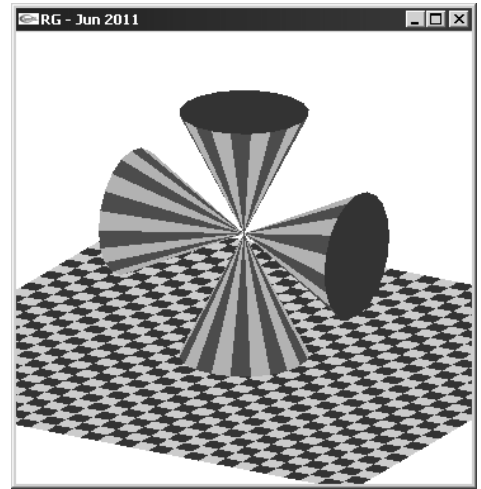


ISPIT IZ RAČUNARSKE GRAFIKE

- 1) [30] **OpenGL:** Napisati na jeziku C ili C++ deo programa za crtanje scene prikazane na slici 1 primenom grafičke biblioteke OpenGL. Scena se sastoji od figure postavljene iznad podloge sa tamnim i svetlim poljima, poput šahovske table. Figura se sastoji od četiri kupa čiji se vrhovi dodiruju (kao na slici). Omotači i osnove kupa su sastavljeni od trouglova. Kod omotača kupa, trouglovi se naizmenično boje svetlijim i tamnijim nijansama sive. Osnove kupa su jednobojne. Scena se crta na beloj podlozi. Koristi se ortografska projekcija. Napisati posebnu funkciju koja vrši osnovnu inicijalizaciju OpenGL sistema potrebnu za crtanje scene. Parametre projekcije podesiti i 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.



- 2) [20] Korišćenjem Bresenham-ovog algoritma za crtanje kružnice kao osnove, potprograma za crtanje linije `Line(xa, ya, xb, yb)` koji crta liniju od tačke (x_a, y_a) do tačke (x_b, y_b) i potprograma za crtanje tačke `Plot(x, y)` koji crta tačku (x, y) , napisati potprogram čija je deklaracija

`void Crtaj(int x1, int y1, int x2, int y2, bool mod);`

gde *mod* određuje da li će se crtati figura 2a ili figura 2b, na slici 2. *mod* je *true* za figuru 2b. Tačka (x_1, y_1) predstavlja donji levi, a (x_2, y_2) gornji desni ugao figure. Pretpostaviti da je pre poziva potprograma izvršena provera validnosti datih koordinata. Crtanje je crnom olovkom na beloj pozadini.

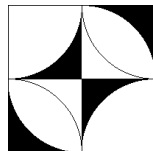


figura 2a

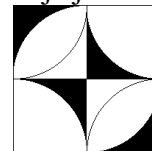


figura 2b

slika 2

- 3) [20] Rekurzivni algoritmi popunjavanja oblasti.
- 4) [30] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Koje razlike postoje između SPHIGS lokatora i SRGP lokatora?
 - Kada bi se ideja *Cohen-Sutherland* algoritma primenila u 3D prostoru, sa koliko bita bi se kodirali položajni kodovi i kako bi izledali uslovi trivijalnog prihvatanja i odbacivanja?
 - Kojom instrukcijom se završava sekvenca instrukcija (programa) za opis slike u memoriji (displej-baferu) vektorskog prikazivača i zašto?

Napomene:

- Ispit traje 180 minuta.
- Nije dozvoljena upotreba literature niti programabilnih kalkulatora.
- Dozvoljena je upotreba OpenGL podsetnika.

Rešenja zadataka

Jun 2011

1) Rešenje

```
#include <stdio.h>
#include <math.h>
#include "GL/glut.h"
#include "GL/GL.H"

#define DEG_2_RAD(x)    (x*3.14159265358979/180)

float tamno_siva[] = {0.2f, 0.2f, 0.2f};
float svetlo_siva[] = {0.8f, 0.8f, 0.8f};

void promenaProzora(int width, int height) {
    // Postavljanje viewport-a
    glViewport(0, 0, (GLint) width, (GLint) height);
}

void crtajKupu(int brSeg) {
    float korak = 360.f / brSeg;
    float xp = 0.5, zp = 0;
    float tekUgao = korak;
    for(int i = 0; i <= brSeg; i++) {
        float x = 0.5*cos( DEG_2_RAD(tekUgao) );
        float z = 0.5*sin( DEG_2_RAD(tekUgao) );

        // omotac
        float intenzitet = 0.3f + 0.4f * (i&1);
        glColor3f(intenzitet, intenzitet, intenzitet);
        glBegin(GL_TRIANGLES);
            glVertex3f(0, 0, 0);
            glVertex3f(xp, -1, zp);
            glVertex3f(x, -1, z);
        glEnd();

        // osnova
        glColor3f(0.2f, 0.2f, 0.2f);
        glBegin(GL_TRIANGLES);
            glVertex3f(0, -1, 0);
            glVertex3f(xp, -1, zp);
            glVertex3f(x, -1, z);
        glEnd();
        tekUgao += korak;
        xp = x; zp = z;
    }
}

void crtajFiguru() {
    glPushMatrix();
    crtajKupu(20);
    glScalef(1, -1, 1);
    crtajKupu(20);

    glRotatef(90, 1, 0, 0);
    crtajKupu(20);
    glScalef(1, -1, 1);
    crtajKupu(20);
    glPopMatrix();
}
```

```
void crtajPodlogu(int vrsta, float sFactor) {
    glPushMatrix();
    glScalef(sFactor, 1, sFactor);
    glTranslatef(-vrsta/2, 0, -vrsta/2);
    for(int i = 0; i < vrsta; i++)
        for(int j = 0; j < vrsta; j++) {
            if( ((i+j)&1) == 0 )
                glColor3fv(tamno_siva);
            else
                glColor3fv(svetlo_siva);
            glBegin(GL_QUADS);
                glVertex3f(i, 0, j);
                glVertex3f(i, 0, j+1);
                glVertex3f(i+1, 0, j+1);
                glVertex3f(i+1, 0, j);
            glEnd();
        }
    glPopMatrix();
}

void crtajScenu(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glTranslatef( 0, 0, -100 );
    glRotatef(20, 1, 0, 0);
    glRotatef(60, 0, 1, 0);
    crtajPodlogu(30, 6);
    glScalef(40, 40, 40);
    glTranslatef(0, 1, 0);
    crtajFiguru();
    glFlush();
}

void init() {
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glDisable(GL_CULL_FACE);
    glClearColor (1.0, 1.0, 1.0, 0.0);
    glEnable(GL_DEPTH_TEST);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-70, 70, -40, 100, -400, 400);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("RG - Jun 2011");
    init();
    glutDisplayFunc(crtajScenu);
    glutReshapeFunc(promenaProzora);
    glutMainLoop();
    return 0;
}
```

2)

```
void crtaj(int x1, int y1, int x2, int y2, bool mod)
{
    int d, x, y, cx, cy;

    line(x1, y1, x2, y1);
    line(x2, y1, x2, y2);
    line(x2, y2, x1, y2);
    line(x1, y2, x1, y1);

    cy = (y2+y1)/2;
    cx = (x2+x1)/2;

    x = (x2-x1)/2;
    y = 0;
    d = 3-2*x;
    while(x >= y)
    {
        if( mod )
        {
            line(x1, cy+y, cx-x, cy+y);
            line(x1, cy+x, cx-y, cy+x);

            line(cx+x, cy-y, x2, cy-y);
            line(cx+y, cy-x, x2, cy-x);

            line(x1+x, y1+y, cx, y1+y);
            line(x1+y, y1+x, cx, y1+x);

            plot(x2-x, y1+y);
            plot(x2-y, y1+x);

            line(cx, y2-y, x2-x, y2-y);
            line(cx, y2-x, x2-y, y2-x);

            plot(x1+x, y2-y);
            plot(x1+y, y2-x);
        }
    }
}
```

```
else
{
    line(cx+x, cy+y, x2, cy+y);
    line(cx+y, cy+x, x2, cy+x);

    line(x1, cy-y, cx-x, cy-y);
    line(x1, cy-x, cx-y, cy-x);

    line(cx, y1+y, x2-x, y1+y);
    line(cx, y1+x, x2-y, y1+x);

    plot(x1+x, y1+y);
    plot(x1+y, y1+x);

    line(x1+x, y2-y, cx, y2-y);
    line(x1+y, y2-x, cx, y2-x);

    plot(x2-x, y2-y);
    plot(x2-y, y2-x);
}

if( d < 0 )
    d += 4*y+6;
else
{
    d += 4 * (y-x)+10;
    x--;
}
y++;
```