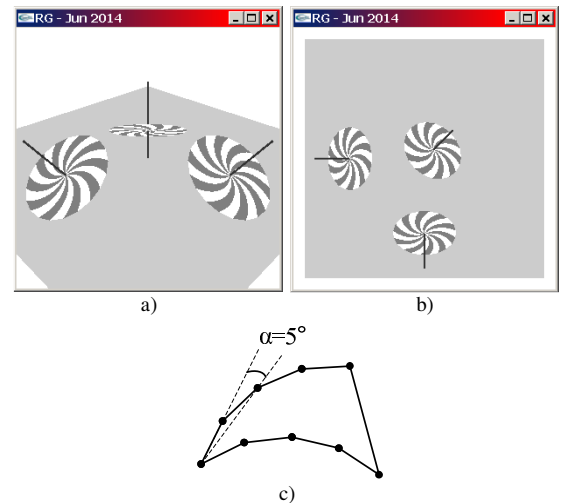


ISPIT IZ RAČUNARSKE GRAFIKE

- 1) [30] **OpenGL:** Napisati na jeziku C ili C++ deo programa za crtanje scene prikazane na slici 1a primenom grafičke biblioteke OpenGL. Scena se sastoji od tri čigre koje leže na podlozi kvadratnog oblika. Telo čigre se sastoji od cilindrične osovine i diska. Disk preseca osovinu na 0.35 dužine osovine, mereno od donje osnovice osovine. Osovina svake čigre je pod uglom od 35° u odnosu na podlogu. Raspored čigri u prostoru prikazan je na slici 1b (pogled odozgo). Osovina čigre se crta tamno sivom bojom. Disk čigre je ukrašen upotrebom 20 spiralnih traka bele i sive boje koje se naizmenično smenjuju. Svaka traka sastoji se od 10 segmenata. Na slici 1c dat je primer izgleda jedne trake koja se sastoji od 4 segmenta. Pritiskom na tastere \leftarrow i \rightarrow čigre rotiraju oko svoje ose za 3° u odgovarajućem smeru. Scena se crta na beloj podlozi. Koristi se projekcija sa perspektivom. 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 sliku 1a. Smatrati da je otvaranje prozora za crtanje realizovano u glavnom programu koji nije potrebno pisati.

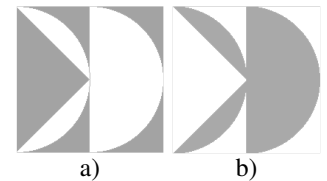


Slika 1

- 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 valjanosti datih koordinata. Crtanje je crnom olovkom na beloj pozadini.



Slika 2

- 3) [20] Opisati detaljno *Sutherland-Hodgmanov* algoritam.
- 4) [30] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:
- Koji je razlog za uvođenje identiteta $1=1$ u sistem jednačina kojim se opisuju transformacije u računarskoj grafici?
 - Šta se postiže prikaznim transformacijama, a šta geometrijskim modelirajućim transformacijama u sistemu SPHIGS? Kojim redosledom se izvršavaju?
 - Koje su prednosti a koji nedostaci algoritma za uklanjanje skrivrenih površi okrenutih naličjem prema posmatraču (*backface removal*)? Da li se i zašto dati algoritam može koristiti za: (1) pojedinačna konkavna tela, (2) scene sa više tela?

Napomene: 1. Ispit traje 180 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

Jun 2014

1) Rešenje (delovi programa obeleženi sivo nisu deo očekivanog rešenja)

```
/* RG 2013-2014: Jun */
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "GL/glut.h"
#include "GL/GL.H"
#define D2R(x) ((x)*3.14159265358979/180)
const float tamno_siva[] =
    {0.2f, 0.2f, 0.2f};
const float svetlo_siva[] =
    {0.8f, 0.8f, 0.8f};
const float siva[] = {0.5f, 0.5f, 0.5f};
const float bela[] = {1, 1, 1};

float rotacija = 0;

void promenaProzora(int width, int height)
{
    glViewport(0, 0, (GLint) width, (GLint)
height);
}

void crtajCilindar(int brSeg,
    const float *boja) {
    static const float r=0.5f, h=0.5f;
    float korak = 360.f / brSeg;
    float x = r, z = 0;
    float tekUgao = 0;
    glColor3fv(boja);

    glBegin(GL_QUAD_STRIP);
    for(int i = 0; i <= brSeg; i++) {
        float x=r*cos(D2R(tekUgao));
        float z=r*sin(D2R(tekUgao));
        glVertex3f(x, -h, z);
        glVertex3f(x, h, z);
        tekUgao += korak;
    }
    glEnd();
}

void crtajSegmentSare(const float opseg,
const float inkUgla, const int brKoraka) {
    float korakR = 0.5f / brKoraka;
    float tekUgao = 0;
    glBegin(GL_TRIANGLE_STRIP);
    glVertex3f(0, 0, 0);
    for(int k=1; k<=brKoraka; k++){
        float r = korakR * k;
        float x = r * cos(D2R(tekUgao));
        float z = r * sin(D2R(tekUgao));
        glVertex3f(x, 0, z);
        x = r * cos(D2R(tekUgao+opseg));
        z = r * sin(D2R(tekUgao+opseg));
        glVertex3f(x, 0, z);
        tekUgao += inkUgla;
    }
    glEnd();
}

void crtajSarenDisk(const float *bojal,
const float *boja2, const int brSeg, const
float inkUgla, const float brKoraka) {
    float opseg = 360.0f / brSeg;
    glPushMatrix();
    glPushAttrib(GL_CURRENT_BIT);

    for(int seg = 0; seg < brSeg; seg++) {
        if( seg%2 == 0 ) glColor3fv(bojal);
        else glColor3fv(boja2);
        crtajSegmentSare(opseg, inkUgla,
            brKoraka);
        glRotatef(opseg, 0, 1, 0);
    }

    glPopAttrib();
    glPopMatrix();
}

void crtajCigru(float velicina)
{
    glPushMatrix();
    glScalef(1, velicina, 1);
    glTranslatef(0, 0.5, 0);

    glPushMatrix();
    glScalef(0.1f, 1, 0.1f);
    crtajCilindar(20, tamno_siva);
    glPopMatrix();

    glTranslatef(0, -0.15f, 0);
    glScalef(velicina, 1, velicina);
    crtajSarenDisk(siva, bela, 20, 5, 10);
    glPopMatrix();
}

void crtajScenu(void) {
    glClear(GL_COLOR_BUFFER_BIT |
        GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glTranslatef( 0, 0, -23 );
    glRotatef(30, 1, 0, 0);
    glRotatef(45, 0, 1, 0);

    glBegin(GL_QUADS);
    glColor3fv(svetlo_siva);
    glVertex3f(-10, 0, -10);
    glVertex3f(10, 0, -10);
    glVertex3f(10, 0, 10);
    glVertex3f(-10, 0, 10);
    glEnd();

    glPushMatrix();
    glTranslatef(0, 0, 5);
    glRotatef(35.f, 1, 0, 0);
    glRotatef(rotacija, 0, 1, 0);
    crtajCigru(5);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(-5, 0, 0);
    glRotatef(35.f, 0, 0, 1);
    glRotatef(rotacija, 0, 1, 0);
    crtajCigru(5);
    glPopMatrix();

    glPushMatrix();
    glRotatef(-35.f, 1, 0, 1);
    glRotatef(rotacija, 0, 1, 0);
    crtajCigru(5);
    glPopMatrix();
    glFlush();
}

void init() {
    glPolygonMode(GL_FRONT_AND_BACK,
        GL_FILL);
    glDisable(GL_CULL_FACE);
    glClearColor (1.0, 1.0, 1.0, 1.0);
    glEnable(GL_DEPTH_TEST);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(40, 1, 1, 600);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void specFunc(int key, int x, int y) {
    switch(key) {
        case GLUT_KEY_LEFT:
            rotacija += 3;
            glutPostRedisplay();
            break;
        case GLUT_KEY_RIGHT:
            rotacija -= 3;
            glutPostRedisplay();
            break;
    }
}

int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_RGB |
        GLUT_DEPTH);
    glutInitWindowSize(300, 300);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("RG - Jun 2014");
    init();
    glutDisplayFunc(crtajScenu);
    glutSpecialFunc(specFunc);
    glutReshapeFunc(promenaProzora);
    glutMainLoop();
    return 0;
}
```

2)

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

    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;
    r = (x2-x1)/2;

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

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

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

            line(cx, cy-x, cx+y, cy-x);
            line(cx, cy+x, cx+y, cy+x);
        }
    }
```

```
else
{
    line(cx-r, cy-y, cx-y, cy-y);
    line(cx-r, cy+y, cx-y, cy+y);

    line(cx-r, cy+x, cx-x, cy+x);
    line(cx-r, cy-x, cx-x, cy-x);

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

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

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

    line(cx+y, cy+x, cx+r, cy+x);
    line(cx+y, cy-x, cx+r, cy-x);
}

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

Odgovori na pitanja:

- 1.
- 2.
- 3.