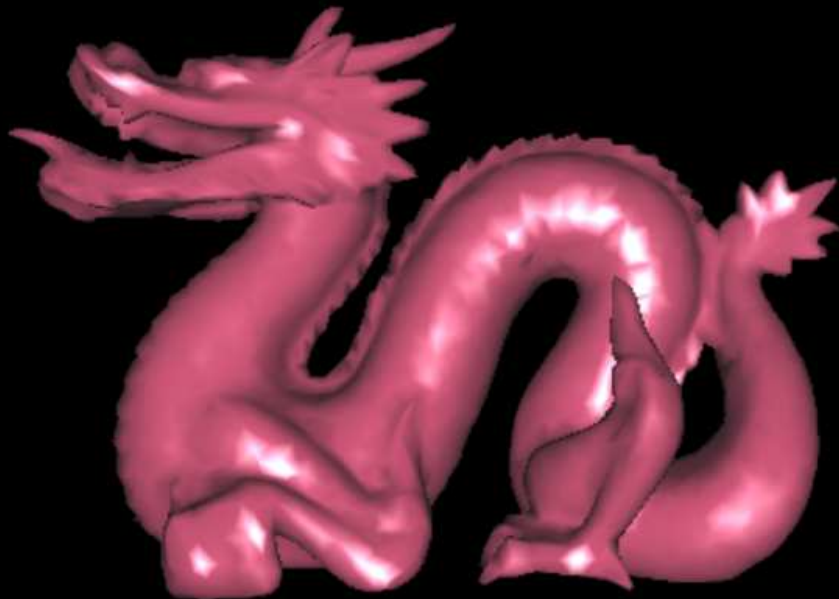
- ◆ **Use keyboard and mouse to control the objects transformation as implemented in assignment #2**
- ◆ **Use some keys to switch between different lights source types with per-vertex lighting or per-pixel lighting**
- ◆ **Display help file, e.g., pressing key 'h', for how to control the actions of your program (display on console window)**



# Requirement

- ◆ Display different lighting model (per-vertex or per-pixel) with different light source types (directional, positional, spot) **side-by-side** for comparing the rendering effects

Per-Vertex Lighting / Point Light



Per-Pixel Lighting / Point Light





# *Input Model Format*

- ◆ **Wavefront 3D Graphics model description file with extension .obj**
- ◆ **Models without vertex normal**
  - The input model contains no vertex normal information
  - Generate the vertex normal by yourself
- ◆ **Models with vertex normal**
  - The input model contains not only the vertex position information but also the normal information for lighting calculation





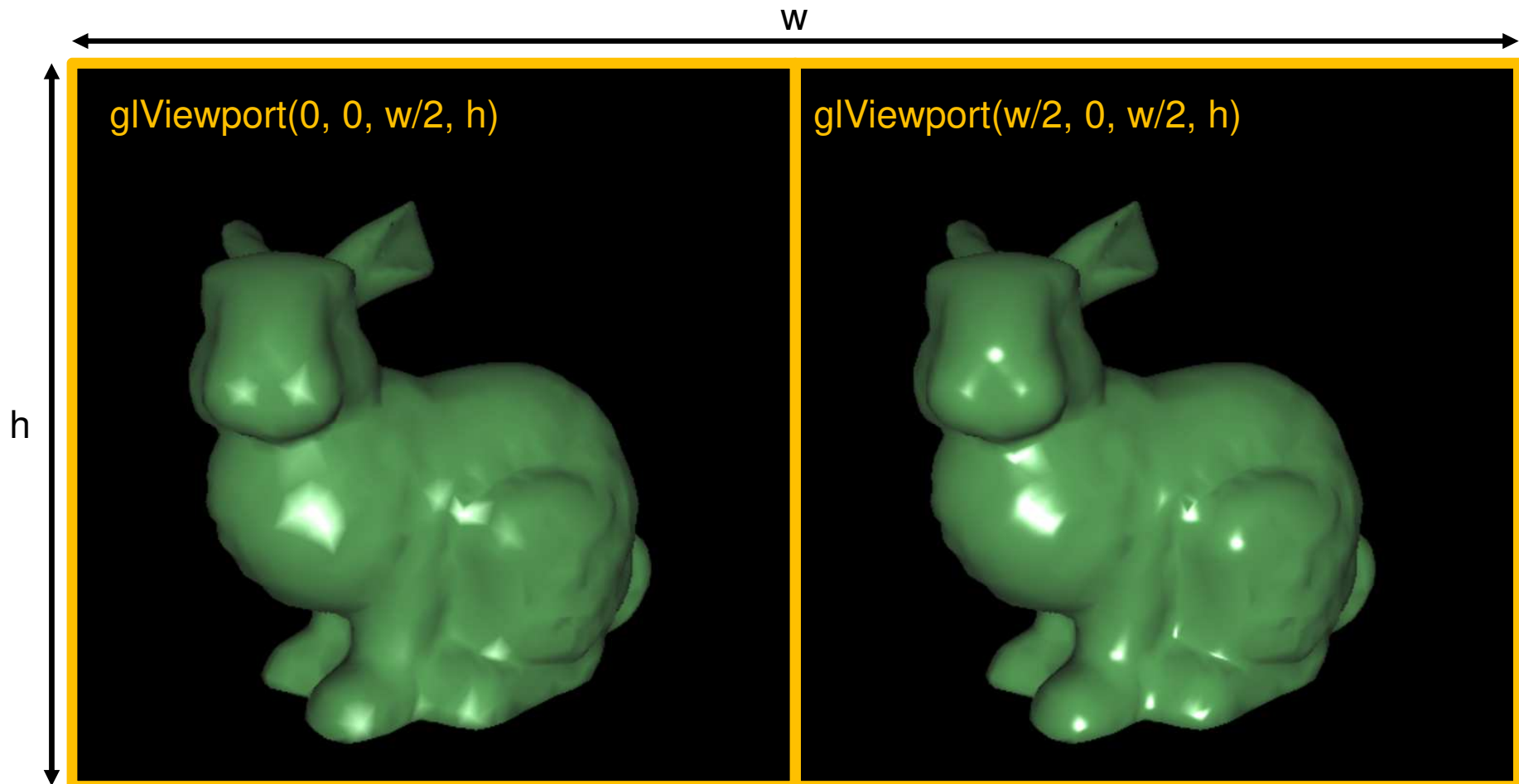
# Hints

- ◆ **Normal transformation** is necessary to derive correct lighting result
- ◆ Normalization to the normals is also necessary to obtain correct lighting result
- ◆ **Per pixel lighting** can be achieved by passing the transformed vertex normals to rasterizer for generating per-pixel normals and then do the lighting calculations in fragment shader
  - Replace the vertex colors in assignment #2 by vertex normals



# Hints

- ◆ Draw two models side-by-side
  - Set the viewport before the draw command



# *Hints*

- ◆ **Some 3D models with vertex normals are provided for verifying your design during program development.**
- ◆ **Please download the test models from iLMS**



# ***Due Date***

- ◆ **Three weeks** after announcement (should be **5/22**)
- ◆ Late submission is allowed with less score
- ◆ **No score if you did not submit your assignment**
- ◆ Plagiarism is strictly forbidden
  - If you copy from others, your score will become zero
  - The score to the one who provides the original copy will also be downgraded

# Submission Guide

- ◆ Please submit to **the course webpage at NTHU iLMS system**
  - *Notice: E-mail submission will not be accepted*
- ◆ Submission should include
  - Source codes (including solution and project files)
  - Executable binary (can be run on PC/windows)
  - Documentation (explain how you did it and how to operate it)
  - *Notice: please do not submit any 3D models to save the disk space*
- ◆ Contact with TAs if you have problem in submission

# Q&A

