Experiment No: 11

Experiment Name:

Draw a Flag and Animate it using Bezier Curves

Objective:

The objective of this experiment is to draw a **flag** and create a **waving animation** using **Bezier curves** in OpenGL to understand curve representation and animation in computer graphics.

Theory:

A Bezier curve is a parametric curve widely used in computer graphics to model smooth and continuous shapes. It is defined using a set of control points, and the shape of the curve depends on these points.

Bezier curves are especially useful for:

- Drawing smooth surfaces
- Animating cloth-like objects
- Modeling realistic motion

To animate a flag:

- A Bezier surface is used to represent the cloth
- Control points are slightly moved over time
- This creates a waving effect, similar to wind

Software Requirements:

- Code::Blocks
- C++
- OpenGL, GLUT Library

Procedure / Description of Code:

1. Initialize OpenGL and create a window using GLUT.
2. Define control points for the Bezier surface representing the flag.
3. Use `glMap2f()` to map the Bezier surface.
4. Draw the flag surface using `glEvalMesh2()`.
5. Draw the flag pole using line primitives.
6. Modify the control points periodically using a timer function.
7. Update the display continuously to create a waving animation.
8. Display the animated flag on the screen.

SOURCE CODE:

```
#include <windows.h>

#include <GL/glut.h>

#include <cmath>

float wave = 0.0f;

void drawCircle(float cx, float cy, float r)

{

   glBegin(GL_TRIANGLE_FAN);

   glVertex2f(cx, cy);

   for (int i = 0; i <= 360; i++)

   {

      float angle = i * 3.1416f / 180;

      glVertex2f(cx + r * cos(angle), cy + r * sin(angle));

   }

   glEnd();
```

```
}
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();

    glTranslatef(0.0f, sin(wave) * 0.02f, 0.0f); // waving effect

    // Green rectangle (flag)
    glColor3f(0.0f, 0.6f, 0.0f);
    glBegin(GL_QUADS);
        glVertex2f(-0.6f,  0.3f);
        glVertex2f( 0.6f,  0.3f);
        glVertex2f( 0.6f, -0.3f);
        glVertex2f(-0.6f, -0.3f);
    glEnd();
    // Red circle (slightly left)
    glColor3f(0.8f, 0.0f, 0.0f);
    drawCircle(-0.1f, 0.0f, 0.15f);
    glFlush();
}
```

```
void timer(int)
{
    wave += 0.05f;
    glutPostRedisplay();
    glutTimerFunc(30, timer, 0);
}
void init()
{
    glClearColor(1, 1, 1, 1);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-1, 1, -1, 1);
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 300);
    glutCreateWindow("Bangladesh Flag Animation");

    init();
```

```
    glutDisplayFunc(display);
    glutTimerFunc(0, timer, 0);
    glutMainLoop();
    return 0;
}
```

Conclusion / Discussion:

This experiment shows how **Bezier curves** can be used to model and animate smooth surfaces in computer graphics. By dynamically changing control points, realistic motion such as a waving flag can be achieved. This technique is commonly used in animations, simulations, and game graphics.