Harsh Kasliwal
2009085
(2)

Page No.
Date

Experiment 9

Aim: Implement Bezier curve.

Theory: A Bezier curve is a parametric curve which can be controlled using any number of control points. It is extensively used in computer graphics and engineering.

Algorithm:

Step 1: Take the control points coordinates as input (P_{0x}, P_{1x} ... P_{0y}, P_{1y} ...)

step 2: Set $t=0$

step 3: Set $i=0$, $P_x=0$, $P_y=0$

step 4: Set $P_x = P_x + P_{ix} * n_{C_i} * t^i * (1-t)^{(n-i)}$

Set $P_y = P_y + P_{iy} * n_{C_i} * t^i * (1-t)^{(n-i)}$.

step 5: Repeat step 4 until $i \leq n$ that is repeat step 4 for all control points.

step 6: Put pixel at P_x, P_y.

step 7: Set $t = t + 0.001$ [or whichever value is wanted. Lower the value, more points of the curve will be plotted].

step 8: If $t \leq 1$, goto step 5.

The Bezier curve has been plotted !!.

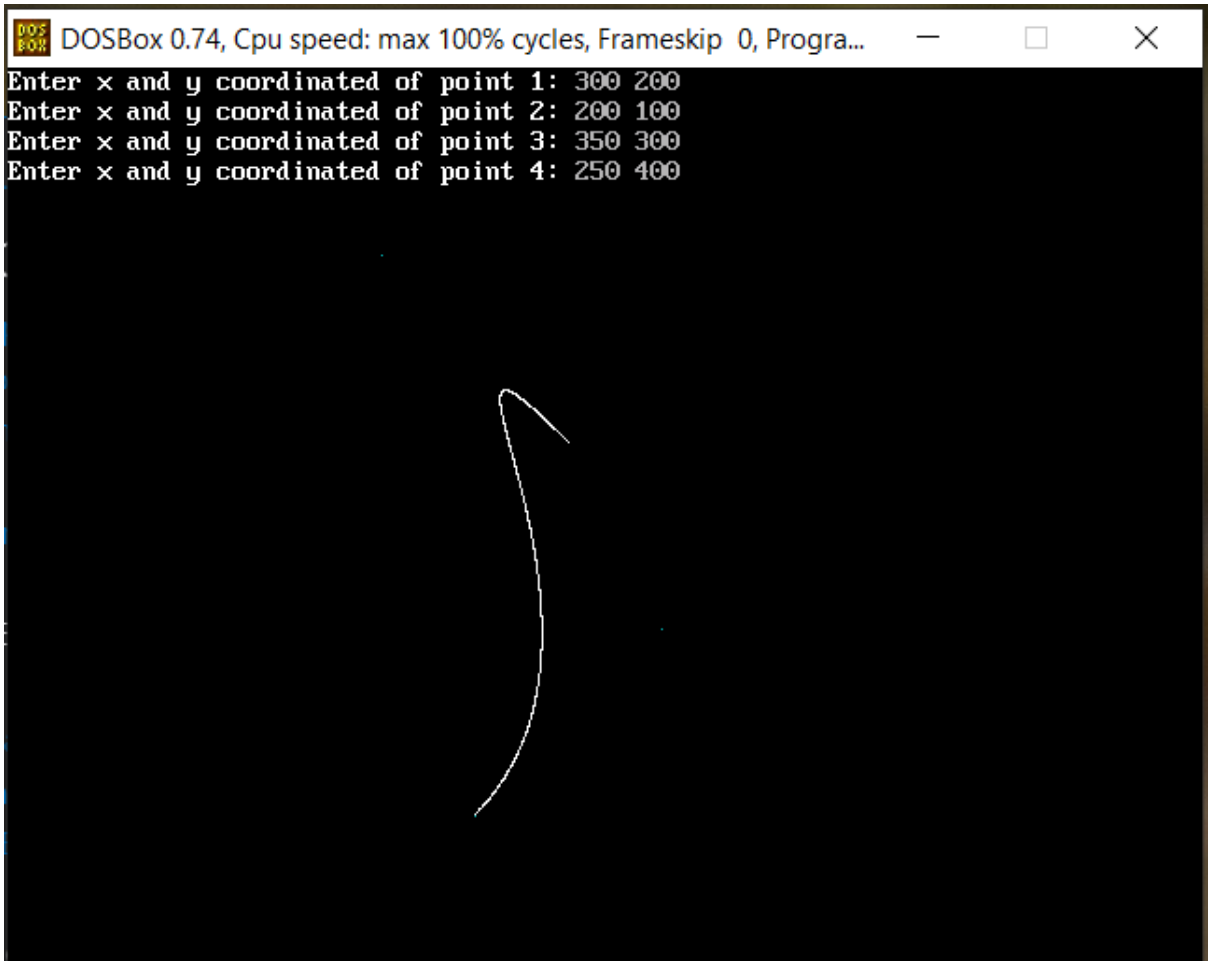
PROGRAM:

Code:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<math.h>
void main(){
   int x[4],y[4],i;
   double puty,putx,t;
   int gd=DETECT,gm;
   initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
   for ( i = 0; i < 4; i++)
   {
      printf("Enter x and y coordinated of point %d: ",i+1);
      scanf("%d%d",&x[i],&y[i]);
      putpixel(x[i],y[i],3);


   }
   for(t=0.0;t<=1.0;t=t+0.001){
      putx=pow(1-t,3)*x[0]+ 3*t*pow(1-t,2)*x[1]+ 3*t*t*pow(1-t,1)*x[2]+
pow(t,3)*x[3];
      puty=pow(1-t,3)*y[0]+ 3*t*pow(1-t,2)*y[1]+ 3*t*t*pow(1-t,1)*y[2]+
pow(t,3)*y[3];
      putpixel(putx,puty,WHITE);
   }
   getch();
   closegraph();
   }
```

OUTPUT:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Enter x and y coordinated of point 1: 300 200
Enter x and y coordinated of point 2: 200 100
Enter x and y coordinated of point 3: 350 300
Enter x and y coordinated of point 4: 250 400