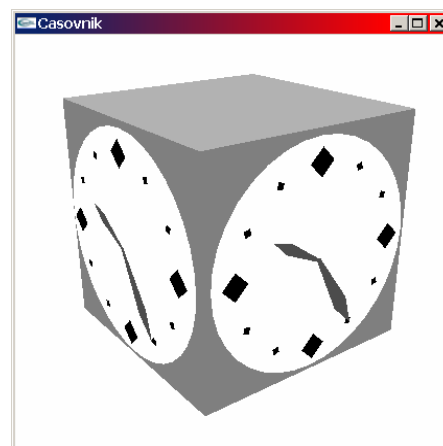


RAČUNARSKA GRAFIKA

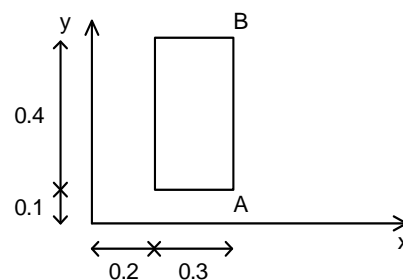
drugi kolokvijum

1) [25] **OpenGL:** Napisati na jeziku C ili C++ deo programa za crtanje scene prikazane na slici 1 primenom grafičke biblioteke OpenGL. Scenu čini telo oblika kocke na čijim su bočnim stranama raspoređeni časovnici. Kocka je obojena sivom, s tim što su gornja i donja strana svetlije od bočnih strana. Oznake sati na časovniku su oblika romba, crne boje, rotirane tako da jedno teme bude okrenuto ka centru datog časovnika. Oznake za 12, 3, 6 i 9 sati su 3 puta većih dimenzija od ostalih. Časovnik ima kazaljke za sate i minute, obojene tamno sivom. Ugao koji kazaljke zaklapaju sa vertikalnom osom zadaju se kao parametar prilikom crtanja svakog časovnika. Koristi se projekcija sa perspektivom. Napisati posebnu funkciju koja vrši osnovnu inicijalizaciju OpenGL sistema potrebnu za crtanje scene. Posmatračku kameru približno postaviti u položaj koji bi proizveo prikazanu sliku. 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] Posmatraju se objekti u desnom 3D koordinatnom sistemu. Vršni se projekcija sa perspektivom za posmatrača koji se nalazi u tački $P(-1, 0, 0)$ na ravan $z = 5 - x$. Delovi projekcije objekata koji se nalaze u prozoru definisanom sa $x \in [-2.5, 2.5]$, $y \in [-2.5, 2.5]$ se potom preslikavaju u prikazni prozor definisan slikom 2, tako da se stranica prozora $y = -2.5$ preslika u stranicu AB, pri čemu se tačka prozora $(-2.5, -2.5)$ preslikava u tačku B. Postaviti matičnu jednačinu preslikavanja iz 3D prostora u zadati prikazni prozor. U sve matrice elementarnih transformacija uvrstiti konkretne vrednosti. Nije potrebno množiti matrice.



slika 2

- 3) [50] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Čime je određen broj osnih iščezavajućih tačaka?
 - Kojim projekcijama pripada (navesti celu klasifikacionu putanju) i kako je definisana izometrijska projekcija?
 - Kako se definišu i čemu su namenjene prednja i zadnja odsecajuća ravan u SPHIGS-u?
 - Ako se u SPHIGS strukturi sa više elemenata nalazi i poliedar koji definiše kuću sa krovom na dve vode (dva paralelna petougla čija su korespondentna temena spojena ivicama), kako se model kuće može promeniti tako da krov ima četiri vode?
 - Da li je vrednost SPHIGS pik-identifikatora modalna u strukturi i da li se nasleđuje u podstrukturi?

Napomene: 1. Kolokvijum traje 120 minuta.

- Rad se predaje isključivo u vežbanci za ispite. Nije dozvoljeno imati pored sebe druge listove papira, niti uz sebe imati mobilni telefon, bez obzira da li je uključen ili isključen.
- Nije dozvoljena upotreba literature niti programabilnih kalkulatora.
- Dozvoljena je upotreba OpenGL podsetnika.
- Voditi računa o urednosti. Nečitki delovi teksta će biti tretirani kao nepostojeći. Rešenja zadataka navesti po gornjem redosledu (-1 poen za loš redosled). Preporučuje se rad običnom grafitnom olovkom.

Rešenja zadataka

drugi kolokvijum 2014

1) Rešenje

```
#include <stdlib.h>
#include <GL/glut.h>
#include <GL/GL.H>
#define STEP_U_RAD(x) ((x)*PI/180)
const float crna[] = { 0, 0, 0 };
const float tamno_siva[] = { 0.3f, 0.3f, 0.3f };
const float siva[] = { 0.5f, 0.5f, 0.5f };
const float svetlo_siva[] = { 0.7f, 0.7f, 0.7f };
float rotSati, rotMinuta;
void crtajKvadrat(const float *boja) {
    const float a = 0.5f;
    glPushAttrib(GL_ALL_ATTRIB_BITS);
    glBegin(GL_QUADS);
    glColor3fv(boja);
    glVertex2f(+a, +a);    glVertex2f(+a, -a);
    glVertex2f(-a, -a);    glVertex2f(-a, +a);
    glEnd();
    glPopAttrib();
}
void crtajKrug(const float *boja, int brSeg) {
    const float r = 0.5f;
    glPushMatrix();
    glPushAttrib(GL_ALL_ATTRIB_BITS);
    glColor3fv(boja);
    glBegin(GL_TRIANGLE_FAN);
    float korak = 360.0f / brSeg;
    glVertex2f(0,0);
    for(int i = 0; i < brSeg+1; i++) {
        float x = r * cos( STEP_U_RAD(i*korak) );
        float y = r * sin( STEP_U_RAD(i*korak) );
        glVertex2f(x, y);
    }
    glEnd();
    glPopAttrib();
    glPopMatrix();
}
void crtajRomb() {
    glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(0,-1);    glVertex2f(1,0);
    glVertex2f(-1,0);    glVertex2f(0,1);
    glEnd();
}
void crtajKazaljku(const float *boja) {
    glPushAttrib(GL_ALL_ATTRIB_BITS);
    glColor3fv(boja);
    glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(-0.01f, -0.01f);    glVertex2f(0.01f, -0.01f);
    glVertex2f(-0.04f, 0.3f);    glVertex2f(0.04f, 0.3f);
    glVertex2f(0.0f, 0.45f);
    glEnd();
    glPopAttrib();
}
void crtajOznake(const float *boja) {
    glPushMatrix();
    glPushAttrib(GL_ALL_ATTRIB_BITS);
    glColor3fv(boja);
    for(int i = 0; i < 12; i++) {
        glPushMatrix();
        glRotatef(-30*i, 0, 0, 1);    glTranslatef(0, 0.4f, 0);
        glScalef(0.02f, 0.02f, 1);
        if( i % 3 == 0 )                glScalef(3.0f, 3.0f, 1);
        crtajRomb();
        glPopMatrix();
    }
}
void crtajStranuSata(float rotSati, float rotMinuta) {
    glPushAttrib(GL_ALL_ATTRIB_BITS);
    glPushMatrix();
    crtajKvadrat(siva);
    glTranslatef(0,0,0.01f);    glScalef(0.95f, 0.95f, 1);
    crtajKrug(bela, 36);
    glTranslatef(0,0,0.01f);
    crtajOznake(crna);
    glTranslatef(0,0,0.01f);
    glPushMatrix();
    glRotatef(rotSati, 0, 0, 1);    glScalef(0.6f, 0.6f, 1);
    crtajKazaljku(tamno_siva);
    glPopMatrix();
    glRotatef(rotMinuta, 0, 0, 1);    glScalef(0.9f, 0.9f, 1);
    crtajKazaljku(tamno_siva);
    glPopMatrix();
    glPopAttrib();
}
void crtajSat(float rotSati, float rotMinuta) {
    const float a = 0.5f;
    glPushAttrib(GL_ALL_ATTRIB_BITS);
    glPushMatrix();
    glScalef(9.5f, 9.5f, 9.5f);    glTranslatef(0,0,a);
    crtajStranuSata(rotSati, rotMinuta);
    for(int i = 0; i < 3; i++) {
        glTranslatef(a, 0, -a);
        glRotatef(90, 0, 1, 0);
        crtajStranuSata(rotSati, rotMinuta);
    }
    glTranslatef(0, a, -a);
    glRotatef(-90, 1, 0, 0);
    crtajKvadrat(svetlo_siva);
    glTranslatef(0, 0, -2*a);    glRotatef(-180, 0, 1, 0);
    crtajKvadrat(svetlo_siva);
    glPopMatrix();
    glPopAttrib();
}
void crtajScenu(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(20, -10, 15, 0, 0, 5, 0, 0, 1);
    crtajSat(rotSati, rotMinuta);
    glFlush();
}
void init() {
    glClearColor (1.0, 1.0, 1.0, 0.0);
    glEnable(GL_DEPTH_TEST);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

2) Rešenje

$$[X' \ Y' \ W] = [X \ Y \ Z \ 1] * M$$

$$[X'' \ Y'' \ 1] = \begin{bmatrix} X' & Y' & 1 \\ W & W & 1 \end{bmatrix}$$

$$M = \text{Translacija1} * \text{Rotacija1} * \text{Projekcija} * \text{Translacija2} * \text{Skaliranje} * \text{Rotacija2} * \text{Translacija3}$$

$$\text{Translacija1} = T(2,0,3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 0 & -3 & 1 \end{bmatrix}$$

$$\text{Rotacija1} = R_y\left(-\frac{3\pi}{4}\right) = \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Projekcija} = P_p(0,0,3\sqrt{2}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3\sqrt{2}} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Translacija2} = T(-2.5,-2.5) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2.5 & 2.5 & 1 \end{bmatrix}$$

$$\text{Skaliranje} = S\left(-\frac{0.4}{5}, \frac{0.3}{5}\right) = \begin{bmatrix} -0.8 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$\text{Translacija3} = T(-0.5,-0.5) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.5 & 0.5 & 1 \end{bmatrix}$$