

## RAČUNARSKA GRAFIKA drugi kolokvijum

- 1) [25] **OpenGL:** Napisati na jeziku C ili C++ deo programa za crtanje scene prikazane na slici 1 primenom grafičke biblioteke OpenGL. Scena se sastoji od jednostavne jednakokrake vage. Vaga ima postolje kvadratne osnove, a telo vage čine kvadri kvadratnog poprečnog preseka. Jezičak vage je jednakokraki trougao. Na tasovima se nalaze tegovi cilindričnog oblika različitih težina (tamniji teg je teži), zbog čega je vaga nagnuta pod uglom od  $10^\circ$  u odnosu na horizontalnu ravan. Scena se crta na beloj podlozi. Koristi se ortografska projekcija, a vidljiva zapremina je definisana u sledećem opsegu:  $x \in [-5, 5]$ ,  $y \in [-5, 5]$ ,  $z \in [0, 30]$ , pri čemu postolje vage leži u ZoX ravni. Kameru postaviti tako da se formira prikaz koji približno odgovara slici 1. Dimenzije objekata izabrati da približno odgovaraju slici 1. Napisati posebnu funkciju koja vrši osnovnu inicijalizaciju OpenGL sistema potrebnu za crtanje scene. Smatrati da je otvaranje prozora za crtanje realizovano u glavnom programu koji nije potrebno pisati. **Napomena:** rezultujuća slika ne sme da zavisi od redosleda crtanja elemenata scene.



slika 1

- 2) [25] Postaviti jednu matičnu jednačinu koja određuje projekciju sa perspektivom iz centra projekcije  $P(5\sqrt{2}, 10, 5\sqrt{2})$  na projekcionu ravan  $x+z = -3\sqrt{2}$  slike proizvoljne tačke (u desnom 3D koordinatnom sistemu) u ogledalu postavljenom na ravan  $x + \sqrt{2}y + z = 2\sqrt{2}$ . U sve matice elementarnih transformacija koje učestvuju u matičnoj jednačini uvrstiti konkretne vrednosti. Nije potrebno množiti matrice.
- 3) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Navesti sekvencu elementarnih transformacija (opisno, navodeći ciljeve) kojima se određuje rotacija oko proizvoljne ose. Nije potrebno pisati matrice ni računati njihove elemente.
  - Definisati pojam iščezavajuće tačke.
  - Napisati matičnu jednačinu za ortografsku projekciju tačke.
  - Šta se postiže prikaznim transformacijama, a šta geometrijskim modelirajućim transformacijama u sistemu SPHIGS? Kojim redosledom se izvršavaju?
  - Kako se u sistemu SPHIGS ostvaruje hijerarhija struktura?

### Napomene:

- Kolokvijum traje 120 minuta.
- Nije dozvoljena upotreba literature niti programabilnih kalkulatora.
- Dozvoljena je upotreba OpenGL podsetnika.

# Rešenja zadataka

## drugi kolokvijum 2010

### 1) Rešenje

```
void crtajCilindar(float *boja1, float *boja2) {
    glPushMatrix();
    glScaled(0.5, 1, 0.5);   glColor3fv(boja1);
    double korak = 360./BROJ_SEGMENATA;
    glBegin(GL_QUAD_STRIP);
        for(double phi = 0; phi <= 360; phi += korak ) {
            glVertex3d( cos( DEG_2_RAD(phi) ), -0.5, sin( DEG_2_RAD(phi) ) );
            glVertex3d( cos( DEG_2_RAD(phi) ), 0.5, sin( DEG_2_RAD(phi) ) );
        }
    glEnd();
    glColor3fv(boja2);   glTranslated(0,-0.5,0);
    glBegin(GL_TRIANGLE_FAN);
        glVertex3d(0,0,0);
        for(double phi = 0; phi <= 360; phi += korak ) {
            glVertex3d( cos( DEG_2_RAD(phi) ), 0, sin( DEG_2_RAD(phi) ) );
            glVertex3d( cos( DEG_2_RAD(phi) ), 0, sin( DEG_2_RAD(phi) ) );
        }
    glEnd();
    glTranslated(0,1,0);   glRotated(180,1,0,0);
    glBegin(GL_TRIANGLE_FAN);
        glVertex3d(0,0,0);
        for(double phi = 0; phi <= 360; phi += korak ) {
            glVertex3d( cos( DEG_2_RAD(phi) ), 0, sin( DEG_2_RAD(phi) ) );
            glVertex3d( cos( DEG_2_RAD(phi) ), 0, sin( DEG_2_RAD(phi) ) );
        }
    glEnd();
    glPopMatrix();
}

void crtajKocku(float *boja1, float *boja2, float *boja3) {
    glBegin(GL_QUADS);
        glColor3fv(boja1);
        glVertex3d( -0.5, 0.5, 0.5 );   glVertex3d( -0.5, -0.5, 0.5 );
        glVertex3d( 0.5, -0.5, 0.5 );   glVertex3d( 0.5, 0.5, 0.5 );
        glColor3fv(boja2);
        glVertex3d( 0.5, 0.5, 0.5 );   glVertex3d( 0.5, -0.5, 0.5 );
        glVertex3d( 0.5, -0.5, -0.5 );   glVertex3d( 0.5, 0.5, -0.5 );
        glColor3fv(boja1);
        glVertex3d( 0.5, 0.5, -0.5 );   glVertex3d( 0.5, -0.5, -0.5 );
        glVertex3d( -0.5, -0.5, -0.5 );   glVertex3d( -0.5, 0.5, -0.5 );
        glColor3fv(boja2);
        glVertex3d( -0.5, 0.5, -0.5 );   glVertex3d( -0.5, -0.5, -0.5 );
        glVertex3d( -0.5, -0.5, 0.5 );   glVertex3d( -0.5, 0.5, 0.5 );
    glEnd();
    glColor3fv(boja3);
    glBegin(GL_QUADS);
        glVertex3d(-0.5, -0.5, 0.5);   glVertex3d(-0.5, -0.5, -0.5);
        glVertex3d(0.5, -0.5, -0.5);   glVertex3d(0.5, -0.5, 0.5);
        glVertex3d(-0.5, 0.5, 0.5);   glVertex3d(0.5, 0.5, 0.5);
        glVertex3d(0.5, 0.5, -0.5);   glVertex3d(-0.5, 0.5, -0.5);
    glEnd();
}

void crtajTasSaTegom(double tezina, float *boja1, float *boja2) {
    glColor3f(0,0,0);
    glBegin(GL_LINES);
        glVertex3f(0,0,0);   glVertex3f(-1, -3, 1);
        glVertex3f(0,0,0);   glVertex3f(1, -3, 1);
        glVertex3f(0,0,0);   glVertex3f(1, -3, -1);
        glVertex3f(0,0,0);   glVertex3f(-1, -3, -1);
    glEnd();
    glColor3fv(boja1);
    glBegin(GL_QUADS);
        glVertex3f(-1, -3, 1);   glVertex3f(1, -3, 1);
        glVertex3f(1, -3, -1);   glVertex3f(-1, -3, -1);
    glEnd();
}

void crtajVagu() {
    static float crna[] = { 0.f, 0.f, 0.f };
    static float tamno_siva[] = { 0.3f, 0.3f, 0.3f };
    static float siva[] = { 0.5f, 0.5f, 0.5f };
    static float svetlo_siva[] = { 0.7f, 0.7f, 0.7f };
    glPushMatrix();
        glScaled(3, 0.1, 3);
        crtajKocku(tamno_siva, svetlo_siva, siva);
    glPopMatrix();
    glPushMatrix();
        glTranslated(0,3,0);   glScaled(0.2, 6, 0.2);
        crtajKocku(tamno_siva, svetlo_siva, siva);
    glPopMatrix();
    glPushMatrix();
        glTranslated(0,6,0);   glRotated(10,0,0,1);
        glPushMatrix();
            glScaled(8, 0.2, 0.2);
            crtajKocku(tamno_siva, svetlo_siva, siva);
        glPopMatrix();
        glPushMatrix();
            glTranslated(4, 0, 0);   glRotated(-10,0,0,1);
            crtajTasSaTegom(0.5, svetlo_siva, siva);
        glPopMatrix();
        glPushMatrix();
            glTranslated(-4, 0, 0);   glRotated(-10,0,0,1);
            crtajTasSaTegom(0.8, svetlo_siva, tamno_siva);
        glPopMatrix();
        glColor3fv(siva);   glTranslated(0, 0, 0.15);
        glBegin(GL_TRIANGLES);
            glVertex3f(-0.2f, 0, 0);
            glVertex3f(0, -3, 0);
            glVertex3f(0.2f, 0, 0);
        glEnd();
    glPopMatrix();
}

void crtajScenu(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();   gluLookAt(-12, 10, 15, 0, 3, 0, 0, 1, 0);
    crtajVagu();
    glFlush();
}

void init() {
    glPolygonMode(GL_FRONT, GL_FILL);
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glEnable(GL_DEPTH_TEST);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();   glOrtho(-5, 5, -5, 5, 0, 30);
    glMatrixMode(GL_MODELVIEW);   glLoadIdentity();
}
```

$$2) \begin{aligned} [X' \ Y' \ Z' \ W] &= [X \ Y \ Z \ 1] * M \\ [X'' \ Y'' \ Z'' \ 1] &= \begin{bmatrix} X' & Y' & Z' \\ W & W & W \end{bmatrix} \end{aligned}$$

$$M = \text{Rotacija1} * \text{Translacija1} * \text{Rotacija2} * \text{Ogledanje} * \text{Rotacija3} * \text{Translacija2} * \text{Projekcija} * \text{Translacija3} * \text{Rotacija4}$$

$$\text{Rotacija1} = \text{Ry}\left(\frac{\pi}{4}\right) = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Translacija2} = \text{T}(0, 10, -5) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -10 & 5 & 1 \end{bmatrix}$$

$$\text{Translacija1} = \text{T}(0, 0, 2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

$$\text{Projekcija} = \text{Pp}(0, 0, 13) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{13} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotacija2} = \text{Rx}\left(-\frac{\pi}{4}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Translacija3} = \text{T}(0, -10, 3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 10 & -3 & 1 \end{bmatrix}$$

$$\text{Ogledanje} = \text{Oxy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotacija4} = \text{Ry}\left(-\frac{\pi}{4}\right) = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotacija3} = \text{Rx}\left(\frac{\pi}{4}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$