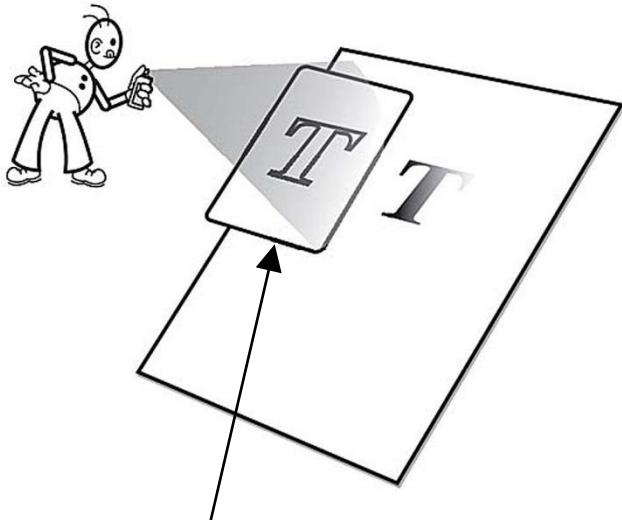
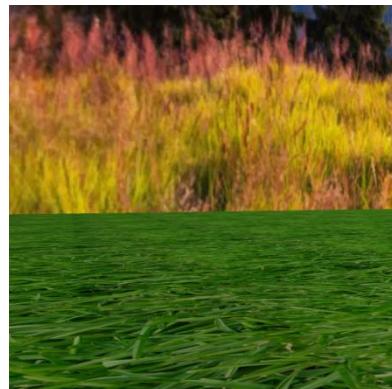


Stensil bafer

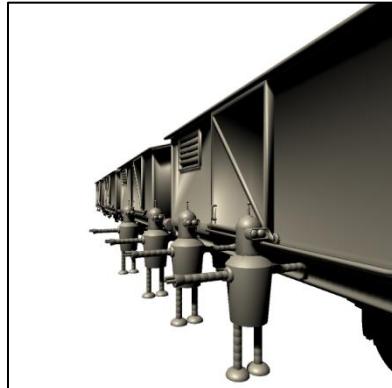


Ovo je "stensil"

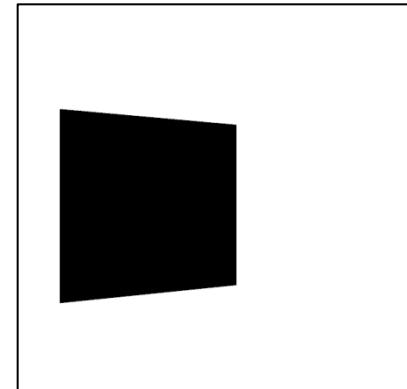
Definicija stensila: tanak karton (...) sa urezanim obrascem, korišćen da se formira specifičan oblik na podlozi ispod.



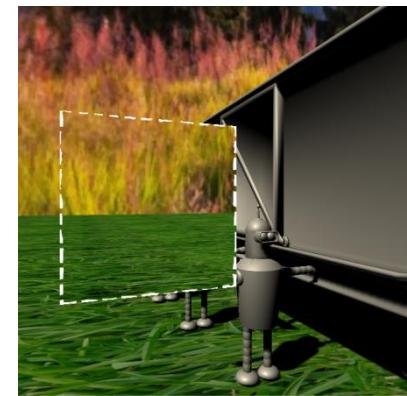
Tekući kolor
bafer



Novo crtanje



Stensil



Rezultat

Jedna od mogućih primena: **shadow volumes**

Stensil (*stencil*) bafer

- *stencil*: šablon za reprodukovanje oblika, najčešće teksta
- stensil bafer
 - slično Z-baferu služi za markiranje specifičnih zona u frejm baferu
 - stencil test se vrši pre crtanja u kolor bafer, čak pre Z-testa
- Veoma korisna alatka, kada je potrebno markirati zone frejm bafera u koje (ne) treba menjati, a Z-test nije dovoljan

Stensil bafer

- **glEnable/glDisable(GL_STENCIL_TEST) ;**
aktivira/deaktivira stencil test
- **void glStencilMask(GLuint mask) ;**
specificira nad kojim bitovima u stensil baferu sme da se vrši izmena

Stensil bafer

- `void glStencilFunc(GLenum func,
 GLint ref,
 GLuint mask);`

Postavlja funkciju poređenja (func) zadate referentne vrednosti (ref) uzimajući u obzir samo one bitove koji su zadati bitskom maskom.

`GL_NEVER, GL_ALWAYS, GL_LESS, GL_EQUAL, GL_EQUAL,
GL_GREQUAL, GL_GREATER, GL_NOTEQUAL.`

- **primer:** ako je funkcija `GL_LESS`, stencil test prolazi ako je ref vrednost manja od vrednosti u stencil baferu (za dati fragment) nakon bitskog maskiranja

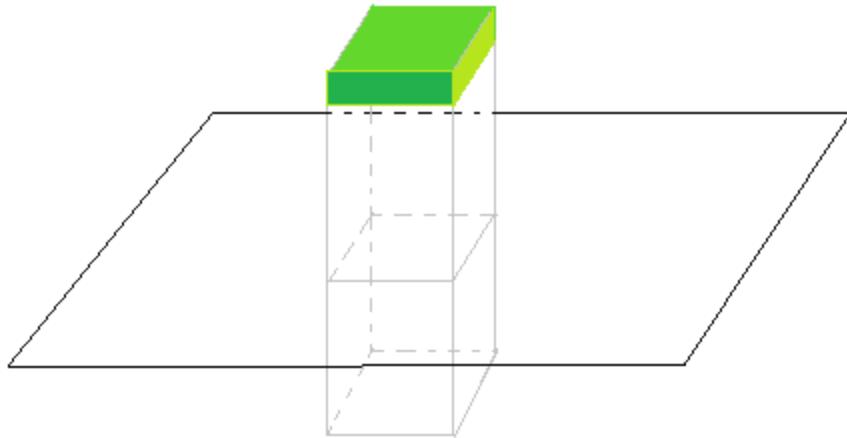
- `void glStencilOp(GLenum fail,
 GLenum zfail,
 GLenum zpass);`

Specificira operaciju nad sadržajem stensil bafera ako ne prođe stensil test (fail), ako prođe stensil ali ne Z-test (zfail) ili ako prođe oba testa (zpass)

`GL_KEEP, GL_ZERO, GL_REPLACE, GL_INCR, GL_DECR, GL_INVERT`

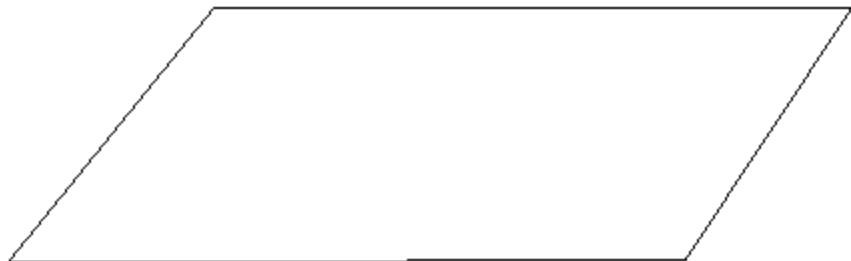
Shadow volumes (1/5)

- Kako nacrtati senku zelene kutije, ukoliko je izvor direkciono svetlo?



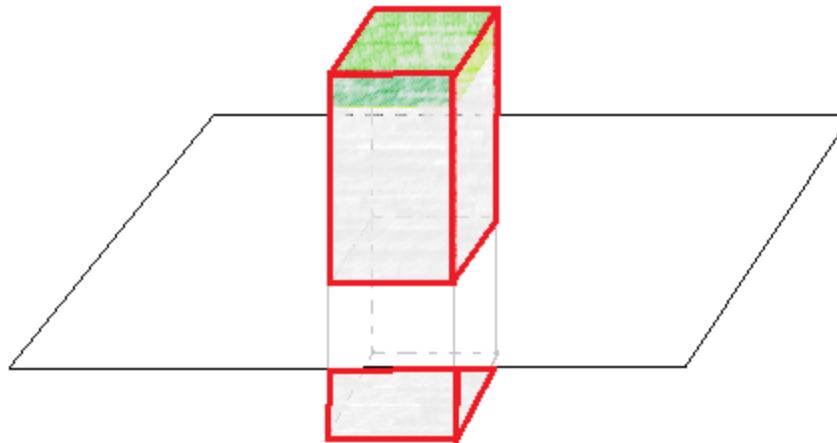
- Potrebno je nacrtati 'zavesu' (sa poklopcima) objekta tako da one seku podlogu na kojoj treba da se pojavi senka
- Prilikom crtanja zavesa utiče se samo na stencil bafer
- Rezultat treba da je stencil bafer sa jedinicama na mestu senke objekta

Shadow volumes (2/5)



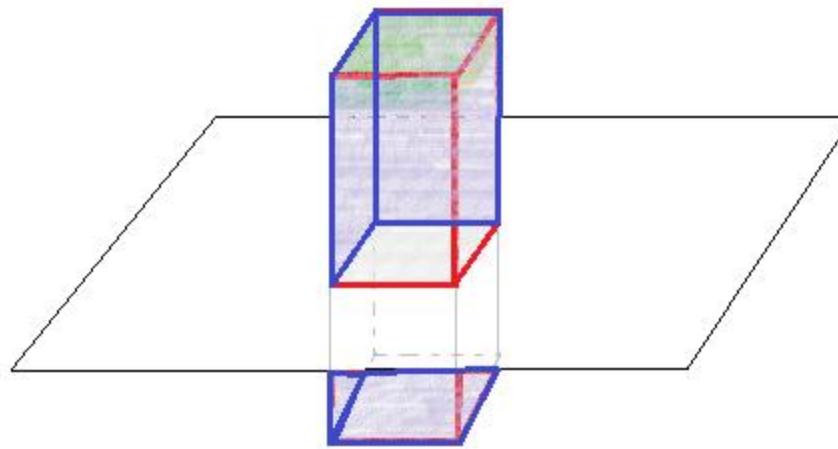
- Crtanje se podloge
- Upis u color i depth bafera

Shadow volumes (3/5)



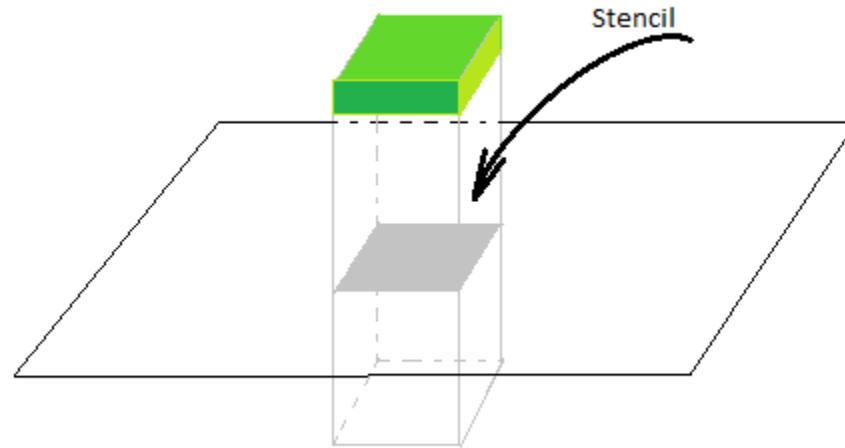
- Crtanje kvadra koji predstavlja zavesu objekta – **prvi prolaz**
- Crtaju se samo prednje stranice (**CCW**)
 - i to oni fragmenti koji nisu zaklonjeni podlogom, tj. koji prolaze depth test
- Fragmenti koji su prošli stencil & depth test utiču na stencil bafer (**incr**)
- Rezultat su **jedinice** u stencil baferu na uokvirenim **crvenim delovima**

Shadow volumes (4/5)



- Crtanje kvadra koji predstavlja zavesu objekta – **drugi prolaz**
- Crtaju se samo prednje stranice (**CW**)
 - i to oni fragmenti koji nisu zaklonjeni podlogom, tj. koji prolaze depth test
- Fragmenti koji su prošli stencil & depth test utiču na stencil bafer (**decr**)
- Rezultat su *ponovo nule* u stencil baferu na delovima uokvirenim plavom bojom

Shadow volumes (5/5)



- Konačni rezultat prethodna dva prolaza su jedinice u stencil baferu, koje odgovaraju obliku i mestu senke objekta
- Da bi se sada nacrtala senka, potrebno je crtati popunjeni pravougaonik preko celog prozora i testirati prolaz fragmenata stencil testom

Pravljenje senke pomoću stensil bafera

```
glCullFace(GL_BACK) ;  
glEnable(GL_CULL_FACE) ;  
glColorMask(GL_FALSE, GL_FALSE, GL_FALSE, GL_FALSE) ;  
glDepthMask(GL_FALSE) ;  
glPolygonMode(GL_FRONT, GL_FILL) ;  
glEnable(GL_STENCIL_TEST) ;  
glStencilFunc(GL_ALWAYS, 1, 0xFFFFFFFFL) ;  
  
/* prvi prolaz */  
glFrontFace(GL_CCW) ;  
glStencilOp(GL_KEEP, GL_KEEP, GL_INCR) ;  
crtajZavese() ;  
  
/* drugi prolaz */  
glFrontFace(GL_CW) ;  
glStencilOp(GL_KEEP, GL_KEEP, GL_DECR) ;  
crtajZavese() ;
```

Pravljenje senke pomoću stensil bafera

```
glFrontFace( GL_CCW );
glColorMask( GL_TRUE, GL_TRUE, GL_TRUE, GL_TRUE );

glColor4f( 0.0f, 0.0f, 0.0f, 0.4f );
 glEnable( GL_BLEND );
 glBlendFunc( GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA );
 glStencilFunc( GL_NOTEQUAL, 0, 0xFFFFFFFFL );
 glStencilOp( GL_KEEP, GL_KEEP, GL_KEEP );

postaviMatriceNaJedinicne();
 glBegin( GL_TRIANGLE_STRIP );
  glVertex2f(-1, 1);
  glVertex2f(-1,-1);
  glVertex2f( 1, 1);
  glVertex2f( 1,-1);
 glEnd();
vratiMatrice();
```