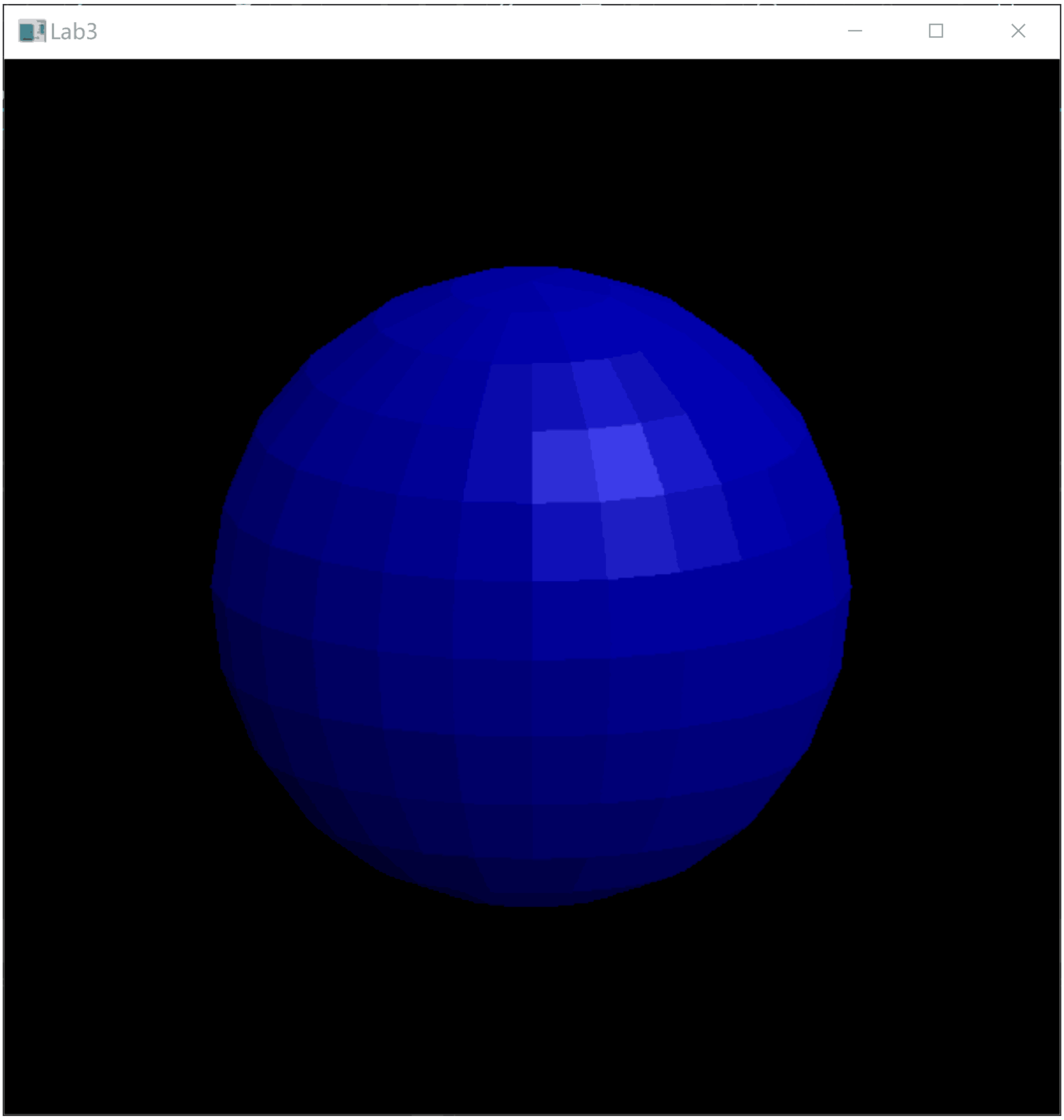
Pixel 2000X2000

```
float h=2.0f, dnear=25.0f, dfar=200.0f;
```

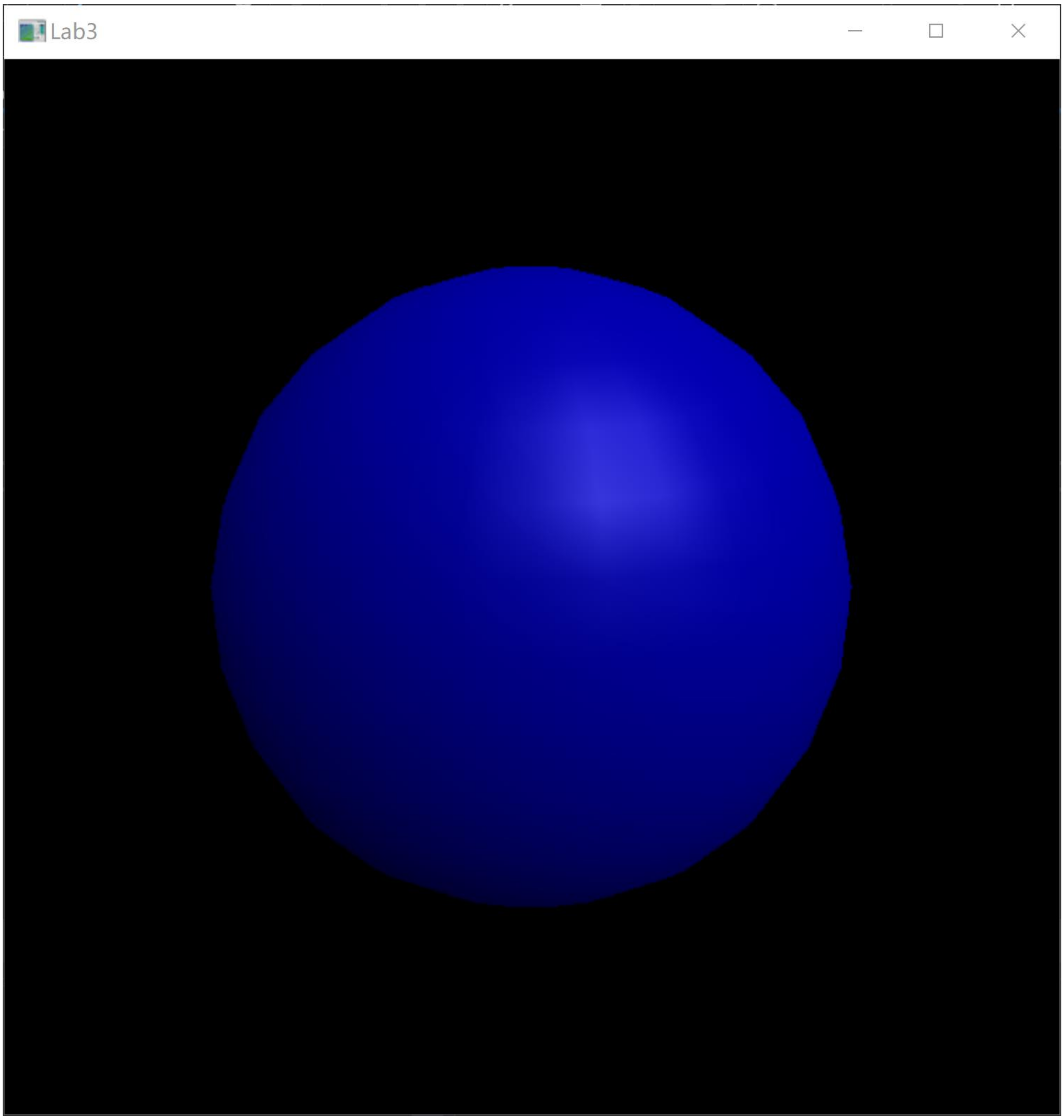
better-ball.d

```
MyVector C = {0,5,-20};
MyVector pref = { 0,0,0 };
MyVector Up = {0,1,0};

MyVector LightDirection = { 60,100,-50 };
MyVector LightIntensity = { 0.5,0.5,0.5 };
MyVector LightColor = { 1,1,1};
MyVector ambientLightIntensity = { 0.3,0.3,1 };
float fatt = 1.0f;

objectColor = MyVector(0,0,1); //RGB
ka = 0.4;
kd = 0.6;
ks = 0.5;
focus = 24;
```

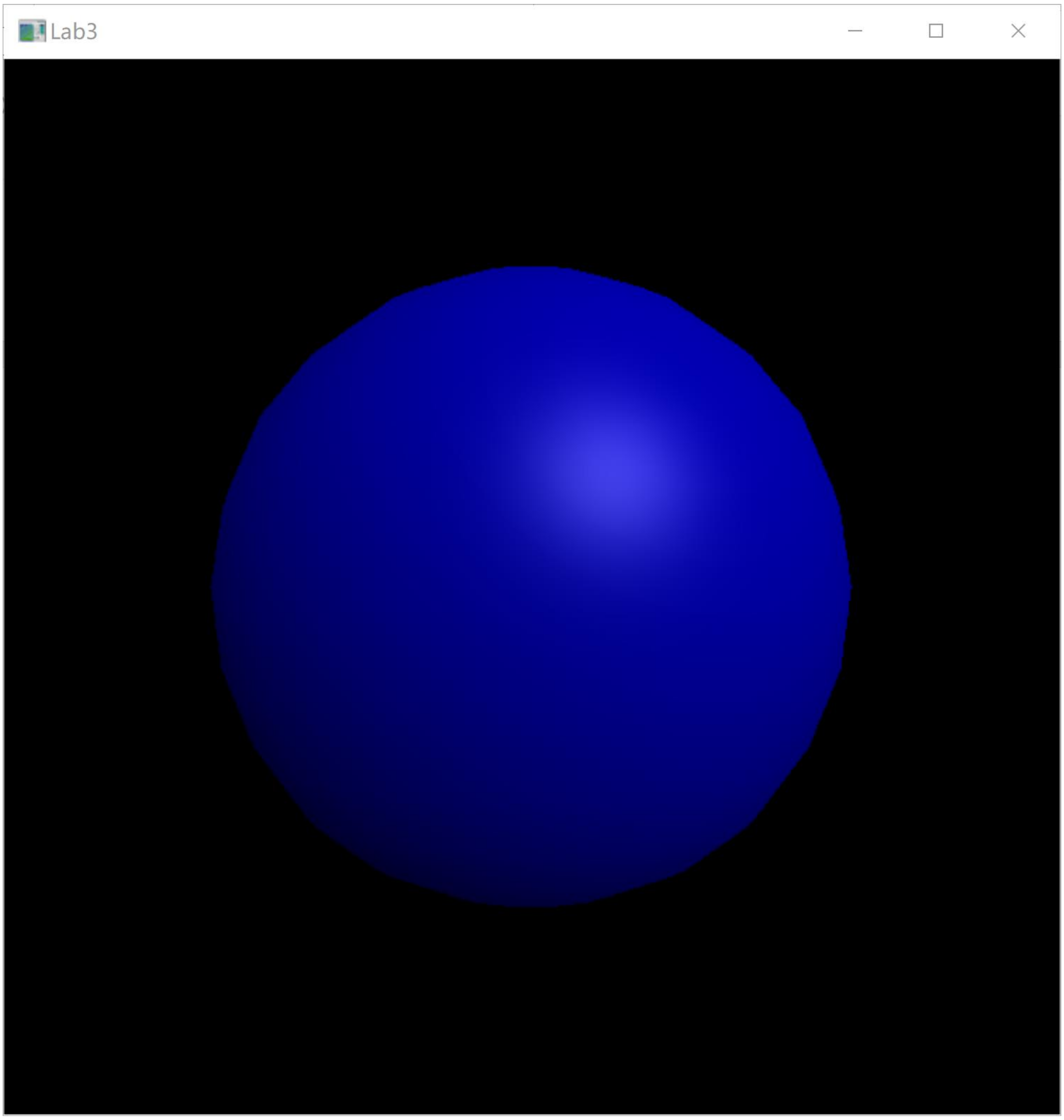
Constant shading



Gouraud shading



Phong shading



For constant shading: calculate the normal of each polygon, then use the illumination model to get the color of one polygon.

For gouraud shading: calculate the normal of each vertex, then use the illumination model to get the color of one vertex, then interpolate the color during scan conversion.

For Phong shading: calculate the normal of each vertex, interpolate the normal during scan conversion, then calculate the color of one pixel.

To calculate polygon normal: Store which vertices each polygon has, and calculate it in the world space coordinate system. For example, polygon1 has three vertices V1, V2, and V3, and calculates (V2-V1) X (V3-V2), and then normalizes.

To calculate vertex normal: Stores which polygons each vertex exists. After calculating the polygon normal, calculate vertex normal. If vertex 1 is on polygon 1, 2, 3, and 4, we added the normal of polygon 1, 2, 3, and 4 and normalized.

How to run:

Edit variables in preference.cpp

```
//Camera
MyVector C = {0,5,-20};
MyVector pref = { 0,0,0 };
MyVector Up = {0,1,0};
```

Edit C,pref,UP to control the Camera.

```
//Light
MyVector LightDirection = { 60,100,-50 };
MyVector LightIntensity = { 0.5,0.5,0.5 };
MyVector LightColor = { 1,1,1};
MyVector ambientLightIntensity = { 0.3,0.3,1 };
float fatt = 1.0f;
```

Edit LightDirection, LightIntensity, LightColor, ambientLightIntensity, fatt to set the Light.

```
//shading modle
int shading_model = 1;
//1 constant shading
//2 gouraud shading
//3 Phong shading
```

Edit shading_model to choose the shading model.

Input the setting of the model in Computer Graphics Lab3.cpp

```
    //LoadData(″D files/house.d″);
    //LoadData(″D files/cow.d″);
    //LoadData(″D files/ball.d″);
    LoadData(″D files/better-ball.d″);
    //LoadData(″D files/bench.d″);
    //LoadData(″D files/rect.d″);
    //LoadData(″D files/car.d″);
    //LoadData(″D files/donut.d″);
    //LoadData(″D files/knight.d″);
```

Edit the filename to select which data file to load.

```
    //first object
    objectColor[0] = MyVector(0,0,1); //RGB
    ka[0] = MyVector(0.4, 0.4, 0.4);  //ka for RGB
    kd[0] = MyVector(0.6, 0.6, 0.6); //kd for RGB
    ks[0] = MyVector(0.5, 0.5, 0.5);//ks for RGB
    focus[0] = 24;
```

Edit the objectColor[i], ka[i], kd[i], ks[i] to set the parameter of each loaded file.

For example, I've loaded 2 files: better-ball.d & rect.d, objectColor[0] is the color of better-ball and objectColor[1] is the color of rect.