// AKG\_2.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#include <iostream>

#include <stdio.h>

#include <stdarg.h>

#include <string.h>

#include <math.h>

#include <glut.h>

void display();

bool isAnimationForward = true;

double figure\_size = 100;

double angle = 0;

double point\_y = 0;

double point\_x = 0;

void axis(GLfloat size) {

glEnable(GL\_COLOR\_MATERIAL);

glLineWidth(2.0f);

glBegin(GL\_LINES);

glColor3f(1.0f, 0.0f, 1.0f);

glVertex3f(size, 0.0f, 0.0f);

glVertex3f(-size, 0.0f, 0.0f);

glEnd();

glBegin(GL\_LINES);

glColor3f(1.0f, 1.0f, 1.0f);

glVertex3f(0.0f, size, 0.0f);

glVertex3f(0.0f, -size, 0.0f);

glEnd();

glBegin(GL\_LINES);

glColor3f(0.0f, 1.0f, 1.0f);

glVertex3f(0.0f, 0.0f, size);

glVertex3f(0.0f, 0.0f, -size);

glEnd();

glDisable(GL\_COLOR\_MATERIAL);

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity();

glTranslatef(0.0f, 0.0f, 0);

glRotatef(20.0, 1.0, 0.0, 0.0);

glRotatef(30.0, 0.0, 1.0, 0.0);

axis(15.0f);

glRasterPos3f(point\_x+1, point\_y + 1, 1.5);

const char\* lastname = "Angel Vinogradova";

for (int i = 0; i < strlen(lastname); i++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, lastname[i]);

}

glColor3f(0.0, 1.0, 0.0);

glTranslatef(point\_x, point\_y, 0.0);

glRotatef(angle++, 0.0, 1.0, 0.0);

if (angle == 360)

angle = 0;

glColor3f(1.0, 0.0, 1.0);

glBegin(GL\_POLYGON);

glVertex3f(-1.5, 0.5, 0.5);

glColor3f(1.0, 1.0, 1.0);

glVertex3f(0, 1.5, 0.5);

glColor3f(0.0, 1.0, 1.0);

glVertex3f(1.5, 0.5, 0.5);

glColor3f(0.0, 0.0, 1.0);

glVertex3f(1.0, -1.0, 0.5);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(-1.0, -1.0, 0.5);

glEnd();

glColor3f(1.0, 0.0, 1.0);

glBegin(GL\_POLYGON);

glVertex3f(-1.5, 0.5, -0.5);

glColor3f(1.0, 1.0, 1.0);

glVertex3f(0, 1.5, -0.5);

glColor3f(0.0, 1.0, 1.0);

glVertex3f(1.5, 0.5, -0.5);

glColor3f(0.0, 0.0, 1.0);

glVertex3f(1.0, -1.0, -0.5);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(-1.0, -1.0, -0.5);

glEnd();

glColor3f(1.0, 0.0, 1.0);

glBegin(GL\_POLYGON);

glVertex3f(-1.5, 0.5, -0.5);

glColor3f(1.0, 0.0, 1.0);

glVertex3f(-1.5, 0.5, 0.5);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(-1.0, -1.0, 0.5);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(-1.0, -1.0, -0.5);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 1.0);

glVertex3f(-1.5, 0.5, -0.5);

glColor3f(1.0, 0.0, 1.0);

glVertex3f(-1.5, 0.5, 0.5);

glColor3f(1.0, 1.0, 1.0);

glVertex3f(0.0, 1.5, 0.5);

glColor3f(1.0, 1.0, 1.0);

glVertex3f(0.0, 1.5, -0.5);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 1.0);

glVertex3f(0.0, 1.5, -0.5);

glColor3f(1.0,1.0, 1.0);

glVertex3f(0.0, 1.5, 0.5);

glColor3f(0.0, 1.0, 1.0);

glVertex3f(1.5, 0.5, 0.5);

glColor3f(0.0, 1.0, 1.0);

glVertex3f(1.5, 0.5, -0.5);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 1.0, 1.0);

glVertex3f(1.5, 0.5, -0.5);

glColor3f(0.0, 1.0, 1.0);

glVertex3f(1.5, 0.5, 0.5);

glColor3f(0.0, 0.0, 1.0);

glVertex3f(1.0, -1.0, 0.5);

glColor3f(0.0, 0.0, 1.0);

glVertex3f(1.0, -1.0, -0.5);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.0, 1.0);

glVertex3f(-1.0, -1.0, -0.5);

glColor3f(0.0, 0.0, 1.0);

glVertex3f(-1.0, -1.0, 0.5);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(1.0, -1.0, 0.5);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(1.0, -1.0, -0.5);

glEnd();

glFlush();

glutSwapBuffers();

}

void ChangeSize(int w, int h)

{

GLfloat nRange = 5.0f;

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

if (w <= h)

glOrtho(-nRange, nRange, -nRange \* h / w, nRange \* h / w, -nRange, nRange);

else

glOrtho(-nRange \* w / h, nRange \* w / h, -nRange, nRange, -nRange, nRange);

glMatrixMode(GL\_MODELVIEW);

}

void updateAnimation(int value) {

if (isAnimationForward && point\_x >= 5.0)

isAnimationForward = false;

if (!isAnimationForward && point\_y <= -5.0)

isAnimationForward = true;

if (isAnimationForward && point\_x <= 0.0)

point\_y += 0.2;

else if (!isAnimationForward && point\_x <= 0.0)

point\_y -= 0.2;

if (isAnimationForward && point\_y >= 0.0) {

point\_x += 0.2; }

else if (!isAnimationForward && point\_y >= 0.0) {

point\_x -= 0.2; }

glutPostRedisplay();

glutTimerFunc(60, updateAnimation, 0);

}

int main(int argc, char\* argv[]) {

glutInit(&argc, argv);

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

glutInitWindowSize(1200, 800);

glutCreateWindow("Lab2");

glutReshapeFunc(ChangeSize);

glEnable(GL\_DEPTH\_TEST);

glutDisplayFunc(display);

glutTimerFunc(60, updateAnimation, 0);

glutMainLoop();

return 0;

}