Experiment No. 8
Implement Curve: Bezier for n control points.
Name: Pratik Sanjay Avhad
Roll Number: 01
Date of Performance:
Date of Submission:

Experiment No. 8

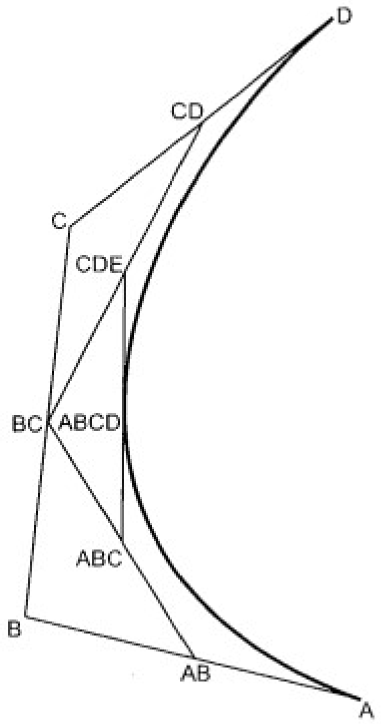
Aim: To implement Bezier curve for n control points. (Midpoint approach)

Objective:

Draw a Bezier curves and surfaces written in Bernstein basis form. The goal of interpolation is to create a smooth curve that passes through an ordered group of points. When used in this fashion, these points are called the control points.

Theory:

In midpoint approach Bezier curve can be constructed simply by taking the midpoints. In this approach midpoints of the line connecting four control points (A, B, C, D) are determined (AB, BC, CD, DA). These midpoints are connected by line segment and their midpoints are ABC and BCD are determined. Finally, these midpoints are connected by line segments and its midpoint ABCD is determined as shown in the figure –



The point ABCD on the Bezier curve divides the original curve in two sections. The original curve gets divided in four different curves. This process can be repeated to split the curve into smaller sections until we have sections so short that they can be replaced by straight lines.

Algorithm:

1) Get four control points say A(xa, ya), B(xb, yb), C(xc, yc), D(xd, yd).

2) Divide the curve represented by points A, B, C, and D in two sections.

xab = (xa + xb) / 2

yab = (ya + yb) / 2

xbc = (xb + xc) / 2

ybc = (yb + yc) / 2

xcd = (xc + xd) / 2

ycd = (yc + yd) / 2

xabc = (xab + xbc) / 2

yabc = (yab + ybc) / 2

xbcd = (xbc + xcd) / 2

ybcd = (ybc + ycd) / 2

xabcd = (xabc + xbcd) / 2

yabcd = (yabc + ybcd) / 2

3) Repeat the step 2 for section A, AB, ABC, ABCD and section ABCD, BCD, CD, D.

4) Repeat step 3 until we have sections so that they can be replaced by straight lines.

5) Repeat small sections by straight lines.

6) Stop.

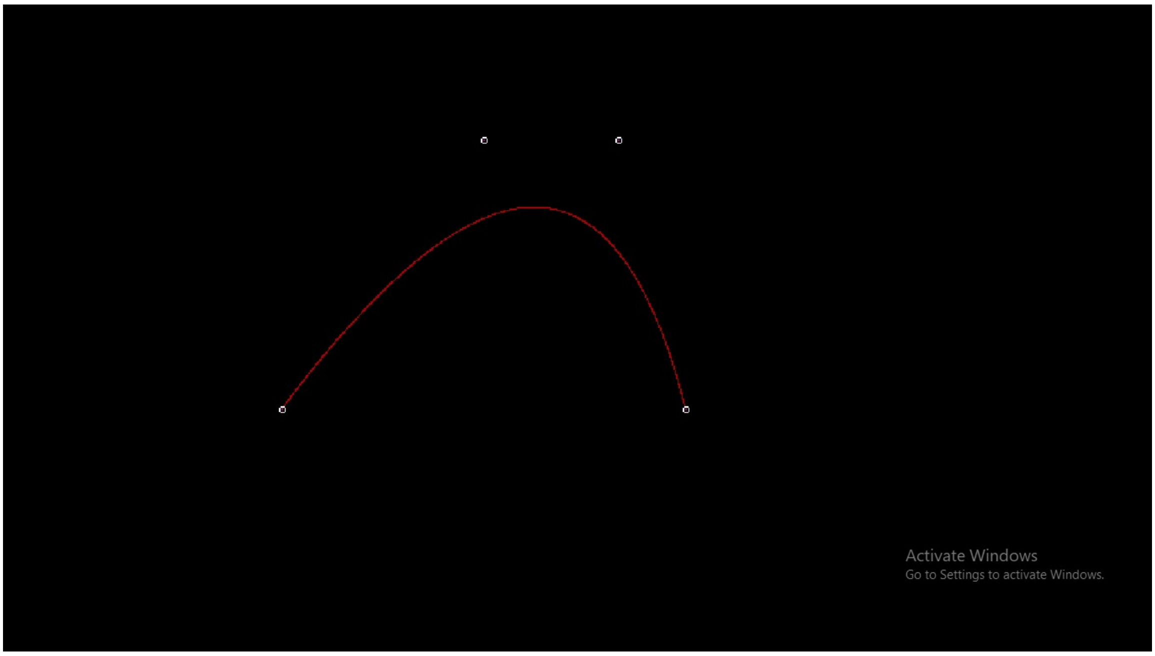
Program:

```
#include<graphics.h>
#include<math.h>
int x[4],y[4];
void bezier(int x[4],int y[4])
{
int gd=DETECT,gm,i;

double t,xt,yt;
initgraph(&gd,&gm," ");
```

```
for(t=0.0;t<1.0;t+=0.0005)
{
xt=pow((1.0-t),3)*x[0]+3*t*pow((1.0-t),2)*x[1]+3*pow(t,2)*(1.0-t)*x[2]+pow(t,3)*x[3];
yt=pow((1.0)-t,3)*y[0]+3*t*pow((1.0)-t,2)*y[1]+3*pow(t,2)*(1.0-t)*y[2]+pow(t,3)*y[3];
putpixel(xt,yt,4);
delay(5);
}
for(i=0;i<4;i++)
{
putpixel(x[i],y[i],5);
circle(x[i],y[i],2);
delay(2);
}
getch();
closegraph();
}
int main()
{
int i,x[4],y[4];
printf("Enter the four control points : ");
for(i=0;i<4;i++)
{
scanf("%d %d",&x[i],&y[i]);
}
bezier(x,y);
}
```

Output:



Conclusion

– Comment on

1. Difference from arc and line- Bezier curves are parametric curves defined by control points and offer more flexibility in defining shapes compared to arcs. Arcs typically represent portions of circles or ellipses and have simpler geometric definitions. Bezier curves can create a wide range of curves with various shapes.

2. Importance of control point- Control points in the Bezier curve determine the shape and behavior of the curve. They provide a high degree of control and allow for precise manipulation of the curve's path. The number and position of control points influence the shape, direction, and curvature of the curve. Control points are essential for defining custom shapes and paths.

3. Applications-

   1. **Graphic Design and Illustration:** Bézier curves are extensively used in graphic design software like Adobe Illustrator and vector-based drawing programs. Graphic designers use Bézier curves to create and edit curves for illustrations, logos, fonts, and complex shapes.

   2. **Computer-Aided Design (CAD):** Engineers and architects use Bézier curves to define curves in CAD software. These curves are essential for creating curves in mechanical parts, architectural designs, and three-dimensional modeling.

3. **Typography:** Bézier curves are used to define the outlines of fonts in digital typefaces. This allows for precise control over the shape of each character, ensuring smooth curves and precise alignment.
4. **Animation:** In animation, Bézier curves are employed for creating and controlling motion paths. Animators use Bézier curves to define the trajectory of objects or characters in both 2D and 3D animations, creating smooth and natural movements.
5. **Video Games:** Bézier curves are used in game development for character movement, camera paths, and other animations. They provide a versatile tool for creating dynamic and visually appealing gameplay.
6. **Computer-Generated Imagery (CGI):** In the film and entertainment industry, Bézier curves are used in CGI to create realistic and detailed objects and characters. They are particularly valuable in modeling complex shapes and surfaces.
7. **Industrial Design:** Industrial designers use Bézier curves to model and refine the shapes of consumer products, from automobiles to consumer electronics. They help in achieving aesthetically pleasing and ergonomic designs.
8. **Aerospace Engineering:** Engineers use Bézier curves to define the shapes of aircraft components, such as wings and fuselages. Bézier surfaces are also used for creating complex three-dimensional shapes in aerospace design.
9. **Medical Imaging:** Bézier curves are employed in medical imaging software for the creation and manipulation of curves and surfaces used in the visualization of anatomical structures, surgical planning, and patient-specific modeling.
10. **Geometric Modeling:** Bézier curves and surfaces are foundational in the field of geometric modeling, where they are used to represent and manipulate complex shapes and surfaces in a variety of industries, including manufacturing and computational geometry.