



ШУТИС, Мэдээлэл Холбооны Технологийн Сургууль

F.CS209 Компьютерийн график

Лекц 11 – 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

Гэрэлтүүлэг - Lighting

- Гэрэл гэдэг нь товчхондоо дулаан биетээс цацрах фотонууд юм. Ямар хэмжээтэйгээр цацаргаснаас 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.
- Компьютер графикт гадаргуу дээр туссан гэрлийн эрчим буюу light intensity утгыг олоход хэрэглэгддэг гурван төрлийн гэрлийн ойлтын утгууд 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



Цэгэн эх үүсгүүр

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

➤ \mathbf{n} нь гадаргуугийн нормал.

\mathbf{n} хавтгай дээрх өөр өөр байршилд ижил байна

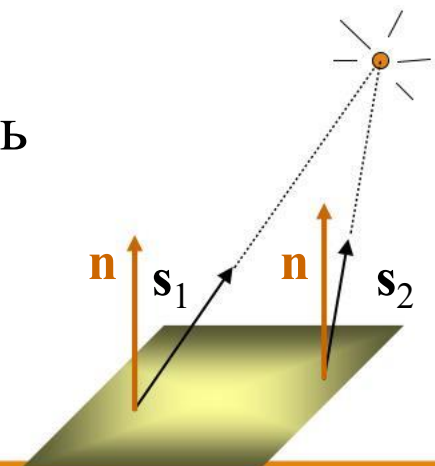
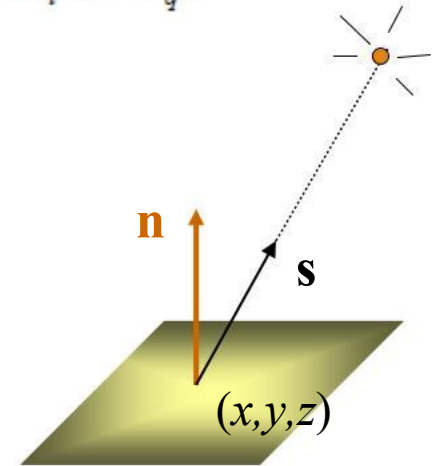
➤ (x, y, z) нь гадаргуу дээрх цэгийн координат

➤ \mathbf{s} нь (x, y, z) дэх гэрлийн эх үүсвэрийн чиглэл юм.

\mathbf{s} нэг байршилаас нөгөөд шилжинэ, жишээ нь

$$\mathbf{s}_1 \neq \mathbf{s}_2$$

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

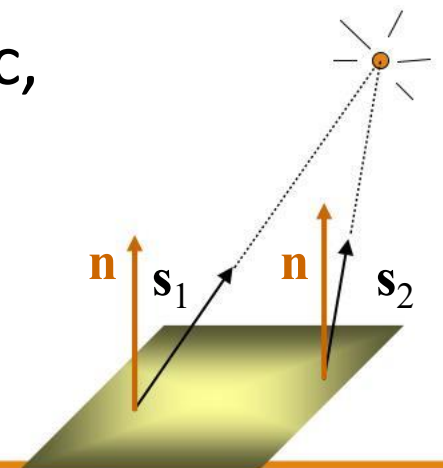
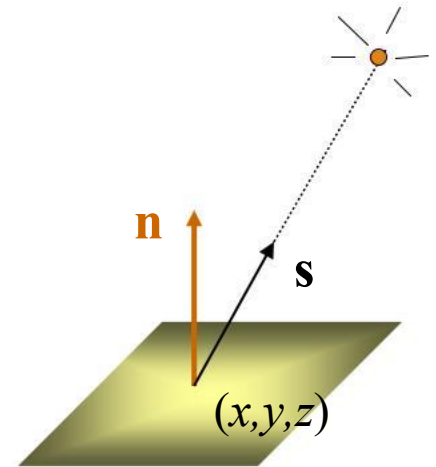


Цэгэн эх үүсгүүр

Объектийн гадаргуу дээрхи цэг рүү хүрч очих гэрлийн intensity утгыг нь дараах томъёогоор тооцон олдог.

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

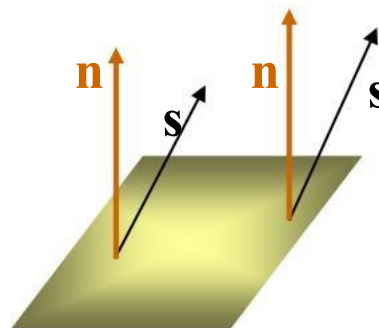
Энд **d** нь гэрлийн гадаргуу дээр туссан цэг болон тэндээс гэрлийн үүсгүүрийн байрлал хүртлэх зайн урт юм. Харин k_c , k_l , k_q утгууд нь харгалзан linear, quadratic, attenuation тогтмолууд юм. attenuation гэдэг нь гэрлийн шингэлт буюу сулрах зэргийг нь илтгэнэ.



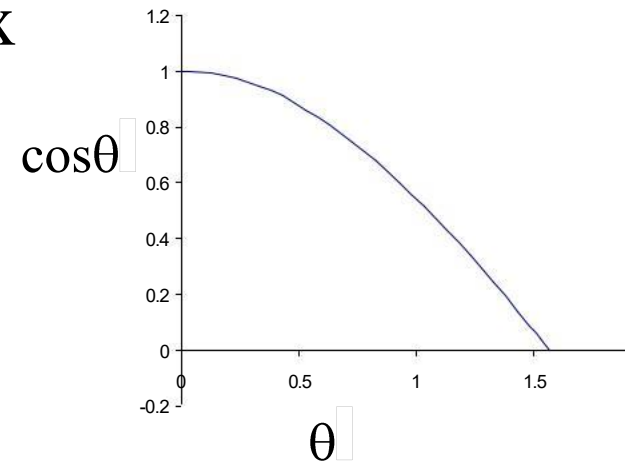
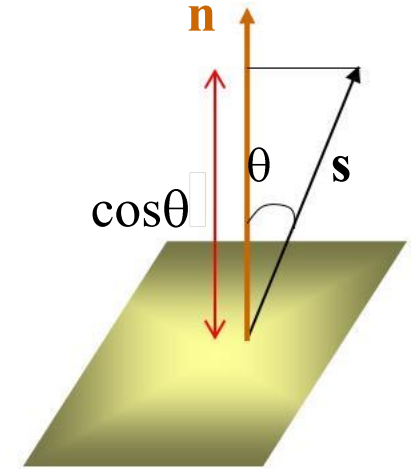


Чиглэлийн үүсгэвэр

- 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.



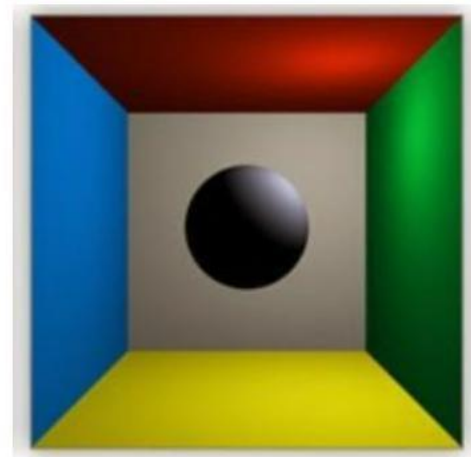
- I_i тусах гэрлийн intensity
- θ , s ба n хоорондох өнцөг, $0^\circ \leq \theta \leq 90^\circ$.
- $I_i \cos\theta$ гадаргуугийн нэгж талбайд хүлээн авсан гэрлийн хэмжээ
- $\theta = 0$ үед, тусах гэрэл нь гадаргууд перпендикуляр, гадаргуу гэрлийг хамгийн дээд хэмжээгээр хүлээн авах бөгөөд жнь үдийн нарны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



Сарнисан ойлт

diffuse гадаргуу дээр гэрлийн цацрагийн хэсгүүд цаашаага 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.

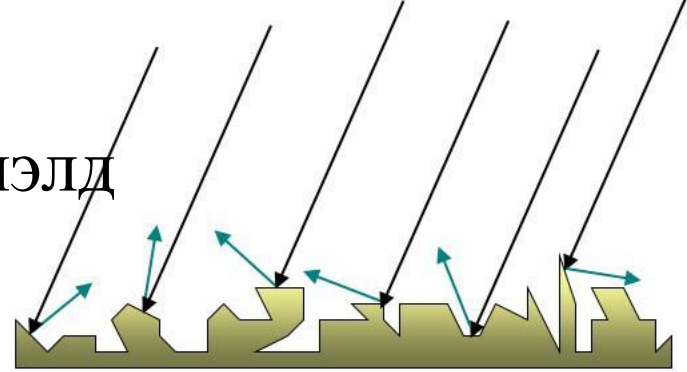
Гэхдээ илүү нарийн бүтэцтэй Oren-Nayar ийн арга ч гэж бий.



Diffuse

Diffuse - сарнисан ойлт

- Тэгш гадаргуу гэрлийг бүх чиглэлд тусгадаг

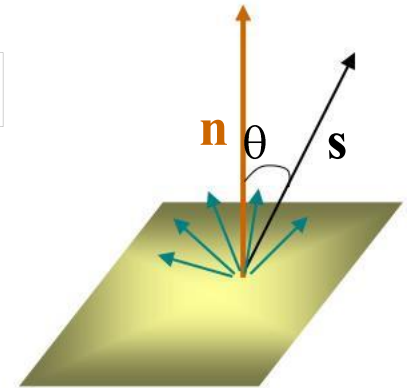


- Ламбертын косинусын хууль

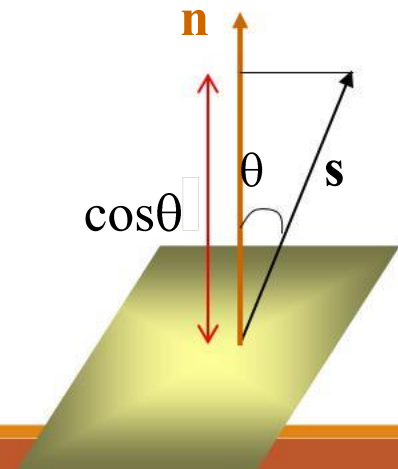
Туссан гэрлийн хэмжээ нь $I_d = R_d I_i \cos\theta$

- R_d , сарнисан тусгалын коэффициент,
 $0 \leq R_d \leq 1$

- I_d нь харах цэг (view point)



- $\cos\theta$ нь \mathbf{n} дээрх \mathbf{s} –ийн проекц.
Хэрэв \mathbf{s} ба \mathbf{n} нормчлогдсон бол,
 $\cos\theta = \mathbf{s} \cdot \mathbf{n}$



Attenuation – гэрлийн сулрал

- 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 параметрууд юм..

- Чиглэлийн эх үүсгэвэрийн хувьд, $A_f = 1$

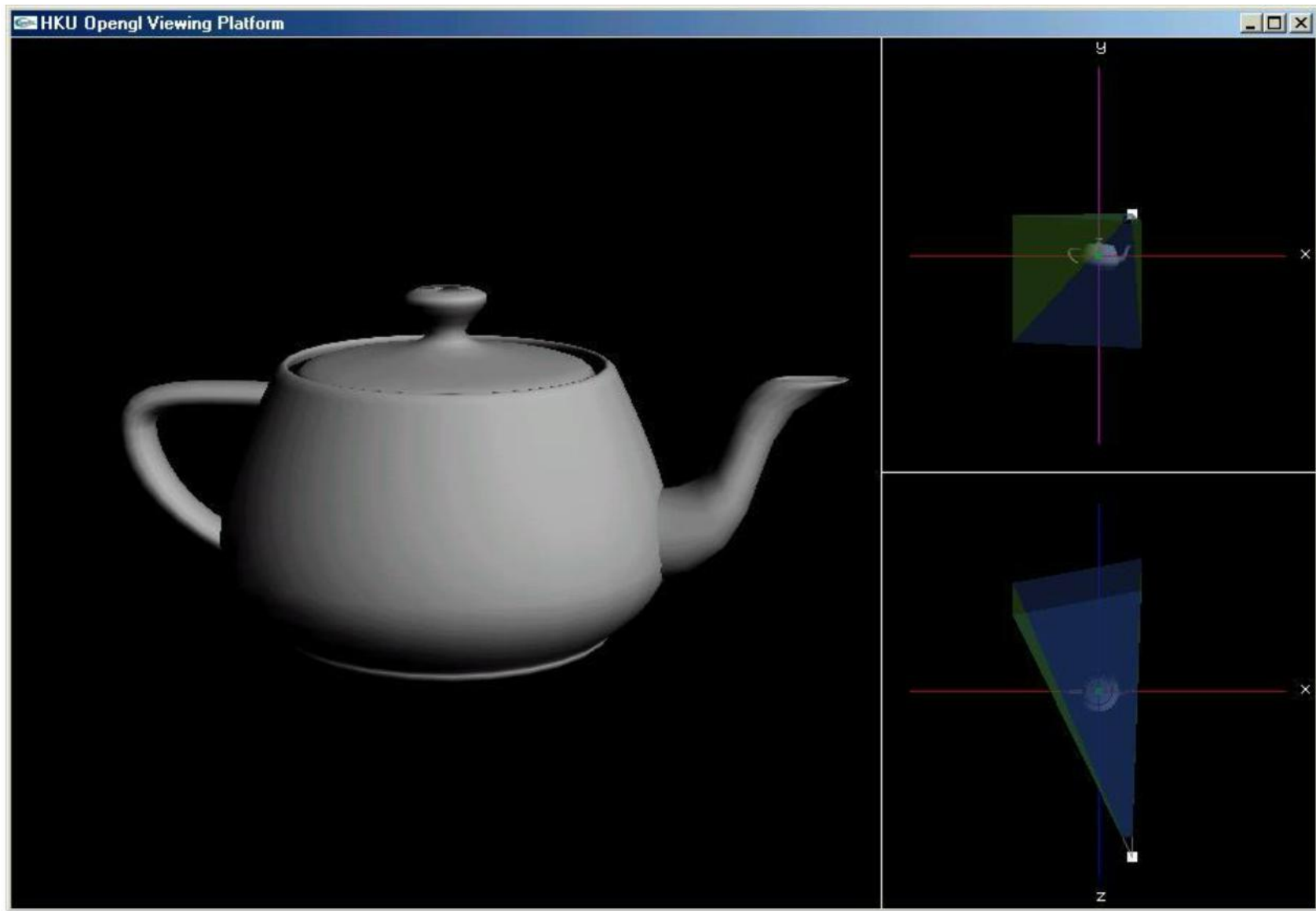
Ерөнхий томъёолол

- 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

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



Чиглэлийн эх үүсвэрээс цэвэр сарнисан тусгал $(1,1,1)$

Specular буюу толин 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.



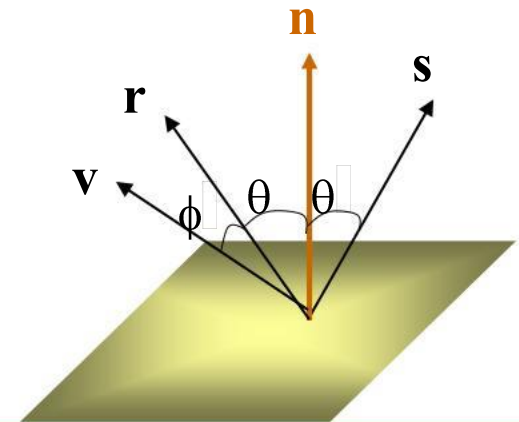
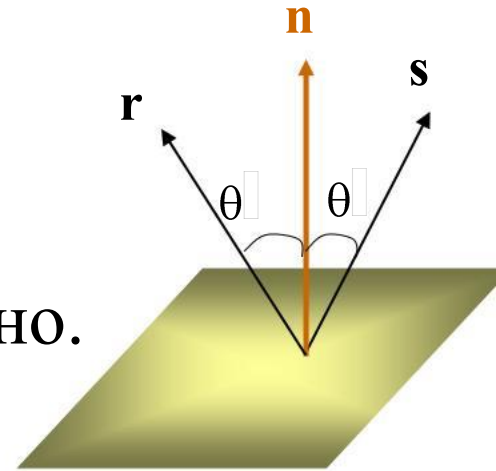
Specular

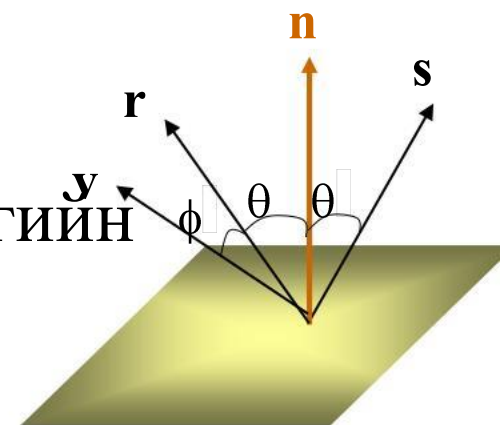
Specular ойлт

- Гялалзсан гадаргуу тусгалын чиглэлд (\mathbf{r}) гэрлийг ойлгодог.
- \mathbf{r} нь \mathbf{n} ба \mathbf{s} ийн тархсан хавтай дээр оршино.
 \mathbf{n} -ийн хоорондох өнцөг нь \mathbf{n} ба \mathbf{s} -ийн хоорондох өнцөгтэй ижил байна,
 $0^\circ \leq \theta \leq 90^\circ$.
- Гадаргуугийн цэгийг \mathbf{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 коэффициент,
 $0 \leq R_s \leq 1$





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

➤ ϕ нь харах чиглэл ба төгс тусгалын цацрагийн хоорондох өнцөг юм.

➤ Гялалзсан гадаргуу дээр \mathbf{v} нь \mathbf{r} бага зэрэг хазайхад туссан гэрэл маш бага үзэгдэнэ. Ө.х, dull surface дээр гэрэл илүү их үзэгдэнэ. Ийм үзэгдлийг $\cos^k \phi$ томъёогоор тодорхойлох ба k нь материалын гялалзсан байдал. k их байх тусам гялалзах чадвар өндөр болно

➤ $\cos \phi = \mathbf{r} \bullet \mathbf{v}$

➤ Сулралтыг diffuse reflection адил загварчилдаг.

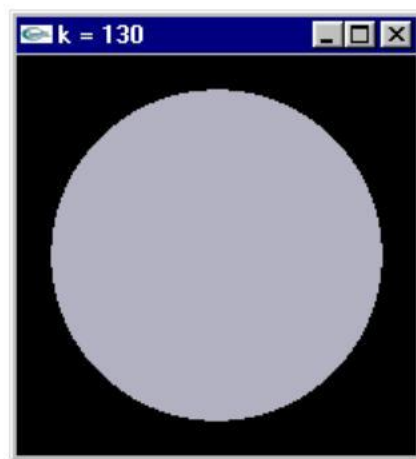
Ерөнхий томъёолол

➤ 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

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



к нь 128-аас бага байна

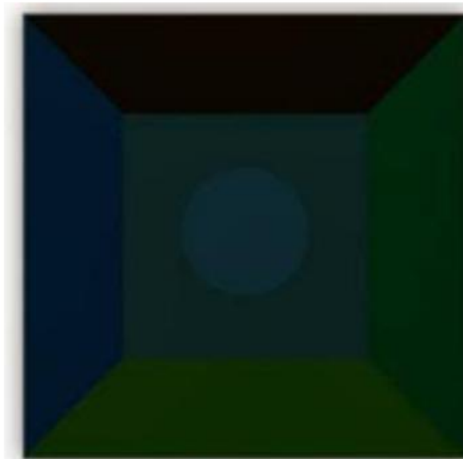
Pure Specular Reflection



Ambient ойлт- орчны тусгалын 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.

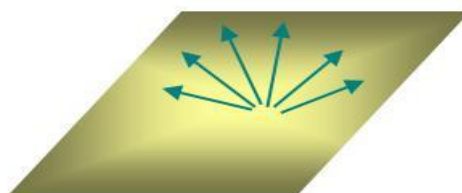


Ambient

ambient light ойлт (background), I_a

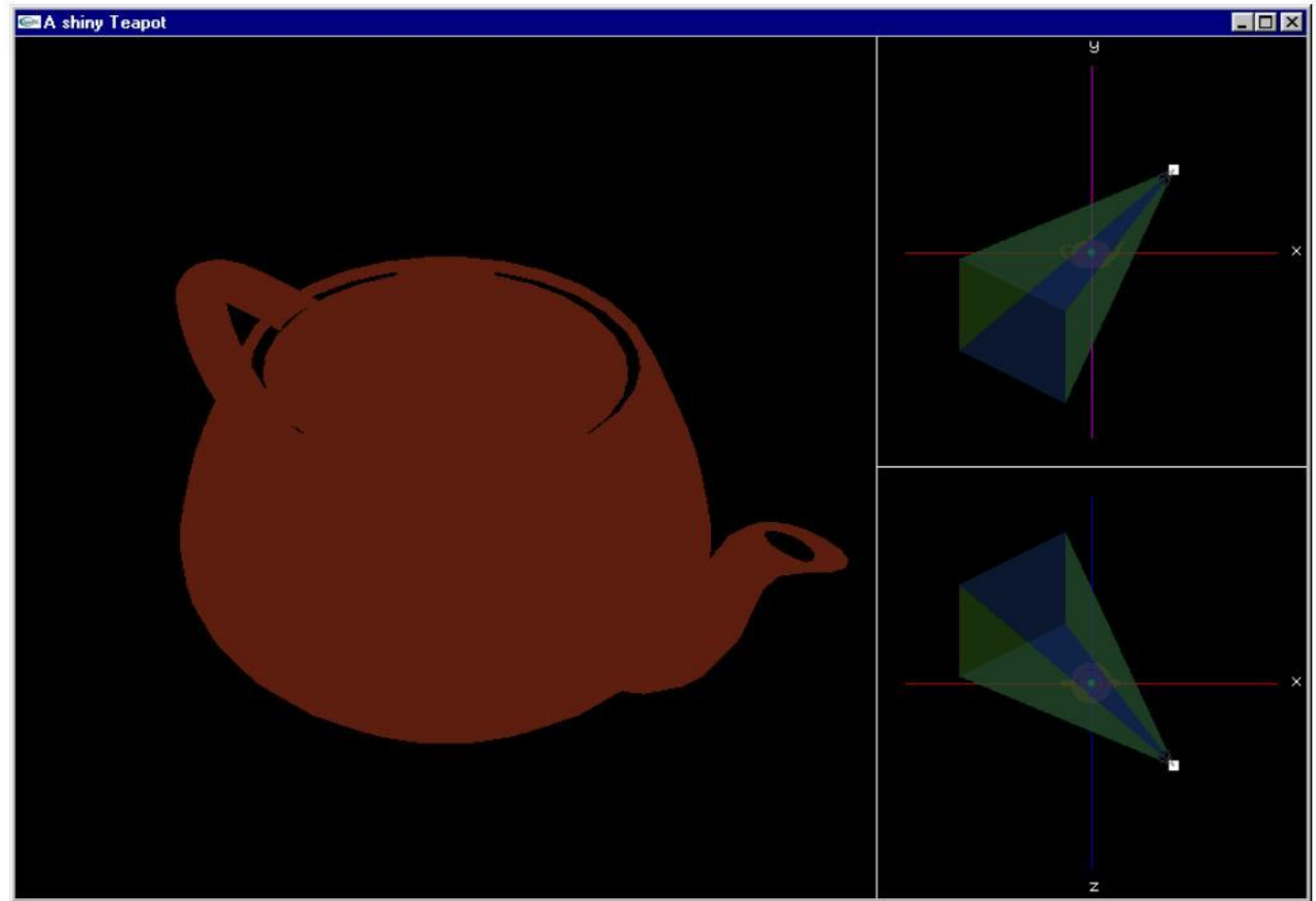
- Ambient light (I_{Am}) нь гадаргуу ба гэрэл хоорондын олон тусгалаас шалгаалж жигд гэрэлтдэг.

$$I_a = R_a I_{Am}$$

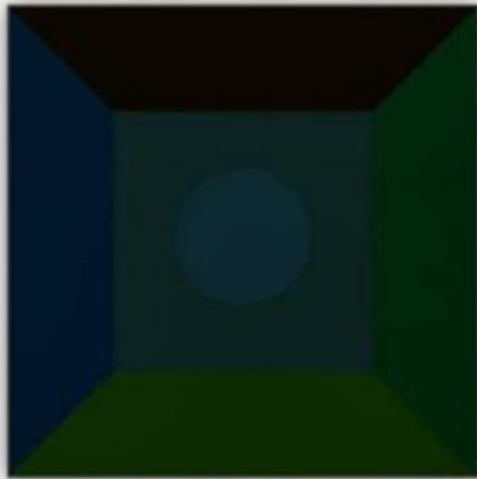


- R_a орчны тусгалын коэффициент,
 $0 \leq R_a \leq 1$
- I_a гадаргуугийн нормал ба харах өнцөг, гэрлийн эх үүсгүүрийн байрлал

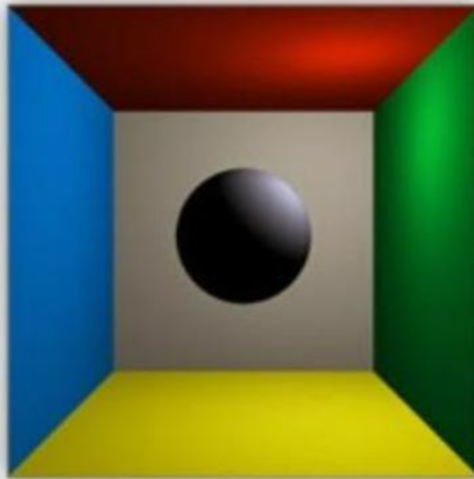
```
float globalAmbient[] = { 0.9, 0.9, 0.9, 1.};  
glLightModelfv( GL_LIGHT_MODEL_AMBIENT, globalAmbient );
```



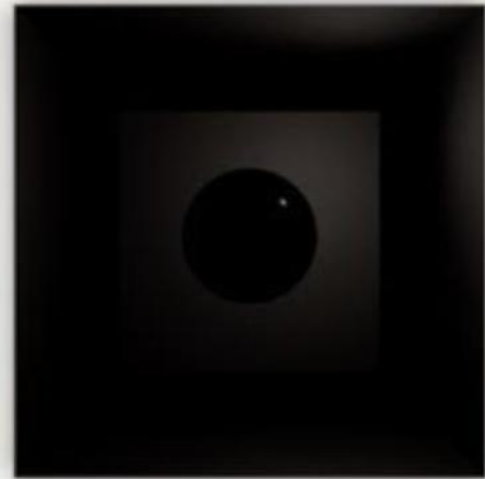
Pure Ambient



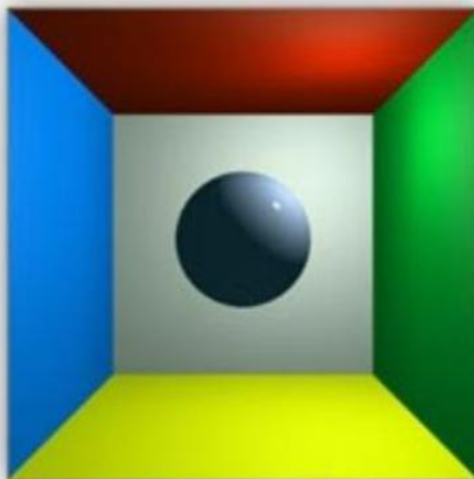
Ambient



Diffuse



Specular



Ambient + Diffuse
+ Specular

OpenGL Хэрэгжүүлэлт (Lightinig)

Гэрлийн эх үүсвэр бий болгох

Гэрлийн эх үүсвэр нь өнгө, байрлал, чиглэл гэсэн параметруудтэй 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);
```

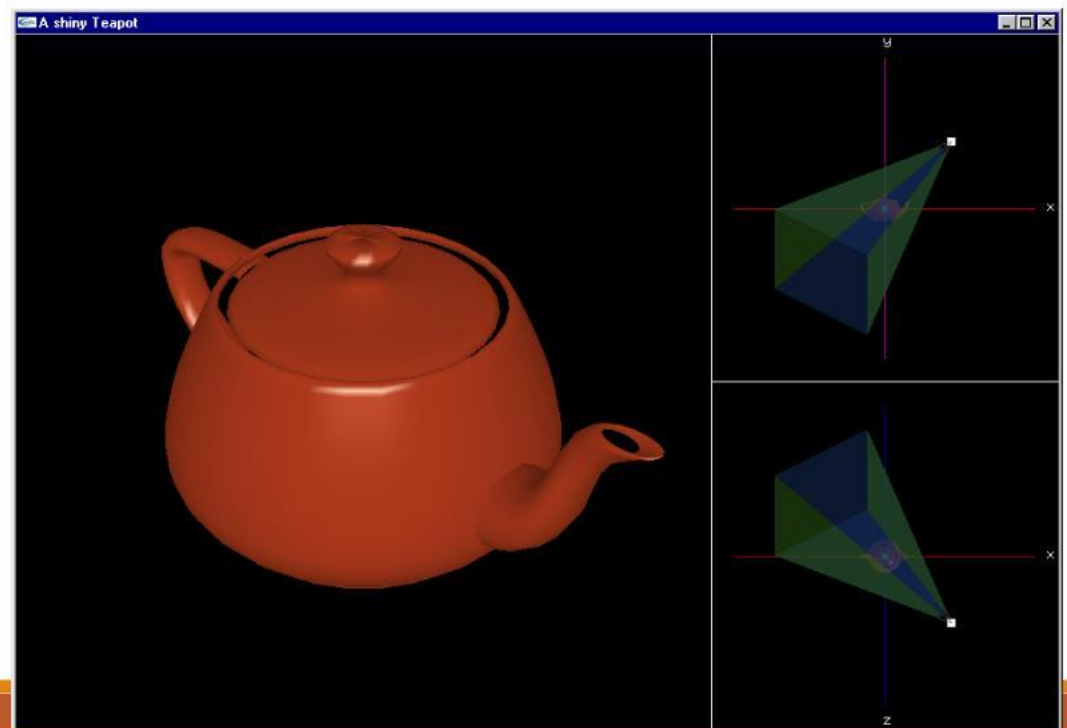
Light параметрийн утга GL_LIGHT0, GL_LIGHT1,..

Name 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

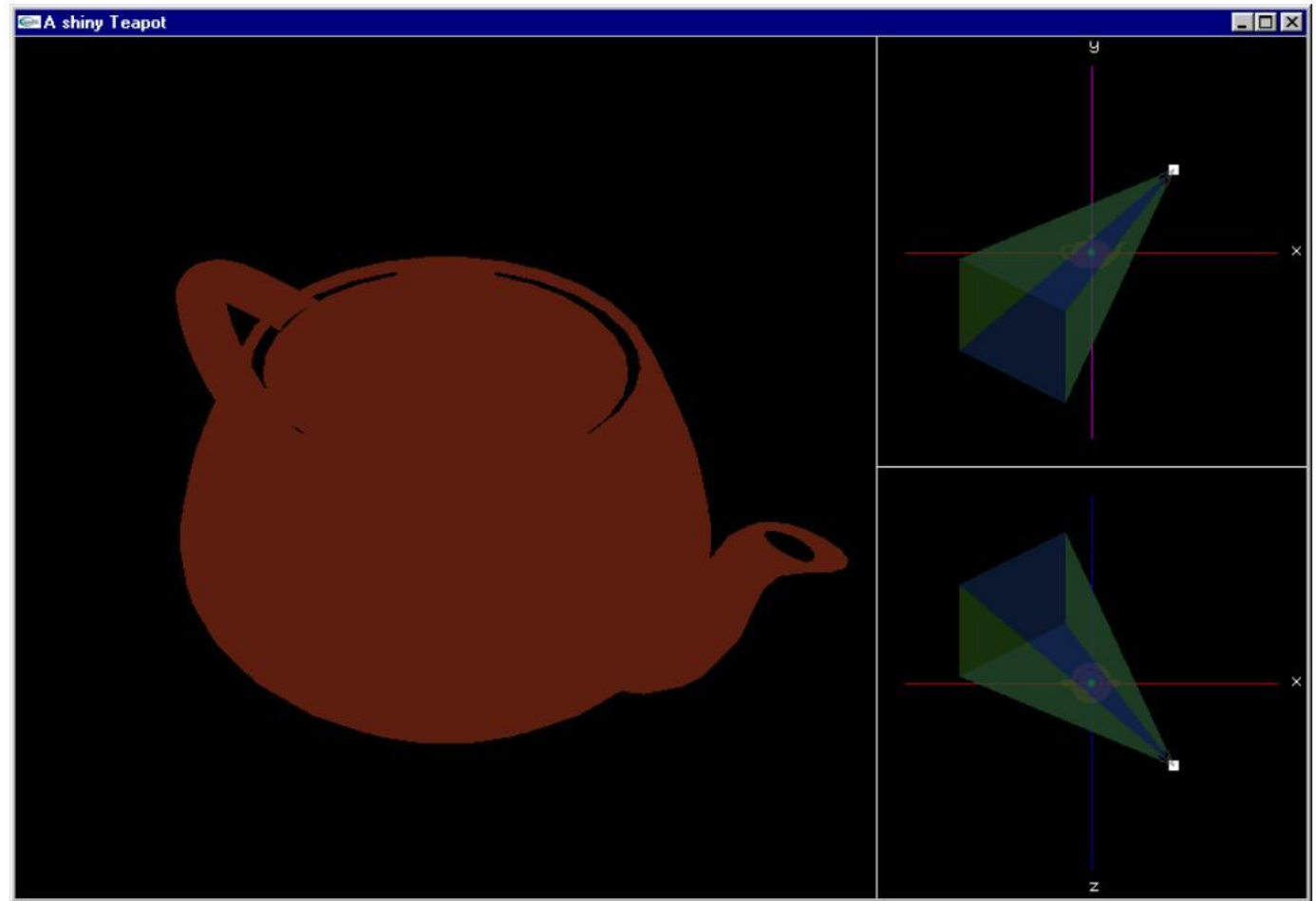
➤ Өнгөт зураг гарган авахын тулд сүүдрийн тооцоолол улаан, ногоон, цэнхэр компонент тус бүрд нь гүйцэтгэнэ.

```
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

```
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) {
```

```
        glColor3f( cos( t), 0., sin( t));
```

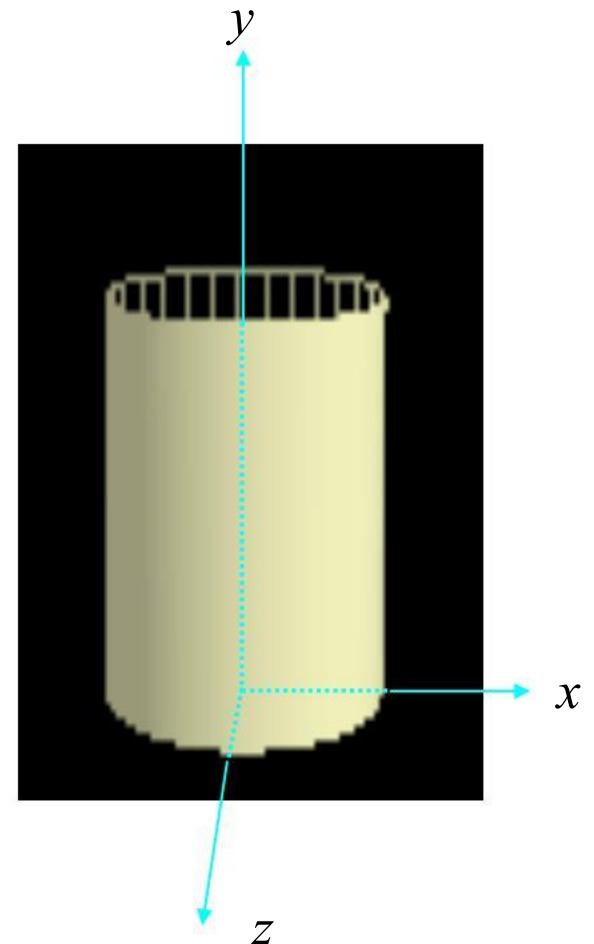
```
        glVertex3f( cos( t), 0., sin( t));
```

```
        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.);

```

```

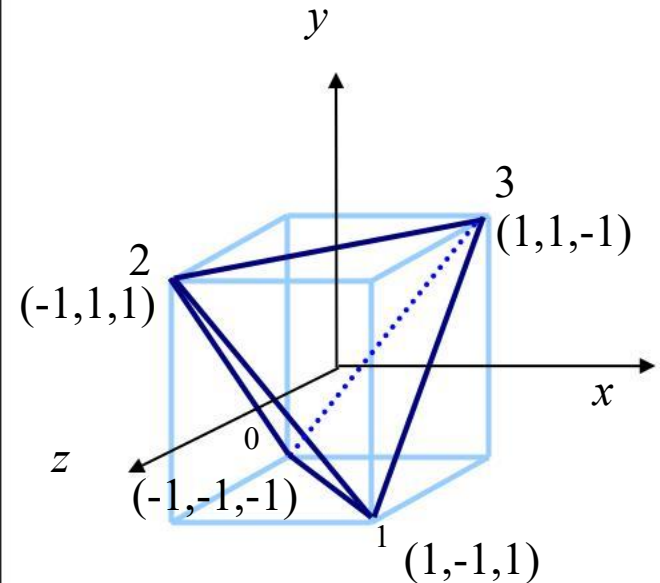
    triangle( v, 0, 3, 1);

```

```

}

```



➤ 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 орчноос хамаарна. 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
```

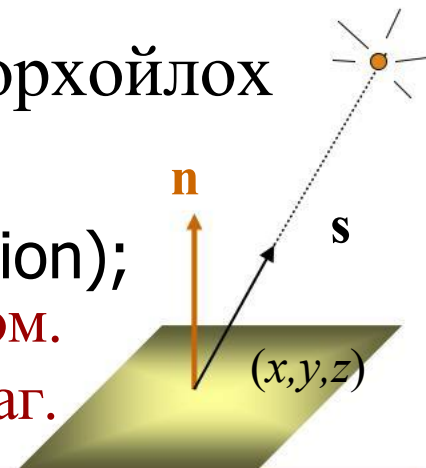
```
GLfloat s[] = { 1., 1., 0., 1.}; //Yellow
```

```
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);

```
1 #include <GL/gl.h>
2 #include <GL/glu.h>
3 #include <GL/glut.h>
4
5 void init(void)
6 {
7     GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
8     GLfloat mat_shininess[] = { 50.0 };
9     GLfloat light_position[] = { 1.0, 1.0, 1.0, 0.0 };
10    glClearColor (0.0, 0.0, 0.0, 0.0);
11    glShadeModel (GL_SMOOTH);
12
13    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
14    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
15    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
16
17    glEnable(GL_LIGHTING);
18    glEnable(GL_LIGHT0);
19    glEnable(GL_DEPTH_TEST);
20 }
```

```

22 void display(void)
23 {
24     glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
25     glutSolidSphere (1.0, 20, 16);
26     glFlush ();
27 }
28
29 void reshape (int w, int h)
30 {
31     glViewport (0, 0, (GLsizei) w, (GLsizei) h);
32     glMatrixMode (GL_PROJECTION);
33     glLoadIdentity();
34     if (w <= h)
35         glOrtho (-1.5, 1.5, -1.5*(GLfloat)h/(GLfloat)w,
36                 1.5*(GLfloat)h/(GLfloat)w, -10.0, 10.0);
37     else
38         glOrtho (-1.5*(GLfloat)w/(GLfloat)h,
39                 1.5*(GLfloat)w/(GLfloat)h, -1.5, 1.5, -10.0, 10.0);
40     glMatrixMode(GL_MODELVIEW);
41     glLoadIdentity();
42 }
43
44 int main(int argc, char** argv)
45 {
46     glutInit(&argc, argv);
47     glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
48     glutInitWindowSize (500, 500);
49     glutInitWindowPosition (100, 100);
50     glutCreateWindow (argv[0]);
51     init ();
52     glutDisplayFunc(display);
53     glutReshapeFunc(reshape);
54     glutMainLoop();
55     return 0;
56 }

```