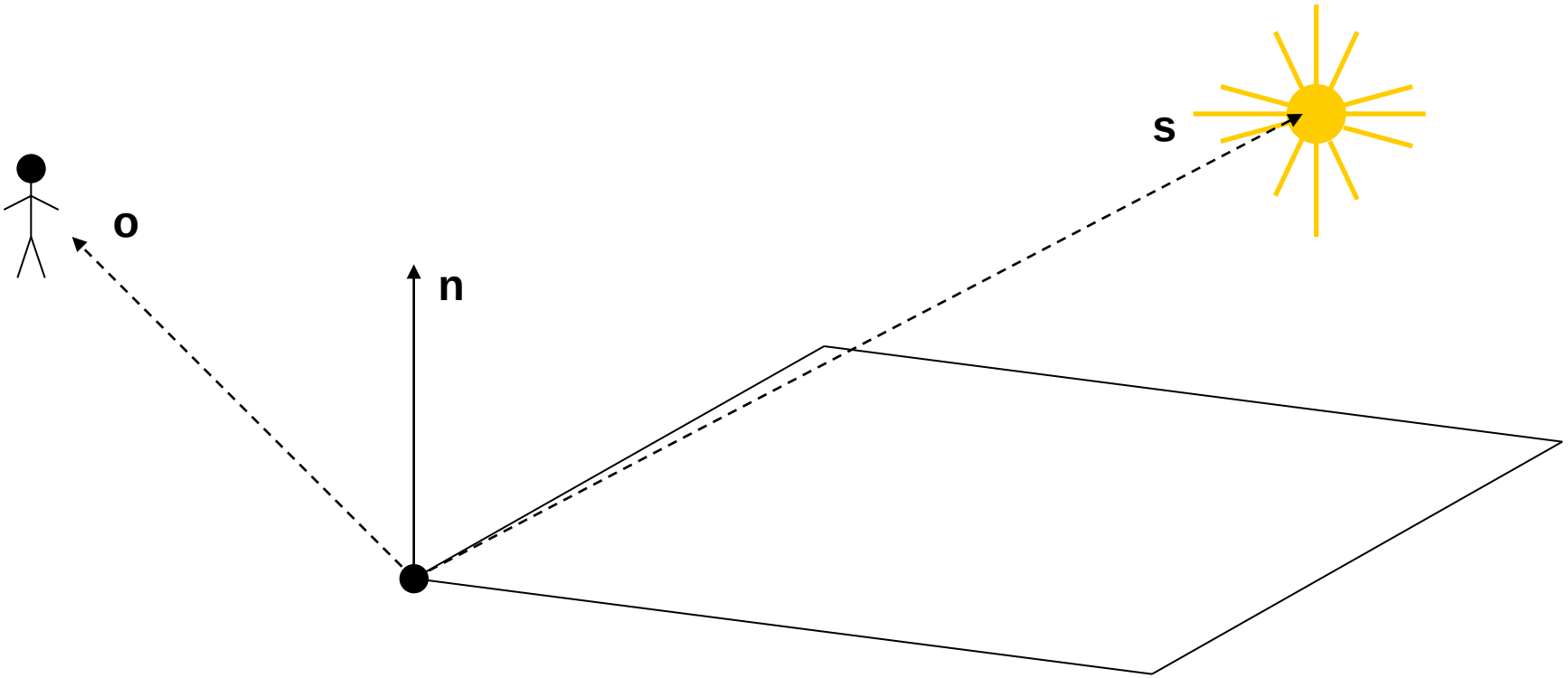
OpenGL: Osvetljenje

- Svetlo se primenjuje na nivou **vertex**-a
- Osvetljaj fragmenata se interpolira (slično kao kod **`glShadeModel( GL_SMOOTH)`**)

- Broj nezavisnih izvora svetla je implementaciono zavisan (OpenGL 2.1: mora biti minimalno 8)

- Karakteristike svetla:
  - pozicija
  - ambijentalna komponenta
  - difuzna komponenta
  - spekularna (blještava, reflektujuća, nalik ogledalu) komponenta
- Svaki **vertex** mora da ima dodeljenu **normalu**
- Vektor normale utiče na osvetljaj

Osvetljenje

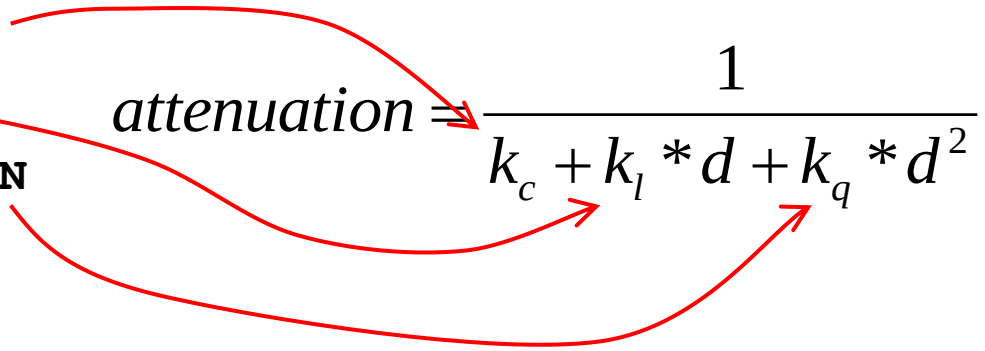


n – vektor normale za dati vertex v
s – vektor položaja izvora svetla u odnosu na vertex
o – vektor položaja posmatrača u odnosu na vertex

-Za računanje efekta difuzne komponente, koristi se skalarni proizvod n*s
-Za računanje efekta spekularne komponente, koriste se sva tri vektora

Osvetljenje

- Zadavanje parametara izvoru svetla:
  - **`void glLight{if}(enum light, enum pname, T param);`**
  - **`void glLight{if}v(enum light, enum pname, T params);`**
- light: **`GL_LIGHT0..GL_LIGHT`*`n`*`, [GL_LIGHT`*`i`*`=GL_LIGHT0+`*`i`*`]`**
- **pname:**
  - **`GL_AMBIENT`**
  - **`GL_DIFFUSE`**
  - **`GL_SPECULAR`**
  - **`GL_POSITION`**
  - **`GL_SPOT_DIRECTION`**
  - **`GL_SPOT_EXPONENT`**
  - **`GL_SPOT_CUTOFF`**
  - **`GL_CONSTANT_ATTENUATION`**
  - **`GL_LINEAR_ATTENUATION`**
  - **`GL_QUADRATIC_ATTENUATION`**



Osvetljenje

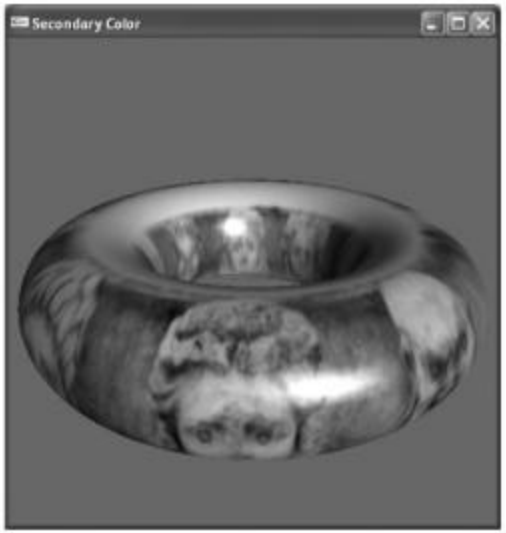
- Zadavanje pozicije svetla
  - **`GLfloat light_position[] = { x, y, z, w };`**
    `glLightfv(GL_LIGHT0, GL_POSITION, light_position);`
- Ako je **`w=0,`** radi se o usmerenom izvoru svetla, **`(x,y,z)`** određuje pravac
- Ako je **`w≠0,`** radi se o neusmerenom izvoru svetla, **`(x,y,z)`** određuje poziciju
- Na zadatu poziciju se primenjuje matrica transformacije modela i pogleda (MODELVIEW), odnosno pozicija svetla se pamti u koordinatnom sistemu pogleda
  - ako izvor svetla treba da bude nepomičan u odnosu na posmatrača, pozicija se zadaje pre transformacije pogleda (**`gluLookAt`**)
  - u suprotnom, pozicija se zadaje nakon transformacije pogleda

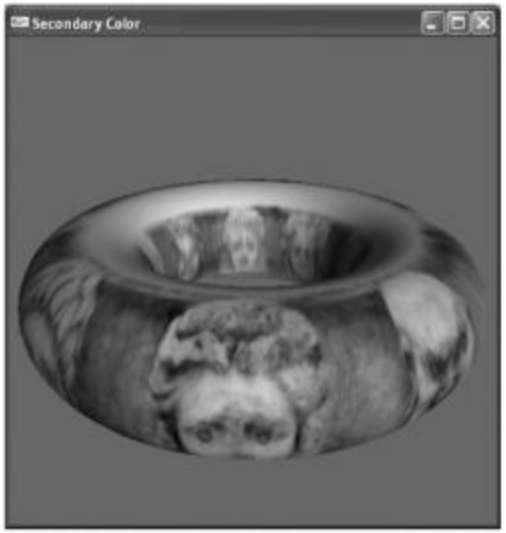
Osvetljenje

- Parametri modela svetla
  - **`void LightModel{if}( enum pname, T param );`**
  - **`void LightModel{if}v( enum pname, T params );`**

pname	**param**
GL_LIGHT_MODEL_AMBIENT	Globalno amb. osvetljenje Default: (0.2, 0.2, 0.2, 1.0)
GL_LIGHT_MODEL_LOCAL_VIEWER	Položaj posmatrača Default: GL_FALSE
GL_LIGHT_MODEL_TWO_SIDE	Primena svetla na prednju i zadnju stranu Default: GL_FALSE
GL_LIGHT_MODEL_COLOR_CONTROL	Kada se primenjuje spekularna komponenta Default: GL_SINGLE_COLOR Nakon tekst.:GL_SEPARATE_SPECULAR_COLOR

Osvetljenje





Primena spekularne komponente pre (levo) i posle (desno) primene teksture
Izvor: Paul Martz, “OpenGL Distilled”, str. 169, 2006.

OpenGL: Stapanje (Blending)

- Mešanje boje dva fragmenta
  - onog koji treba da se doda u bafer (**source**)
  - onog koji se već nalazi u baferu (**destination**)
- Aktiviranje moda za stapanje
  - **`glEnable/glDisable(GL_BLEND);`**
- Zadavanje načina stapanja
  - **`void glBlendFunc(GLenum sfactor,GLenum dfactor);`**

  `sfactor` i **`dfactor`** su komande za računanje faktora stapanja

`GL_ZERO`
`GL_ONE`
`GL_DST_COLOR`
`GL_SRC_COLOR`
`GL_ONE_MINUS_DST_COLOR`
`GL_ONE_MINUS_SRC_COLOR`
`GL_SRC_ALPHA`
`GL_ONE_MINUS_SRC_ALPHA`
`GL_DST_ALPHA`
`GL_ONE_MINUS_DST_ALPHA`
`GL_SRC_ALPHA_SATURATE`

NEMAJU SVE KOMBINACIJE SMISLA!

Stapanje (Blending)

Primer:

```
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glEnable(GL_BLEND);
glBegin(...);
  glColor4f( r, g, b, a );
  ...
glEnd();
```

Nova boja piksela se dobija na sledeći način:
- **`src`** = boja kojom se crta
- **`dst`** = boja koja se već nalazi u kolor-baferu

`nova_boja = src * a + dst * (1-a)`

Pošto se boja i transparencija zadaju **na nivou verteksa**,
za popunjavanje primitiva se koristi **linearna interpolacija vrednosti**.

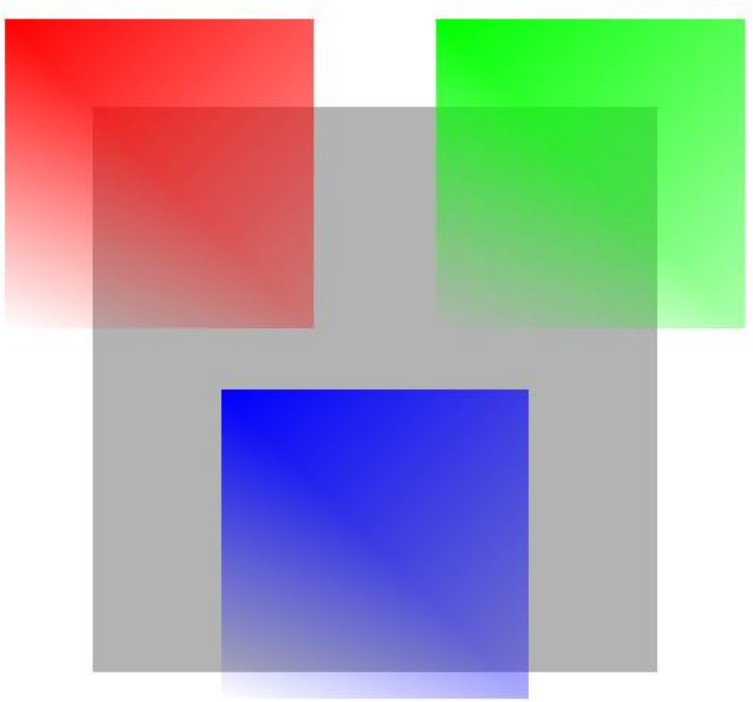
Stapanje (Blending)

```
glBegin(GL_QUADS);
      // crveni kvadrat
     glColor4f( 1, 0, 0, 0 );
     glVertex3f( -3, 0.5f, -5 );
     glColor4f( 1, 0, 0, 0.33f );
     glVertex3f( -0.5f, 0.5f, -5 );
     glColor4f( 1, 0, 0, 0.67f );
     glVertex3f( -0.5f, 3, -5 );
     glColor4f( 1, 0, 0, 1 );
     glVertex3f( -3, 3, -5 );

      // zeleni kvadrat...

      // plavi kvadrat...

glEnd();
```

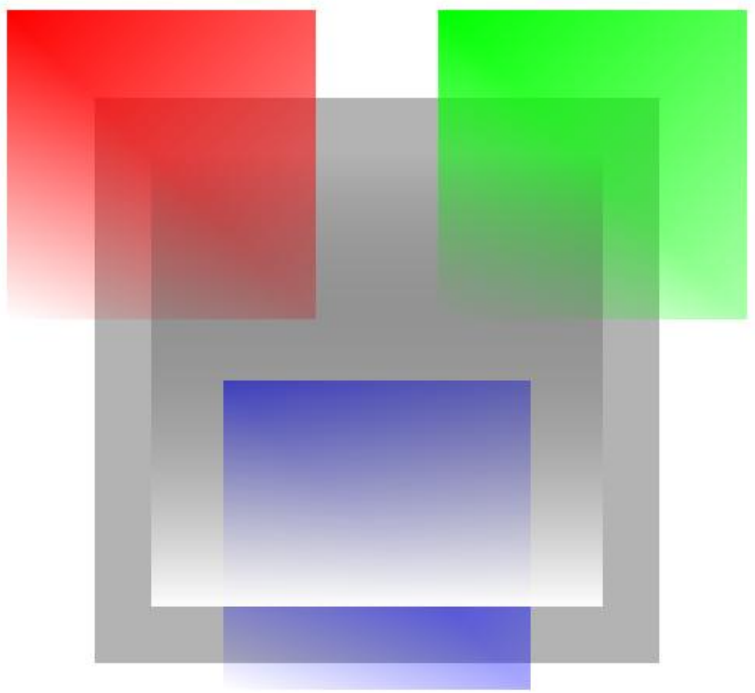


Stapanje (Blending)

Primer:

```
glBlendFunc(GL_SRC_COLOR, GL_ONE_MINUS_SRC_COLOR);
```

```
glBegin(GL_QUADS);
     glColor3f(1.0f, 1.0f, 1.0f);
     glVertex3f( -1.5, -1.5, -5 );
     glVertex3f( 1.5, -1.5, -5 );
     glColor3f(0.0f, 0.0f, 0.0f);
     glVertex3f( 1.5, 1.5, -5 );
     glVertex3f( -1.5, 1.5, -5 );
 glEnd();
```



Stapanje (Blending)

- Kod tekstura
  - može se zadati providnost svakom tekselu
  - obično je tip teksture RGBA
  - novi podatak (A) se tretira ravnopravno sa ostalim:
    - transformacije (skaliranje, filtriranje,...)

Stapanje (Blending)

```
glDisable(GL_BLEND);
```

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

```
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_COLOR,
GL_ONE_MINUS_SRC_COLOR);
```

Stapanje (Blending)

- Moguće je razdvojiti koeficijente
  - posebni koeficijenti za RGB
  - posebni koeficijenti za A
- Slično tome: moguće je definisati **OP**

glBlendFunc

$$RGB_{SRC} \times Faktor_{SRC} \;\; OP \;\; RGB_{DST} \times Faktor_{DST}$$
$$[\; A_{DST} = A_{SRC} \;]$$

glBlendFuncSeparate

$$RGB_{SRC} \times Faktor_{SRC} \;\; OP \;\; RGB_{DST} \times Faktor_{DST}$$
$$A_{DST} = A_{SRC} \times Faktor_{SRC} \;\; OP \;\; A_{DST} \times Faktor_{DST}$$

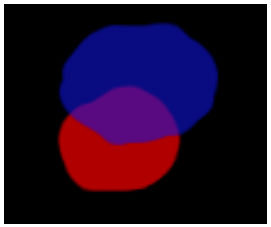
Stapanje (Blending)

- $RGB_{SRC} \times Faktor_{SRC}$ OP1 $RGB_{DST} \times Faktor_{DST}$
- $A_{SRC} \times FaktorA_{SRC}$ OP2 $A_{DST} \times FaktorA_{DST}$

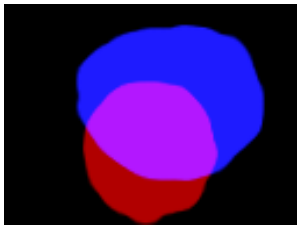
OP
Add (default)
Sub
RevSub
Min
Max
...

Faktor	
One	DstColor
Zero	DstAlpha
SrcColor	OneMinusDstColor
SrcAlpha	OneMinusDstAlpha
OneMinusSrcColor	
OneMinusSrcAlpha	

glBlendEquation
glBlendEquationSeparate



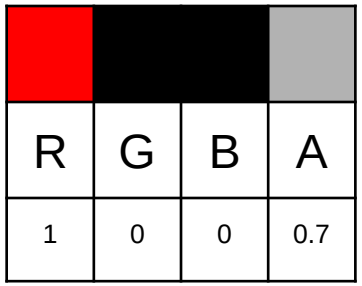
$Faktor_{SRC}$=SrcAlpha
$Faktor_{DST}$=OneMinusSrcAlpha



$Faktor_{SRC}$=One
$Faktor_{DST}$=One

Stapanje (Blending)

SRC



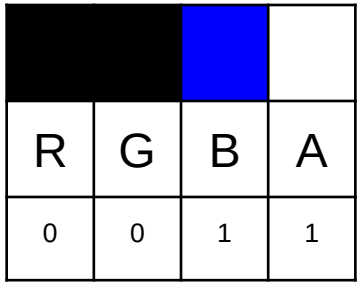
R	G	B	A
1	0	0	0.7

OP = ADD

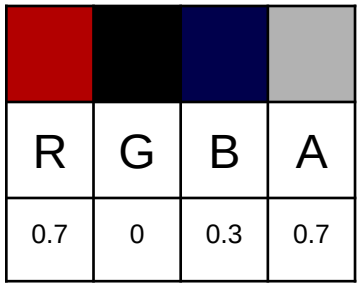
$Factor_{SRC}$ = SrcAlpha

$Factor_{DST}$ = OneMinusSrcAlpha

DST



R	G	B	A
0	0	1	1



R	G	B	A
0.7	0	0.3	0.7

New DST

Interaktivni primeri:
http://www.andersriggelsen.dk/glblendfunc.php