



# ШУТИС, Мэдээлэл Холбооны Технологийн Сургууль

*F.CS209 Компьютерийн график*

Лекц 12 – Гэрэл, Сүүдэр ба Материал  
Боловсруулсан багш: Х.Хулан, Ч.Цэнд-Аюуш

2022 он

# Хичээлийн агуулга

- Гэрлийн үүсгүүр
  - Point sources – Цэгэн эх үүсгэвэр
  - Directional sources – Чиглэлийн эх үүсгэвэр
- Тусгал
  - Diffuse reflection – Сарнисан ойлт
  - Specular reflection – Толин тусгалын ойлт
  - Ambient reflection – Орчны тусгалын ойлт
- Gouraud Shading – Сүүдэрлэлт
- Phong Shading - Сүүдэрлэлт
- Shadows - Сүүдэр

## ➤ mesh sizes (Important) өөрчлөлт

Дараахь хэсэгт гялалзсан цагаан ханыг янз бүрийн хэмжээтэй рендэрлэгдсэн чиглэлтэй цагаан гэрлээр харуулсан болно.

```
GLfloat ambient0[] = { 0., 0., 0., 1.};
```

```
GLfloat diffuse0[] = { .7, .7, .7, 1.};
```

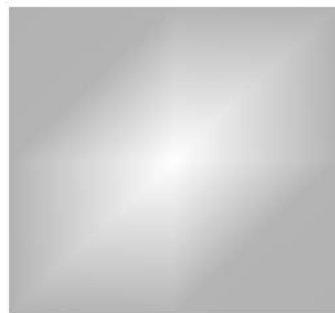
```
GLfloat specular0[] = { 1., 1., 1., 1.};
```

```
position[] = { 0., 0., 1., 0.}; //Light direction
```

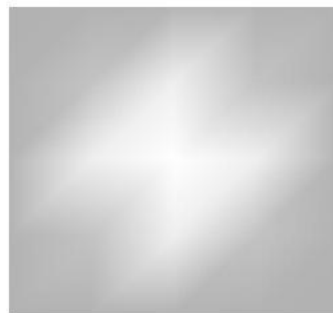
```
//Viewing position (0, 0, 8)
```



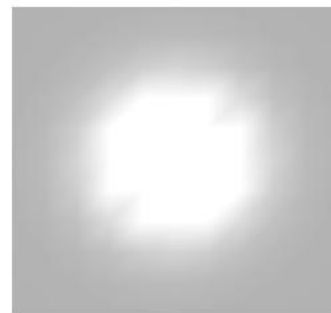
1 × 1



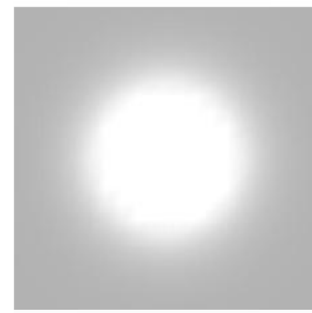
2 × 2



4 × 4



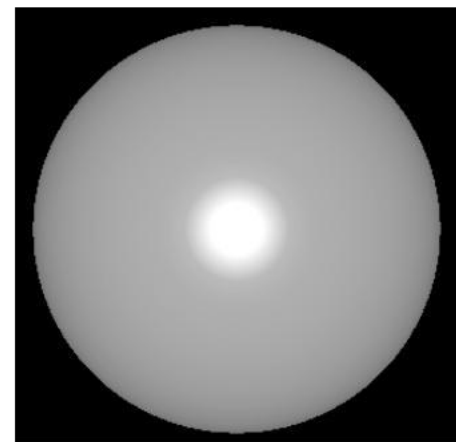
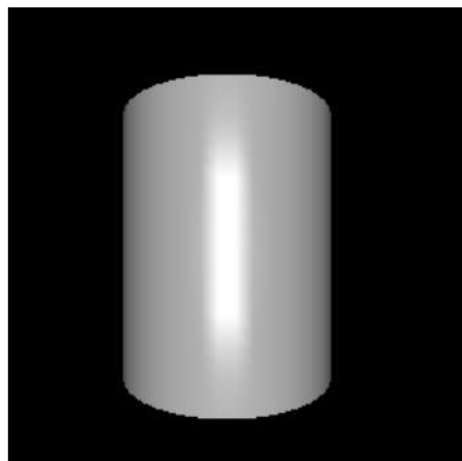
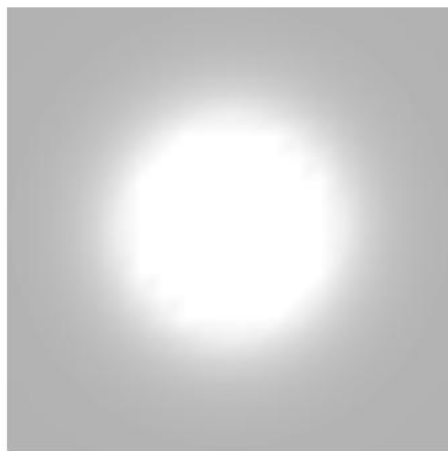
10 × 10



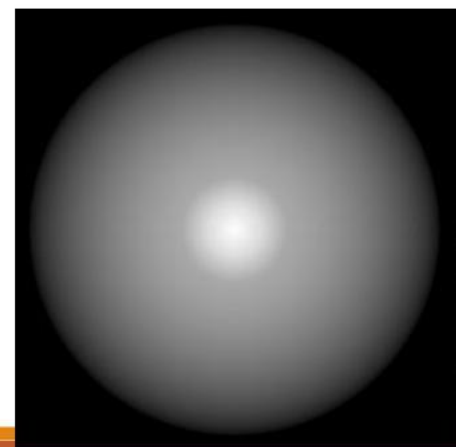
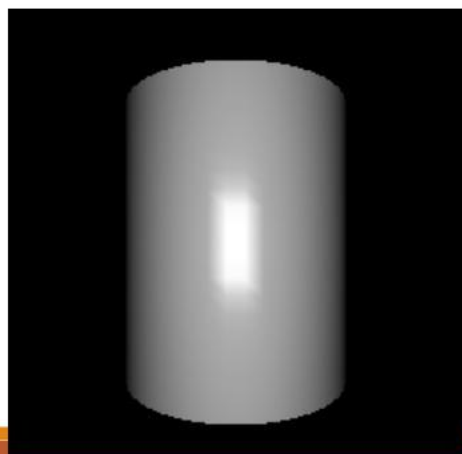
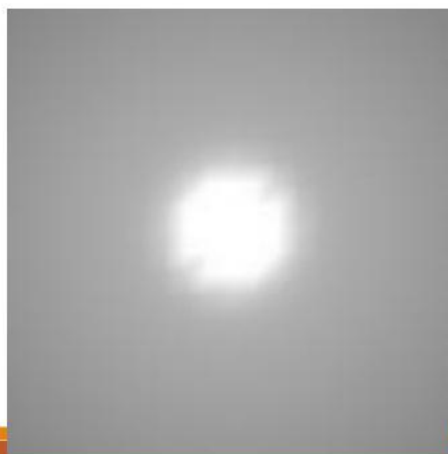
20 × 20

- Гялалзсан цагаан объект дээрх directional source ба point source харьцуулалт. Харах байрлал  $(0, 0, 8)$

Dir. source  
at  $(0,0,1)$



Point source  
at  $(0,0,8)$



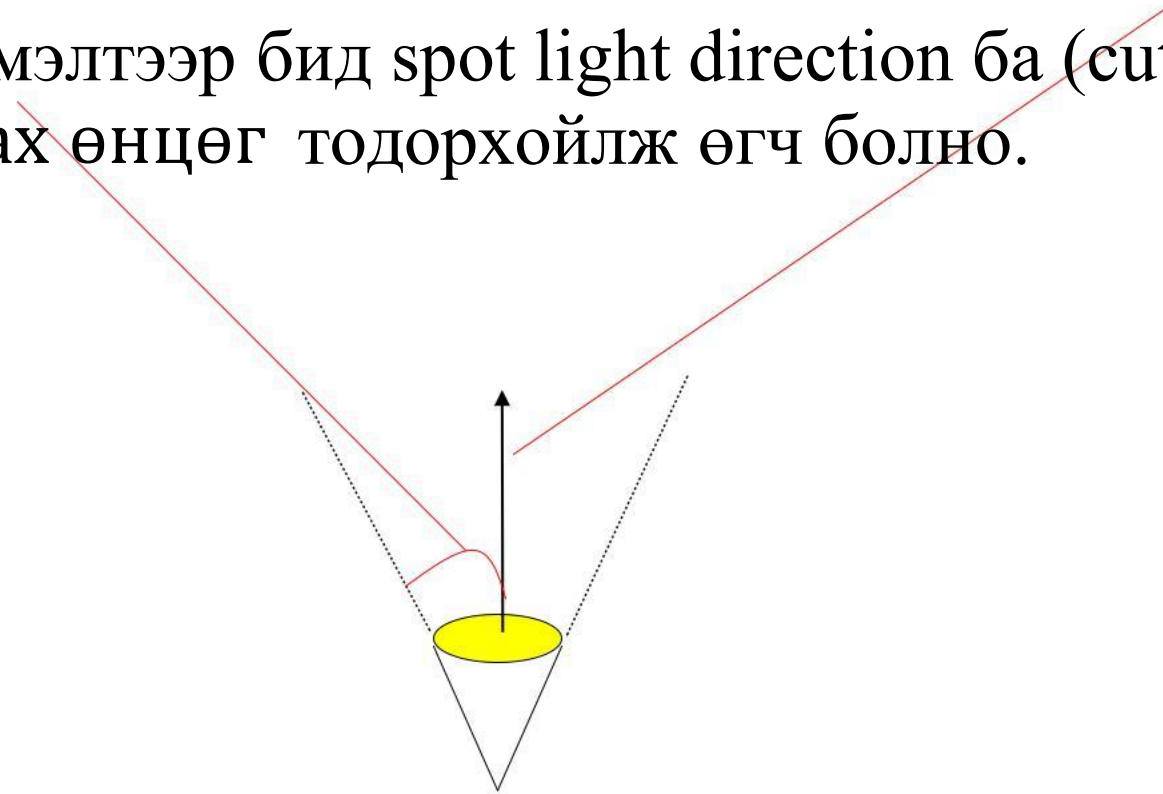
- point light source хувьд , attenuation factor томъёонд тогтмолыг зааж өгч болно

$$A_f = \frac{1}{a + bd + cd^2}$$

```
glLightf( GL_LIGHT1, GL_CONSTANT_ATTENUATION, .0);  
glLightf( GL_LIGHT1, GL_LINEAR_ATTENUATION, .5);  
glLightf( GL_LIGHT1, GL_QUADRATIC_ATTENUATION, 0.);
```

- Default утга нь  $a = 1, b = 0, c = 0$  (no attenuation)
- higher attenuation,  $b$  ба  $c$  хамгийн их утгыг тогтооно жнь, 1, 1.5, 2 эсвэл 3.

- spot light (a point source) тодорхойлох өнгөний компонент, байрлал болон attenuation constants –дыг ердийн point source нэгэн адил тодорхойлно.
- Нэмэлтээр бид spot light direction ба (cut-off angle) таслах өнцөг тодорхойлж өгч болно.

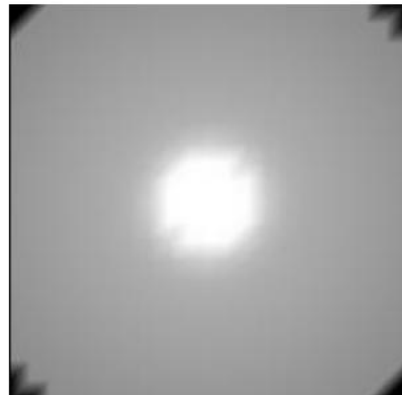


➤ Цагаан ханан дээрх cutoff angle өөрчлөлт

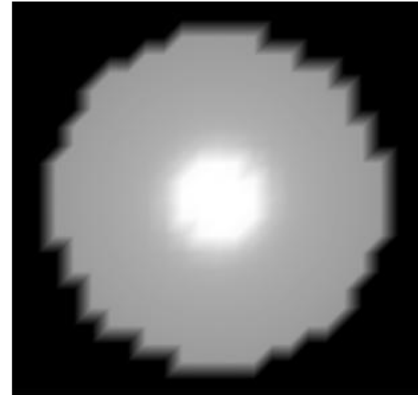
```
GLfloat light_direction[] = {0, 0, -1}; //From the light  
glLightfv( GL_LIGHT2, GL_SPOT_DIRECTION, light_direction);  
  
glLightf( GL_LIGHT2, GL_SPOT_CUTOFF, 20.);
```



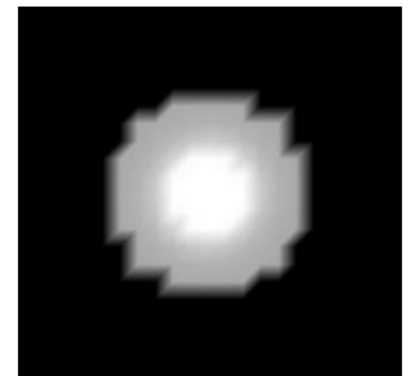
50



40



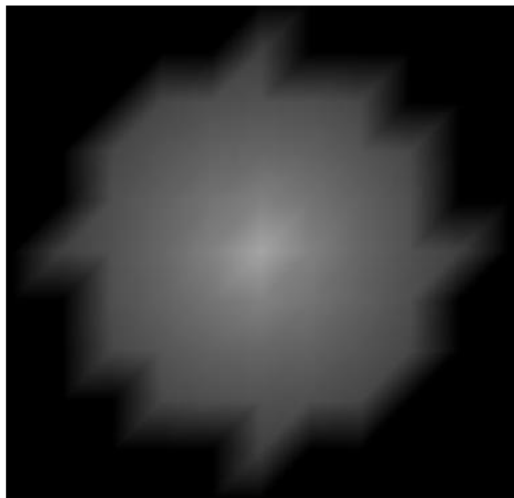
30



20

- GL\_SPOT\_EXPONENT мөн тодорхойлж болно
- Default утга нь 0 ба төвөөс тойрог хүртэлх гэрлийн intensity тархалтыг илэрхийлнэ.
- Өндөр цэгийн илтгэгч үр дүн нь төвийн эргэн тойронд өндөр эрчим (higher intensity) бий болгоно.

```
glLightf( GL_LIGHT2, GL_SPOT_EXPONENT, 10.);
```



$e = 0$



$e = 10$



$e = 100$



➤ shiny red материалын тусгалыг тодорхойлох

```
GLfloat diffuse[] = { .41, .135, .067, 1.};
```

```
GLfloat specular[] = {1., 1., 1., 1.};
```

```
glMaterialfv( GL_FRONT, GL_AMBIENT, diffuse);
```

```
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse);
```

```
glMaterialfv( GL_FRONT, GL_SPECULAR, specular);
```

```
glMaterialf( GL_FRONT, GL_SHININESS, 125.);
```



Цагаан гэрлийн эх  
үүсвэрийн чиглэл  
нь(1,1,1)

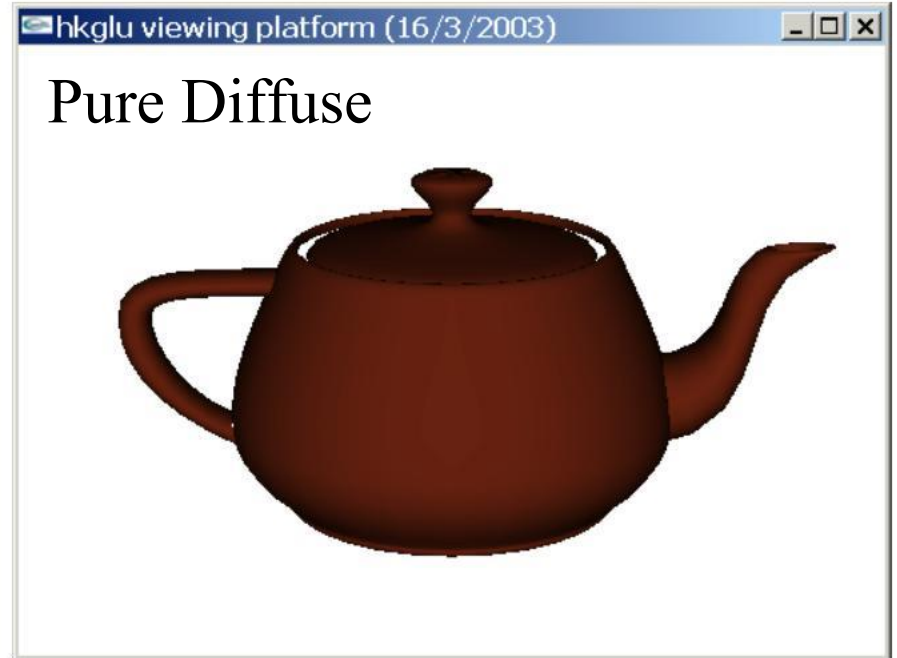
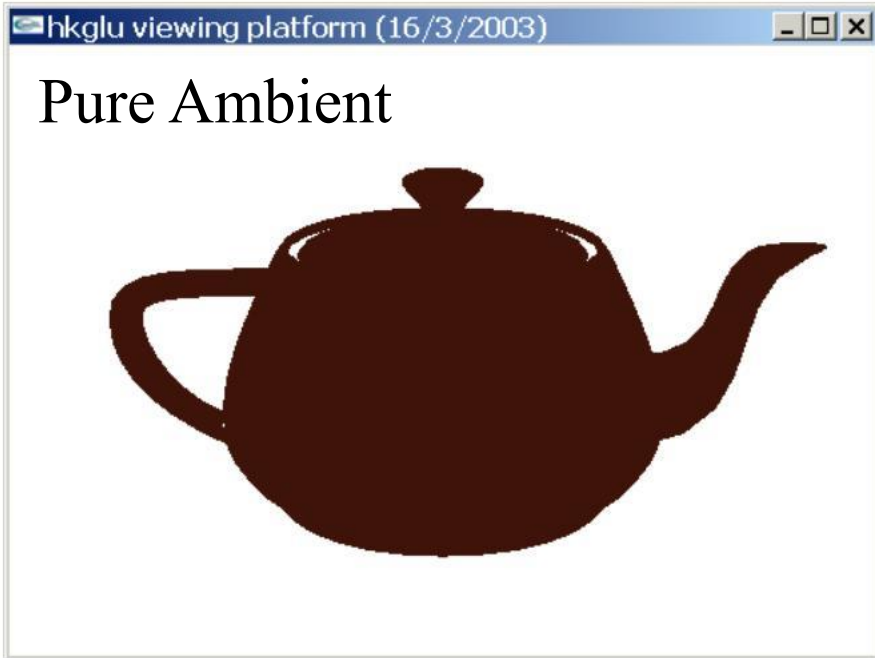
# Материалын тусгал

□ Спекулярын улаан, ногоон, цэнхэр бүрэлдэхүүн хэсгүүдэд тусгал нь ихэвчлэн ижил байдаг тул анхны өнгө нь гэрлийн эх үүсвэр туссан байна.

(Өмнөх зураг дээр цайны аяганы тодотгол нь цагаан, гэрэлтэй адил эх сурвалж юм.)

Цагаан гэрлийн эх үүсвэрийн чиглэл нь 1, гялалзах нь 100 ба түүнээс дээш байж болно. Dull surface гадаргуугийн гэрэл тусгал ба гялалзах байдал 0-тэй ойролцоо байна.

- Материалын өнгө сарнисан тусгалаас хамаарна.
- Зөвхөн сарнисан тусгал нь объектын муруйлт, гүний талаархи ихэнх мэдээллийг өгдөг.
- Материалын орчны тусгал нь ихэвчлэн сарнисан тусгалтай ижил байдаг.



Лок Яу Hall-ын цайвар шаргал ягаан материал

```
GLfloat ambient[] = { .6, .59, .42, 1.};
```

```
GLfloat diffuse[] = { .5, .5, .35, 1.};
```

```
GLfloat specular[] = {.1, .1, .07, 1.};
```

```
glMaterialfv( GL_FRONT, GL_AMBIENT, ambient);
```

```
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse);
```

```
glMaterialfv( GL_FRONT, GL_SPECULAR, specular);
```

```
glMaterialf( GL_FRONT, GL_SHININESS, 1.);
```



# Emission – материалын ялгарал

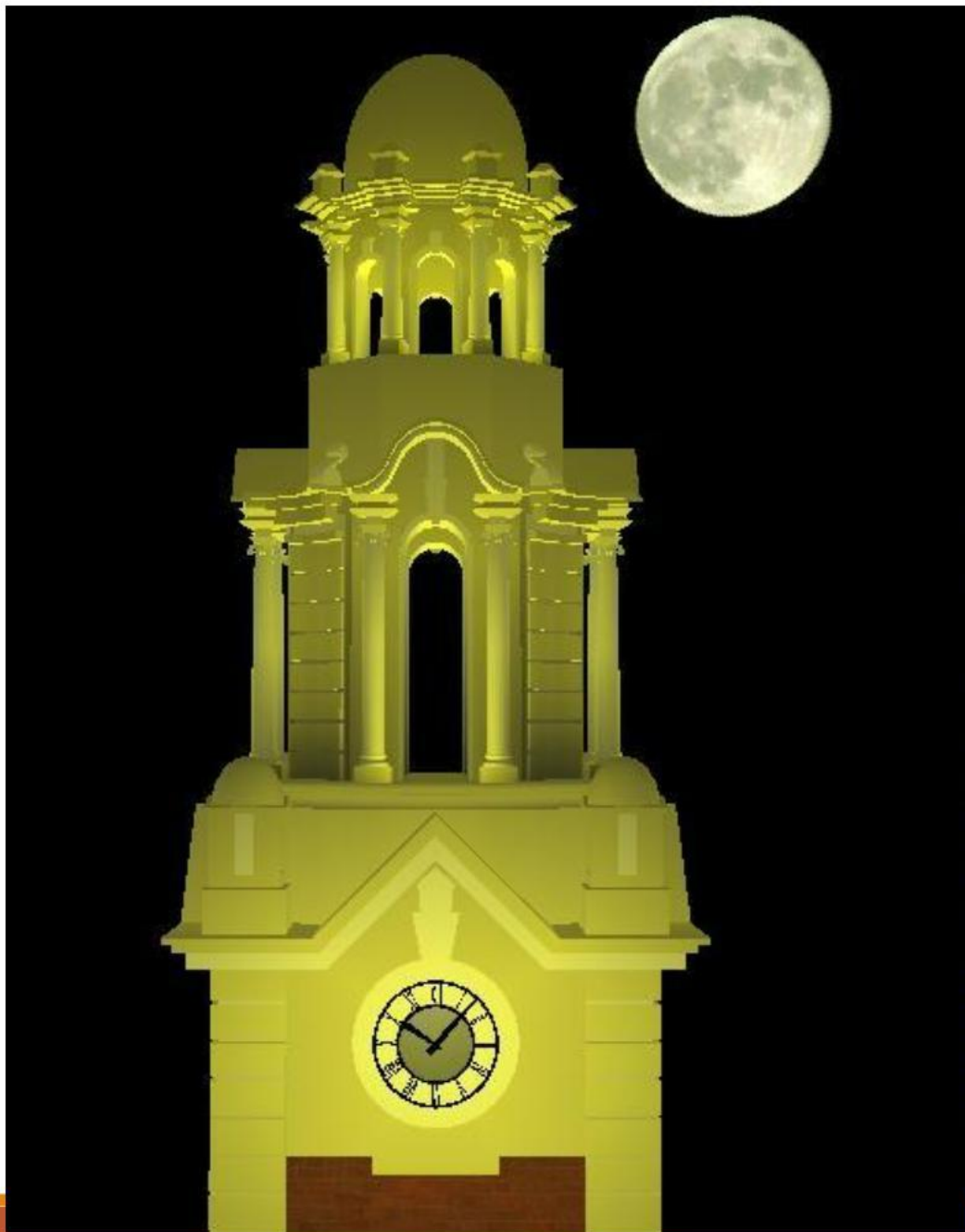
Материалыг гэрэлтэж байгаа мэт харагдуулахын тулд:

```
GLfloat emission[] = { r, g, b, a};  
glMaterialfv( GL_FRONT, GL_EMISSION, emission);
```

Ялгаралтыг унтраах: r-, g-, b-бүрэлдэхүүнүүдэд 0-ийг онооно.

Ялгарлын объект нь гэрлийн эх үүсвэр биш гэдгийг анхаарна уу. Ялгарал бол зөвхөн материалын өмч юм.

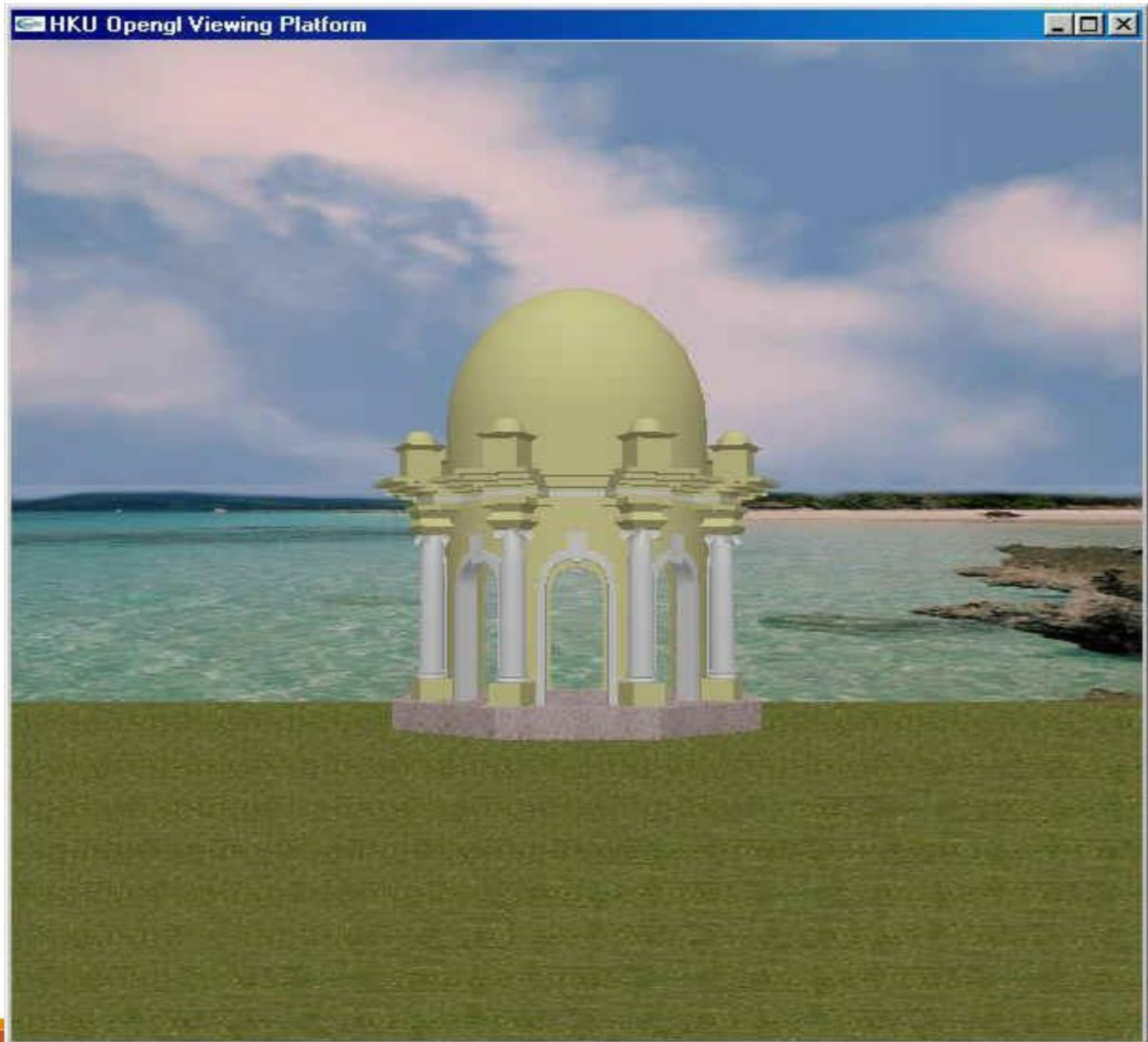
```
GLfloat emission[] = { 1., 1., .75, 1.};  
glMaterialfv( GL_FRONT, GL_EMISSION, emission);  
draw_moon();  
  
GLfloat no_em[] = { 0., 0., 0., 1.};  
glMaterialfv( GL_FRONT, GL_EMISSION, no_em);
```



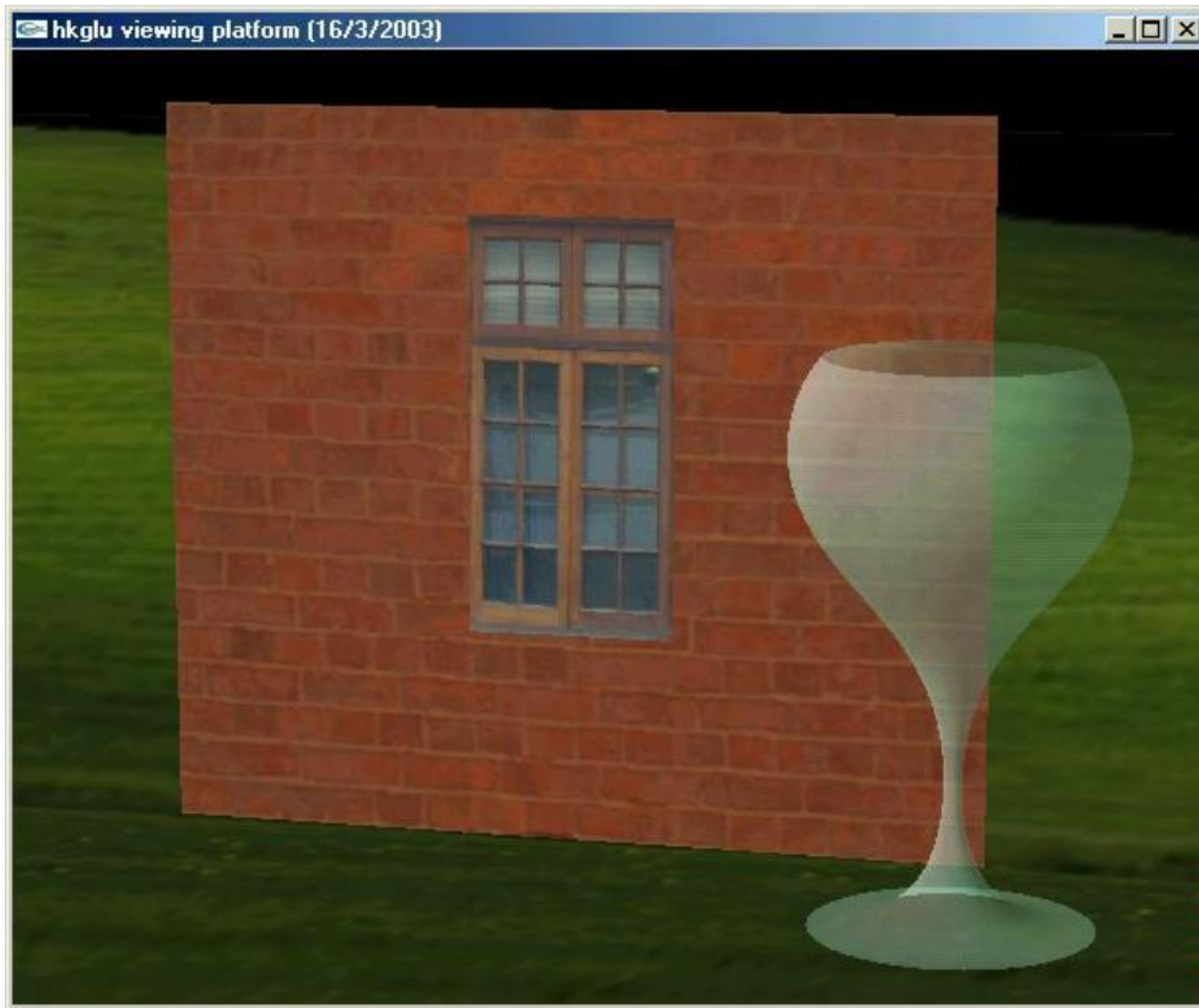
зурган дээрх сар.

5 төрлийн материал,  
Сар бол ялгаруулах объект  
юм.





# Нэвт гэрэлтдэг объектууд



Transparent

# Нэвт гэрэлтдэг объектууд

## 1. Блендинг хийх

```
glBlendFunc( GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);  
glEnable( GL_BLEND);
```

## 2. Define the material with the opacity less than 1.

```
void transMaterial1() {  
    GLfloat a[] = { .7, 1., .85, .3};  
    GLfloat s[] = { 1., 1., 1., 1.};  
  
    glMaterialfv( GL_FRONT, GL_AMBIENT, a);  
    glMaterialfv( GL_FRONT, GL_DIFFUSE, a);  
    glMaterialfv( GL_FRONT, GL_SPECULAR, s);  
  
    glMaterialf( GL_FRONT, GL_SHININESS, 100.);  
}
```

## 3. Draw the background objects before drawing the transparent objects. Draw both front and back faces as filled polygons.





Бүх OpenGL параметрууд, түүний дотор гэрэлтүүлэг нь анхдагч утгатай байна. Параметрийн утга нь шинэ утгад шилжих хүртэл хүчинтэй байна. Жишээлбэл, өгөгдмөл GL\_LIGHT0 нь цагаан гэрлийг гаргаж, харах цэг дээр байрладаг.

Гэрэлтүүлгийн үр нөлөөг урьдчилан таамаглахад хэцүү байдаг. Туршилт, туршлага нь гэрэлтүүлгийн параметруудийг тохируулахад маш чухал юм.

Компьютер дээрх OpenGL-д өгсөн гэрэлтүүлгийн дэмжлэг хамгийн бага байдаг.

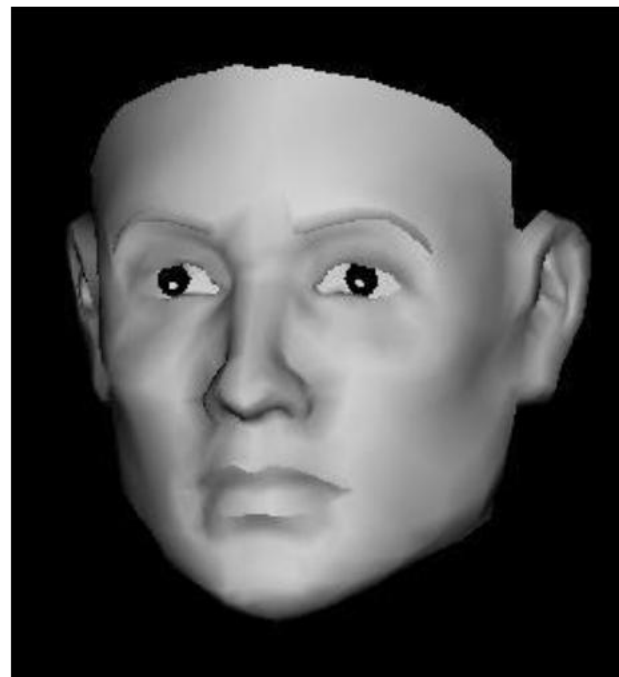
# Сүүдэр



Shadow

# Сүүдэрлүүлэх

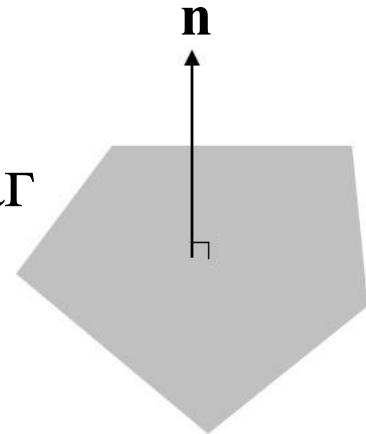
Геометрийн загвараас гэрэл зургийн бодит график үзэгдлүүдийг бий болгохын тулд бид физикийн хуулиудын дагуу материалын оптик шинж чанар, гэрлийн байрлал, харах цэг дээр үндэслэн гадаргуугийн гэрлийн эрчимийг тооцоолно.





# Constant (Flat) Shading - Тогтмол (хавтгай) сүүдэрлэх

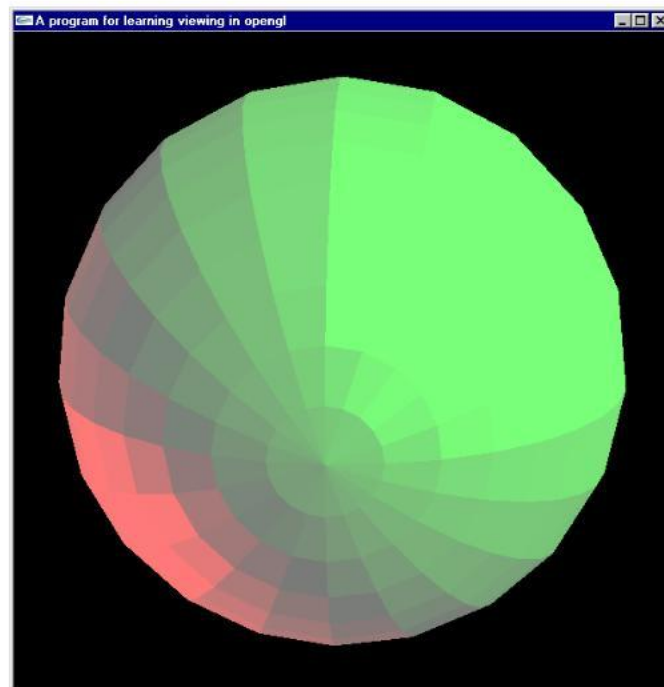
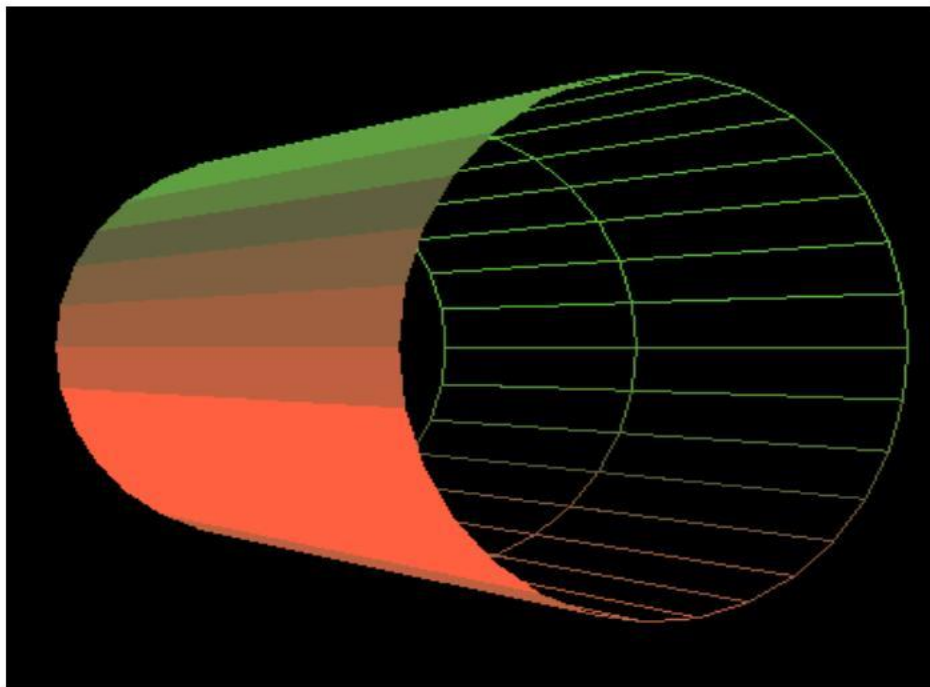
- Гадаргуугийн сүүдэр нь гадаргууд туссан ambient, diffuse ба specular гэрлийн хэмжээнээс хамаарах боломжтой. Мөн constant shading хувьд тусгалын тооцоог зөвхөн олон өнцөгт дээр жишээлбэл эхний оройн цэг дээр нэг удаа гүйцэтгэнэ. Үүссэн сүүдэр нь олон өнцөгтийг бүрхсэн бүх цэгүүдийн хувьд оногдоно.
- Тооцоололд гадаргуугийн нормал ашиглана.
- Олон өнцөгт бүхэлдээ сүүдэр ижил байна
- Үр дүнтэй хэдий ч болхи бүдүүлэг харагддаг

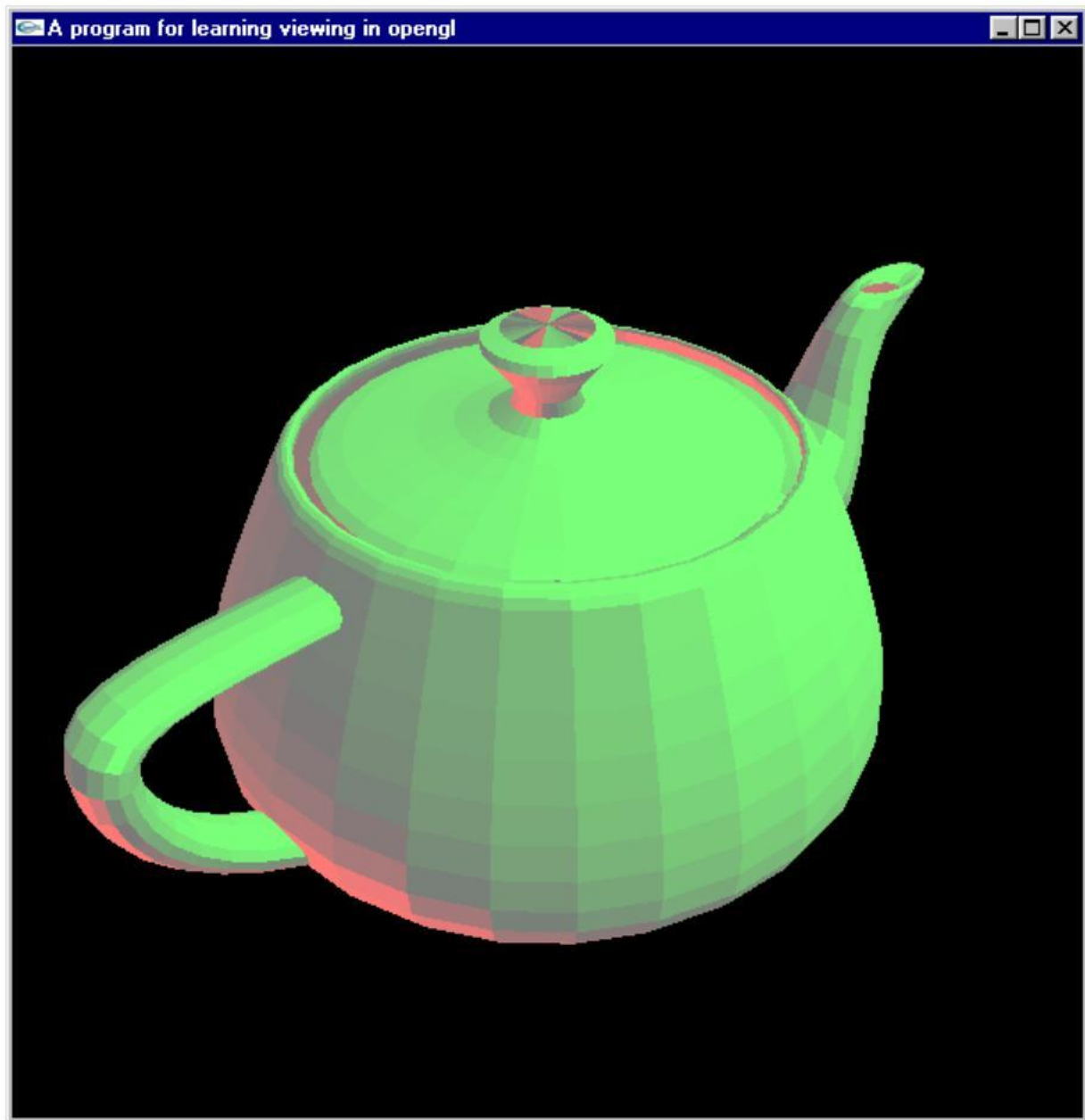




□ *A Mach band* гэдэг нь өөр өөр сүүдэртэй олон өнцөгтийн ирмэгээс олдсон судал юм. Мах хамтлагууд нь ихэвчлэн байнгын сүүдэрт үүсдэг.

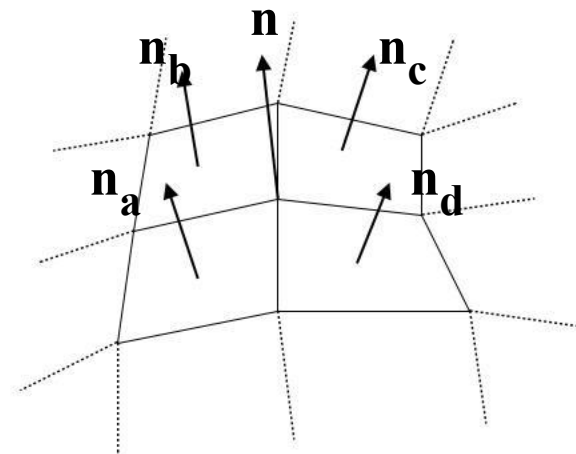
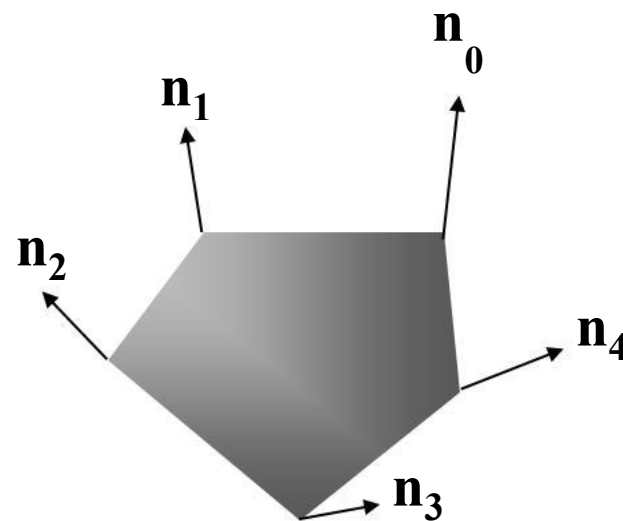
Бидний нүд сүүдрийн салангид өөрчлөлтөд онцгой мэдрэмтгий байдаг.





# Gouraud-ын сүүдэрлэлт


- Shading олон өнцөгтийн орой бүрд тооцоолж авч үзнэ.
- Орой бүрт олон өнцөгтийн нормал буюу зэргэлдээ олон өнцөгтийн дундаж нормалаар хуваарилагдана.
- Орой дахь цэгийн shade нь орой бүрт хувиарлагдсан нормалыг ашиглана.



- Олон өнцөгтийг бүрхсэн пикселийн сүүдрийг өнгөний нэгэн адил интерполяцаар гарган авч болно.



- Цэг хоорондын өнгө C дараах байдлаар тодорхойлогдоно

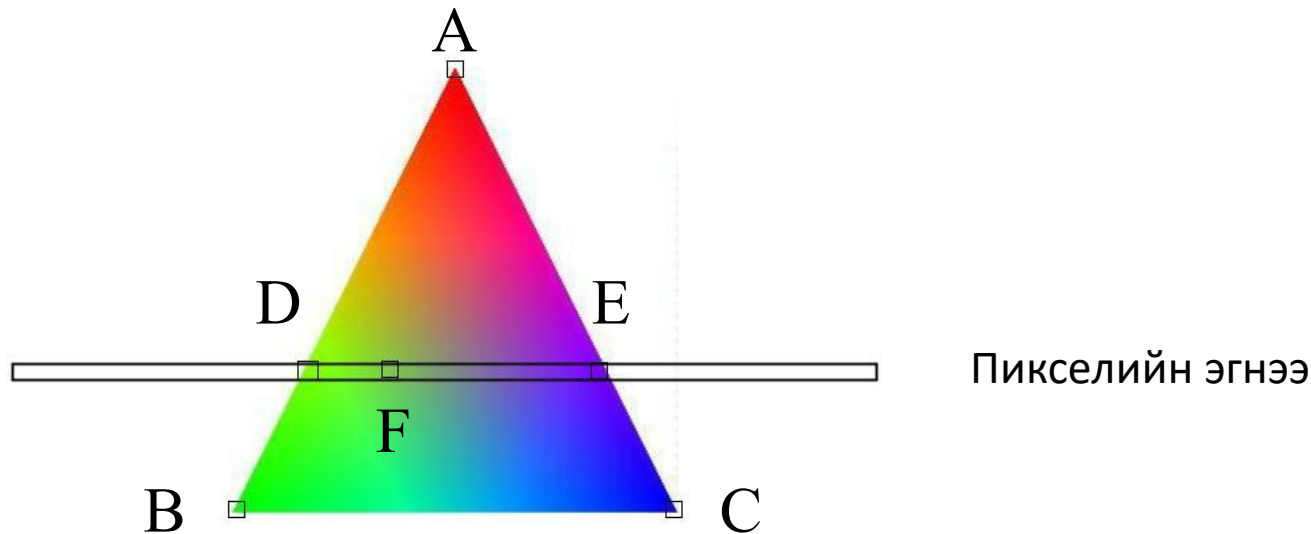


A diagram showing a horizontal line segment with three points marked by small squares. The leftmost point is red and labeled 'A' below it. The middle point is orange and labeled 'C' below it. The rightmost point is green and labeled 'B' below it. Below the diagram, the formula for the interpolation parameter  $\alpha$  is given:

$$\alpha = \frac{\text{distance of AC}}{\text{distance of AB}}$$

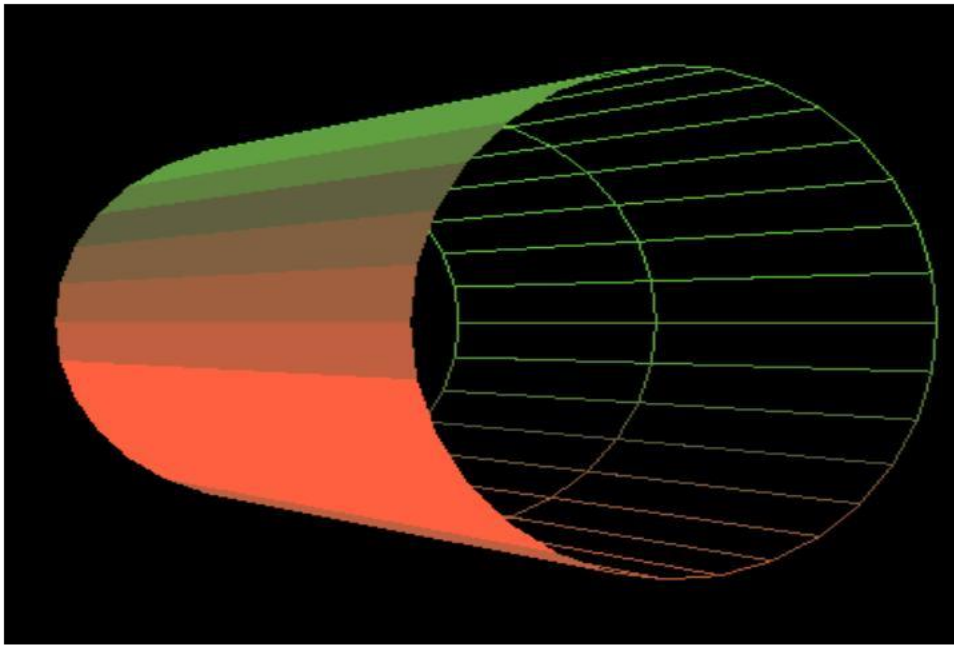
$$Color(C) = (1 - \alpha) \times Color(A) + \alpha \times Color(B)$$

➤ Олон өнцөгтийн пикселийн өнгийг тодорхойлох

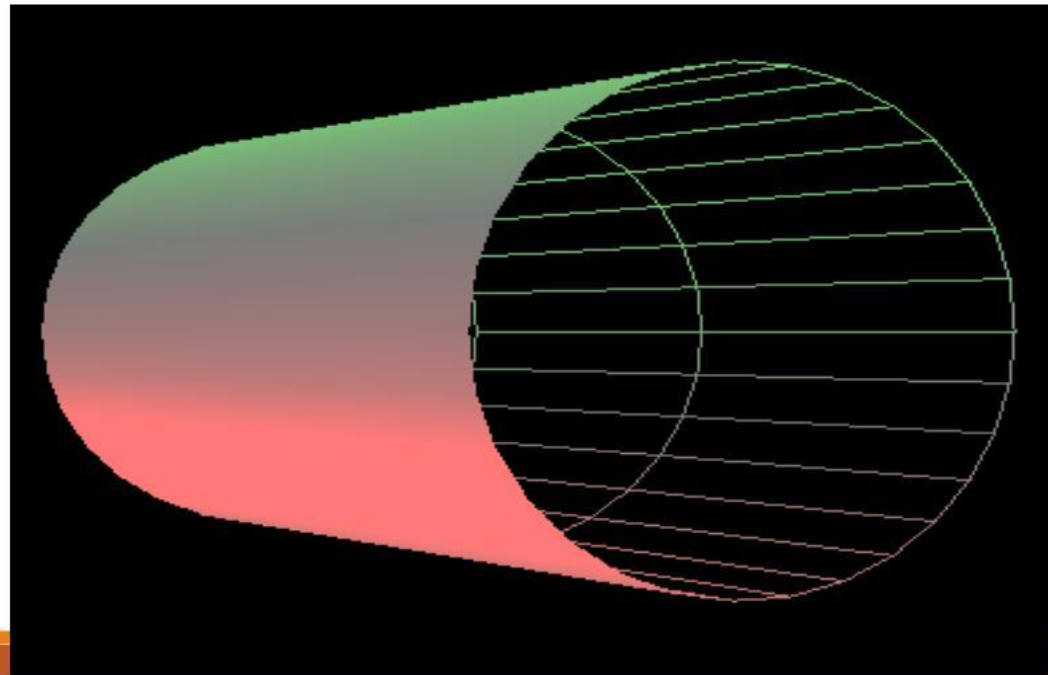


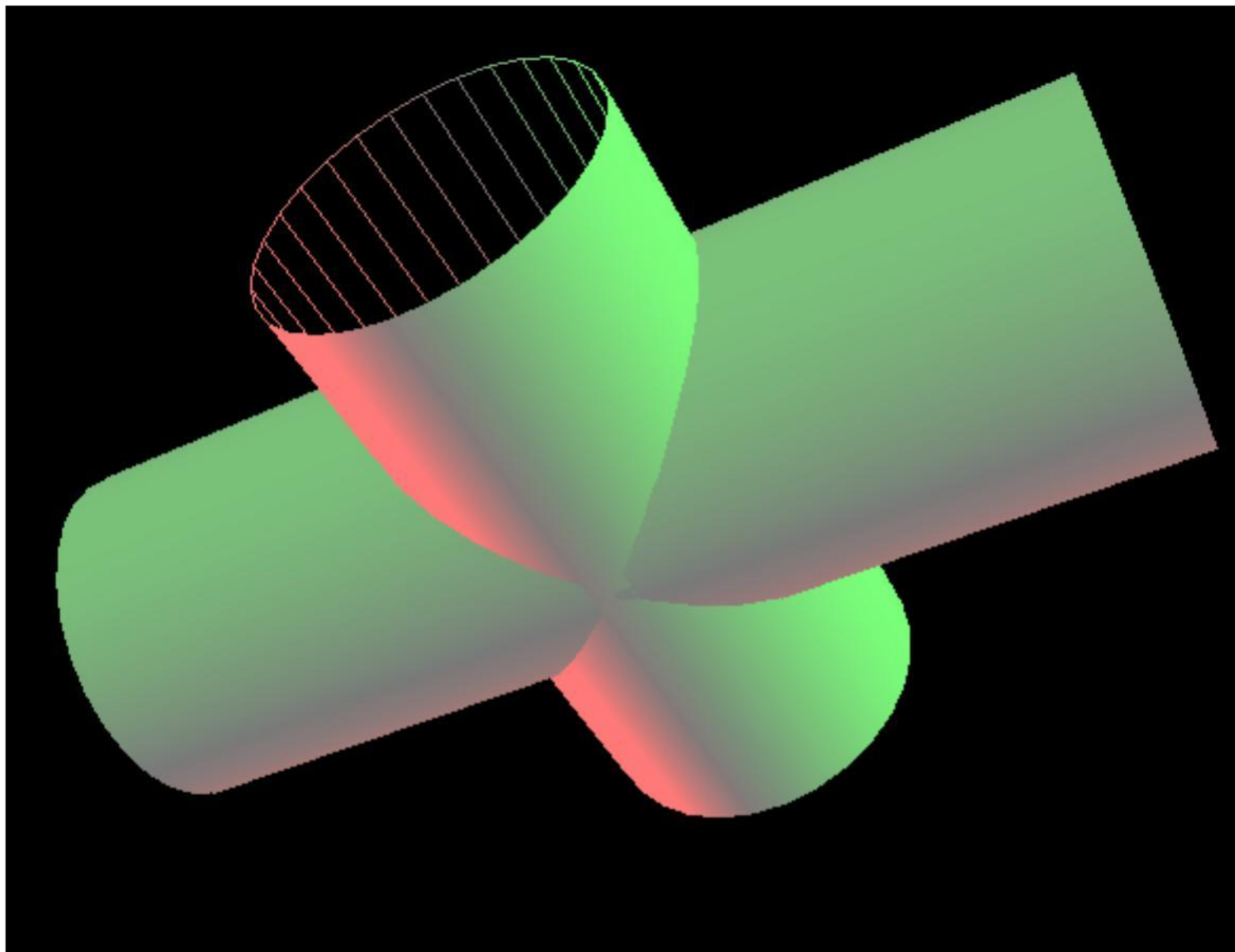
- D нь A ба B шулуун дээрх өнгөөр тодорхойлогдоно
- E нь A ба C шулууны хоорондох өнгөөр
- F нь D ба E шулууны хоорондох өнгөөр тодорхойлогдоно

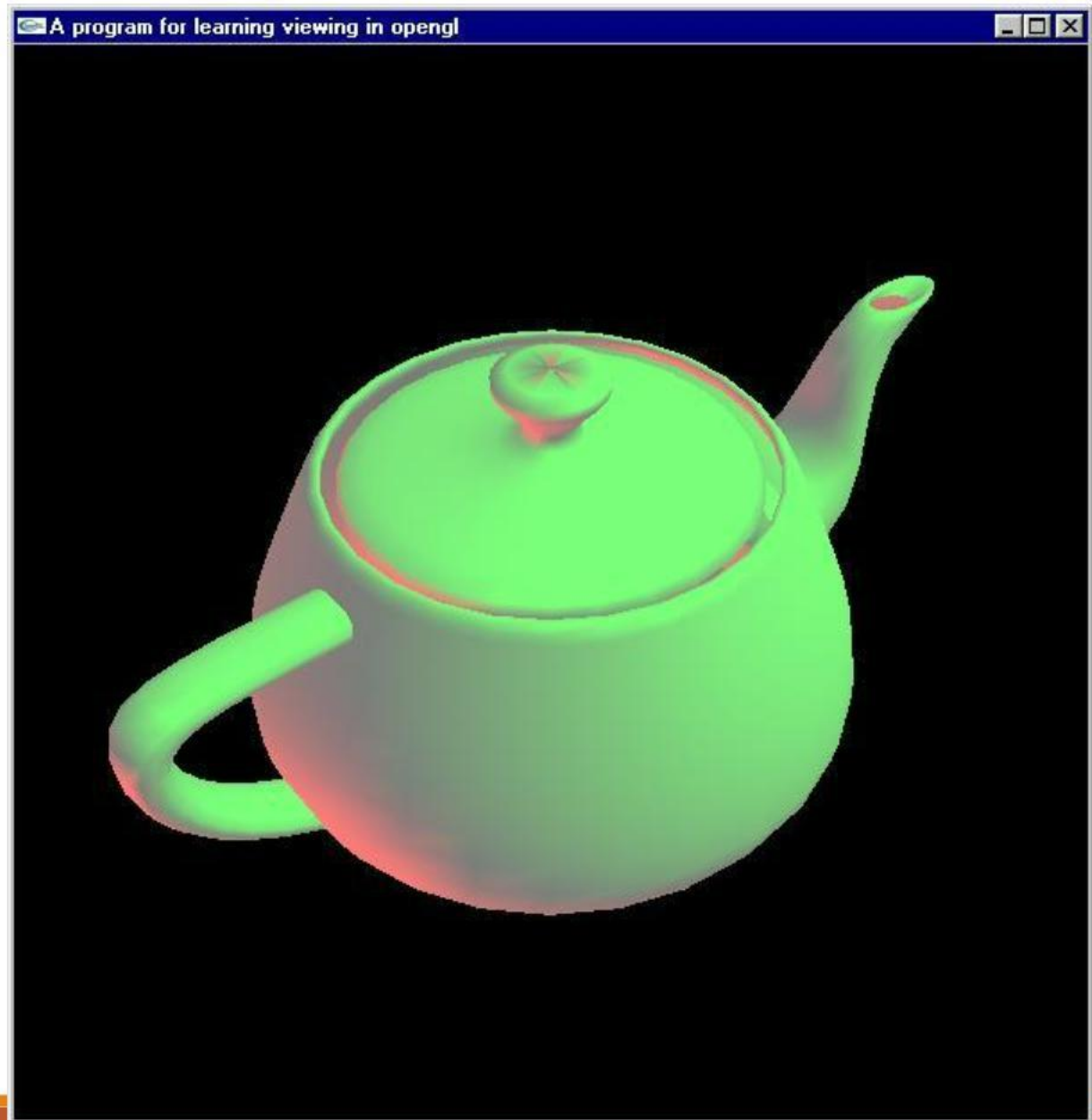
Constant Shading



Gouraud Shading



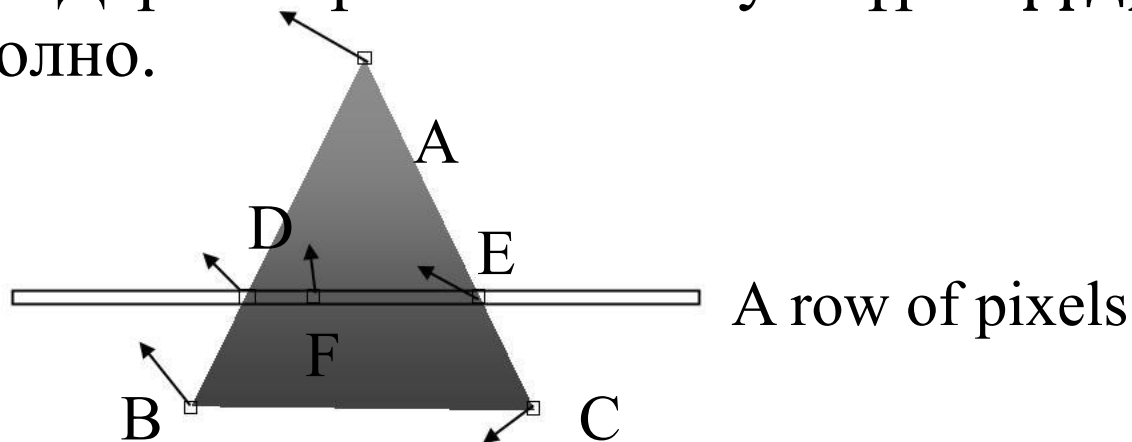






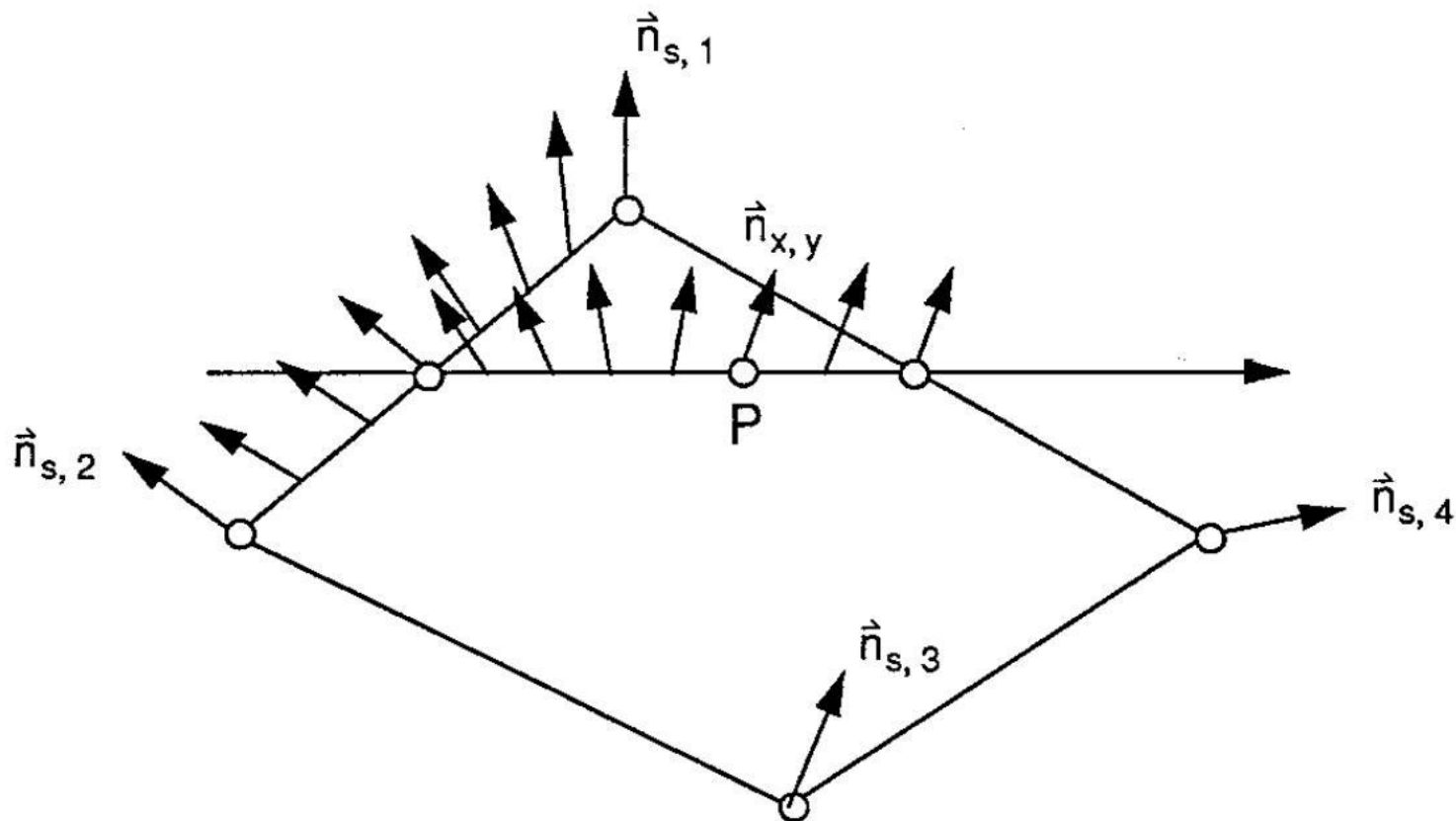
# Понг сүүдэр

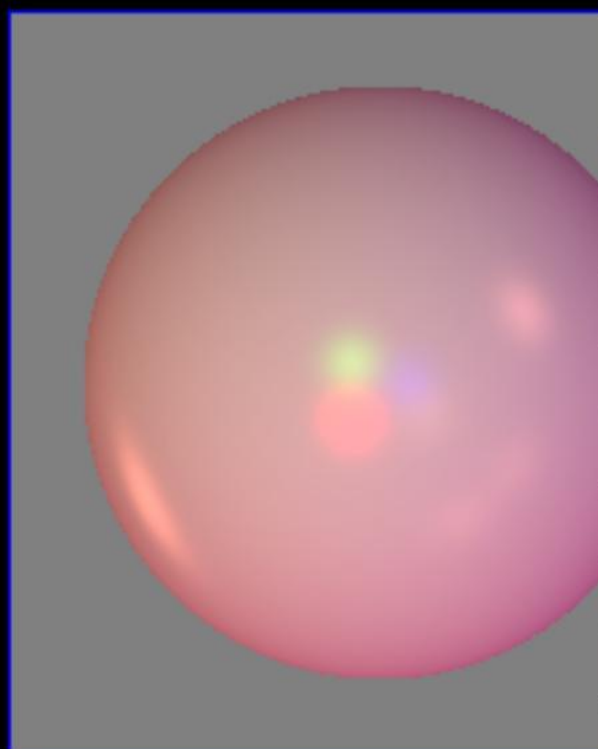
- Gouraud shading адил оройн эрчмийг интерполяц хийхийн оронд Phong олон өнцөгтийн нормалийг интерполяц хийдэг. Дараагаар нь пиксел тус бүрт сүүдрийг тооцоолно.



- Gouraud сүүдэрлэх аргыг ашиглан бид хангалттай үр дүнд хүрэхийн тулд гадаргууг маш жижиг хэсгүүдэд хуваах хэрэгтэй. Phong shading нь хэт олон зүсэлтийг бууруулдаг боловч илүү их тооцоо шаарддаг.
- Энэ нь чанартай зураг гаргаж болох боловч илүү удаан байдаг

- Ирмэгийн дагуух нормал болон пикселийн мөрийг интерполяци хийх





-----LIGHTING CONTROLS-----

R	G	B	X	Y	Z
0.3	0.3	0.3	2.0	1.0	1.0
0.5	0.5	0.5	-2.0	-1.0	-1.5
0.04	0.08	0.04	-1.0	1.0	7.0
0.1	0.13	0.1	2.0	-1.3	6.0
0.09	0.09	0.09	1.9	-3.1	1.7
0.09	0.09	0.09	3.1	-1.9	1.7
0.8	0.0	0.0	-1.0	-1.3	5.0
0.0	0.8	0.0	-1.0	0.7	5.0
0.0	0.0	0.9	1.0	0.0	5.0

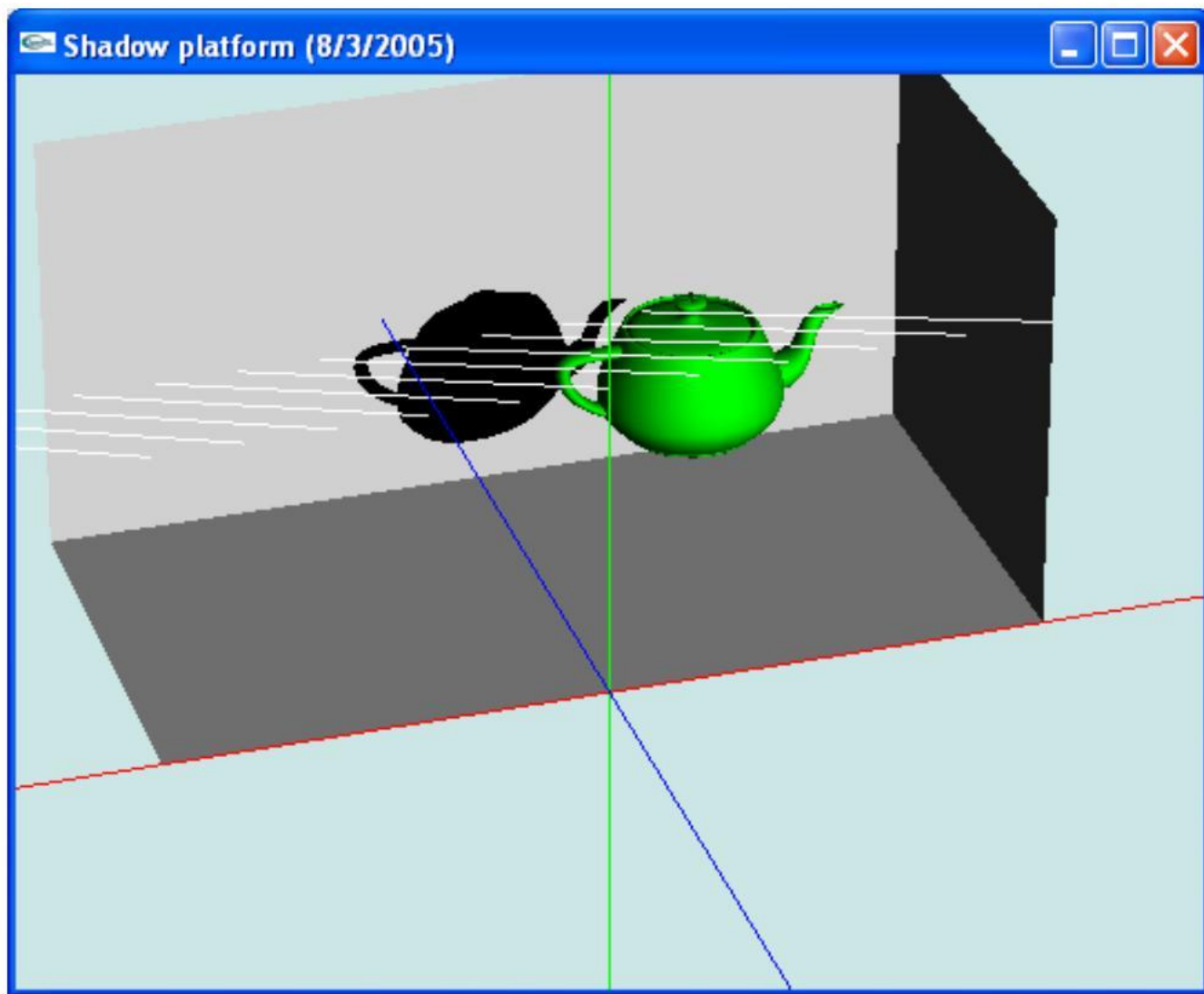
Render

Ambient Color: R  G  B

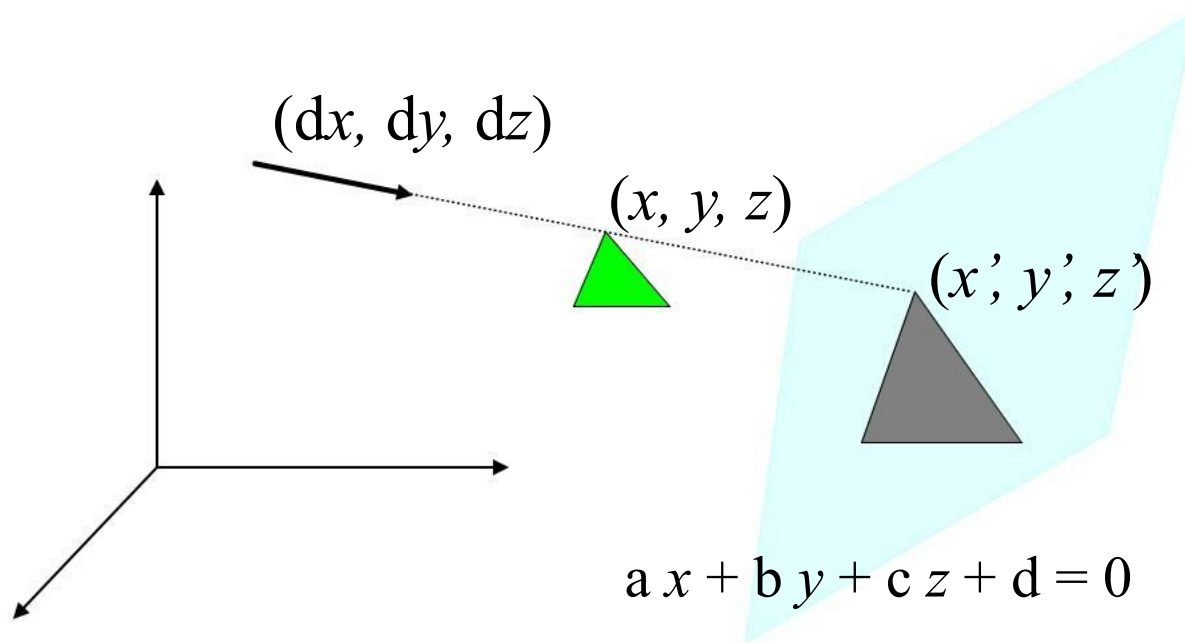
Diffuse Color: R  G  B

Specular Color: R  G  B  Exponent

➤ чиглэлтэй гэрлийн эх үүсвэр-с үүссэн сүүдэр



ShadowB



Гэрлийн чиглэл нь  $(dx, dy, dz)$ .

$(x', y', z')$  дах хавтгай дээрх  $(x, y, z)$  оройн сүүдэр.

Эндээс ,  $x' = x + \alpha dx$ ,  $y' = y + \alpha dy$ ,  $z' = z + \alpha dz$ .

Үүнээс гадна  $a(x + \alpha dx) + b(y + \alpha dy) + c(z + \alpha dz) + d = 0$ .

$\alpha = -(ax + by + cz + d) / (a dx + b dy + c dz)$

$$\alpha = - (ax + by + cz + d) / (adx + bdy + cdz)$$

$$x' = x - \frac{ax + by + cz + d}{adx + bdy + cdz} dx$$

$$x' = \frac{(bdy + cdz)x - (bdx)y - (cdx)z - d dx}{adx + bdy + cdz}$$

$$y' = \dots$$

$$z' = \dots$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} b \times dy + c \times dz & -b \times dx & -c \times dx & -d \times dx \\ -a \times dy & a \times dx + c \times dz & -c \times dy & -d \times dy \\ -a \times dz & -b \times dz & a \times dx + b \times dy & -d \times dz \\ 0 & 0 & 0 & a \times dx + b \times dy + c \times dz \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

M матрицыг хавтгай дээр оройн проекц гаргахаар тохируул

```
// ax + by + cz + d = 0.  
// Гэрлийн чиглэл нь(dx, dy, dz).
```

```
void directionalLightShadow( double dx, double dy, double dz,  
                             double a, double b, double c, double d, GLfloat M[16])  
{  
    M[0] = b*dy+c*dz; M[4] = -b*dx;      M[8] = -c*dx; M[12] = -d*dx;  
    M[1] = -a*dy;      M[5] = a*dx+c*dz; M[9] = -c*dy; M[13] = -d*dy;  
    M[2] = -a*dz;      M[6] = -b*dz;     M[10] = a*dx+b*dy; M[14] = -d*dz;  
    M[3] = 0;          M[7] = 0;         M[11] = 0; M[15] = a*dx+b*dy+c*dz;  
}
```

$z + 5 = 0$  хавтгай дээр shadow зурах.  
Гэрлийн чиглэл нь  $(dx, dy, dz)$  байна.

```
GLfloat M[16];  
double  dx = ...,  
        dy = ..., dz = ... ;
```

```
glDisable( GL_LIGHTING);
```

```
glColor4f( 0., 0., 0., 1.);
```

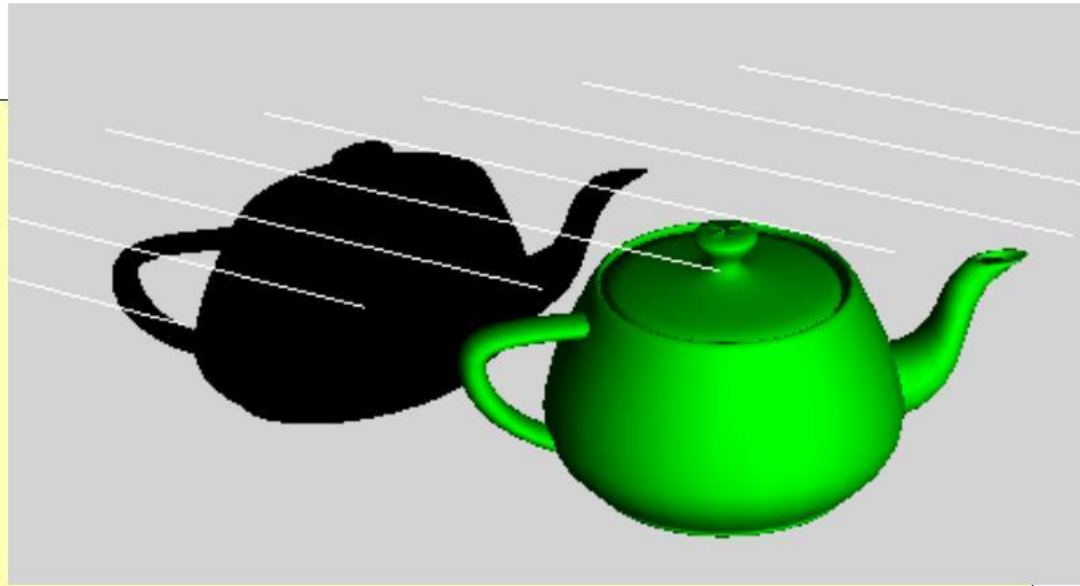
```
glPushMatrix();
```

```
    directionalLightShadow( dx, dy, dz, 0, 0, 1, 5, M);
```

```
glMultMatrixf( M);
```

```
drawTeapot();    //teapot зурах
```

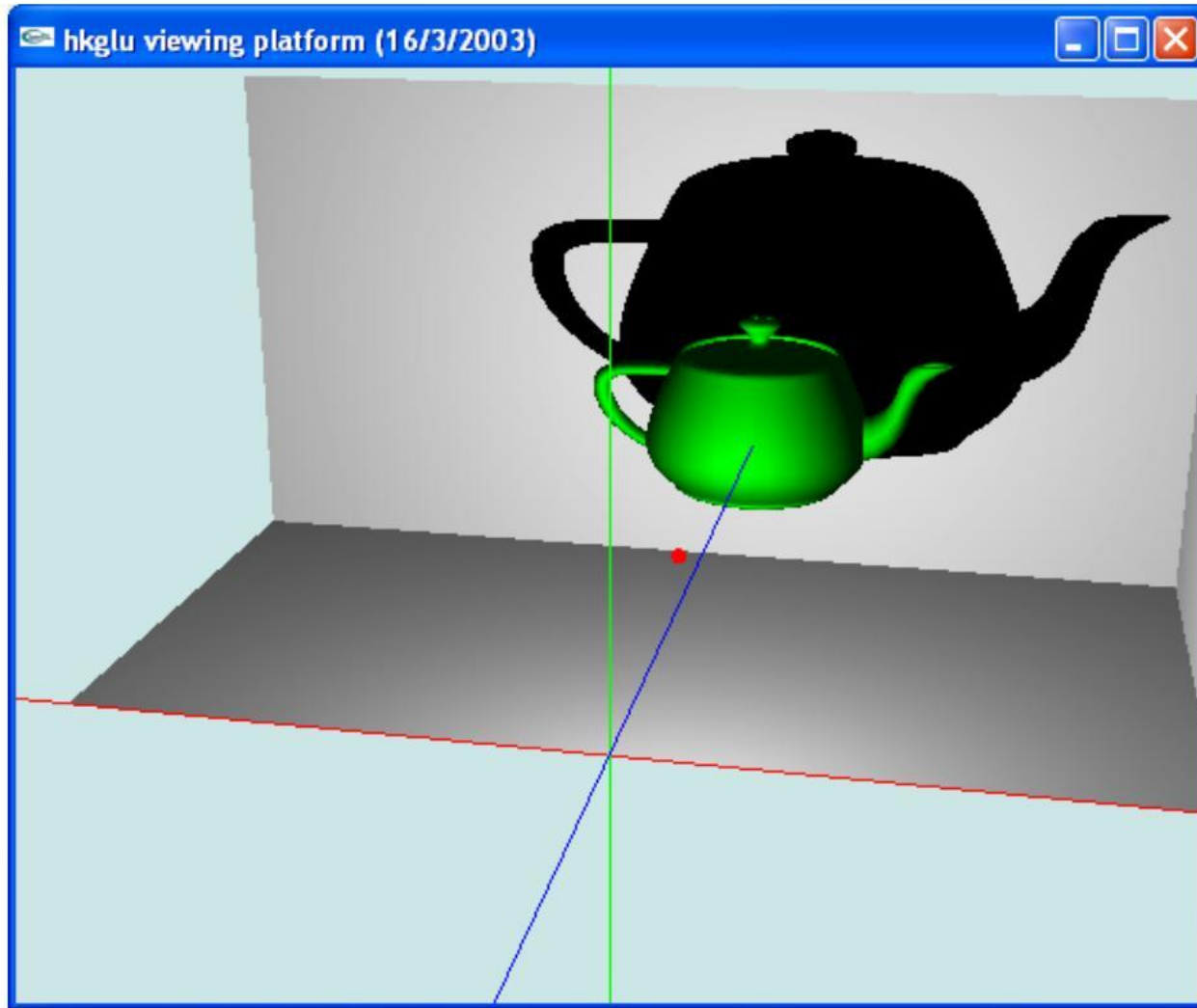
```
glPopMatrix();
```



ShadowB



➤ point light source үүсэх сүүдэр



ShadowA

point light source –г координатын эх дээр байна гэж үзье.

$(x', y', z')$  хавтгай дээрх  $(x, y, z)$  оройн сүүдэр

Эндээс  $x' = \alpha x$ ,  $y' = \alpha y$ ,  $z' = \alpha z$ .

Түүнчлэн  $(\alpha x) + b(\alpha y) + c(\alpha z) + d = 0$ .  $ax + by + cz + d = 0$

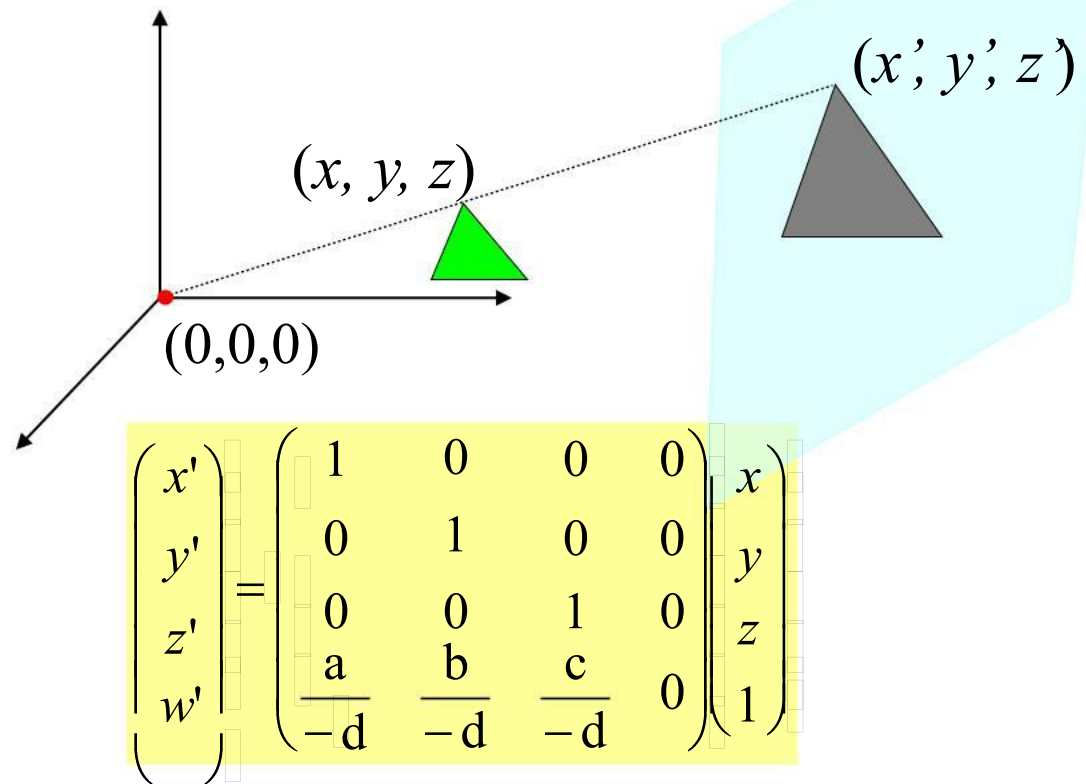
$$\alpha = -d / (ax + by + cz)$$

$$x' = -d x / (ax + by + cz)$$

$$x' = \frac{x}{\frac{a}{-d}x + \frac{b}{-d}y + \frac{c}{-d}z}$$

$$y' = \dots$$

$$z' = \dots$$



Хэрэв light source координатын эх дээр биш  $(l_x, l_y, l_z)$  дээр байвал координатын эхийг light source рүү шилжүүлнэ.

$$ax + by + cz + d = 0$$

$$x = x - l_x$$

$$y = y - l_y$$

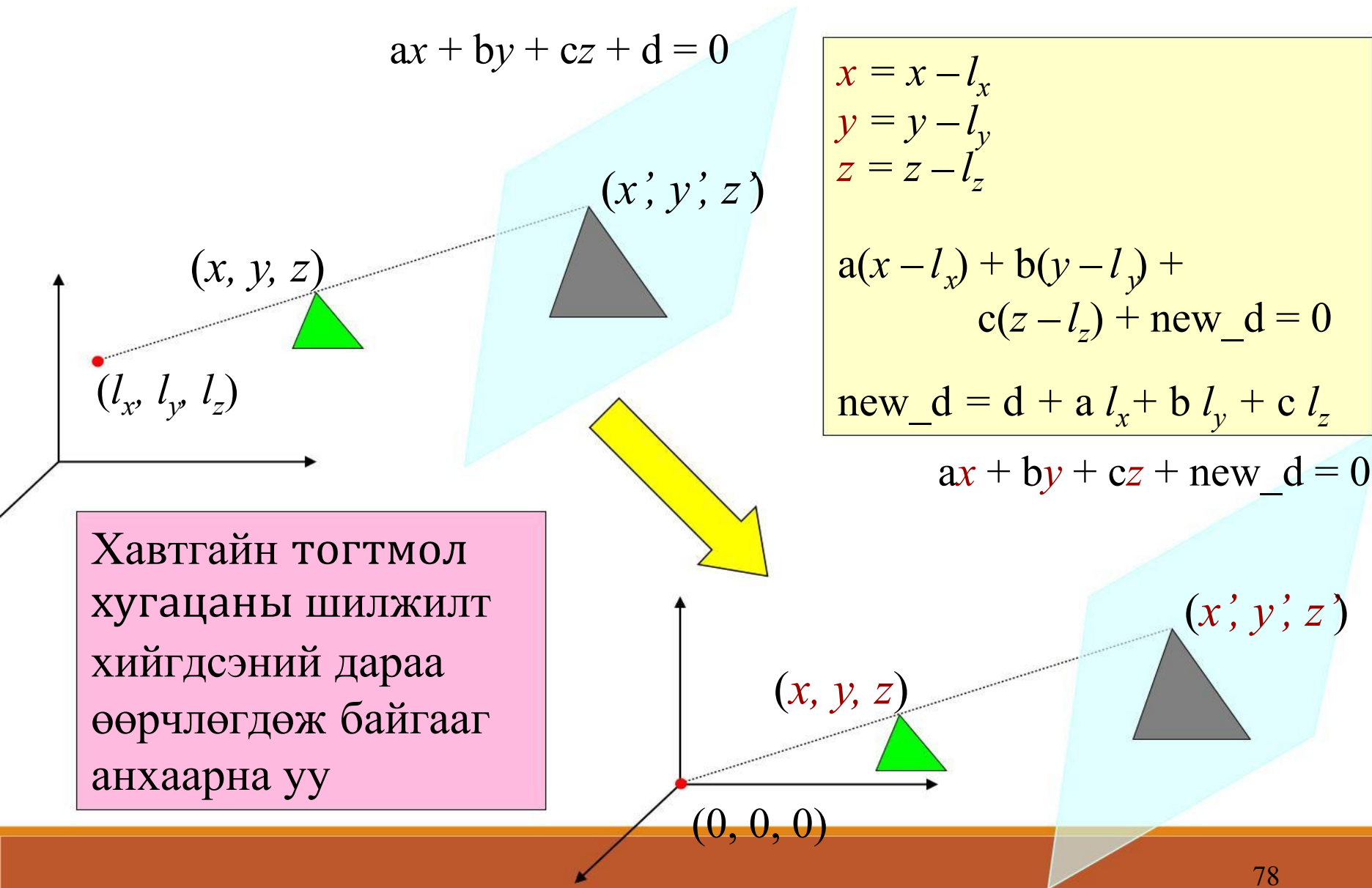
$$z = z - l_z$$

$$a(x - l_x) + b(y - l_y) + c(z - l_z) + \text{new\_d} = 0$$

$$\text{new\_d} = d + a l_x + b l_y + c l_z$$

$$ax + by + cz + \text{new\_d} = 0$$

Хавтгайн тогтмол хугацаны шилжилт хийгдсэний дараа өөрчлөгдөж байгааг анхаарна уу



```
void pointSourceShadow( double lx, double ly, double lz,  
                        double a, double b, double c, double d, GLfloat M[16]) {  
    double new_d = d + (a * lx + b * ly + c * lz);  
    M[0] = 1;           M[4] = 0;           M[8] = 0;           M[12] = 0;  
    M[1] = 0;           M[5] = 1;           M[9] = 0;           M[13] = 0;  
    M[2] = 0;           M[6] = 0;           M[10] = 1;          M[14] = 0;  
    M[3] = -a/new_d;    M[7] = -b/new_d;    M[11] = -c/new_d;    M[15] = 0;  
  
}
```

$z + 5 = 0$  хавтгай дээр сүүдэр байгуулах.  
Гэрэл ( $light\_x, light\_y, light\_z$ ) байна.

```
GLfloat M[16];  
double light_x = ..., light_y = ...,  
        light_z = ...;  
  
glDisable( GL_LIGHTING);  
glColor4f( 0., 0., 0., 1.);  
  
glPushMatrix();  
    glTranslatef( light_x, light_y, light_z);  
        pointSourceShadow( light_x, light_y, light_z, 0, 0, 1, 5, M);  
    glMultMatrixf( M);  
    glTranslatef( -light_x, -light_y, -light_z);  
        drawTeapot();    //teapot зурх  
glPopMatrix();
```

