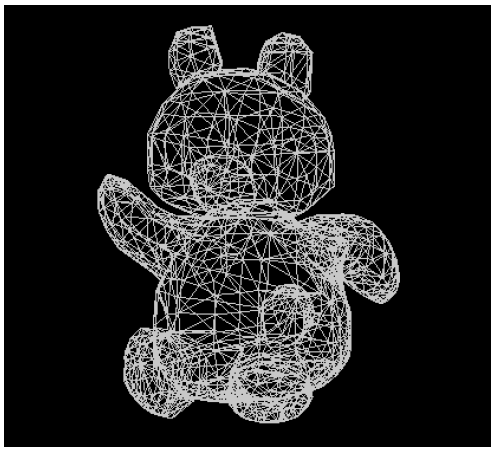
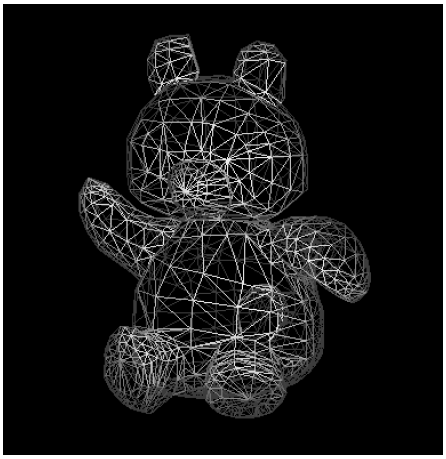
Instructions

As an extension from project 3, for this project, you need to do the coloring on the object surface to make a more realistic 3D look. Recall what we have learned in the class, there are three types of illumination models: diffuse, ambient, and phong terms. So you need to apply these three illumination principles to generate a plausible RGB color for each 3D unit. Since the Teddy bear object consists of multiple triangles. For this assignment, you need to use the flat shading mode to render the Teddy bear object. In other word, the shading unit is a triangle. All the surface points from the same triangle have the same RGB color. Here are some example pictures:



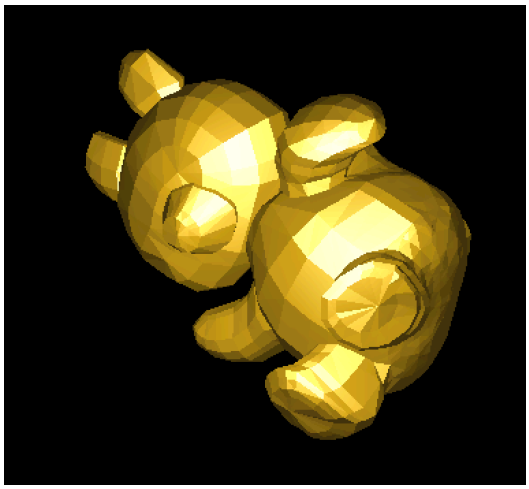
Mesh without Shading



Mesh with Shading

Plaster Version (diffuse + ambient)

China Version (diffuse + ambient + phong = 50)



Gold Version (diffuse + ambient + phong = 0.5)

Requirements:

(1) Correct computation on cross-product between vectors (15%)

(2) Correct computation on view, light, and normal vectors (15%)

(3) Correct illumination model for the ambient term (15%)

(4) Correct illumination model for the diffuse term (20%)

(5) Correct illumination model for the phong term (25%)

(6) Correct keyboard callback function to switch between different views (10%)

- hit “0” to show the Teddy bear with no shading at all (a single color for all the triangles).
- hit “1” to show the Teddy bear with only diffuse and ambient terms. The result looks like a plaster material.
- hit “2” to show the Teddy bear with diffuse, ambient, and phong terms. The result may look like different types of materials, depending on your choice of the exponential p for the phong terms, e.g. china material (p = [55, 100], metal material p = [0.1, 1.0]).
- hit key “M” or “m" only to show the mesh lines (no triangle surface).

Some Hints:

Below are some hints to achieve this project:

(1) Normal Computation

To compute the normal for each triangle, you can use the anti-clockwise product to compute it:

n = (V2 - V1) x (V3 - V1);

The index of V1, V2, and V3 are obtained from the .obj file:

……

f 353 223 187

f 73 237 223

f 234 187 317

…

(2) Cross Product

To make your code more effiecient and cleaner, you can create a cross product function in your defined Vertex class/structure. Remember in last project, you create the Vertex class to store every vertex loaded from the .obj file. Now you can implement the cross product function to do the cross product between two vectors and return another vector that is perpendicular to the first two vectors. It looks like:

```
struct Vertex{

 float x, y, z;

 Vertex cross(const Vertex& right)

 {

     Vertex result;

     …

     return result;

 }



}
```

(3) Inner (dot) product, +, - operators

Similar to the cross product, you can implement the reloading operator for +, -, dot product inside the Vertex structure.

```
   const Vertex operator - (const Vertex& right) const

   {
```

```
    Vertex result;
    ...
  return result;
}



/*Reload + operator*/
const Vertex operator + (const Vertex& right) const
{
    Vertex result;
    ...
    return result;
}

/*Reload * operator to times a number*/
const Vertex operator * (const float right) const
{
    Vertex result;
     ...
    return result;
}

/*For normalized vector dot product use*/
float dotProduct(const  Vertex& right) const
{
    float result;
    ...
```

```
    return result;
}
```

(4) Where is the light? What is the color?

The light position can be very flexibly define. You can put the light anywhere you like. If you don't have any preference. Below are some of my suggested data:

light position: (0, 0, 5) light

color: (0.9, 0.9, 0.9) ambient

color: (0.2, 0.2, 0.2) phong

color:(0.8, 0.8, 0.8)

Teddy object surface color: (0.5, 0.5, 0.5) for dark white Teddy bear, (0.9, 0.7, 0.1) for gold Teddy

Submission:

1. As previous projects, only source files are needed to be submitted via Isidore. You can create as many files as needed.
2. Please provide a .docx or .pdf file showing several screen captured images of your running results: the Teddy bear has different materials, like metal, china, and plaster.