

3^η Εργαστηριακή Ενότητα

Άσκηση 6: Δημιουργήστε ένα αρχείο με όνομα 'exer06.c' και χρησιμοποιείτε τον παρακάτω κώδικα.

α) Τι πρόκειται να εμφανίσει ο παρακάτω κώδικας;

β) Μεγεθύνετε το εμφανιζόμενο σχήμα κατά 20% χρησιμοποιώντας τη συνάρτηση `glScalef`.

γ) Περιστρέψτε το σχήμα κατά γωνία 45/2 μοίρες χρησιμοποιώντας τη συνάρτηση `glRotatef`.

```
#include <GL/freeglut.h>
#include <math.h>

#define PI 3.1415926

void drawSimplePolygon(int n, float width) {
    glBegin(GL_POLYGON);
    for (int i = 0; i < n; ++i) {
        float theta = 2.0 * PI * i / n; // Calculate angle for each vertex
        float x = cos(theta) * width; // Calculate x-coordinate
        float y = sin(theta) * width; // Calculate y-coordinate
        glVertex2f(x, y); // Draw the vertex
    }
    glEnd();
}

void display() {
    glClearColor(1.0f, 1.0f, 1.0f, 0);
    glClear(GL_COLOR_BUFFER_BIT);

    // Draw a black vertical line
    glColor3f(0.0f, 0.0f, 0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(-50, 0);
        glVertex2i(50, 0);
    glEnd();
    // Draw a black horizontal line
    glColor3f(0.0f, 0.0f, 0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(0, -50);
        glVertex2i(0, 50);
    glEnd();
    // Draw a string with black Roman fonts
    glColor3f(0.0f, 0.0f, 0.0f);
    glRasterPos2f(-50, 50);
    glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (void *)"Exercise 6");

    glColor3f(1.0f, 0.5f, 0.0f);
    glMatrixMode(GL_MODELVIEW);

    glTranslatef(25, 25, 0);
    drawSimplePolygon(5, 20);
    glTranslatef(-25, -25, 0);

    glFlush();
}
```

```

int main(int argc, char** argv) {
    glutInit(&argc,argv);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(800,800);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow("A sample 2D OpenGL application");
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-5,105,-5,105);
    glTranslatef(50,50,0);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Μεταγλωττίστε τον κώδικα με την εντολή:

```
gcc -Wall -o exer06.run exer06.c -lglut -lGLU -lGL -lm
```

Άσκηση 7: Δημιουργήστε ένα αρχείο με όνομα 'exer07.c' και χρησιμοποιείτε τον παρακάτω κώδικα.

α) Τι πρόκειται να εμφανίσει ο παρακάτω κώδικας;

β) Μετατοπίστε το σχήμα κατά 25 μονάδες στον άξονα των X και κατά 25 μονάδες στον άξονα των Y.

γ) Μεγεθύνετε το εμφανιζόμενο σχήμα κατά 20% χρησιμοποιώντας τη συνάρτηση glScalef.

δ) Περιστρέψτε το σχήμα κατά γωνία 45 μοίρες χρησιμοποιώντας τη συνάρτηση glRotatef.

```

#include <GL/freeglut.h>
#include <math.h>

void drawCube3d(float size) {
    float halfSize = size / 2.0f;

    glBegin(GL_QUADS);           // Begin drawing the color cube with 6 quads
    // Top face (y = 1.0f)
    // Define vertices in counter-clockwise (CCW) order with normal pointing out
    glColor3f(0.0f, 1.0f, 0.0f); // Green
    glVertex3f( halfSize, halfSize, -halfSize);
    glVertex3f(-halfSize, halfSize, -halfSize);
    glVertex3f(-halfSize, halfSize, halfSize);
    glVertex3f( halfSize, halfSize, halfSize);
    // Bottom face (y = -1.0f)
    glColor3f(1.0f, 0.5f, 0.0f); // Orange
    glVertex3f( halfSize, -halfSize, halfSize);
    glVertex3f(-halfSize, -halfSize, halfSize);
    glVertex3f(-halfSize, -halfSize, -halfSize);
    glVertex3f( halfSize, -halfSize, -halfSize);
    // Front face (z = 1.0f)
    glColor3f(1.0f, 0.0f, 0.0f); // Red
    glVertex3f( halfSize, halfSize, halfSize);
    glVertex3f(-halfSize, halfSize, halfSize);
    glVertex3f(-halfSize, -halfSize, halfSize);
    glVertex3f( halfSize, -halfSize, halfSize);
    // Back face (z = -1.0f)
    glColor3f(1.0f, 1.0f, 0.0f); // Yellow
    glVertex3f( halfSize, -halfSize, -halfSize);
    glVertex3f(-halfSize, -halfSize, -halfSize);
    glVertex3f(-halfSize, halfSize, -halfSize);
    glVertex3f( halfSize, halfSize, -halfSize);
    // Left face (x = -1.0f)
    glColor3f(0.0f, 0.0f, 1.0f); // Blue
    glVertex3f(-halfSize, halfSize, halfSize);
    glVertex3f(-halfSize, halfSize, -halfSize);
    glVertex3f(-halfSize, -halfSize, -halfSize);
    glVertex3f(-halfSize, -halfSize, halfSize);
    // Right face (x = 1.0f)
    glColor3f(1.0f, 0.0f, 1.0f); // Magenta
    glVertex3f(halfSize, halfSize, -halfSize);
    glVertex3f(halfSize, halfSize, halfSize);
    glVertex3f(halfSize, -halfSize, halfSize);
    glVertex3f(halfSize, -halfSize, -halfSize);
}

```

```

        glVertex3f(halfSize, -halfSize, -halfSize);
    glEnd(); // End of drawing color-cube
}

void display() {
    glClearColor(1.0f, 1.0f, 1.0f, 0);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Draw a black vertical line
    glColor3f(0.0f, 0.0f, 0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(-50, 0);
        glVertex2i(50, 0);
    glEnd();
    // Draw a black horizontal line
    glColor3f(0.0f, 0.0f, 0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(0, -50);
        glVertex2i(0, 50);
    glEnd();
    // Draw a string with black Roman fonts
    glColor3f(0.0f, 0.0f, 0.0f);
    glRasterPos2f(-50, 50);
    glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (void *)"Exercise 7");

    drawCube3d(20);
    glFlush();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(800, 800);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutCreateWindow("A sample 3D OpenGL application");
    glMatrixMode(GL_PROJECTION);
    glOrtho(-5, 105, -5, 105, -100, 100);
    glTranslatef(50, 50, 0);
    glEnable(GL_DEPTH_TEST);
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Άσκηση 8: Δημιουργήστε ένα αρχείο με όνομα `exer08.c` και χρησιμοποιείτε τον παρακάτω κώδικα.

- α) Τι πρόκειται να εμφανίσει ο παρακάτω κώδικας;
- β) Μετατοπίστε το σχήμα κατά 25 μονάδες στον άξονα των X και κατά 25 μονάδες στον άξονα των Y.
- γ) Μεγεθύνετε το εμφανιζόμενο σχήμα κατά 20% χρησιμοποιώντας τη συνάρτηση `glScalef`.
- δ) Περιστρέψτε το σχήμα κατά γωνία 45 μοίρες χρησιμοποιώντας τη συνάρτηση `glRotatef`.

```
#include <GL/freeglut.h>
#include <math.h>

void drawPyramid3d(float size) {
    float halfSize = size / 2.0f;

    glBegin(GL_TRIANGLES);           // Begin drawing the pyramid with 4 triangles
    // Front
    glColor3f(1.0f, 0.0f, 0.0f);    // Red
    glVertex3f( 0.0f, halfSize, 0.0f);
    glColor3f(0.0f, 1.0f, 0.0f);    // Green
    glVertex3f(-halfSize, -halfSize, halfSize);
    glColor3f(0.0f, 0.0f, 1.0f);    // Blue
    glVertex3f(halfSize, -halfSize, halfSize);

    // Right
    glColor3f(1.0f, 0.0f, 0.0f);    // Red
    glVertex3f(0.0f, halfSize, 0.0f);
    glColor3f(0.0f, 0.0f, 1.0f);    // Blue
    glVertex3f(halfSize, -halfSize, halfSize);
    glColor3f(0.0f, 1.0f, 0.0f);    // Green
    glVertex3f(halfSize, -halfSize, -halfSize);

    // Back
    glColor3f(1.0f, 0.0f, 0.0f);    // Red
    glVertex3f(0.0f, halfSize, 0.0f);
    glColor3f(0.0f, 1.0f, 0.0f);    // Green
    glVertex3f(halfSize, -halfSize, -halfSize);
    glColor3f(0.0f, 0.0f, 1.0f);    // Blue
    glVertex3f(-halfSize, -halfSize, -halfSize);

    // Left
    glColor3f(1.0f,0.0f,0.0f);       // Red
    glVertex3f( 0.0f, halfSize, 0.0f);
    glColor3f(0.0f,0.0f,1.0f);      // Blue
    glVertex3f(-halfSize,-halfSize,-halfSize);
    glColor3f(0.0f,1.0f,0.0f);       // Green
    glVertex3f(-halfSize,-halfSize, halfSize);
    glEnd();    // Done drawing the pyramid
}

void display() {
    glClearColor(1.0f,1.0f,1.0f,0);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Draw a black vertical line
    glColor3f(0.0f,0.0f,0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(-50,0);
        glVertex2i(50,0);
    glEnd();
    // Draw a black horizontal line
    glColor3f(0.0f,0.0f,0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(0,-50);
        glVertex2i(0,50);
    glEnd();
    // Draw a string with black Roman fonts
```

```

    glColor3f(0.0f,0.0f,0.0f);
    glRasterPos2f(-50, 50);
    glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (void *)"Exercise 8");

    drawPyramid3d(20);

    glFlush();
}

int main(int argc, char** argv) {
    glutInit(&argc,argv);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(800,800);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutCreateWindow("A sample 3D OpenGL application");
    glMatrixMode(GL_PROJECTION);
    glOrtho(-5,105,-5,105, -100, 100);
    glTranslatef(50,50,0);
    glEnable(GL_DEPTH_TEST);
    glClearColor(0.0f,0.0f,0.0f,1.0f);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Άσκηση 9: Δημιουργήστε ένα αρχείο με όνομα `exer08.c` και χρησιμοποιείτε τον παρακάτω κώδικα.

- α) Τι πρόκειται να εμφανίσει ο παρακάτω κώδικας;
- β) Μετατοπίστε το σχήμα κατά 25 μονάδες στον άξονα των X και κατά 25 μονάδες στον άξονα των Y.
- γ) Μεγεθύνετε το εμφανιζόμενο σχήμα κατά 20% χρησιμοποιώντας τη συνάρτηση `glScalef`.
- δ) Περιστρέψτε το σχήμα κατά γωνία 45 μοίρες χρησιμοποιώντας τη συνάρτηση `glRotatef`.

```

#include <GL/freeglut.h>
#include <math.h>

void drawCube3d(float size) {
    float halfSize = size / 2.0f;
    GLfloat vertices[][3] = {
        // Front face
        {-halfSize, -halfSize, halfSize}, // 0
        {halfSize, -halfSize, halfSize}, // 1
        {halfSize, halfSize, halfSize}, // 2
        {-halfSize, halfSize, halfSize}, // 3

        // Back face
        {-halfSize, -halfSize, -halfSize}, // 4
        {halfSize, -halfSize, -halfSize}, // 5
        {halfSize, halfSize, -halfSize}, // 6
        {-halfSize, halfSize, -halfSize} // 7
    };
    GLuint faces[][4] = {
        {0, 1, 2, 3}, // Front face
        {7, 6, 5, 4}, // Back face
        {0, 3, 7, 4}, // Left face
        {1, 5, 6, 2}, // Right face
        {3, 2, 6, 7}, // Top face
        {0, 4, 5, 1} // Bottom face
    };
    // Define a single color for each face
    GLfloat colors[][3] = {
        {1.0f, 0.0f, 0.0f}, // Front face (red)
        {0.0f, 1.0f, 0.0f}, // Back face (green)
        {0.0f, 0.0f, 1.0f}, // Left face (blue)
        {1.0f, 1.0f, 0.0f}, // Right face (yellow)
        {1.0f, 0.0f, 1.0f}, // Top face (magenta)
    };
}

```

```

        {0.0f, 1.0f, 1.0f} // Bottom face (cyan)
    };

    // Draw the cube with colors
    for(int i = 0; i < 6; i++) {
        glBegin(GL_POLYGON);
        glColor3fv(colors[i]);
        for(int j = 0; j < 4; j++) {
            glVertex3fv(vertices[faces[i][j]]);
        }
        glEnd();
    }
}

void display() {
    glClearColor(1.0f, 1.0f, 1.0f, 0);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Draw a black vertical line
    glColor3f(0.0f, 0.0f, 0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(-50, 0);
        glVertex2i(50, 0);
    glEnd();
    // Draw a black horizontal line
    glColor3f(0.0f, 0.0f, 0.0f);
    glLineWidth(2); // size of the line
    glBegin(GL_LINES);
        glVertex2i(0, -50);
        glVertex2i(0, 50);
    glEnd();
    // Draw a string with black Roman fonts
    glColor3f(0.0f, 0.0f, 0.0f);
    glRasterPos2f(-50, 50);
    glutBitmapString(GLUT_BITMAP_TIMES_ROMAN_24, (void *)"Exercise 9");

    drawCube3d(20);

    glFlush();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(800, 800);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutCreateWindow("A sample 3D OpenGL application");
    glMatrixMode(GL_PROJECTION);
    glOrtho(-5, 105, -5, 105, -100, 100);
    glTranslatef(50, 50, 0);
    glEnable(GL_DEPTH_TEST);
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```