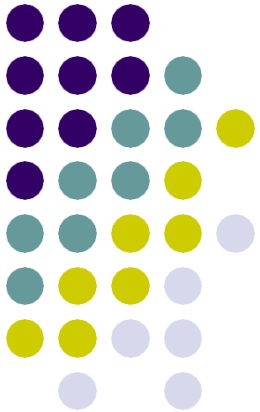
Компьютерийн график (CS209)
Лекц 14: Гэрэл, Сүүдэр ба Материал
Lighting, Shading and Materials

Ч.Цэнд-Аюуш

Компьютерийн ухааны салбар

ШУТИС, МХТС



Хичээлийн агуулга

- Гэрлийн үүсгүүр
  - Point sources
  - Directional sources
- Тусгал
  - Diffuse reflection
  - Specular reflection
  - Ambient reflection
- Gouraud Shading
- Phong Shading
- Shadows

Гэрэлтүүлэг - Lighting

- Гэрэл гэдэг нь товчхондоо дулаан биетээс цацрах фотонууд юм. Ямар хэмжээтэйгээр цацаргасанаас хамаараад гэрэл өөр нэг гадаргуу дээр очихдоо янз бүрийн тоо хэмжээтэйгээр очдог.
- 3D програмчлалд гэрэлтүүлгийг загварчлан тооцон олох нь эдгээр фотонуудын тоо хэмжээг ойролцоогоор загварчлан олох явдал юм.
- Компютер графикт гадаргуу дээр туссан гэрлийн эрчим буюу light intensity утгыг олоход хэрэглэгддэг гурвантөрлийн гэрлийн ойлтын утггууд байдаг.

Гэрэлтүүлэг - Lighting

- Нартай, эсвэл үүлэрхэг өдөр ус, далай өөр өөр өнгөтэй байдаг. Гэрэлтэйд ус тод цэнхэр, харин үүлтэйд бохир ногоон харагдах нь бий.
- Ихэнх объектууд гэрэлтүүлэггүйгээр гурван хэмжээст харагдаж чаддаггүй байна. Гэрэлтүүлэг нь материал, гэрлийн эх үүсвэр хоорондын харилцааг тодорхойлно.
- Гэрэлтүүлэг: гадаргуу дээрх цэгийн өнгөний эрчимжилтийг тооцоолно.

Гэрэлтүүлсэн ба гэрэлтүүлээгүй

Биет дээр туссан гэрлийн цацраг хаачдаг юм бол?

- Хугарна
- Ойно
- шингэнэ

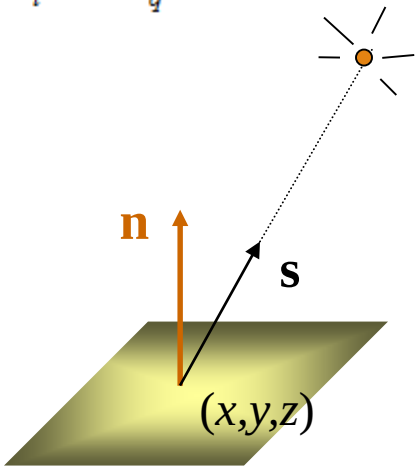
Point resource

$$attenuation = \frac{1}{k_c + k_l d + k_q d^2}$$

- **n** нь гадаргуугийн нормал.

  n хавтгай дээрх өөр өөр байршилд ижил байна

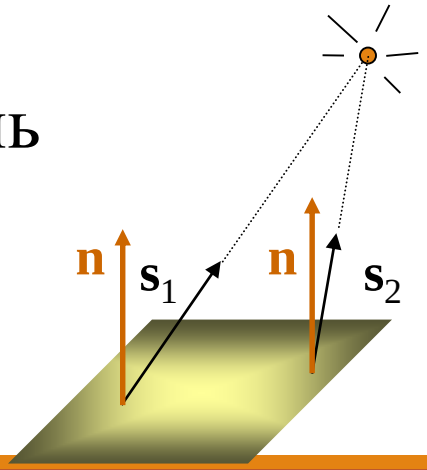
- (*x*, *y*, *z*) нь гадаргуу дээрх цэгийн координат



- **s** нь (x, y, z) дэх гэрлийн эх үүсвэрийн чиглэл юм.

  s нэг байршилаас нөгөөд шилжинэ, жнь

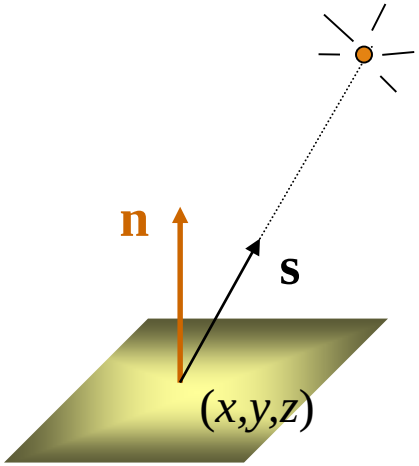
  $\mathbf{s}_1 \neq \mathbf{s}_2$. Хэрэв гэрлийн эх үүсвэр гадаргууд ойрхон байвал s-ийн өөрчлөлт их байна.



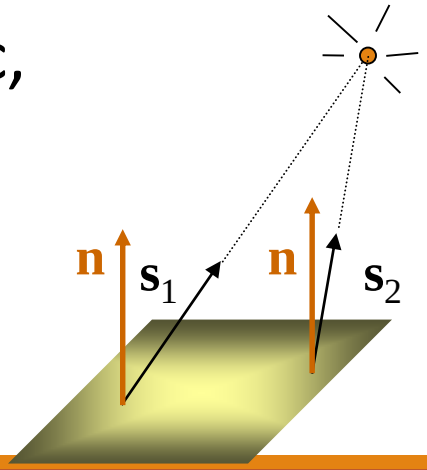
Point sources

Объектийн гадаргуу дээрхи point дээр хүрч очих гэрлийн intensity утгыг нь дараах томъёогоор тооцон олдог.

$$attenuation = \frac{1}{k_c + k_l d + k_q d^2}$$

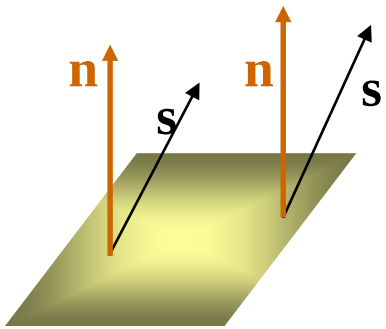


Энд **d** нь гэрлийн гадаргуу дээр туссан цэг болон тэндээс гэрлийн үүсгүүрийн байрлал хүртлэх зайн урт юм. Харин Kc, Kl, Kq утгууд нь харгалзан linear, quadratic, attenuation тогтмолууд юм. attenuation гэдэг нь гэрлийн шингэлт буюу сулрах зэргийг нь илтгэнэ.

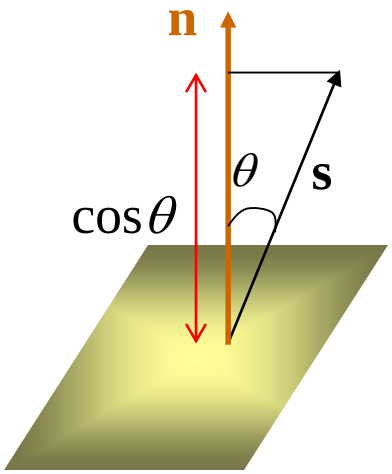


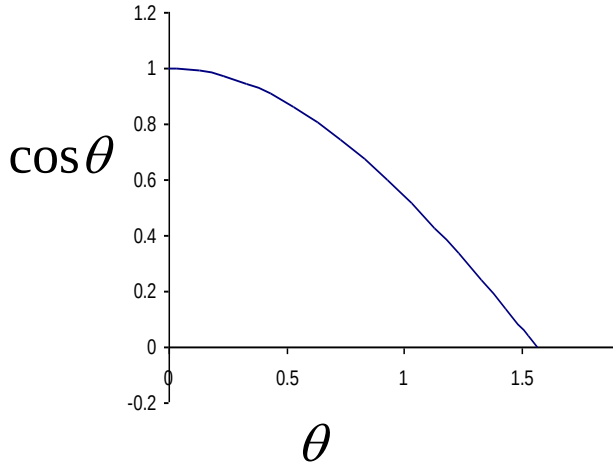
Directional sources

- Гэрлийн эх үүсвэр хол байх үед s-ийн өөрчлөлт бага байна. Зай холдоход s гадаргуу дээр өөрчлөгдөхгүй, жишээлбэл, нар. Үүнийг гэрлийн чиглэлийн эх үүсвэр (эсвэл зэрэгцээ эх үүсвэр, эсвэл тархсан эх үүсвэр) гэж нэрлэдэг.



- I_i be the тусах гэрлийн intensity
- θ, **s** ба **n** хоорондох өнцөг, $0^o \le \theta \le 90^o$.
- $I_i \cos\theta$ гадаргуугийн нэгж талбайд хүлээн авсан гэрлийн хэмжээ
- $\theta = 0$ *үед*, тусах гэрэл нь гадаргууд перпендикуляр, гадаргуу гэрлийг хамгийн дээд хэмжээгээр хүлээн авах бөгөөд жнь үдийн нарны гэрэл
- $\theta = 90^o$ үед, гэрэл хүлээн авахгүй.
- $\theta > 90^o$ үед, гэрлийн үүсгүүр гадаргуугийн нөгөө талд бий тул гэрэл хүлээн авахгүй





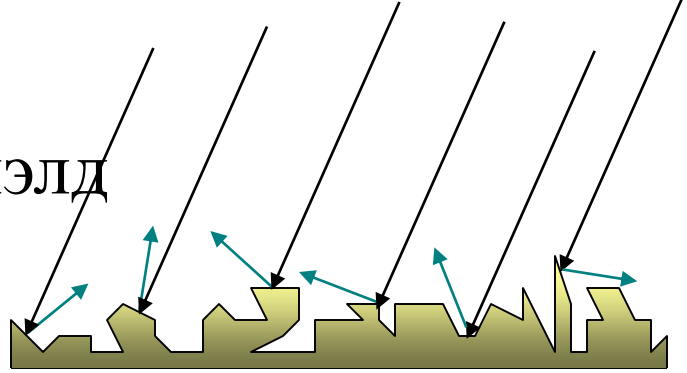
Diffuse ойлт

diffuse гадаргуу дээр гэрлийн цацрагийн хэсгүүд цаашаагаа янз бүрийн чиглэлд санамсаргүйгээр ойн цацардаг. Үүний үр дүнд гэрэл чиглэл бүрт нэгэн хэвийн ойж харагддаг. diffuse ойлтыг тооцож олох олон аргууд байдгаас lambertian reflection арга нь хамгийн хурдан нь юм.

Гэхдээ илүү нарийн бүтэцтэй Oren-Nayar ийн арга ч гэж бий.

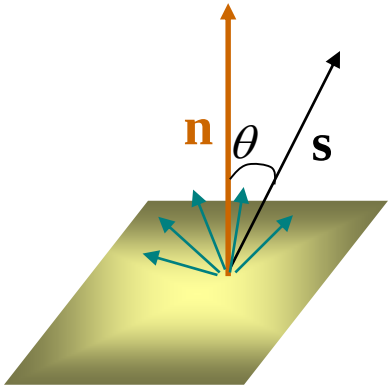
Diffuse

Diffuse ойлт



- Тэгш гадаргуу гэрлийг бүх чиглэлд тусгадаг
- Lambert cosine law

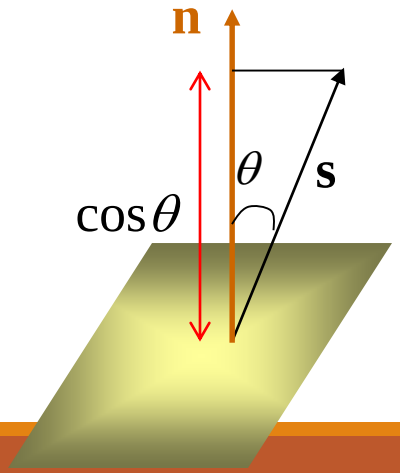
  Туссан гэрлийн хэмжээ нь $I_d = R_d I_i \cos\theta$



- R_d , the diffuse reflection coefficient, $0 \leq R_d \leq 1$
- I_d нь харах цэг (view point) тусдаа
- $\cos\theta$ *нь* ***n*** *дээрх* **s** –ийн проекц.

  Хэрэв **s** ба **n** нормчлогдсон бол,

  $\cos\theta$ = **s** • **n** (the dot–product)



Attenuation

- Гэрэл гадаргуугаас холдох үед, хүлээн авах эрчим буурдаг.
- point light source-ийн хувьд attenuation factor нь

$$A_f = \frac{1}{d^2} \quad \text{(онолоор)}$$

$$A_f = \frac{1}{a + bd + cd^2} \quad \text{(практикт)}$$

d нь point light source-с алслагдсан зай

a, b, ба c нь эмпирик байдлаар сонгосон эвристик параметрүүд юм..

- directional light source хувьд, $A_f = 1$

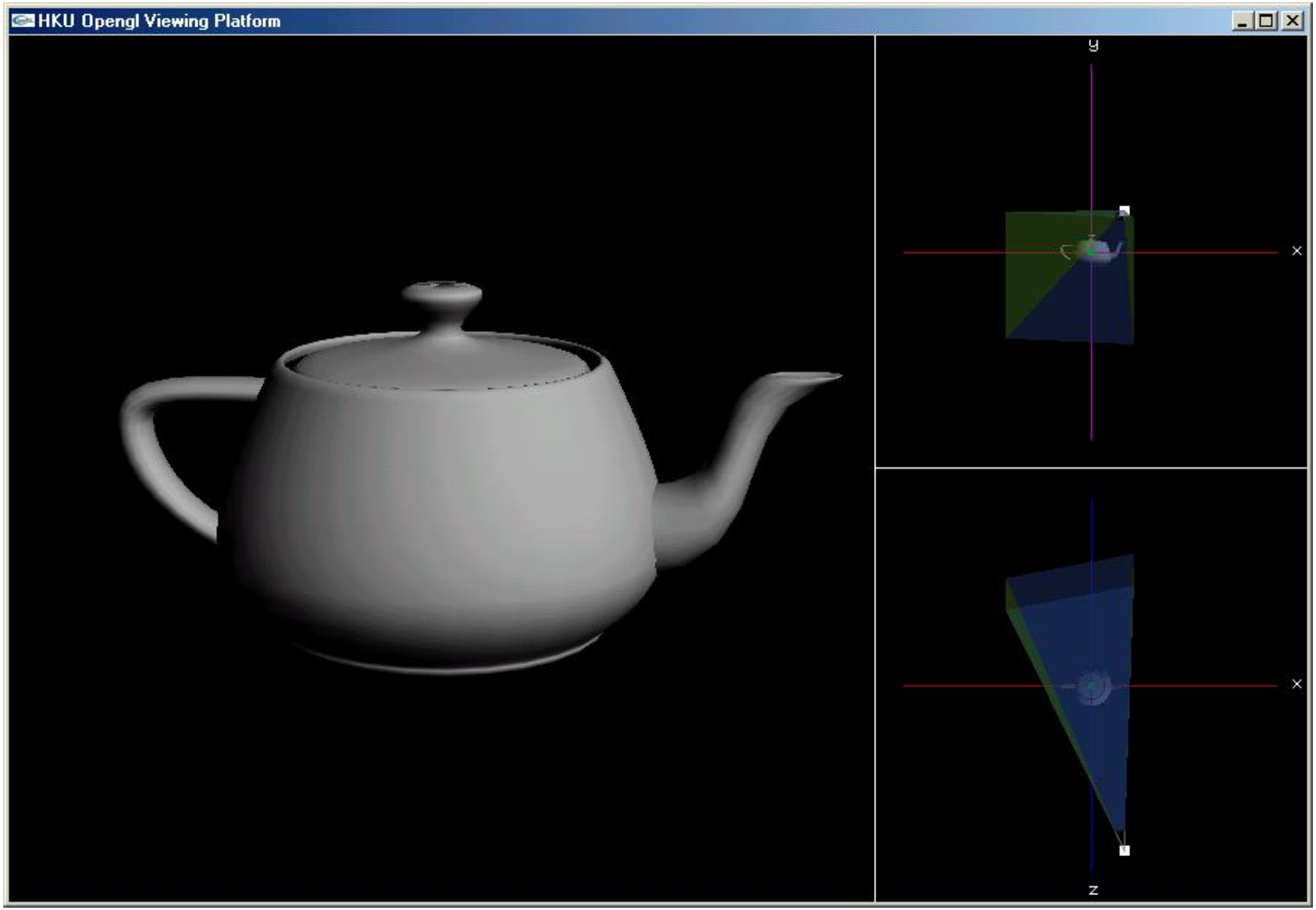
Ерөнхий

- point source-с сарнисан тусгал (Diffuse reflection)

$$I_d = \frac{R_d I_i \cos\theta}{a + bd + cd^2} = \frac{R_d I_i}{a + bd + cd^2}(\mathbf{s} \bullet \mathbf{n})$$

- directional source-с сарнисан тусгал

$$I_d = R_d I_i \cos\theta = R_d I_i (\mathbf{s} \bullet \mathbf{n})$$

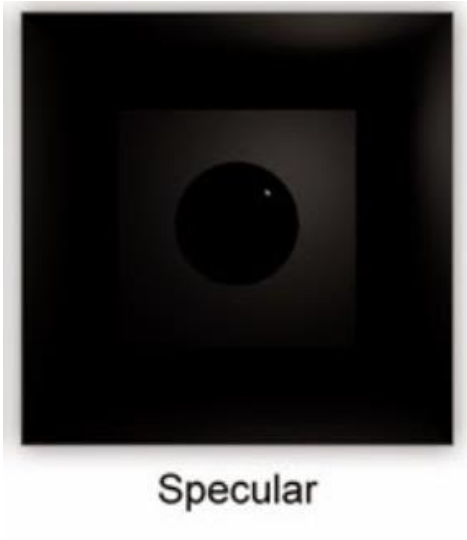


Pure diffuse reflection from directional source pointing (1,1,1)

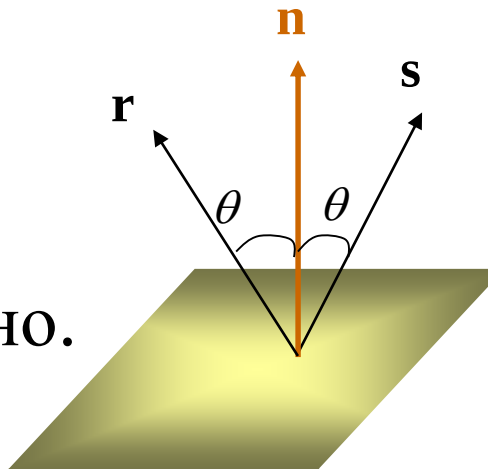
Specular Ойлт

Мөлгөр мөн гялалзсан гадаргууг харах юм бол гэрлийн ойлт нь гадаргууны тодорхой нэг цэг дээр илүү тод гялтганаж харагддагийг ажиглах боломжтой.

diffuse ойлтоос ялгаатай тал нь гэвэл specular ойлтын утга ажиглагчийн байрлалаас үргэлж хамааралтай байдаг.



Specular ойлт

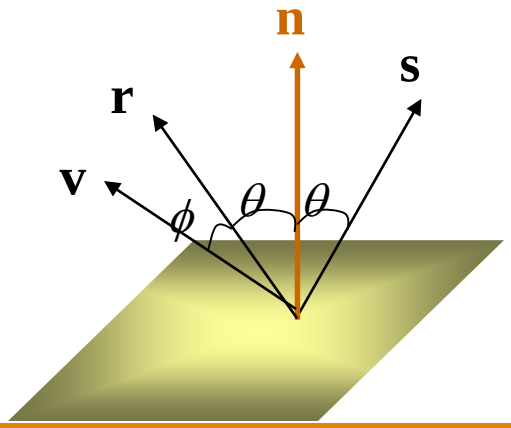


- Гялалзсан гадаргуу тусгалын чиглэлд (**r**) гэрлийг ойлгодог.
- **r** нь **n** ба **s** ийн тархсан хавтай дээр оршино.

  n-ийн хоорондох өнцөг нь n ба s-ийн хоорондох өнцөгтэй ижил байна,

  $0^o \le \theta \le 90^o$.
- Гадаргуугийн цэгийг **v** чиглэлд харах үед, ойсон гэрлийн хэмжээ

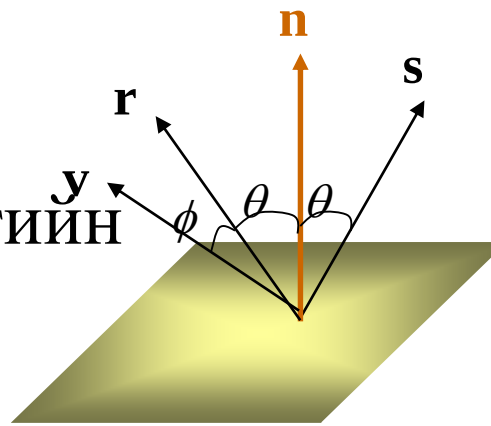
$$I_s = R_s I_i \cos^k \phi$$



- I_i , ойсон гэрлийн эрчим (intensity)
- R_s , ойлтын коэфициент,

$0 \le R_s \le 1$

$$I_s = R_s I_i \cos^k \phi$$



- ϕ нь харах чиглэл ба төгс тусгалын цацрагийн хоорондох өнцөг юм.
- Гялалзсан гадаргуу дээр **v** нь **r** бага зэрэг хазайхад туссан гэрэл маш бага үзэгдэнэ. Ө.х, dull surface дээр гэрэл илүү их үзэгдэнэ. Иймүзэгдлийг $\cos^k \phi$ томъёогоор тодорхойлох ба k нь материалын гялалзсан байдал. К их байх тусам гялалзах чадвар өндөр болно
- $\cos \phi = \mathbf{r} \bullet \mathbf{v}$
- attenuation –г diffuse reflection адил загварчилдаг.

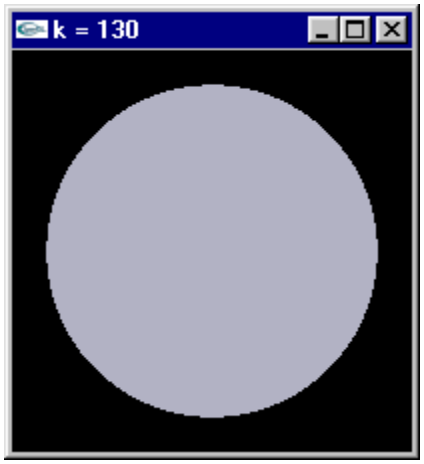
Ерөнхий

- point source-ийн Specular ойлт

$$I_s = \frac{R_s I_i \cos^k \phi}{a + bd + cd^2} = \frac{R_s I_i}{a + bd + cd^2} (\mathbf{r} \bullet \mathbf{v})^k$$

- directional source-ийн Specular ойлт

$$I_s = R_s I_i \cos^k \phi = R_s I_i (\mathbf{r} \bullet \mathbf{v})^k$$



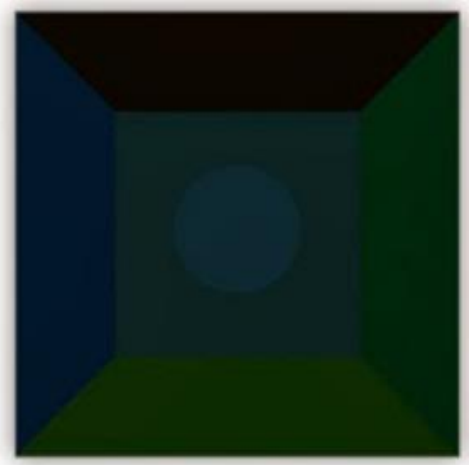
k нь 128-аас бага байна

Pure Specular Reflection

Ambient ойлт

Ambient ойлт бол гэрэл маш олон гадаргуу дээр тусан цааш ойж явсаар бий болох ерөнхий дунджилсан утга юм. Объектийн гадаргуугийн цэг дээрхи чиглэл бүрт ижилхэн intensity утгатай байдаг гэсэн үг юм.

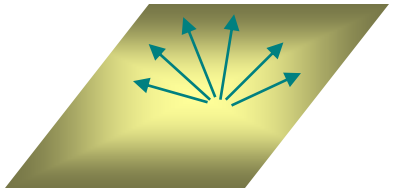
Ambient гэрэл нь объектийн гадаргуун цэг бүр дээр ижилхэн тусаж байдаг тул чиглэл байрлал гэсэн ойлголт байдаггүй, харин зөвхөн өнгө болон intensity гэсэн утга байна.



ambient light ойлт (background), I_a

- Ambient light (I_{Am}) нь гадаргуу ба гэрэл хоорондын олон тусгалаас шалгаалж жигд гэрэлтдэг.

$$I_a = R_a I_{Am}$$

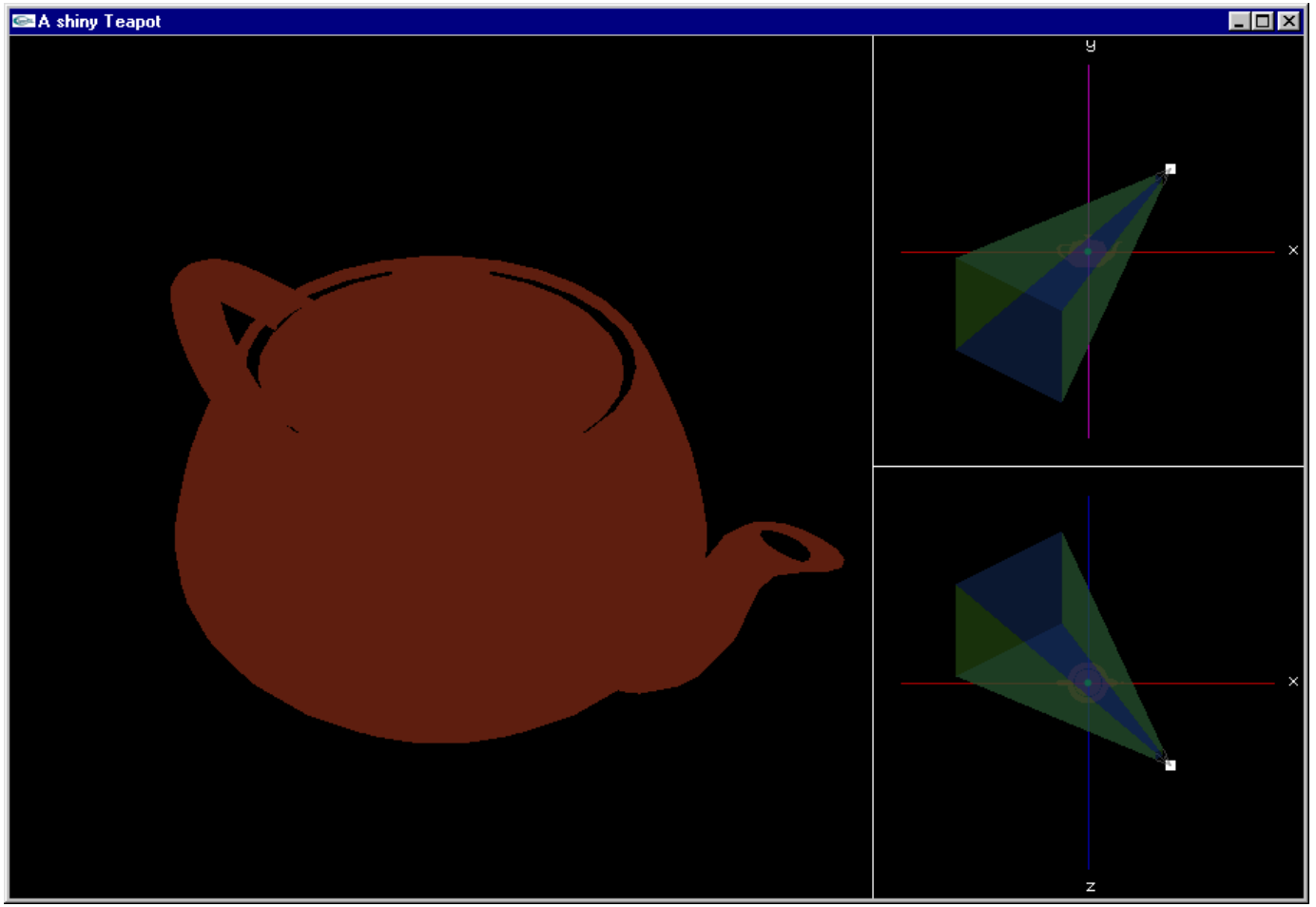


- R_a ambient reflection coefficient,

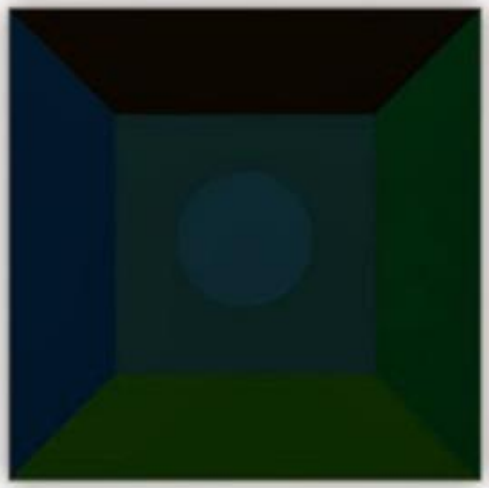
  $0 \leq R_a \leq 1$

- I_a гадаргуугийн нормал ба харах өнцөг, гэрлийн эх үүсгүүрийн байрлал

```
float  globalAmbient[] = { 0.9, 0.9, 0.9, 1.};
glLightModelfv( GL_LIGHT_MODEL_AMBIENT,  globalAmbient );
```



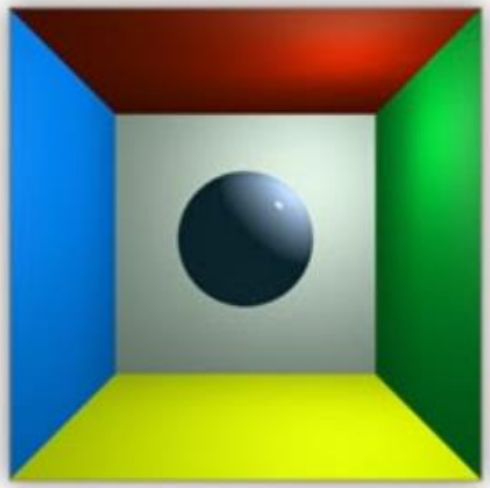
Pure Ambient



Ambient

Diffuse

Specular



Ambient + Diffuse
+ Specular

OpenGL Хэрэгжүүлэлт (Lightinig)

Гэрлийн эх үүсвэр бий болгох

Гэрлийн эх үүсвэр нь өнгө, байрлал, чиглэл гэсэн параметрүүдтэй байна.

Гэрлийн эх үүсгэврийн дугаар, шинж чанар, авах утга

```
void glLight{if} (GLenum light, GLenum pname, TYPE param);

void glLight{if}v (GLenum light, GLenum pname, TYPE *param);
```

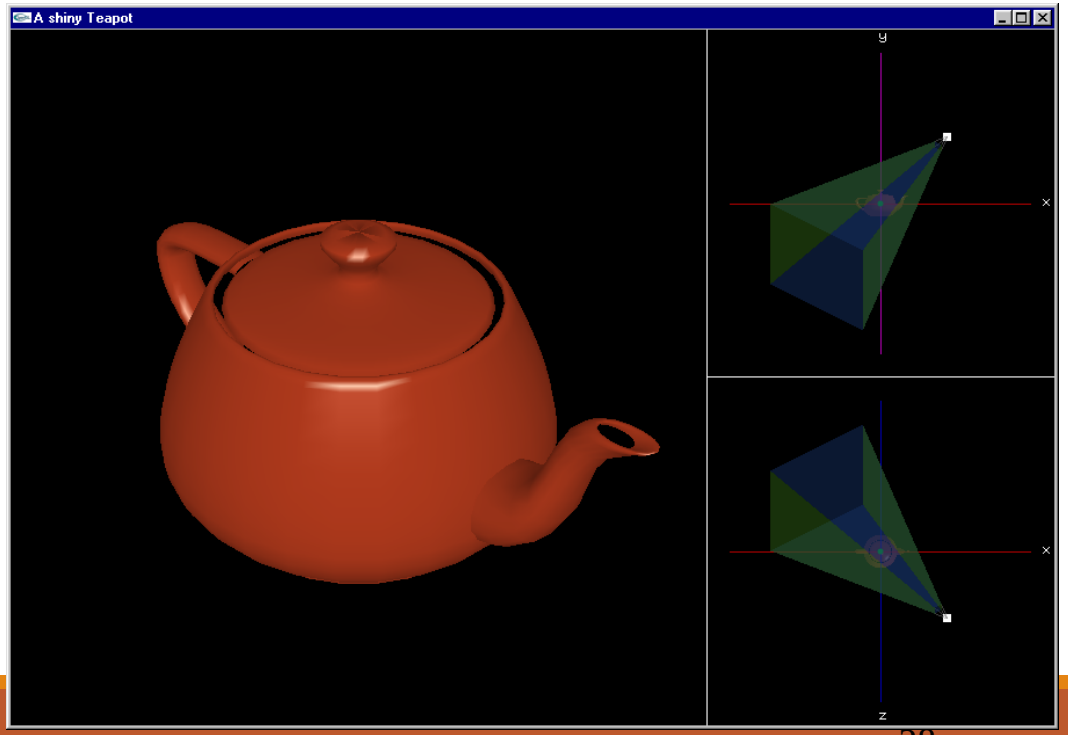
Light параметрийн утга: GL_LIGHT0, GL_LIGHT1, ..GL_LIGHT7)

Pname параметрийн утга

параметрүүдийн нэр	Үндсэн өгөгдмөл утга	Тайлбар
GL_AMBIENT	(0.0,0.0,0.0,1.0)	Intensity of background light
GL_DIFFUSE	(1.0,1.0,1.0,1.0) or (0.0,0.0,0.0,1.0)	Intensity of diffuse light - сарнисан гэрлийн эрчимжилт (0 үүсгэвэр- цагаан гэрэл, бусад хар)
GL_SPECULAR	(1.0,1.0,1.0,1.0) or (0.0,0.0,0.0,1.0)	Intensity of mirror light - толин буюу гялтгар гэрлийн эрчимжилт (0 үүсгэвэр- цагаан гэрэл, бусад хар)
GL_POSITION	(0.0,0.0,1.0,0.0)	the position of the light source (x,y,z,w)
GL_SPOT_DIRECTION	(0.0,0.0,-1.0)	spotlight direction-Гэрэлтүүлэх чиглэл (x,y,z)
GL_SPOT_EXPONENT	0.0	гэрлийн цацрагийн агууламж
GL_SPOT_CUTOFF	180.0	гэрлийн цацрагийн өнцгийн өргөн
GL_CONSTANT_ATTENUATION	1.0	гэрэлтүүлгийг сулруудах фактор
GL_LINEAR_ATTENUATION	0.0	шугаман сулруулагч
GL_QUADRATIC_ATTENUATION	0.0	квадратан сулруулагч

- Өнгөт зураг гарган авахын тулд shading тооцоолол улаан, ногоон, цэнхэр компонент тус бүрд нь гүйцэтгэнэ.

```
GLfloat specular[] = {1., 1., 1., 1.};
GLfloat ambient[] = { .41, .135, .067, 1.};
GLfloat diffuse[] = { .41, .135, .067, 1.};
glMaterialf( GL_FRONT,  GL_SHININESS, 125.);
```

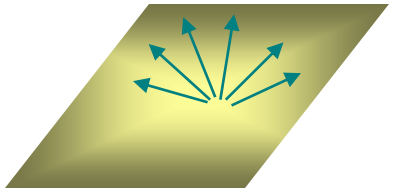


Shiny Teapot

Орчны гэрэл-ambient light тусгал (background), I_a

- Ambient light (I_{Am}) нь гадаргуу ба гэрэл хоорондын олон тусгалаас шалгаалж жигд гэрэлтдэг.

$$I_a = R_a I_{Am}$$

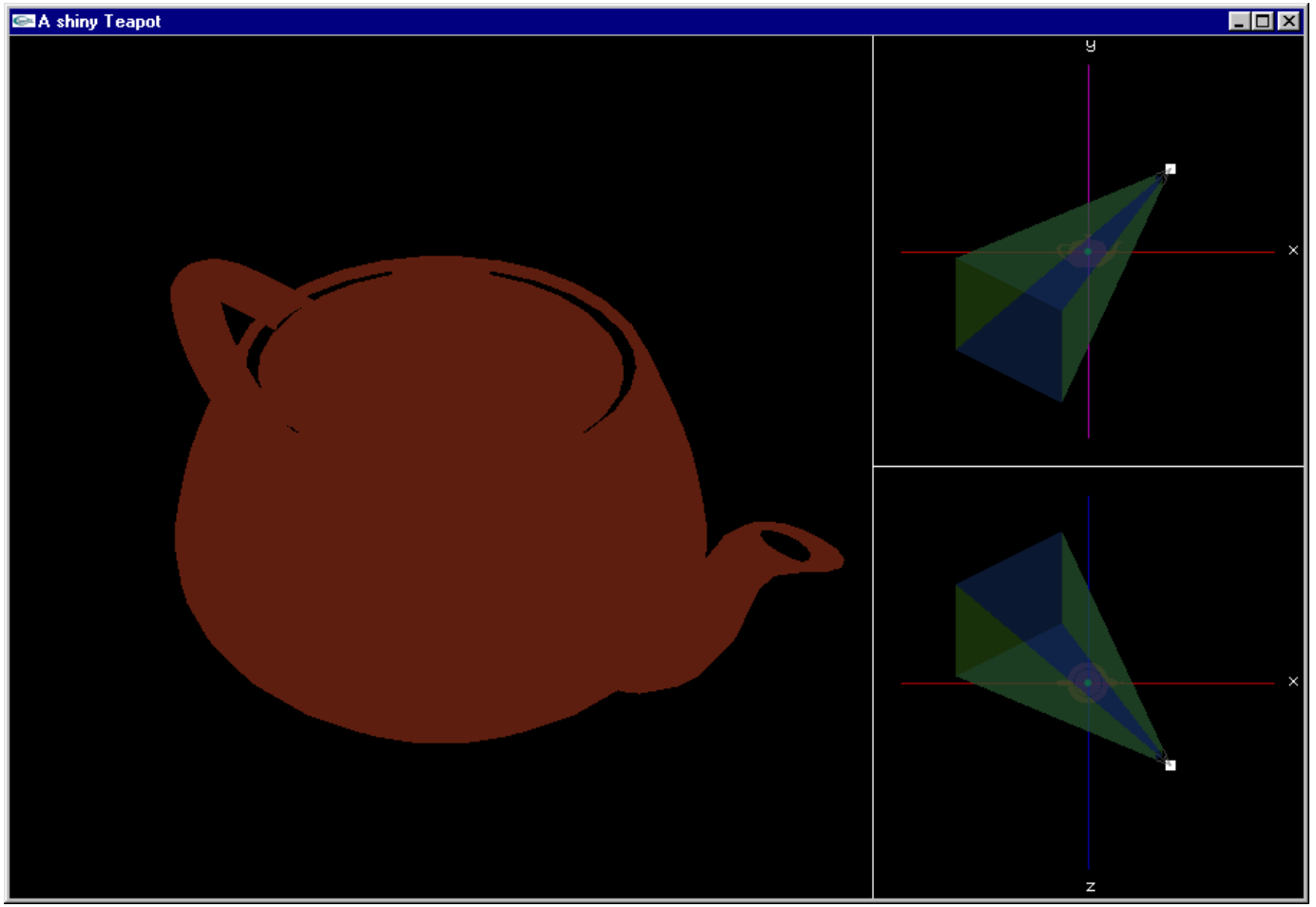


- R_a ambient reflection coefficient,

  $0 \leq R_a \leq 1$

- I_a гадаргуугийн нормал ба харах өнцөг, гэрлийн эх үүсгүүрийн байрлал

```
float  globalAmbient[] = { 0.9, 0.9, 0.9, 1.};
glLightModelfv( GL_LIGHT_MODEL_AMBIENT,  globalAmbient );
```



Pure Ambient

*OpenGL дэх гэрэлтүүлэг (*Lighting)

- Enable/Disable lighting

```
glEnable( GL_LIGHTING);    //colors are ignored
glDisable( GL_LIGHTING);
```

- shading model тодорхойлох

  flat shading эсвэл smooth (Gouraud) shading

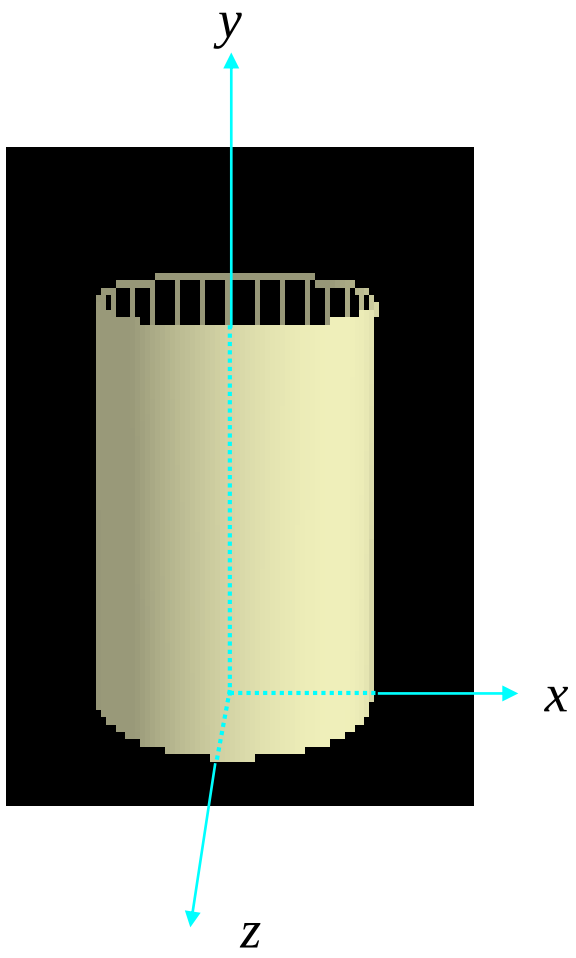
```
glShadeModel( GL_FLAT);
glShadeModel( GL_SMOOTH);
```

- Орой тус бүрийн хувьд нормалийг тодорхойлох. Жнь ., координатын эх дээр бөмбөрцөг зурах.

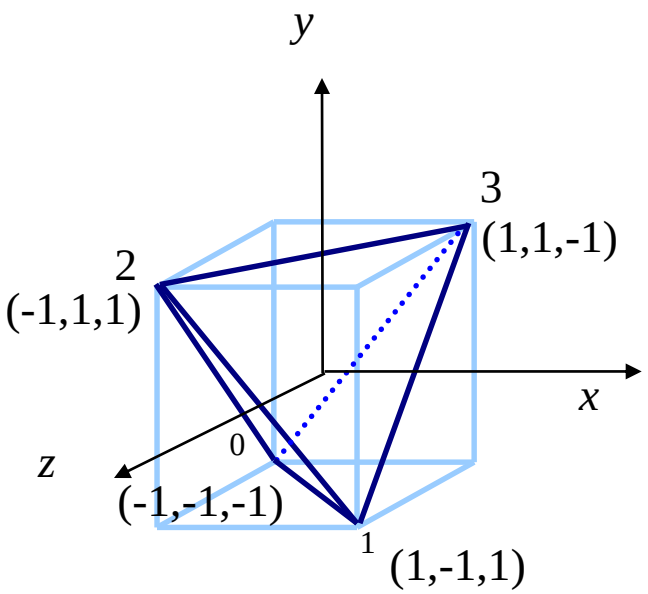
```
glNormal3f( x, y, z);
    glVertex3f( r*x, r*y, r*z);
```

- Нормал нь өөр нэг нэгж нормалийг тодорхойлох хүртэл хүчин төгөлдөр байна.

```
//Draw a cylinder
glBegin( GL_QUAD_STRIP);
        t = 0.;
       dt = (360. / nslice) * 3.1416 / 180.;
        for  (j = 0; j <= nslice; ++j) {
             glNormal3f( cos( t), 0., sin( t));

             glVertex3f( cos( t), 0., sin( t));

        glVertex3f( cos( t), 2., sin( t));

              t = t + dt;
        }
glEnd();
```



```
void triangle( float v[][3],  int a,  int b,  int c) {
      glBegin(GL_TRIANGLES);
      glVertex3fv( v[a]);
            glVertex3fv( v[b]);
            glVertex3fv( v[c]);
     glEnd();
     }

void  tetrahedron()   {
           float v[][3] = {   { -1., -1., -1.},
           { 1., -1., 1.}, { -1., 1., 1.},
                          { 1., 1., -1.}   };
     glNormal3f( -1., -1., 1.);
     triangle( v, 0, 1, 2);
     glNormal3f( 1., 1., 1.);
     triangle( v, 1, 3, 2);
     glNormal3f( -1., 1., -1.);
     triangle( v, 0, 2, 3);
     glNormal3f( 1., -1., -1.);
     triangle( v, 0, 3, 1);
}
```



- нэгж вектор бүрийн нормалыг тооцоолох бэрхшээлтэй асуудлаас зайлхийхийн тулд векторыг нормчлох систем бий бөгөөд дараах үйдлийг *init()* нэмнэ.

```
glEnable( GL_NORMALIZE);
```

- Гэрлийн үүсгүүрийн өнгөний компонентууд (r, g, b, a)

```
GLfloat ambient0[] = { 0.1, 0.1, 0.1, 1.};      //Grey
GLfloat diffuse0[] = { 1., 1., 1., 1.};         //White
GLfloat specular0[] = { 1., 1., 1., 1.};        //White

glLightfv( GL_LIGHT0, GL_AMBIENT, ambient0);
glLightfv( GL_LIGHT0, GL_DIFFUSE, diffuse0);
glLightfv( GL_LIGHT0, GL_SPECULAR, specular0);
```

- Бодит амьдралд гэрлийн үүсгүүрийн diffuse болон specular компонентууд нь ижил байдаг.

- ambient компонент нь орноос хамаарна. component depends on the environment. Хэрэв хүрээлэн буй орчин нь олон улаан объектоос бүрддэг бол ambient component нь бага зэрэг улаан өнгөтэй байх хандлагатай байдаг.

```
GLfloat a[] = { 0.1, 0., 0., 1.}; //Very pale red
GLfloat d[] = { 1., 1., 0., 1.};        //Yellow
GLfloat s[] = { 1., 1., 0., 1.};   //Yellow

glLightfv( GL_LIGHT0, GL_AMBIENT, a);
glLightfv( GL_LIGHT0, GL_DIFFUSE, d);
glLightfv( GL_LIGHT0, GL_SPECULAR, s);
```

- *Opengl* нь global ambient source-г хар саарал (.2, .2, .2) тодорхойлсон байдаг. Эдгээрийг (r, g, b) руу өөрчлөхийн тулд

```
float globalAmbient[] = { r, g, b, 1.};
glLightModelfv( GL_LIGHT_MODEL_AMBIENT, globalAmbient );
```

дуудна

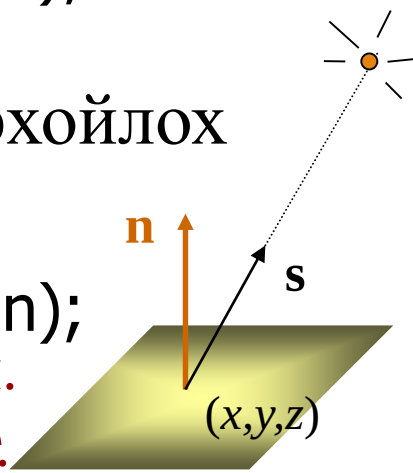
- point light source-ийн байршилыг тодорхойлох

```
GLfloat light_position[] = { 2., 4., 6., 1.};
glLightfv( GL_LIGHT0, GL_POSITION, light_position);
```

- directional light source-ийн чиглэлийг тодорхойлох

```
GLfloat light_position[] = { 1., 1., 1., 0.};
glLightfv( GL_LIGHT1, GL_POSITION, light_position);
```

s нь объектоос эх үүсгүүр рүү чиглэсэн чиглэл юм.
Энэ нь гэрлийн цацрагийн эсрэг чиглэлтэй байдаг.

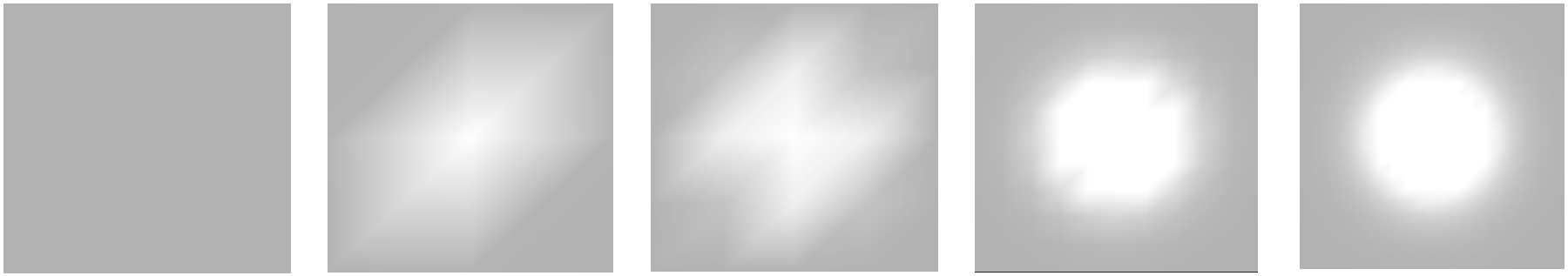


- Хамгийн багадаа 8 гэрлийн үүсгүүр байдаг
  GL_LIGHT0, GL_LIGHT1, …, GL_LIGHT7
- Тодорхой гэрлийн үүсгүүрийг асаах, унтраах (right before/after the drawing)
  glEnable(GL_LIGHT0);
  glDisable(GL_LIGHT0);
- specular reflection тооцоолохд true viewing angle ашиглахыг хүсвэл.
  glLightModeli(GL_LIGHT_MODEL_LOCAL_VIEWER, GL_TRUE);

- mesh sizes (Important) өөрчлөлт

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

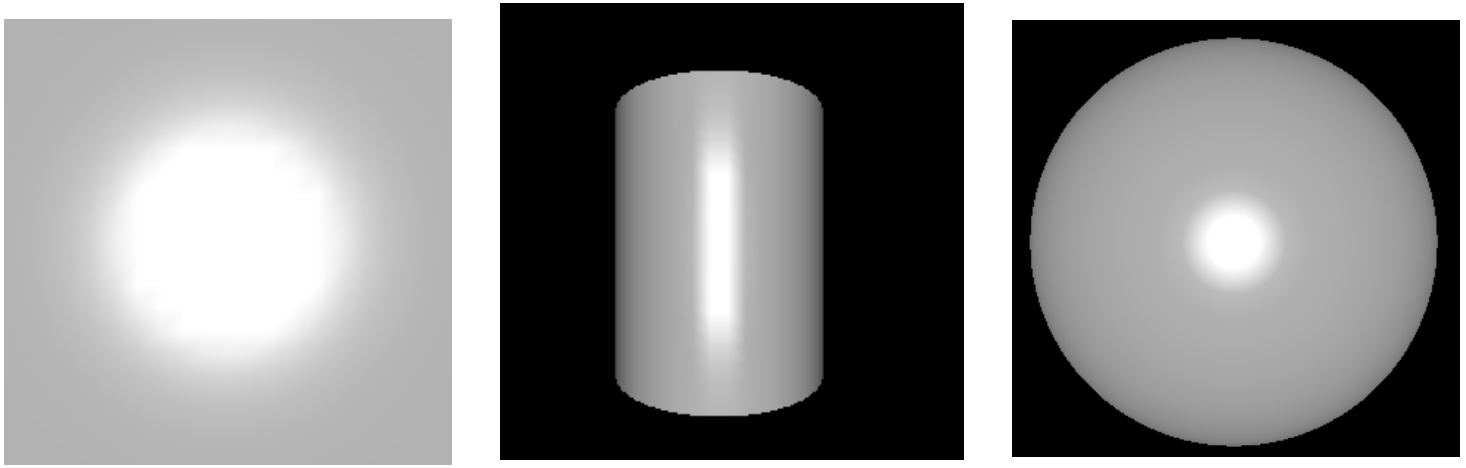
```
GLfloat ambient0[] = { 0., 0., 0., 1.};
GLfloat diffuse0[] = { .7, .7, .7, 1.};
GLfloat specular0[] = { 1., 1., 1., 1.};
position[] = { 0., 0., 1., 0.};     //Light direction
//Viewing position (0, 0, 8)
```



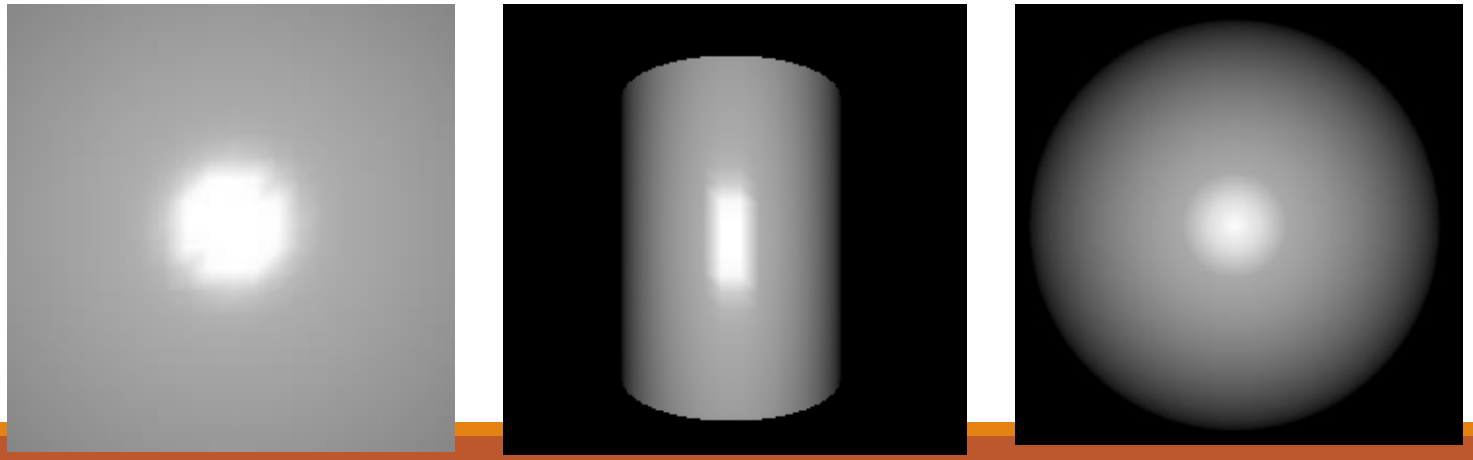
1 × 1 2 × 2 4 × 4 10 × 10 20 × 20

- 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

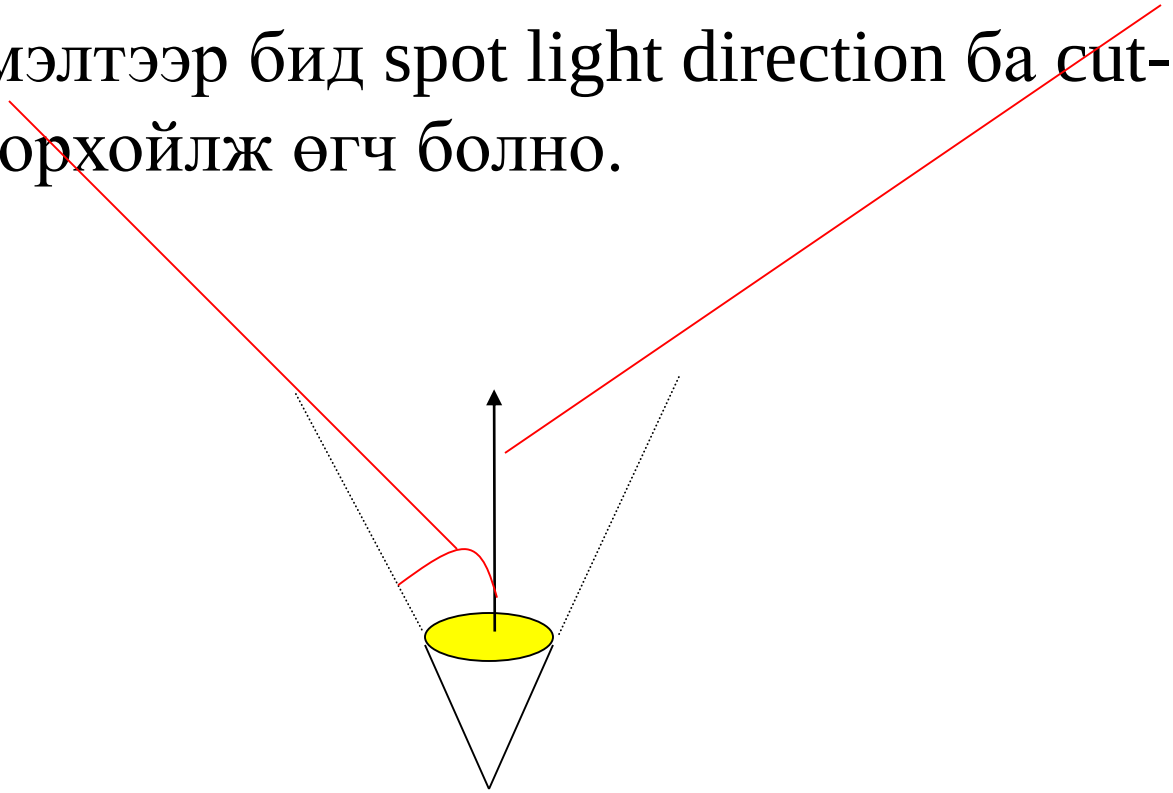
$$A_f = \frac{1}{a + bd + cd^2}$$

```
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)

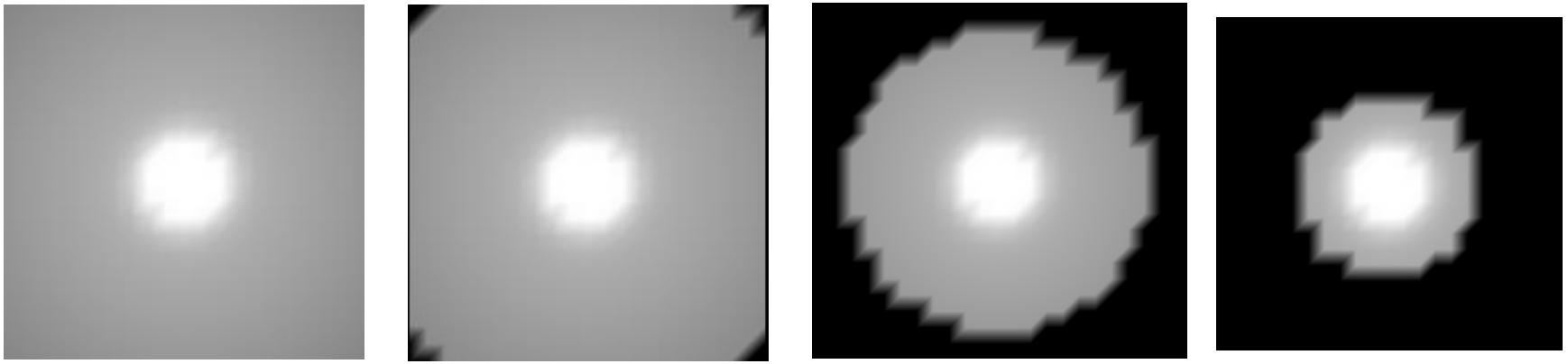
- higher attenuation, b ба *c хамгийн их утгыг тогтооно жнь*, *1*, *1.5*, *2* эсвэл *3*.

- 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 тодорхойлж өгч болно.



- Цагаан ханан дээрх cutoff angle өөрчлөлт

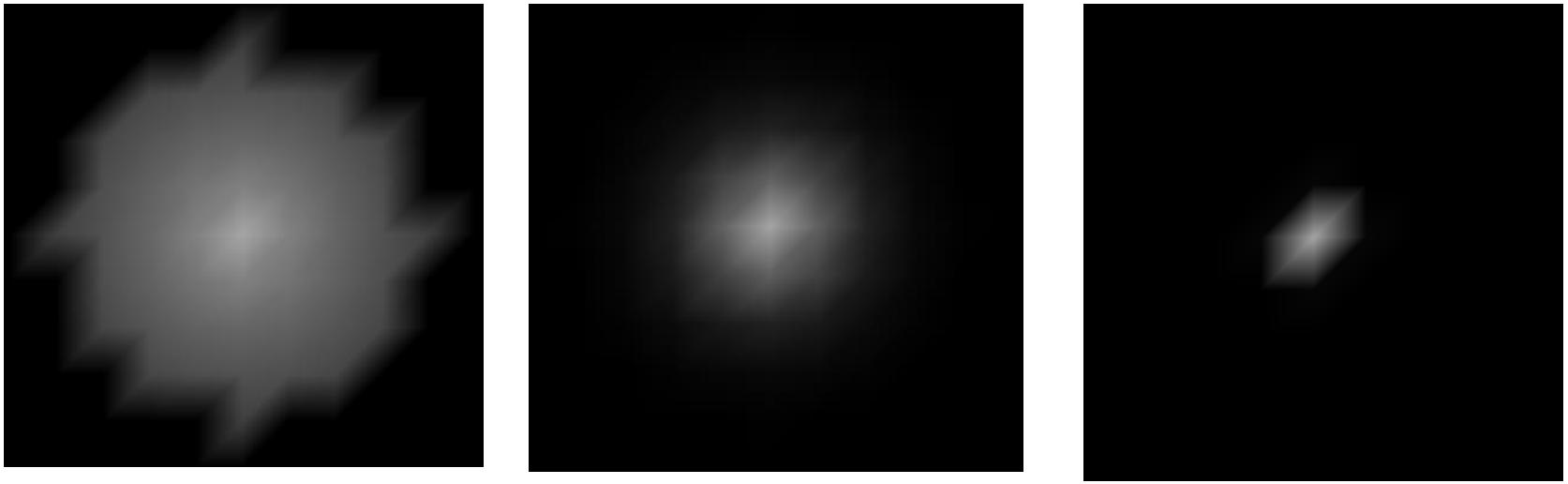
```
GLfloat light_direction[] = {0, 0, -1};   //From the light
glLightfv( GL_LIGHT2, GL_SPOT_DIRECTION, light_direction);

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 spot exponent 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.);
```

The direction of the white light source is (1,1,1)

Материалын тусгал

- The red, green and blue components of the specular reflectance are often the same so that the original color of a light source is reflected. (In the preceding picture of the teapot, the highlight is white, the same as the light source.)
- For highly shiny material, the specular reflectance is close to 1 and the shininess can be as high as 100 or more.
- The specular reflectance and shininess of a dull surface are close to 0.
- A material property, eg, reflectance, remains in effect until a new value is specified.

- Материалын өнгө diffuse reflectance-с хамаарна.
- The diffuse reflection alone provides most information about the curvature and the depth of an object
- The ambient reflectance of a material is often the same as the diffuse reflectance
- The highlights produced in specular reflections are the blurred images of the light sources.

hkglu viewing platform (16/3/2003)

Pure Ambient

hkglu viewing platform (16/3/2003)

Pure Diffuse

hkglu viewing platform (16/3/2003)

Pure Specular

hkglu viewing platform (16/3/2003)

Overall

- The pale yellowish pink material of Lok Yau Hall

```
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.);
```

- The light source is a directional white light coming from (1, 1, 1)

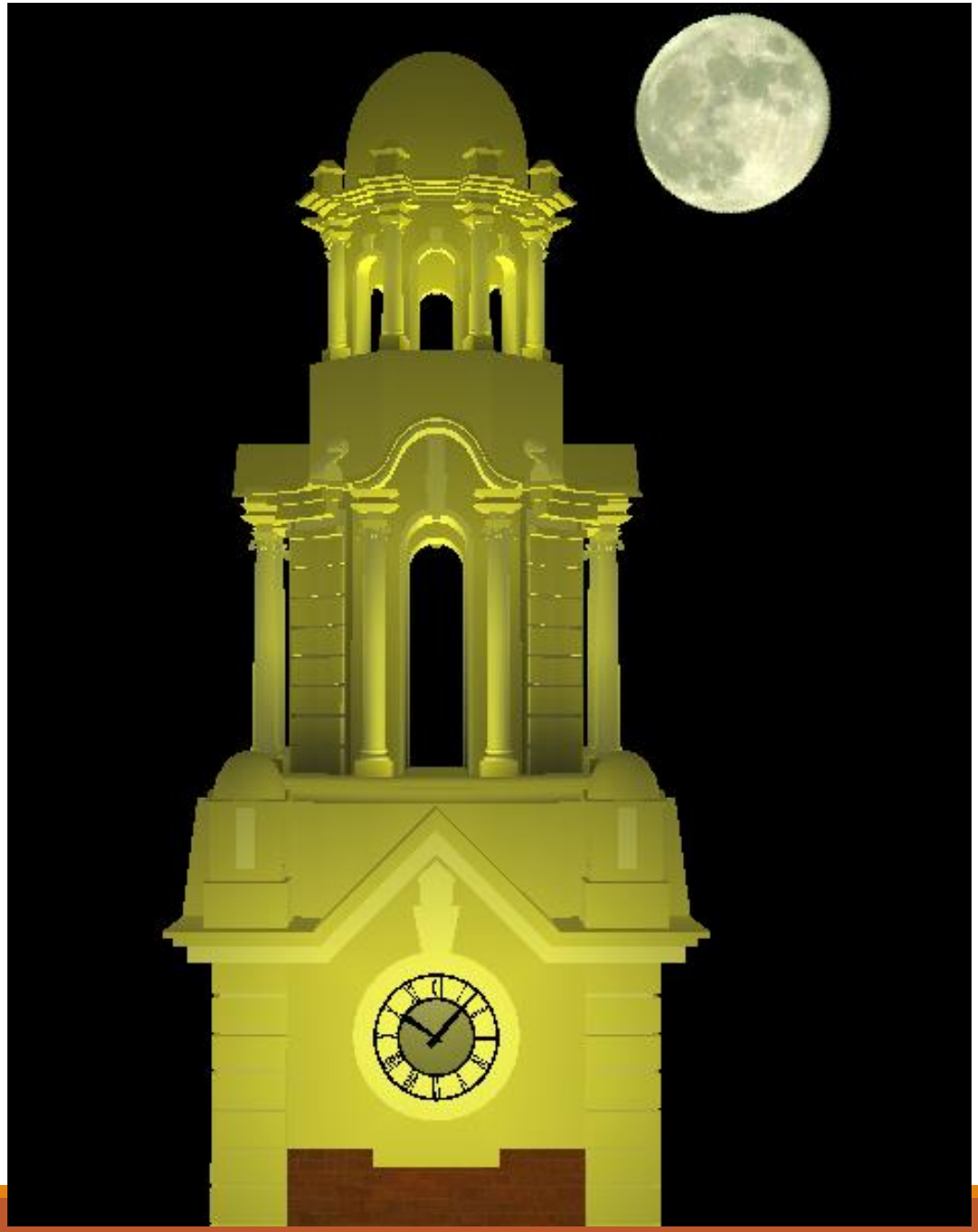
Emission

- To make a material appear to be glowing, eg, the moon in the picture next page.

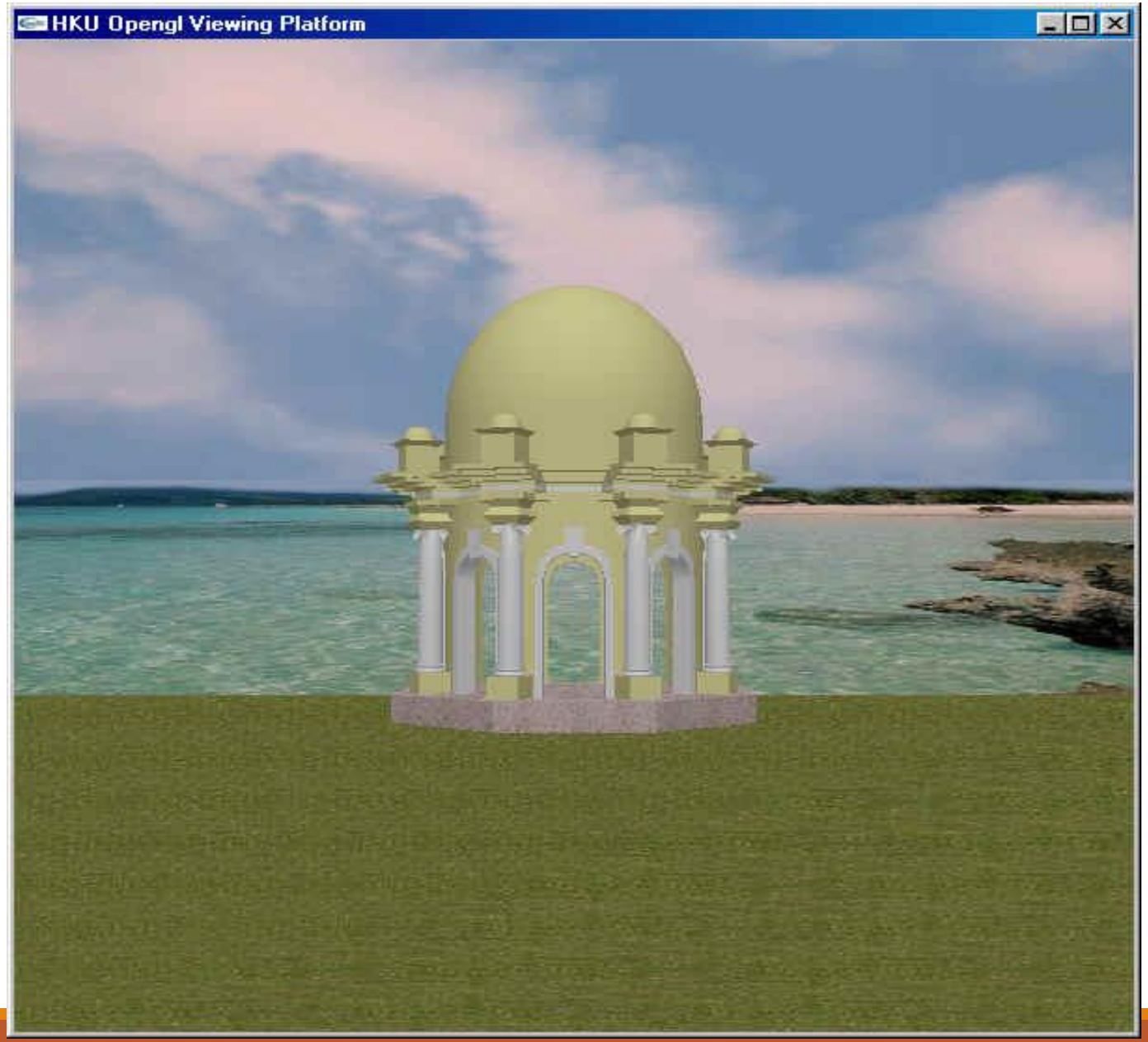
```
GLfloat emission[] = { r, g, b, a};
    glMaterialfv( GL_FRONT, GL_EMISSION, emission);
```

- Turn off the emission: assign 0 to the r-, g-, b-components.
- Note that an emission object is NOT a light source. Emission is just a property of material.

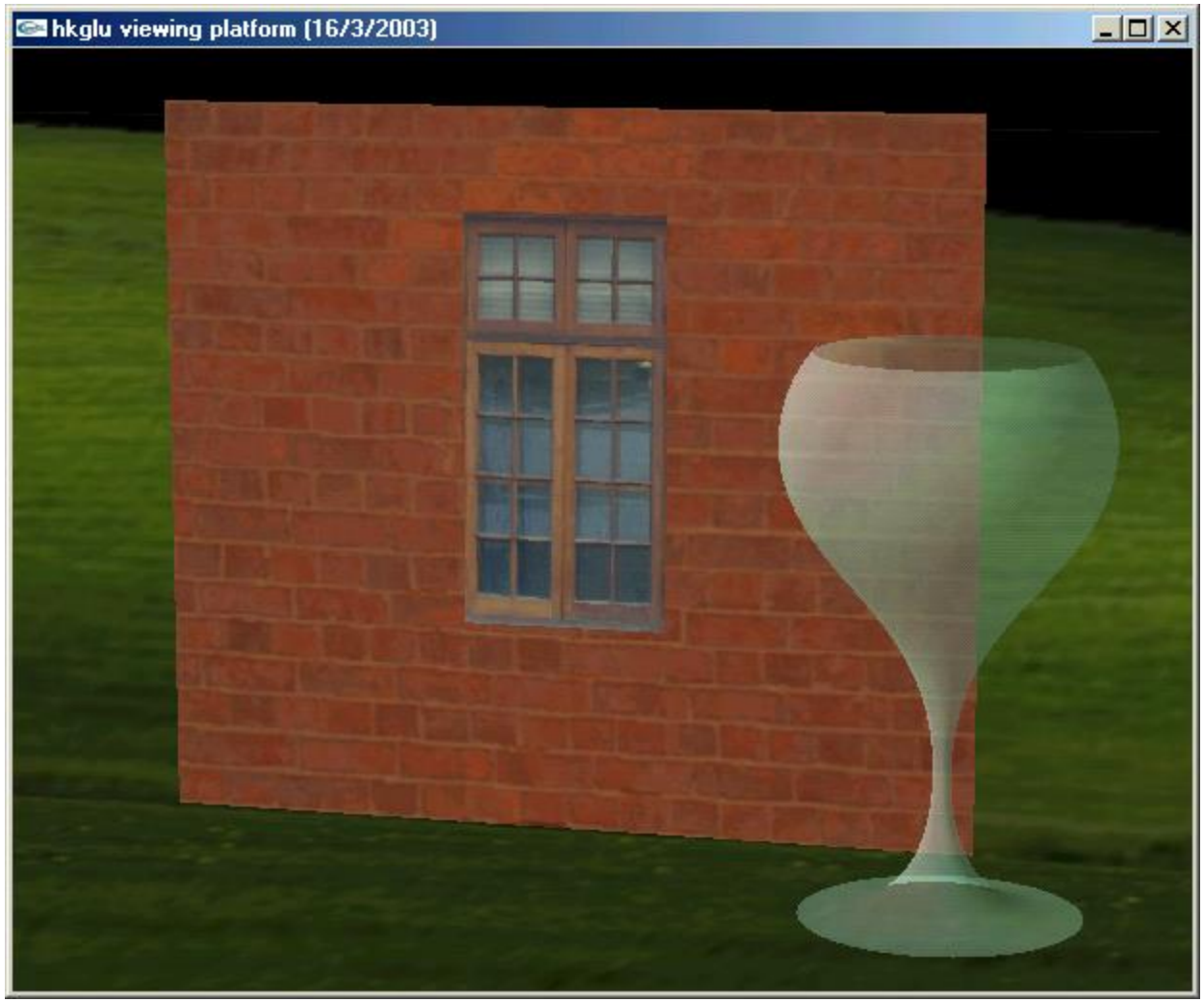
```
GLfloat emission[] = { 1., 1., .75, 1.};
glMaterialfv( GL_FRONT, GL_EMISSION, emission);

draw_moon();

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



A pictures rendered with 5 kinds of material, two textures, 16 spot lights. The moon is an emission object.


HKU Opengl Viewing Platform

Transparent Objects



Transparent

Transparent Objects

1. Define the blending function and turn on blending

```
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.

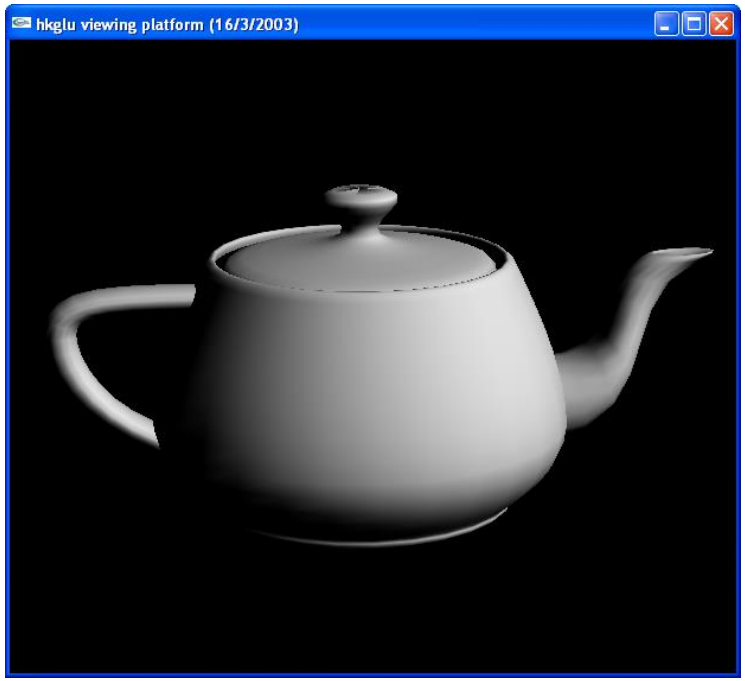
- All *OpenGL* parameters, including the lightings, have default values. A parameter's value is in effect until it is reassigning to a new value. Eg, the default `GL_LIGHT0` gives out white light and is positioned at the viewing point.
- The exact effect of lighting is hard to predict. Experiments and experience are crucial in tuning lighting parameters.
- The lighting support provided in *OpenGL* on a PC is minimal.
- On my own notebook PC, specular reflection on transparent objects seems mistakenly affected by the opacity specified in defuse reflection.

Shadows

Shadow

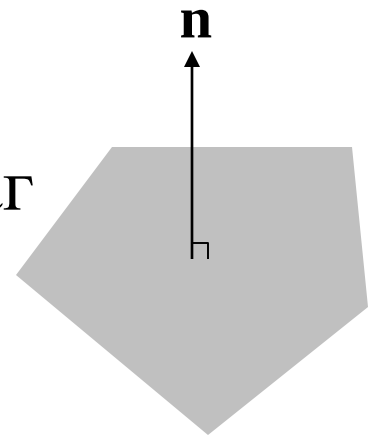
Shading

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.

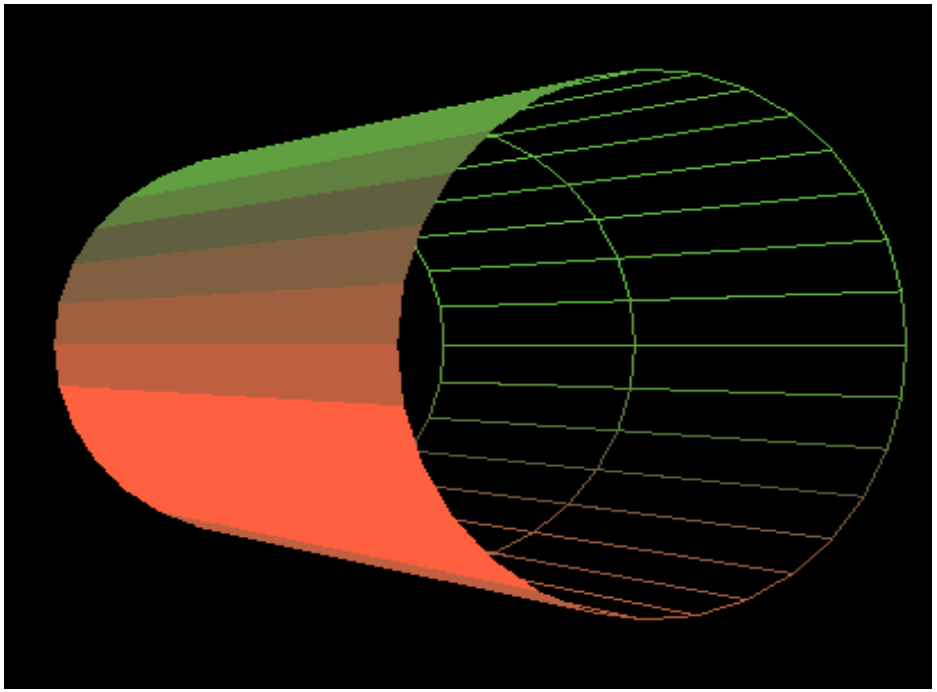


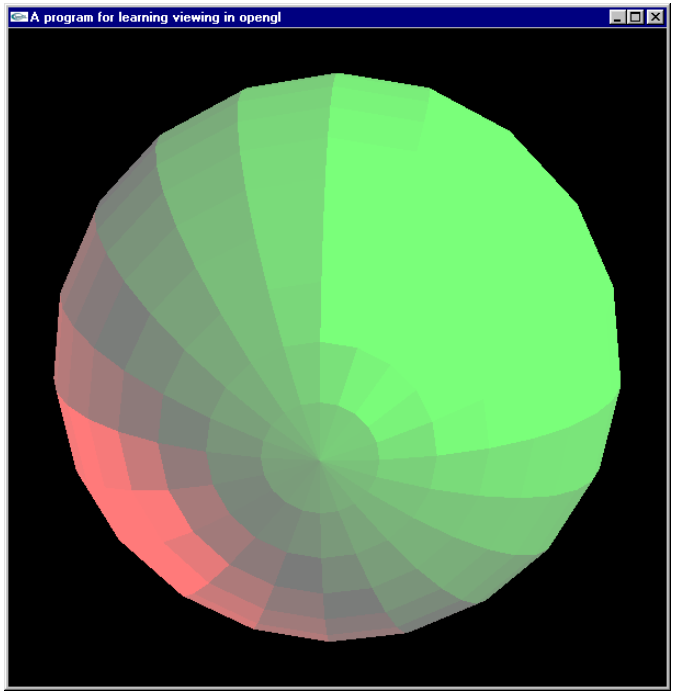
Constant (Flat) Shading

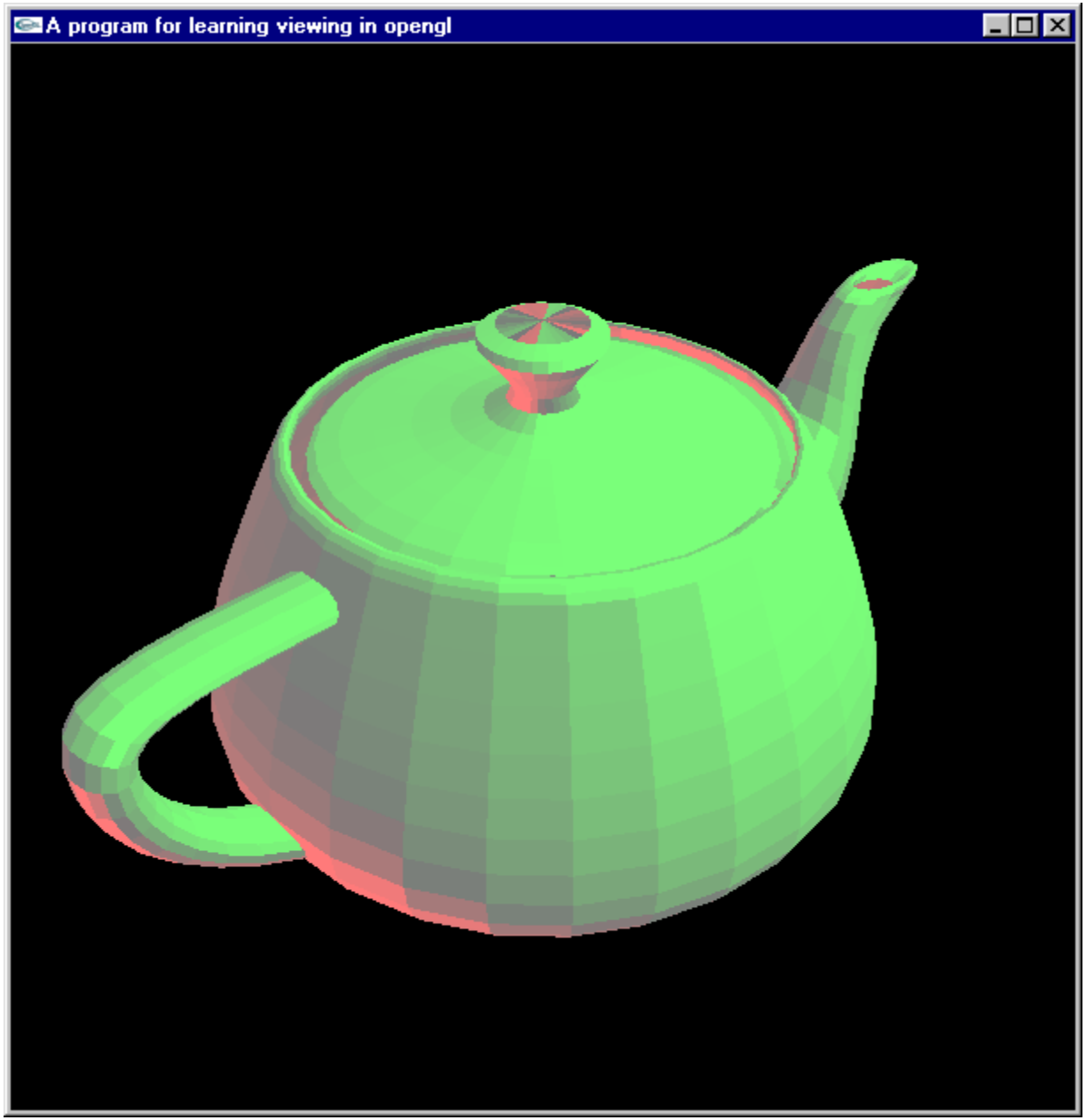
- Гадаргуугийн сүүдэр нь гадаргууд туссан ambient, diffuse ба specular гэрлийн хэмжээ
- constant shading хувьд тусгалын тооцоог зөвхөн олон өнцөгт дээр жишээлбэл эхний оройн цэг дээр нэг удаа гүйцэтгэнэ. Үүссэн сүүдэр нь олон өнцөгтийг бүрхсэн бүх цэгүүдийн хувьд оногдоно.
- Тооцоололд гадаргуугийн нормал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



- A *Mach band* is a stripe found along the edge of two polygons of different shades. Mach bands often occur in constant shading.
- Our eyes are particularly sensitive to the discrete change of shade





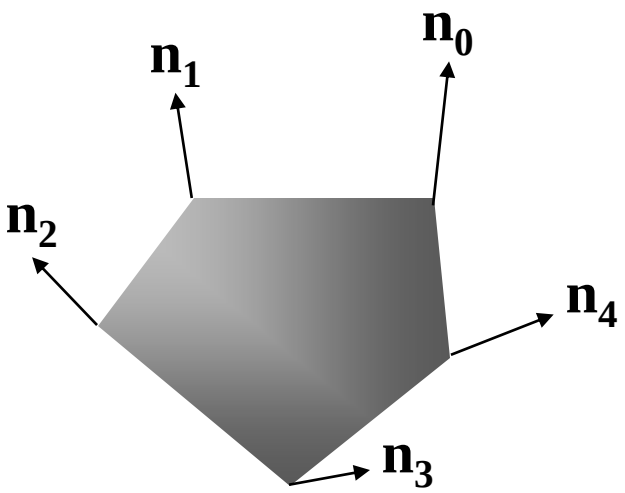

A program for learning viewing in opengl

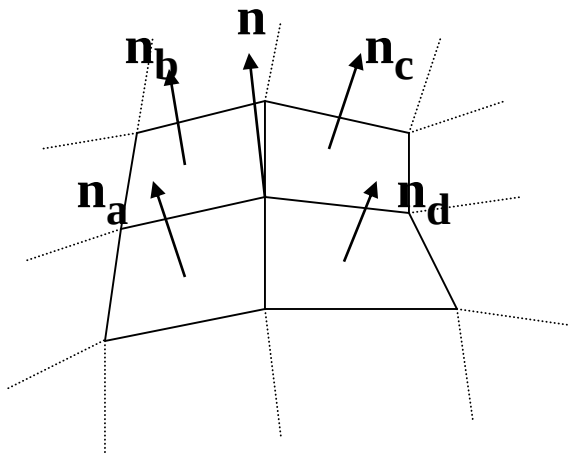
Gouraud shading

- Shading олон өнцөгтийн орой бүрд тооцоолж авч үзнэ.

- Орой бүрт олон өнцөгтийн нормал буюу зэргэлдээ олон өнцөгтийн дундаж нормал хувиарлагдана.

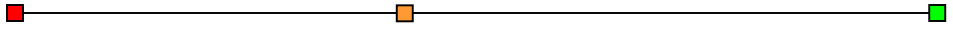
- Орой дахь цэгийн shade нь орой бүрт хувиарлагдсан нормалыг ашиглана.





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

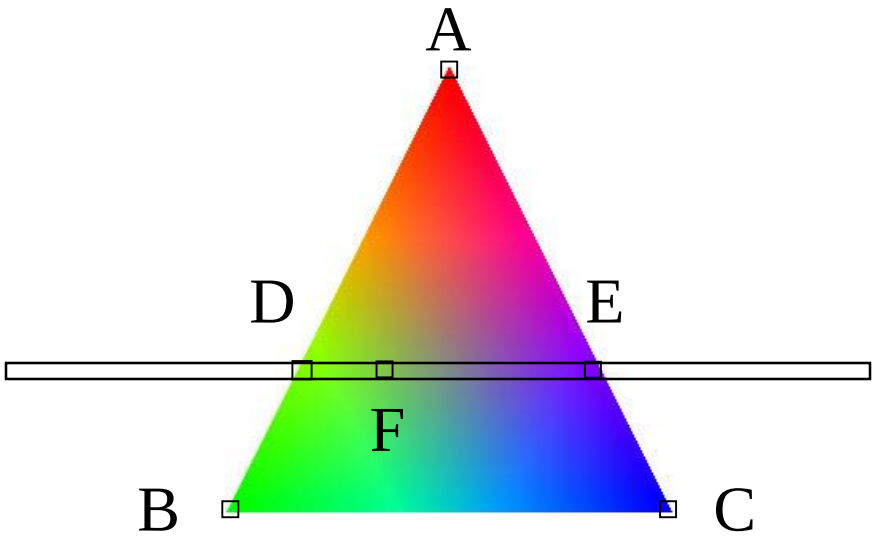
- Цэг хоорондын өнг C дараах байдлаар тодорхойлогдоно



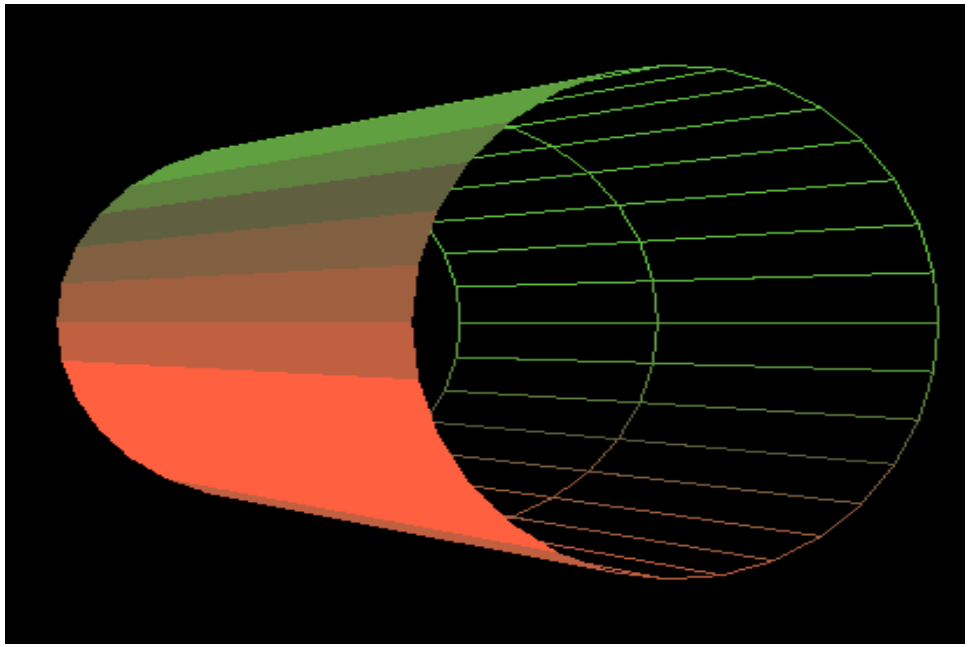
$$\alpha = \frac{\text{distance of AC}}{\text{distance of AB}}$$

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

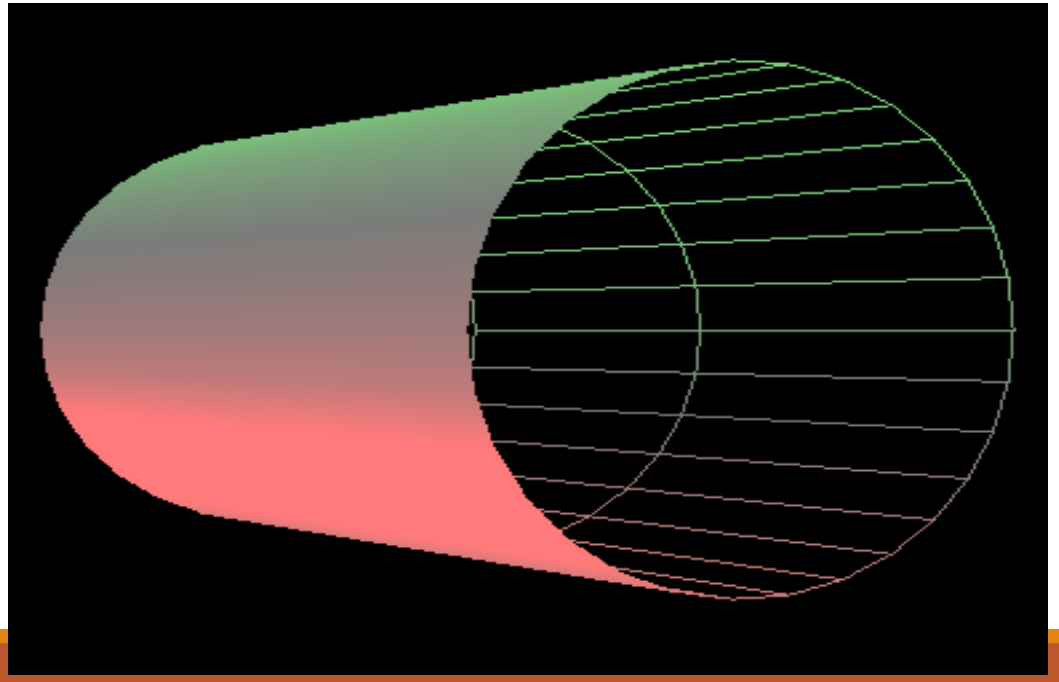
Олон өнцөгтийн пикселийн өнгийг тодорхойлох



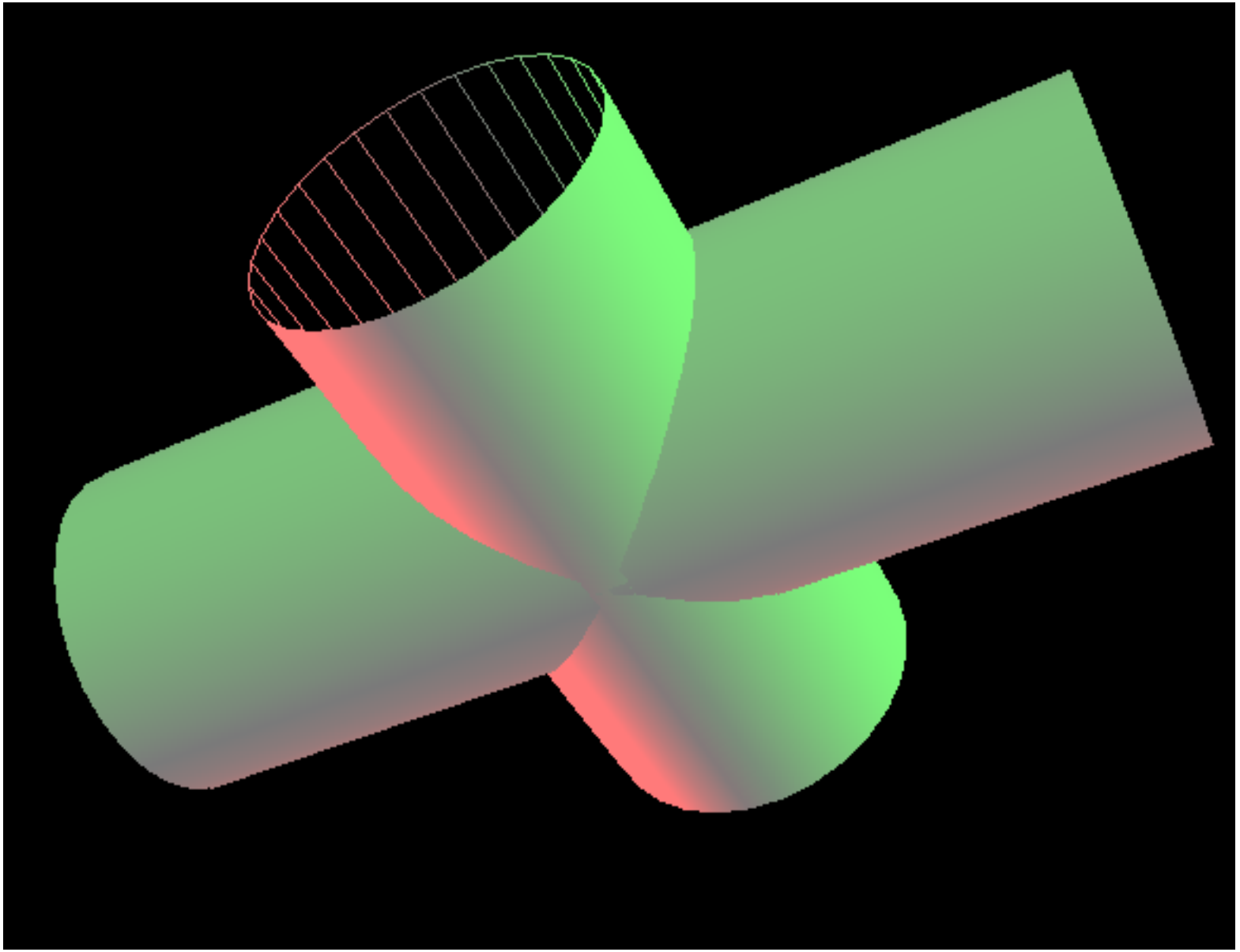
- D нь A ба B шулуун дээрх өнгөөр тодорхойлогдоно
- E нь A ба C шулууны хоорондох өнгөөр
- F нь D ба E шулууны хоорондох өнгөөр тодорхойлогдоно

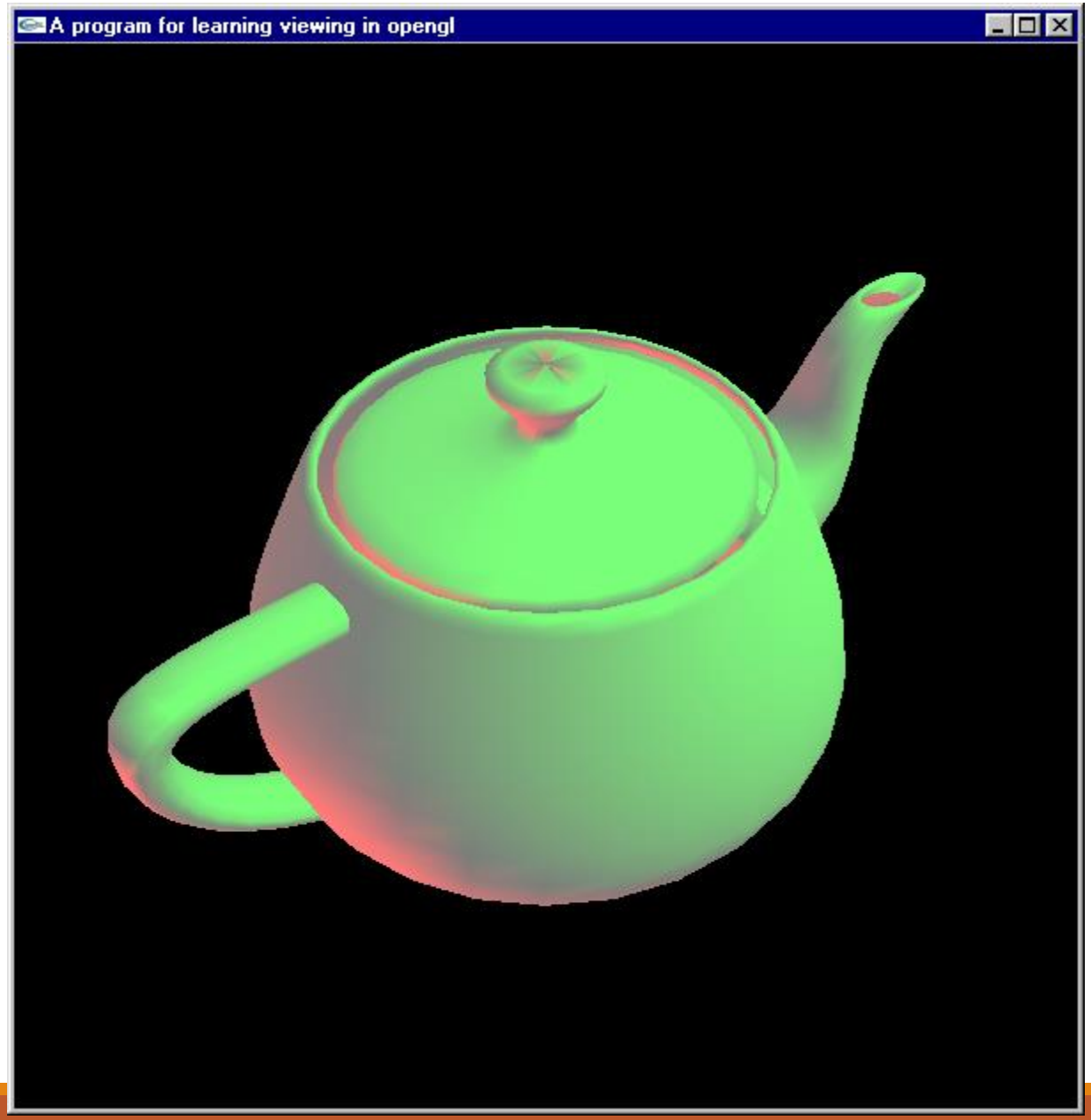


Constant Shading



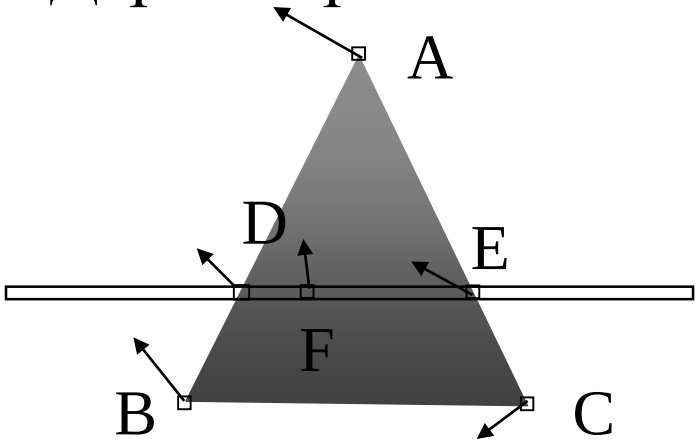
Gouraud Shading




A program for learning viewing in opengl

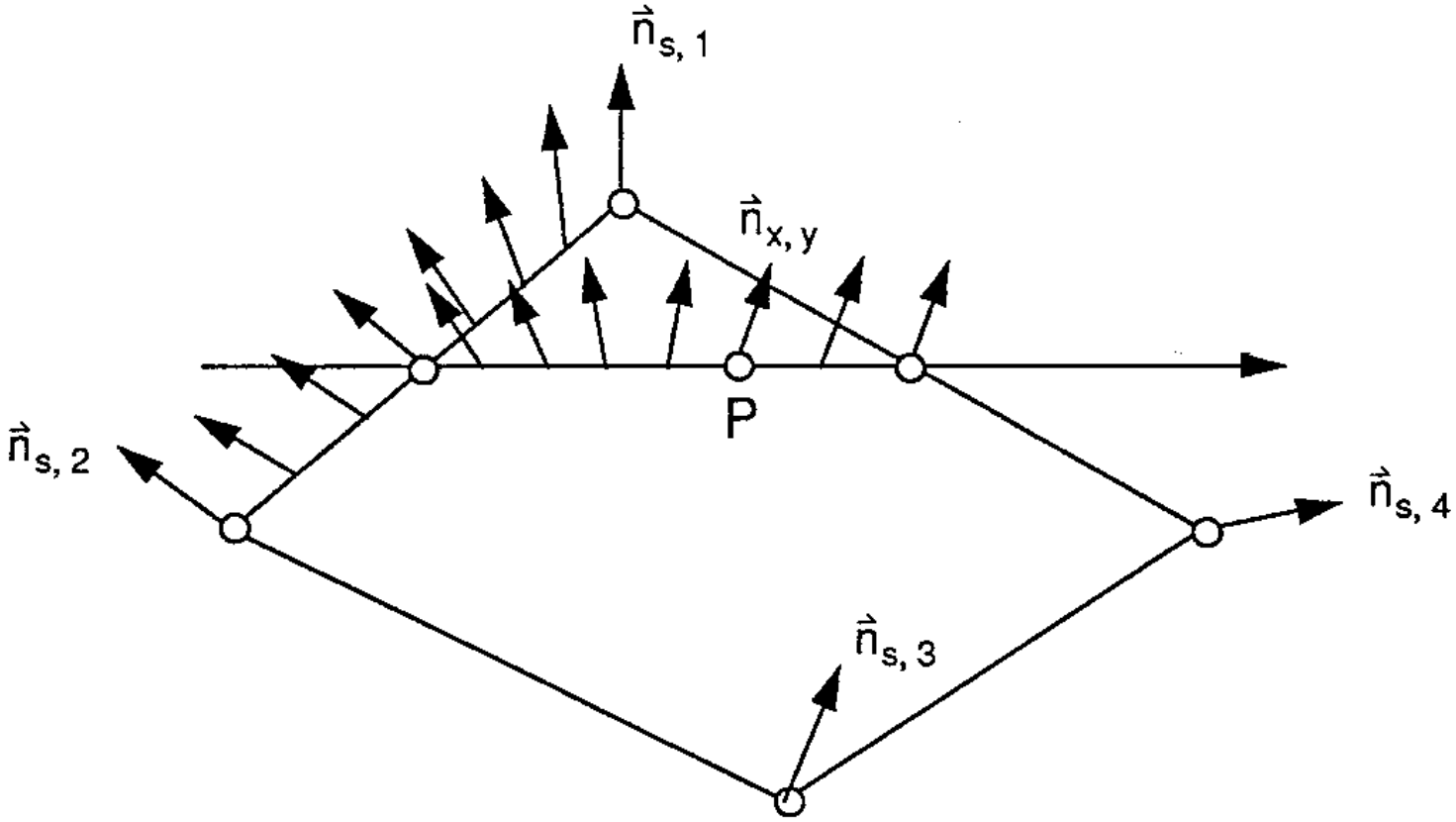
Phong Shading

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



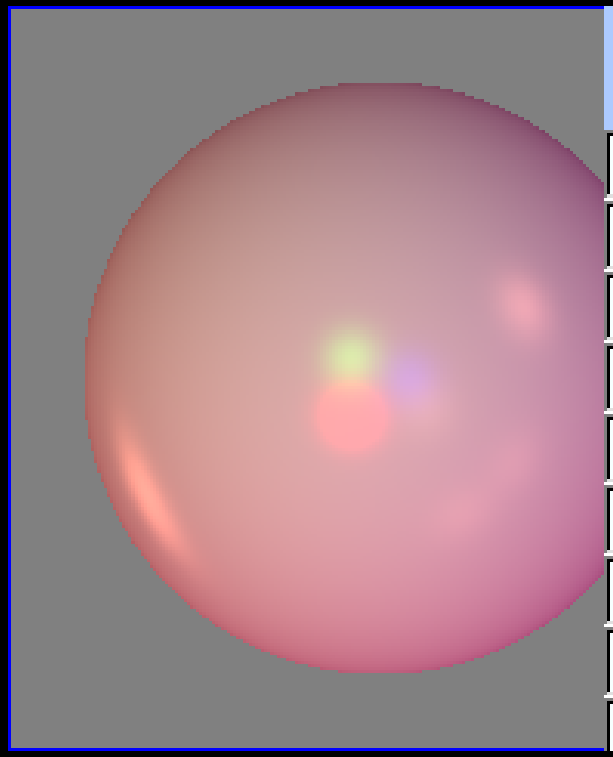
- Gouraud сүүдэрлэх аргыг ашиглан бид хангалттай үр дүнд хүрэхийн тулд гадаргууг маш жижиг хэсгүүдэд хуваах хэрэгтэй. Phong shading нь хэт олон зүсэлтийг бууруулдаг боловч илүү их тооцоо шаарддаг.
- Энэ нь чанартай зураг гаргаж болох боловч илүү удаан байдаг

- Ирмэгийн дагуух нормал болон пикселийн мөрийг интерполяци хийх



http://cs1.cs.nyu.edu/ms_students/chan7637/shade.html

Contact People Yellow Pages Download Channels



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

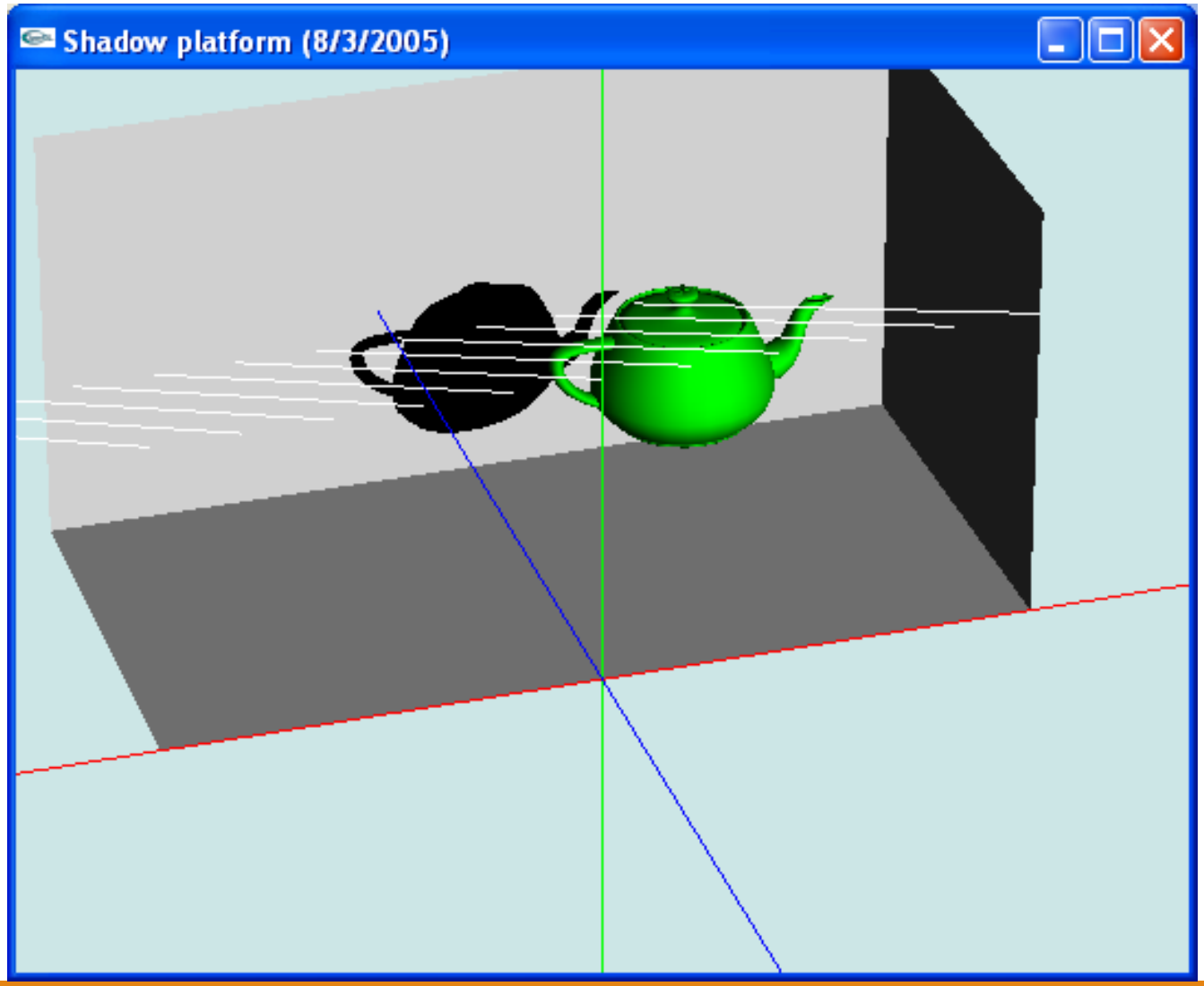
R	G	B	X	Y	Z
0.3	0.3	0.3	2.0	1.0	1.0
0.5	0.5	0.5	-2.0	-1.0	-1.5
0.04	0.08	0.04	-1.0	1.0	7.0
0.1	0.13	0.1	2.0	-1.3	6.0
0.09	0.09	0.09	1.9	-3.1	1.7
0.09	0.09	0.09	3.1	-1.9	1.7
0.8	0.0	0.0	-1.0	-1.3	5.0
0.0	0.8	0.0	-1.0	0.7	5.0
0.0	0.0	0.9	1.0	0.0	5.0

Render **Ambient Color:** R 0.6 G 0.3 B 0.4

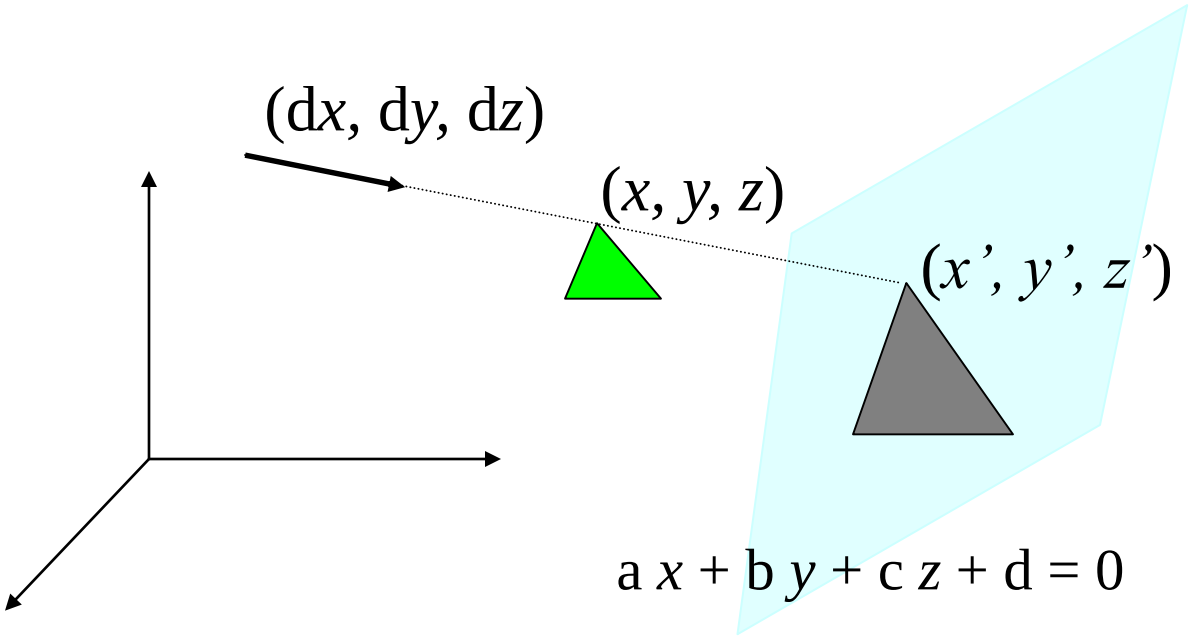
Diffuse Color: R 0.3 G 0.4 B 0.3

Specular Color: R 0.6 G 0.3 B 0.2 **Exponent** 50

- directional light source-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.

$\alpha = -(a\,x + b\,y + c\,z + d) / (a\,dx + b\,dy + c\,dz)$

$$\alpha = -(a\,x + b\,y + c\,z + d) / (a\,dx + b\,dy + c\,dz)$$

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

$$x' = \frac{(b \times dy + c \times dx)x - (b \times dx)y - (c \times dx)z - d \times dx}{a \times dx + b \times dy + c \times dx}$$

$$y' = \ldots$$

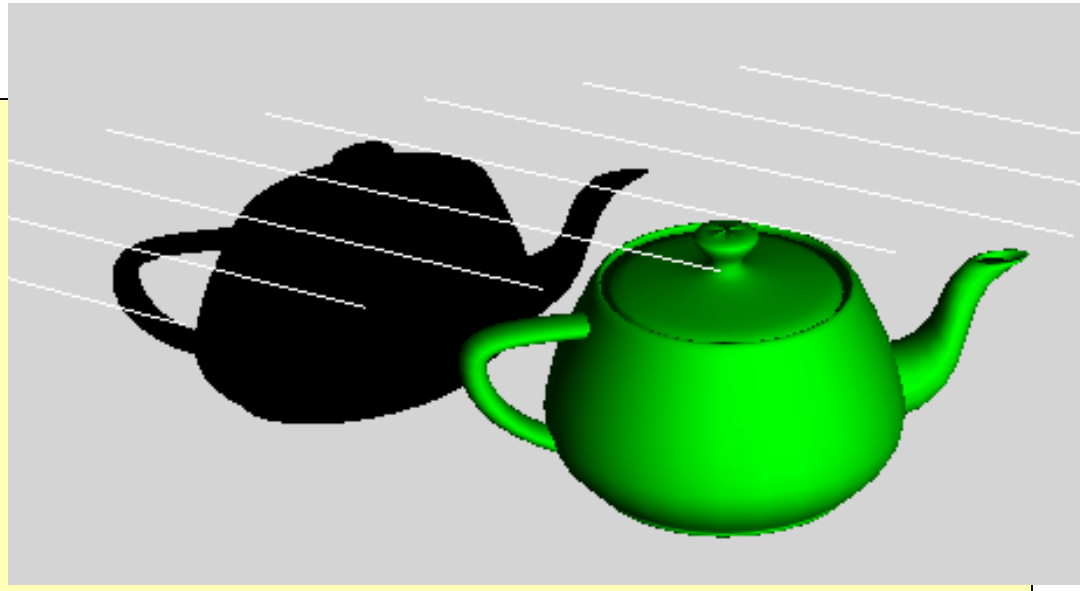
$$z' = \ldots$$

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

```
//Set up the matrix M such that it projects a vertex on the plane
//      ax + by + cz + d = 0.
//      The direction of light is (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.

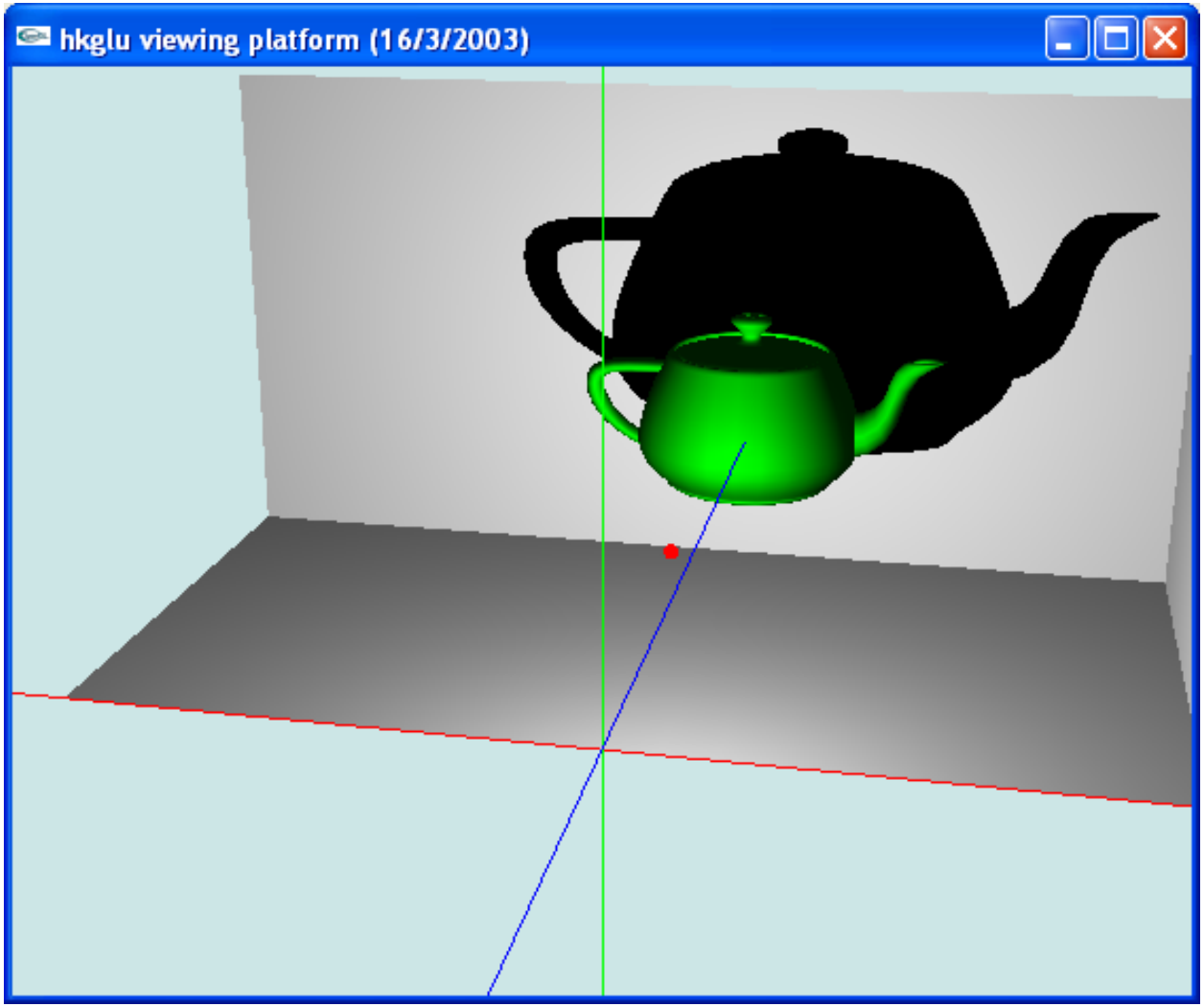
Гэрлийн чиглэл нь (dx, dy, dz) байна.



```
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();     //Draw the teapot

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 $(\alpha x) + b\,(\alpha y) + c\,(\alpha z) + d = 0.$

$ax + by + cz + d = 0$

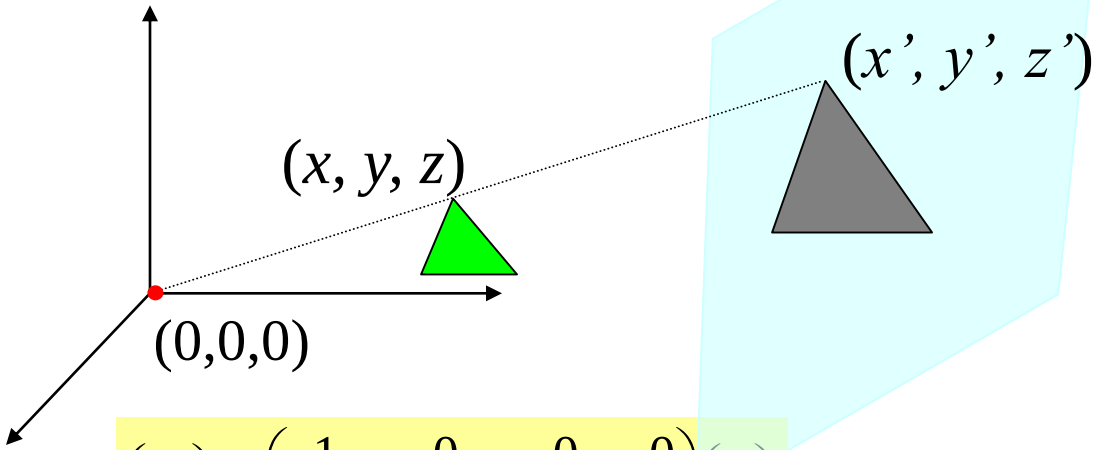
$\alpha = -d / (ax + by + cz)$

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

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

$y' = \ldots$

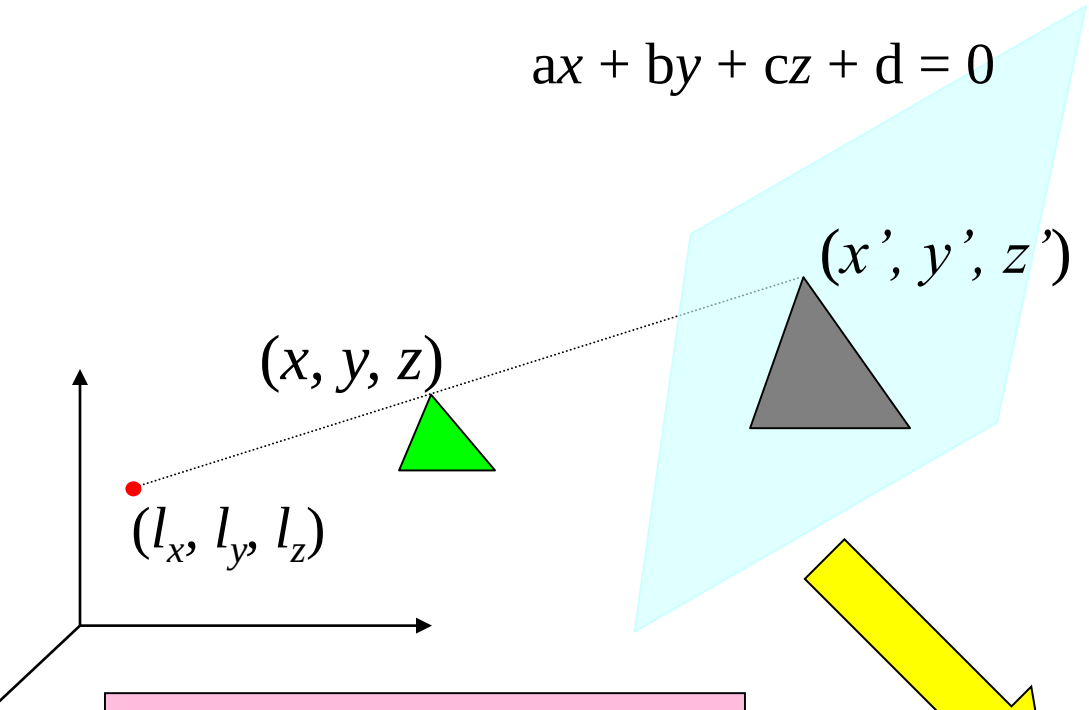
$z' = \ldots$



$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{a}{-d} & \frac{b}{-d} & \frac{c}{-d} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Хэрэв light source координатын эх дээр биш (*lx*, *ly*, *lz*) дээр байвал координатын эхийг light source рүү шилжүүлнэ.

$ax + by + cz + d = 0$



$x = x - l_x$

$y = y - l_y$

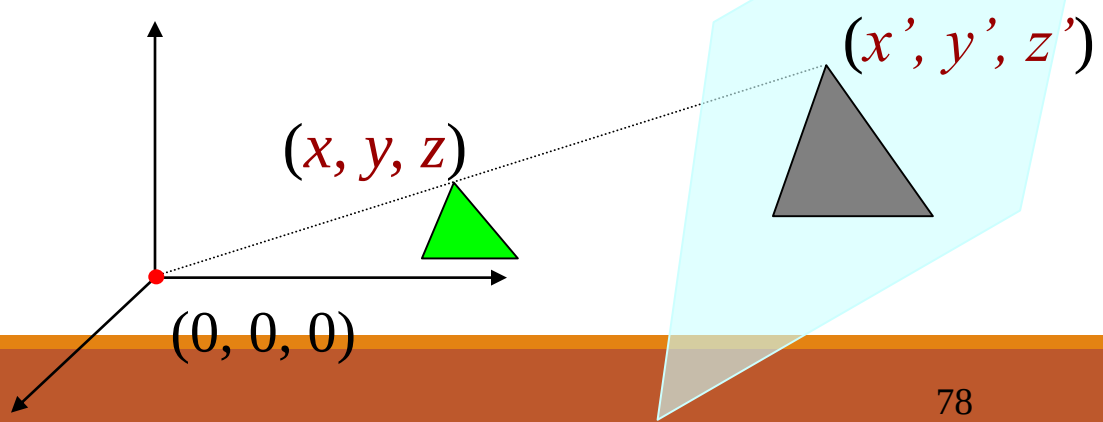
$z = z - l_z$

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

$new\_d = d + a\,l_x + b\,l_y + c\,l_z$

$ax + by + cz + new\_d = 0$

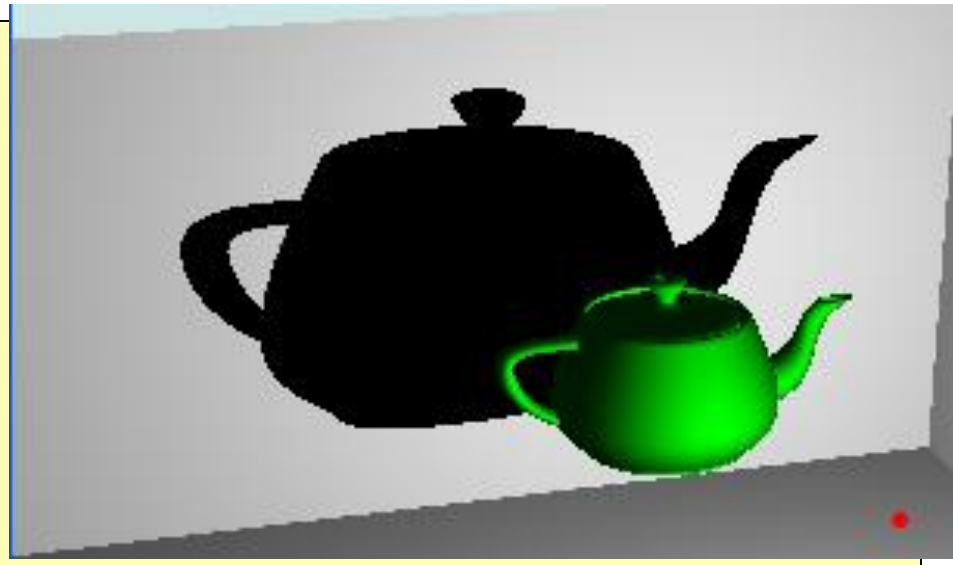
Хавтгайн constant term шилжилт хийгдсэний дараа өөрчлөгдөж байгааг анхаарна уу



```
//Set up the matrix M such that it projects a vertex on the plane
//      ax + by + cz + d = 0.
//      The light source is at (lx, ly, xz).

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();     //Draw the teapot
glPopMatrix();
```

ShadowA