

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI - 590018, KARNATAKA



A Mini Project Report on

“Scaling of shapes”

A Mini project report submitted for the 5th semester Bachelor of Engineering in Computer Science of Visvesvaraya Technological University, Belagavi

Submitted by

SRILATHA S.R [4UB22CS049]

SUHANA K.Z [4UB22CS050]

SUJAL K. B [4UB22CS051]

SUKANYA N [4UB22CS052]

Dr. K S Shreedhara

Professor

Department of Computer Science



UBDT COLLEGE OF ENGINEERING DAVANGERE-577004

DEPARTMENT OF COMUTER SCIENCE & ENGINEERING

2024-25

Introduction

Scaling of shapes is a fundamental concept in geometry, computer graphics, and various fields of design and engineering. It refers to the process of resizing a shape by a specific factor, either uniformly or non-uniformly, while maintaining its geometric proportions. Scaling is used to create models, resize images, and manipulate graphical objects in a way that retains their original characteristics.

Program Overview

Global Variables:

- `float scale = 1.0f;` (Scale factor)
- `int currentShape = 1;` (1: Triangle, 2: Rectangle, 3: Circle)

Initialization:

- `init()`: Sets black background and orthographic projection.

Drawing Functions:

- `drawTriangle()`: Draws a scaled triangle.
- `drawRectangle()`: Draws a scaled rectangle.
- `drawCircle()`: Draws a scaled circle with 100 segments for smoothness.

Display Callback:

- `display()`: Clears screen, sets color to red, and draws the current shape.

Keyboard Interaction:

- `handleKeyPress()`: Handles key input to select shape ('t', 'r', 'c'), scale shapes up ('+') or down ('-'), and quit ('q').

Main Function:

Initializes GLUT, sets up the window, and enters the main event loop.

Key Features

- **Global Variables:**

- `scale` (Scale factor)

- currentShape (1: Triangle, 2: Rectangle, 3: Circle)
- **Initialization (init):**
 - Black background
 - Orthographic projection setup
- **Shape Drawing:**
 - drawTriangle(): Draws a scaled triangle
 - drawRectangle(): Draws a scaled rectangle
 - drawCircle(): Draws a scaled circle with 100 segments for smoothness
- **Display Function:**
 - Clears the screen
 - Sets color to red
 - Draws the current shape based on currentShape
- **Keyboard Interaction:**
 - Selects shape: 't' (triangle), 'r' (rectangle), 'c' (circle)
 - Scales shapes: '+' to scale up, '-' to scale down
 - Quits: 'q' to exit the program
- **Main Function:**
 - Initializes GLUT
 - Sets up the window
 - Enters the main event loop

Graphical Components

- **Initialization (init):**
 - Sets a black background and orthographic projection.
- **Drawing Functions:**
 - drawTriangle(): Draws a scaled triangle.
 - drawRectangle(): Draws a scaled rectangle.

- drawCircle(): Draws a scaled circle with 100 segments for smoothness.
- **Display Callback (display):**
 - Clears the screen.
 - Sets the drawing color to red.
 - Draws the current shape based on currentShape.
- **Keyboard Interaction (handleKeyPress):**
 - Switch shapes: 't' (triangle), 'r' (rectangle), 'c' (circle).
 - Scale shapes: '+' to scale up, '-' to scale down.
 - Quit the program: 'q'.
- **Main Function:**
 - Initializes GLUT.
 - Sets up the window.
 - Enters the main event loop.

• Code

```
#include <GL/glut.h>
#include <cmath>

// Global variables for scaling
float scale = 1.0f; // Scale factor
int currentShape = 1; // 1: Triangle, 2: Rectangle, 3: Circle

// Function to initialize OpenGL settings
void init() {
    glClearColor(0.0, 0.0, 0.0, 1.0); // Black background
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-10, 10, -10, 10); // Orthographic projection
}

// Function to draw a triangle
void drawTriangle() {
    glBegin(GL_TRIANGLES);
    glVertex2f(0.0 * scale, 5.0 * scale);
    glVertex2f(-5.0 * scale, -5.0 * scale);
    glVertex2f(5.0 * scale, -5.0 * scale);
    glEnd();
}

// Function to draw a rectangle
void drawRectangle() {
    glBegin(GL_QUADS);
    glVertex2f(-5.0 * scale, -3.0 * scale);
    glVertex2f(5.0 * scale, -3.0 * scale);
    glVertex2f(5.0 * scale, 3.0 * scale);
    glVertex2f(-5.0 * scale, 3.0 * scale);
    glEnd();
}
```

```

}

// Function to draw a circle
void drawCircle() {
    int segments = 100; // More segments for smoothness
    float angleStep = 2 * 3.14159 / segments;
    glBegin(GL_POLYGON);
    for (int i = 0; i < segments; ++i) {
        float x = cos(i * angleStep) * scale * 5.0;
        float y = sin(i * angleStep) * scale * 5.0;
        glVertex2f(x, y);
    }
    glEnd();
}

// Display callback function
void display() {
    glClear(GL_COLOR_BUFFER_BIT); // Clear the screen

    glColor3f(1.0, 0.0, 0.0); // Red color

    switch (currentShape) {
        case 1: drawTriangle(); break;
        case 2: drawRectangle(); break;
        case 3: drawCircle(); break;
    }

    glFlush(); // Render everything
}

// Keyboard interaction function
void handleKeyPress(unsigned char key, int x, int y) {
    switch (key) {
        case 't': // Select Triangle
            currentShape = 1;
            break;
        case 'r': // Select Rectangle
            currentShape = 2;
            break;
        case 'c': // Select Circle
            currentShape = 3;
            break;
        case '+': // Scale up
            scale += 0.1f;
            break;
        case '-': // Scale down
            if (scale > 0.1f) scale -= 0.1f;
            break;
        case 'q': // Quit the program
            exit(0);
            break;
    }
    glutPostRedisplay(); // Redraw with updated parameters
}

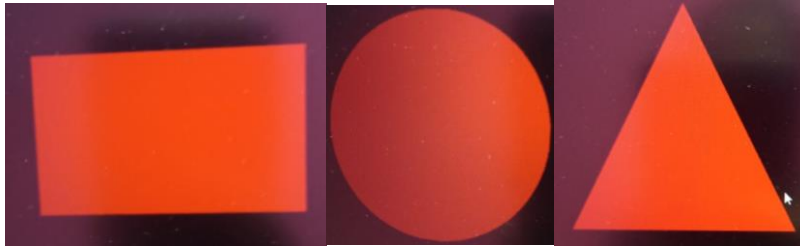
// Main function
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Interactive Shapes with Scaling");

    init();
    glutDisplayFunc(display);
    glutKeyboardFunc(handleKeyPress); // Register keyboard callback
}

```

```
glutMainLoop();  
    return 0;  
}
```

- **Output:**



Execution Instructions

- **Compile the Code:**

```
Bash  
g++ -o shapes_program shapes_program.cpp -lGL -lGLU -  
lglut
```

- **Run the Program:**

```
bash
```

```
./shapes_program
```

Expected Output

A black window with a red shape (triangle, rectangle, or circle) that can be scaled up or down using keyboard inputs.

Conclusion

This program demonstrates interactive scaling and switching between triangle, rectangle, and circle shapes using OpenGL and GLUT with simple keyboard controls.