

Kamera u OpenGL-u

- `gluLookAt(eyeX, eyeY, eyeZ, centerX, centerY, centerZ, upX, upY, upZ)` - definiše poziciju, smer gledanja i orijentaciju kamere
- Parametri:
 - eye - pozicija kamere
 - center - tačka u koju kamera gleda
 - up - definiše orijentaciju kamere (šta je gore)
- LookAt funkcija kreira view matrix koja vrši transformaciju verteksa iz World Space u View Space koordinatni sistem

Kamera u OpenGL-u

- LookAt funkcija kreira matricu transformacije koja se može razložiti na matricu rotacije i matricu translacije

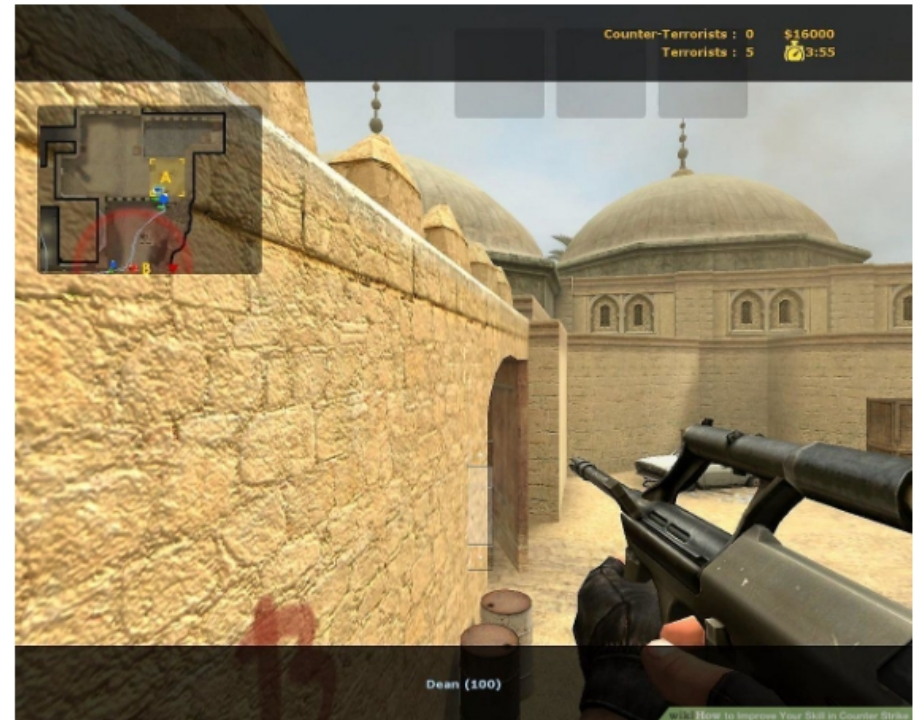
$$M_{\text{view}} = M_R M_T = \begin{pmatrix} r_0 & r_4 & r_8 & 0 \\ r_1 & r_5 & r_9 & 0 \\ r_2 & r_6 & r_{10} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_0 & r_4 & r_8 & r_0 t_x + r_4 t_y + r_8 t_z \\ r_1 & r_5 & r_9 & r_1 t_x + r_5 t_y + r_9 t_z \\ r_2 & r_6 & r_{10} & r_2 t_x + r_6 t_y + r_{10} t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- ```
glRotate(-rotX, 1, 0, 0);
glRotate(-rotY, 0, 1, 0);
glRotate(-rotZ, 0, 0, 1);
glTranslate(-eyeX, -eyeY, -eyeZ);
```

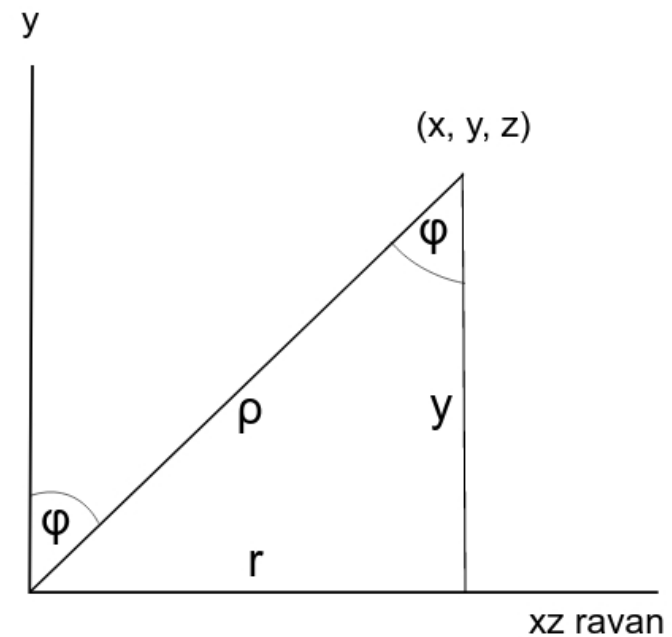
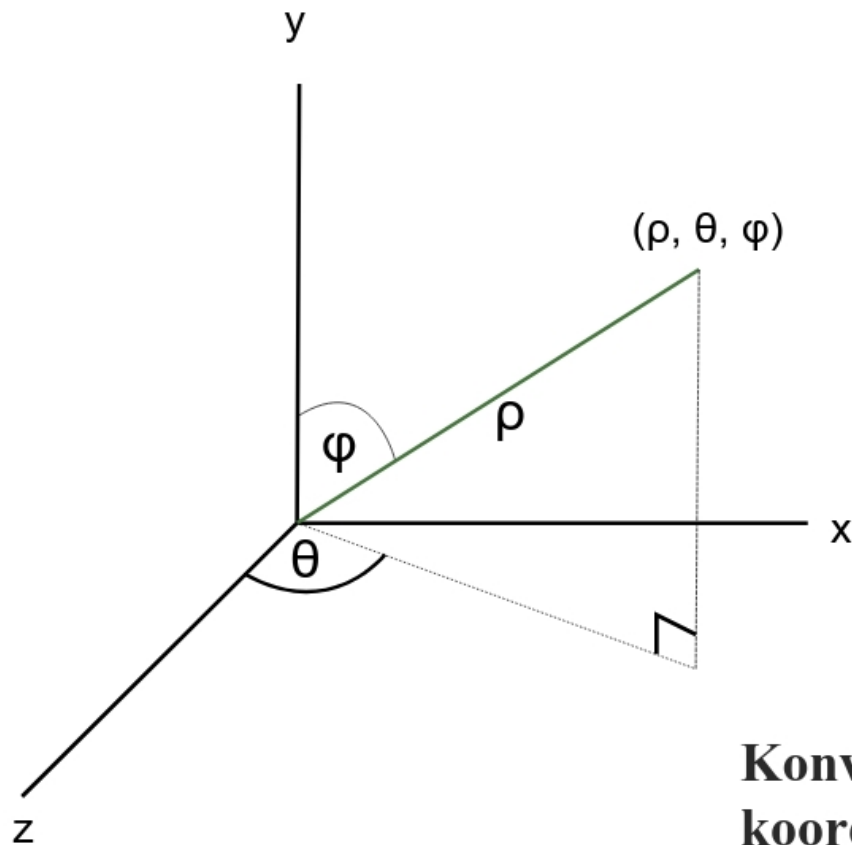
[[http://www.songho.ca/opengl/gl\\_camera.html](http://www.songho.ca/opengl/gl_camera.html)]

# FPS Camera

- First Person Shoot camera
- Rotiramo kameru pomeranjem miša
- Pozicija kamere se menja pritiskom tastera na tastaturi(WASD, strelice)
- Česta pojava u game developmentu



# Sferični koordinatni sistem



**Konverzija u Kartezijev  
koordinatni sistema:**

$$x = \rho \sin \phi \cos \theta$$

$$y = \rho \cos \phi$$

$$z = \rho \sin \phi \sin \theta$$

# Animacija

- Tehnika prikazivanja više uzastopnih slika (frejmova) u pokušaju "oživljanja" scene
- **DispatcherTimer class**

A timer that is integrated into the Dispatcher queue which is processed at a specified interval of time and a specified priority.

[<https://docs.microsoft.com/en-us/dotnet/api/system.windows.threading.dispatchertimer?redirectedfrom=MSDN&view=net-5.0>]

# Boja verteksa

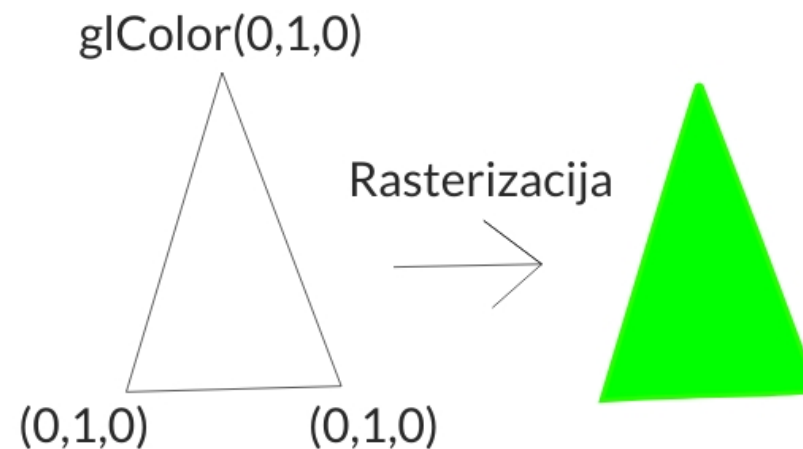
- glColor(float r, float g, float b);
- Materijali
- Shading modeli
- Modeli osvetljenja (normale)
- Teksture

# Shading modeli

- `void glShadeModel(GLenum mode)`
- `mode:`
  - `GL_FLAT`
  - `GL_SMOOTH`
- Definiše način određivanja boja piksela prilikom rasterizacije

# Flat shading

- Konstantno senčenje
- Prilikom rasterizacije flat shaded poligona uzima boju jednog od verteksa poligona i dodeljuje tu boju svim pikselima/fragmentima rasterizovanog poligona

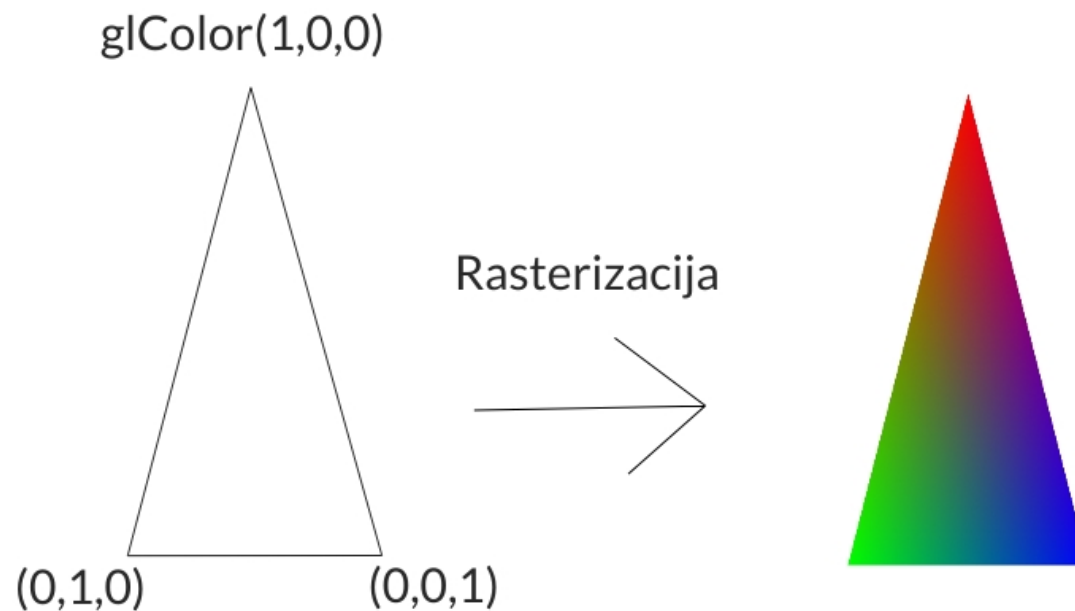


[<https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/glShadeModel.xml>]



# Smooth shading

- Prilikom rasterizacije poligona boje piksela se određuju interpolacijom između boja verteksa



# Reflektivna svojstva objekata/materijala

- Svaki objekat, odnosno materijal od kog je sačinjen taj objekat, ima reflektivna svojstva
- Objekat/materijal delom apsorbuje svetlosne zrake, delom ih odbija
- Odbijeni zraci svetlosti od objekta koji stižu do našeg oka predstavljaju boju kojom ćemo videti taj objekat
- Kako bi dobili realističnije scene, potrebno je da simuliramo interakciju svetlosti i materijala objekata

# Materijali u OpenGL-u

- Materijali definišu reflektivna svojstva objekata
- Potrebno je definisati model kojim će se simulirati različiti materijali kroz interakciju sa modelom osvetljenja
- Komponente koje čine model materijala:
  - **Ambijentna**
  - **Difuzna**
  - **Spekularna**
  - **Shininess**
- Svaka od komponenti interaguje sa odgovarajućom komponentom svetlosnog modela
- Primeri materijala:  
[<http://devernay.free.fr/cours/opengl/materials.html>]

# Materijali u OpenGL-u - glMaterial

● **void glMaterial(GLenum face, GLenum pname, const GLfloat \* params)**

- **face** - GL\_FRONT, GL\_BACK, GL\_FRONT\_AND\_BACK
- **pname** - GL\_AMBIENT, GL\_DIFFUSE, GL\_SPECULAR, GL\_EMISSION, GL\_SHININESS, GL\_AMBIENT\_AND\_DIFFUSE
- **params** - definiše vrednost/boju dodeljenu komponentama

<https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/glMaterial.xml>

# Materijali u OpenGL - Color Tracking

- Color Tracking mehanizam omogućuje da boja određene komponente materijala bude određena sa glColor() pozivom

**void glColorMaterial(GLenum face, GLenum mode)**

- face - GL\_FRONT, GL\_BACK, GL\_FRONT\_AND\_BACK
  - mode - GL\_AMBIENT, GL\_DIFFUSE, GL\_SPECULAR, GL\_AMBIENT\_AND\_DIFFUSE
- Olakšava definisanje materijala objekata (na nivou verteksa) u odnosu na glMaterial() metodu