

# Pitanja

---

1. Što je to **volumen** pogleda?
  - Prostor koji je vidljiv iz pogleda. Točke unutar njega će se projicirati na platno, a one koje su izvan će biti zanemarene.
2. Kako je određen **inicijalni** volumen pogleda kod OpenGL-a?
  - Inicijalni volumen kod OpenGLa je određen sljedećim rasponom po osima:  $([-1, 1], [-1, 1], [-1, 1])$
3. U kakvoj su vezi volumen pogleda i pojam odsijecanja? Što OpenGL odsijeca? Dajte primjer.
  - Ae..
4. Napišite matricu operatora translacije za  $(dx, dy, dz)$  uz obje konvencije. Kako glase matrice koje točku vraćaju u početnu?
  - TLDR:
    - Vertex x TranslationMatrix
    - TranslationMatrix x Vertex
  - Da bi se vratila točka u početnu, potrebno je pomnožiti s inverznom matricom. Inverzna matrica na mjestu dx, dy i dz ima negirane vrijednosti.
5. Napišite matricu operatora skaliranja s faktorima sx, sy i sz uz obje konvencije. Kako glase matrice koje točku vraćaju u početnu?
  - Neću crtati matrice, ae..
  - Inverzne imaju  $1/sx$ ,  $1/sy$  i  $1/sz$
6. Je li transformacija pogleda jednoznačno definirana zadavanjem točke očišta i gledišta? Objasnite.
  - Nije. time je samo definiran smjer z osi koordinatnog sustava kamere. Da bi se odredili smjer i orijentacija vektora y odnosno x, potrebno je definirati takozvani **view up** vektor. On kaže gdje je smjer "gore" za kameru. View up vektor ne mora biti u ravnini XY, zajedno s jediničnim vektorom osi z definira YZ ravninu na temelju koje se odrede jedinični vektori ostalih osi.
7. Što je to **view-up** vektor i čemu služi? Leži li on u ravnini projekcije?
  - Pitanje ^. Nije nužno u ravnini projekcije (XY), dovoljno je da zajedno sa jediničnim vektorom z osi definira ravninu YZ.
8. OpenGL nudi naredbu `gluLookAt`. Čemu služi ta naredba, kakvu transformaciju pogleda radi, koji su njezini argumenti i kako izgleda pripadna matrica, uz obje konvencije (matricu ne trebate učiti napamet)?
  - `void gluLookAt(GLdouble eyex, GLdouble eyey, GLdouble eyez, GLdouble centerx, GLdouble centery, GLdouble centerz, GLdouble vupx, GLdouble vupy, GLdouble vupz);`
  - Naredba omogućuje jednostavno definiranje transformacije pogleda zadavanjem očišta (eye), gledišta (center) i view-up vektora (vup). Ae..
9. OpenGL nudi naredbu `glFrustum`. Čemu služi ta naredba, kako izgleda volumen pogleda koji definira, koji su njezini argumenti i kako izgleda pripadna matrica, uz obje konvencije (matricu ne trebate učiti napamet)?
  - `void glFrustum (GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);`
  - Volumen izgleda kao krnja piramida. Naredba omogućuje jednostavno definiranje volumena pogleda kod perspektivne projekcije. Ae..

10. OpenGL nudi naredbu `gluPerspective`. Čemu služi ta naredba, kako izgleda volumen pogleda koji definira, koji su njezini argumenti i kako izgleda pripadna matrica, uz obje konvencije (matricu ne trebate učiti napamet)?
- Naredba je još jedan način definiranja matrice perspektivne projekcije (prvi način je naredba u pitanju iznad ^). Prototip naredbe: `void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble near, GLdouble far);`
  - **fovy** (engl. field-of-view) je kut u stupnjevima
  - **aspect** određuje omjer širine i visine, aka:  $w = aspect * h$
11. OpenGL nudi naredbu `glViewport`. Čemu služi ta naredba i koji su njezini argumenti?
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height);`
  - Proćiri, proviri
  - Consider the following examples:
    - the whole window: `glViewport(0, 0, (GLsizei) width, (GLsizei) height);`
    - the upper right quarter of the window: `glViewport((GLsizei) width / 2, (GLsizei) height / 2, (GLsizei) width / 2, (GLsizei) height / 2);`
    - the lower left quarter of the window: `glViewport(0, 0, (GLsizei) width / 2, (GLsizei) height / 2);`
    - position the image so that it occupies a quarter of the window but is placed in the center of the window: `glViewport((GLsizei) width / 4, (GLsizei) height / 4, (GLsizei) width / 2, (GLsizei) height / 2);`
12. Uporabom OpenGL naredbe `glMatrixMode` do sada ste odabirali dva moda: **GL\_PROJECTION** i **GL\_MODEL**. Što se podešava u jednom modu a što u drugom?
- U modu **GL\_PROJECTION** se podešava matrica projekcije, dok se u **GL\_MODEL** odnosno **GL\_VIEWMODEL** podešava matrica modela u globalnom koordinatnom sustavu.

by m43 on 01. 05. 2020