**Universitatea Tehnica Cluj Napoca**

**Facultatea de Automatica si Calculatoare**

**Automatica si Informatica aplicata**

**Proiect**

**Grafica asistata de calculator**

Ciungan Andrei-Robert

Automatica romana, an 1, grupa 1

**2020**



Cuprins

Continut :

Coperta proiect.………………………………………………………..……………………….…..1

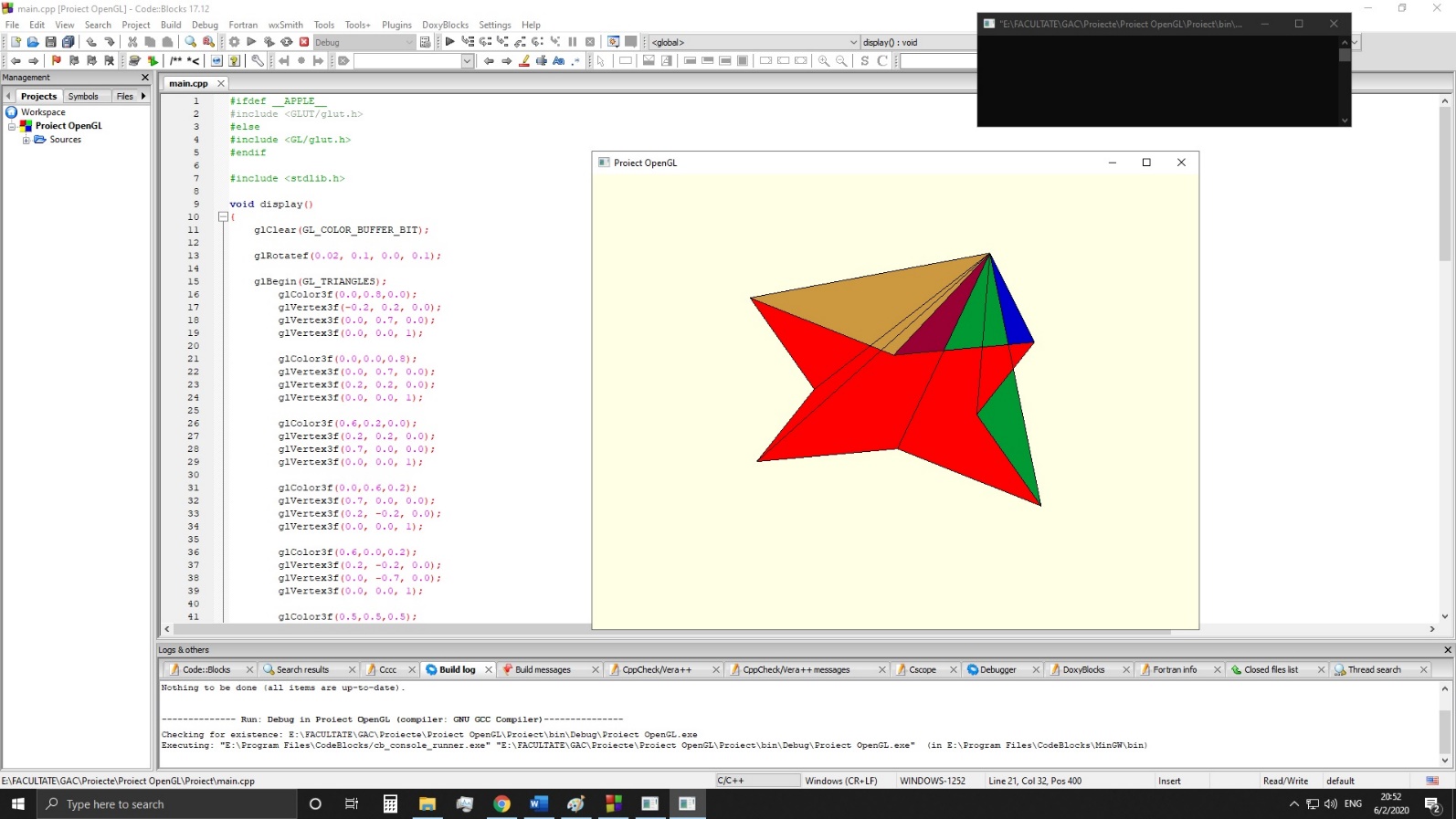
Descriere proiect OpenGL…………………………………………………………………...…3

Printscreen rezultat proiect.……………………….……………………………….……..…4

Cod …………..………………………………………………………………………………….…….5-9

Proiectul consta intr-o piramida, 3D, cu 8 fete de diverse culori. Piramida are la baza o stea rosie cu 4 colturi. Muchiile sunt evidentiate cu linii de culoare neagra.

Figura se roteste automat dupa axa Ox si Oz.



#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glRotatef(0.02, 0.1, 0.0, 0.1);

glBegin(GL\_TRIANGLES);

glColor3f(0.0,0.8,0.0);

glVertex3f(-0.2, 0.2, 0.0);

glVertex3f(0.0, 0.7, 0.0);

glVertex3f(0.0, 0.0, 1);

glColor3f(0.0,0.0,0.8);

glVertex3f(0.0, 0.7, 0.0);

glVertex3f(0.2, 0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glColor3f(0.6,0.2,0.0);

glVertex3f(0.2, 0.2, 0.0);

glVertex3f(0.7, 0.0, 0.0);

glVertex3f(0.0, 0.0, 1);

glColor3f(0.0,0.6,0.2);

glVertex3f(0.7, 0.0, 0.0);

glVertex3f(0.2, -0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glColor3f(0.6,0.0,0.2);

glVertex3f(0.2, -0.2, 0.0);

glVertex3f(0.0, -0.7, 0.0);

glVertex3f(0.0, 0.0, 1);

glColor3f(0.5,0.5,0.5);

glVertex3f(0.0, -0.7, 0.0);

glVertex3f(-0.2, -0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glColor3f(0.25,0.6,0.8);

glVertex3f(-0.2, -0.2, 0.0);

glVertex3f(-0.7, 0.0, 0.0);

glVertex3f(0.0, 0.0, 1);

glColor3f(0.8,0.6,0.25);

glVertex3f(-0.7, 0.0, 0.0);

glVertex3f(-0.2, 0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(1,0.0,0.0);

glVertex3f(-0.2, 0.2, 0.0);

glVertex3f(0.0, 0.7, 0.0);

glVertex3f(0.2, 0.2, 0.0);

glVertex3f(0.7, 0.0, 0.0);

glVertex3f(0.2, -0.2, 0.0);

glVertex3f(0.0, -0.7, 0.0);

glVertex3f(-0.2, -0.2, 0.0);

glVertex3f(-0.7, 0.0, 0.0);

glEnd();

glBegin(GL\_LINE\_STRIP );

glColor3f(0.0,0.0,0.0);

glVertex3f(-0.2, 0.2, 0.0);

glVertex3f(0.0, 0.7, 0.0);

glVertex3f(0.2, 0.2, 0.0);

glVertex3f(0.7, 0.0, 0.0);

glVertex3f(0.2, -0.2, 0.0);

glVertex3f(0.0, -0.7, 0.0);

glVertex3f(-0.2, -0.2, 0.0);

glVertex3f(-0.7, 0.0, 0.0);

glVertex3f(-0.2, 0.2, 0.0);

glVertex3f(0.0, 0.7, 0.0);

glVertex3f(0.0, 0.0, 1);

glVertex3f(0.0, 0.7, 0.0);

glVertex3f(0.2, 0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glVertex3f(0.2, 0.2, 0.0);

glVertex3f(0.7, 0.0, 0.0);

glVertex3f(0.0, 0.0, 1);

glVertex3f(0.7, 0.0, 0.0);

glVertex3f(0.2, -0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glVertex3f(0.2, -0.2, 0.0);

glVertex3f(0.0, -0.7, 0.0);

glVertex3f(0.0, 0.0, 1);

glVertex3f(0.0, -0.7, 0.0);

glVertex3f(-0.2, -0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glVertex3f(-0.2, -0.2, 0.0);

glVertex3f(-0.7, 0.0, 0.0);

glVertex3f(0.0, 0.0, 1);

glVertex3f(-0.7, 0.0, 0.0);

glVertex3f(-0.2, 0.2, 0.0);

glVertex3f(0.0, 0.0, 1);

glEnd();

glutSwapBuffers();

}

void reshape( int w, int h)

{

glViewport(0,0,w,h);

}

int main(int argc, char \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB | GLUT\_DOUBLE | GLUT\_DEPTH);

glutInitWindowSize(800,600);

glutInitWindowPosition(100,100);

glutCreateWindow("Proiect OpenGL");

glClearColor(1.0,1.0,0.9,1.0);

glutDisplayFunc( display );

glutIdleFunc( display );

glutReshapeFunc( reshape );

glutMainLoop();

return 0;

}