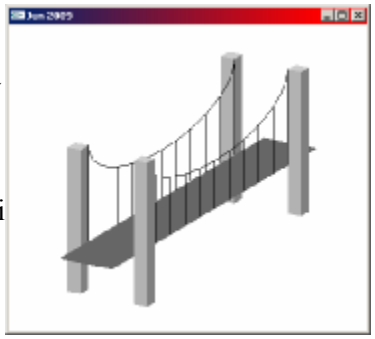
ELEKTROTEHNIČKI FAKULTET 12.06.2009.
Katedra za RTI

ISPIT IZ RAČUNARSKE GRAFIKE

1) [25] **OpenGL:**



slika 1

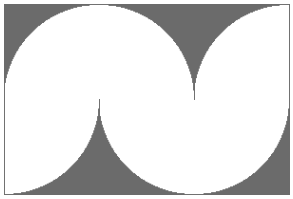
Napisati na jeziku C ili C++ deo programa za crtanje scene prikazane na slici 1 primenom grafičke biblioteke OpenGL. Scena predstavlja drumski most, sledećih dimenzija izraženih u jedinicama dužine: širina kolovoza 24; dužina kolovoza 160; visina stubova 60; visina kolovoza u odnosu na dno stuba 20; osnova stuba je kvadrat 6x6;. rastojanje između vertikalnih simetrala stubova na različitim krajevima mosta 120. Povijeni deo ograde (luk) se crta pravolinijskim segmentima koji prate donju polovinu elipse dužine 114 i visine 20. Ograda svake strane mosta sadrži 11 vertikalnih veza, na jednakom međusobnom rastojanju. 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) [25] Korišćenjem Bresenham-ovog algoritma za crtanje kružnice kao osnove i potprograma za crtanje linije *Line(x1, y1, x2, y2)* koji crta liniju od tačke (*x1, y1*) do tačke (*x2, y2*), 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 3a ili figura 3b (*mod* je `true` za figuru 3b). (*x1, y1*) predstavlja donji levi, a (*x2, y2*) gornji desni ugao figure. Pretpostaviti da je pre poziva potprograma izvršena provera valjanosti datih koordinata. Crtanje je crnom olovkom na beloj pozadini.

3a)



3b)

3) [20] Izvesti matričnu jednačinu za izračunavanje projekcije sa perspektivom za tačku rotiranu oko proizvoljne ose koordinatnog 3D sistema. Nije potrebno izvoditi matricu za projekciju sa perspektivom.

4) [30] Odgovoriti koncizno (jedna do dve rečenice) na sledeća pitanja:

a) Napisati o objasniti uslove trivijalnog prihvatanja i odbacivanja linije u *Cohen-Sutherland* algoritmu.
b) Navesti dobre i loše strane algoritama za uklanjanje skrivenih površina.
c) Da li se mora osvežavati ekran izrađen u tehnologiji (1) CRT, (2) LCD TFT, (3) plazma i zašto?

Napomene:

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

Rešenja zadataka
Jun 2009

1) Rešenje

```
/* RG 2008-2009: Jun */
#include "GL/glut.h"
#include "GL/GL.H"
#define DEG_2_RAD(x)    (x*3.14159265358979/180)
// Funkcija koja se poziva svaki put kada prozor promeni velicinu
void promenaProzora(int width, int height) {
  // Postavljanje viewport-a
  glViewport(0, 0, (GLint) width, (GLint) height);
}
void crtajStub() {
      glPushAttrib(GL_ALL_ATTRIB_BITS);
            glColor3f(0.5f, 0.5f, 0.5f);
            glPushMatrix();
                  glTranslatef(0, 10, 3);
                  glRectf(-3, -30, 3, 30);
                  glTranslatef(0, 0, -6);
                  glRotatef(180, 0, 1, 0);
                  glRectf(-3, -30, 3, 30);
            glPopMatrix();

            glColor3f(0.7f, 0.7f, 0.7f);
            glPushMatrix();
                  glRotatef(90, 0, 1, 0);
                  glTranslatef(0, 10, 3);
                  glRectf(-3, -30, 3, 30);
                  glTranslatef(0, 0, -6);
                  glRotatef(180, 0, 1, 0);
                  glRectf(-3, -30, 3, 30);
            glPopMatrix();

            glColor3f(0.8f, 0.8f, 0.8f);
            glPushMatrix();
                  glRotatef(-90, 1, 0, 0);
                  glTranslatef(0, 0, 40);
                  glRectf(-3, -3, 3, 3);
                  glTranslatef(0, 0, -60);
                  glRotatef(180, 1, 0, 0);
                  glRectf(-3, -3, 3, 3);
            glPopMatrix();
      glPopAttrib();
}
void crtajSajle() {
      glColor3f(0,0,0);
      glBegin(GL_LINE_STRIP);
            for(float i = -180; i <= 0; i+=6) {
            float x = cos( DEG_2_RAD(i) );
            float y = 0.5f*sin( DEG_2_RAD(i) );
                  glVertex3f(x, y, 0);
            }
      glEnd();
      glBegin(GL_LINES);
      for(float i = -1; i <= 1; i+=0.2)   {
      float a = acos(i);
      float x = cos( a );
      float y = 0.5f*sin( a );

            glVertex3f(x, y, 0);
            glVertex3f(x, -1, 0);
      }
      glEnd();
}

void crtajPut() {
      glColor3f(0.4f, 0.4f, 0.4f);
      glRotatef(-90, 1, 0, 0);
      glRectf(-80, -12, 80, 12);
}

void crtajMost() {
      glPushMatrix();
            glPushMatrix();
                  glTranslatef(-60, 0, 15);
                  crtajStub();
                  glTranslatef(0, 0, -30);
                  crtajStub();
                  glTranslatef(120, 0, 0);
                  crtajStub();
                  glTranslatef(0, 0, 30);
                  crtajStub();
            glPopMatrix();

            glPushMatrix();
                  glTranslatef(0, 40, 12);
                  glScalef(57, 40, 1);
                  crtajSajle();
                  glTranslatef(0, 0, -24);
                  crtajSajle();
            glPopMatrix();

            glPushMatrix();
                  crtajPut();
            glPopMatrix();
      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);
  crtajMost();
  glFlush();
}


void init()
{
      glPolygonMode(GL_FRONT, GL_FILL);
      glEnable(GL_CULL_FACE);
      glClearColor (1.0, 1.0, 1.0, 0.0);
      glEnable(GL_DEPTH_TEST);
      glMatrixMode(GL_PROJECTION);
      glLoadIdentity();
      glOrtho(-70, 70, -50, 70, 0, 200);
      glMatrixMode(GL_MODELVIEW);
      glLoadIdentity();
}
```

2) Rešenje zadatka – rasterizacija

```
void crtaj(int x1, int y1, int x2, int y2, bool mod)
{
int d, x, y, cx1, cx2, 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;
	cx1 = x1 + (x2-x1)/3;
	cx2 = x2 - (x2-x1)/3;

	x = (x2-x1)/3;
	y = 0;
	d = 3-2*x;
	while(x >= y)
	{
		if( ! mod )
		{
			line(x1, cy-x, cx1-y, cy-x);
			line(x1, cy-y, cx1-x, cy-y);
			line(cx1+y, cy-x, x2-y, cy-x);
			line(cx1+x, cy-y, x2-x, cy-y);

			line(cx2+y, cy+x, x2, cy+x);
			line(cx2+x, cy+y, x2, cy+y);
			line(x1+y, cy+x, cx2-y, cy+x);
			line(x1+x, cy+y, cx2-x, cy+y);
		}
		else
		{
			line(cx1-y, cy-x, cx1+y, cy-x);
			line(cx1-x, cy-y, cx1+x, cy-y);
			line(x2-y, cy-x, x2, cy-x);
			line(x2-x, cy-y, x2, cy-y);

			line(cx2-y, cy+x, cx2+y, cy+x);
			line(cx2-x, cy+y, cx2+x, cy+y);
			line(x1, cy+x, x1+y, cy+x);
			line(x1, cy+y, x1+x, cy+y);
		}

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