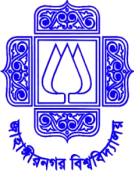
Lab Report-7

*Course title: Computer Graphics Lab*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

Date of Submission: 27/8/2023



***Prof. Mohammad Shorif Uddin***

***Professor***

***Dr. Morium Akter***

***Assistant Professor***

| **Name** | **Class Roll** | **Exam Roll** |
| --- | --- | --- |
| **Samia Alam** | **348** | **202160** |

Department Of Computer Science And Engineering

Jahangirnagar University

Savar,Dhaka

**Experiment Name:**:Bezier Bernstein Algorithm Implementation.

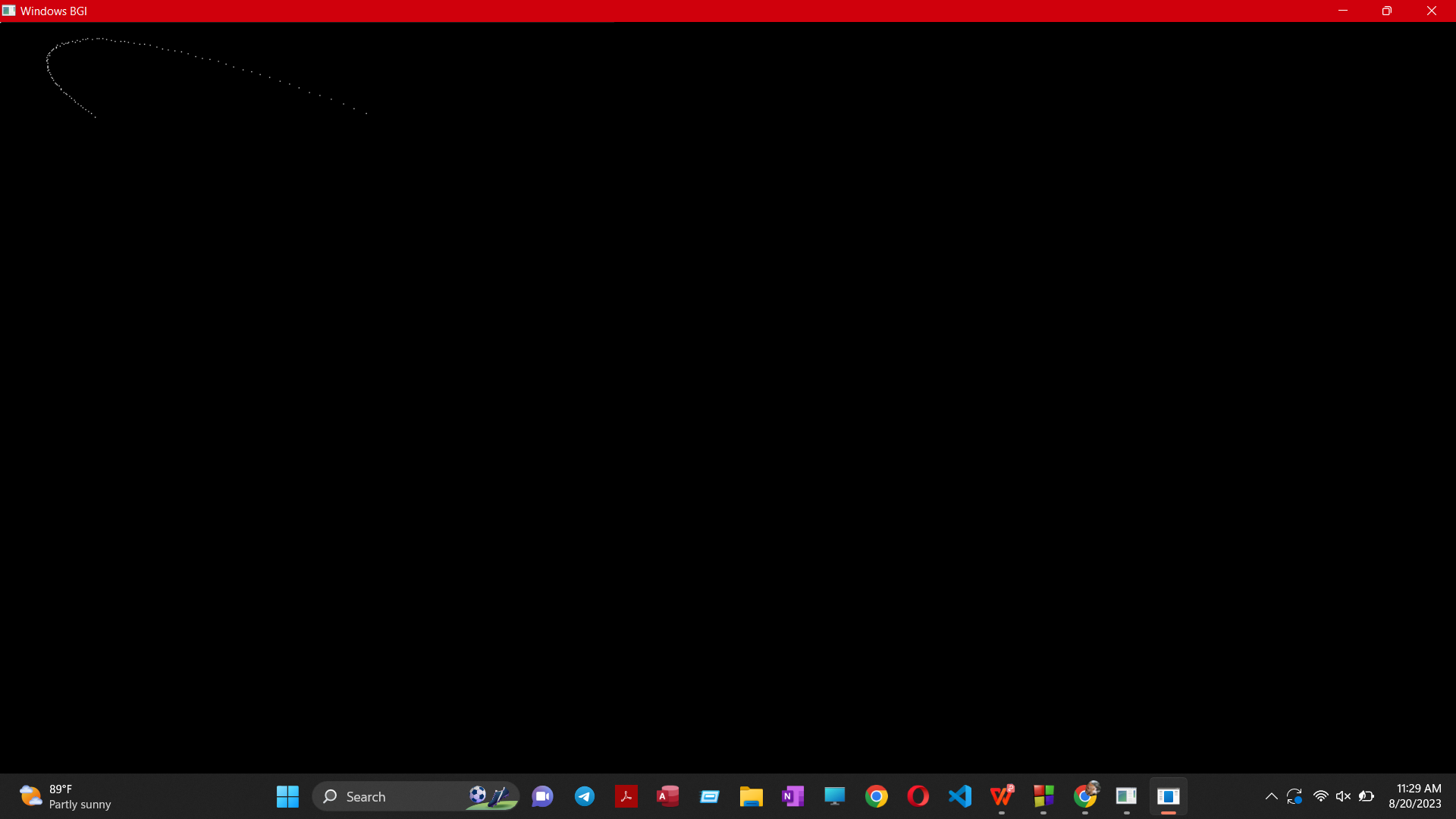
**Introduction:**

The Bezier-Bernstein algorithm is a fundamental technique used in computer graphics and computational geometry for representing and manipulating parametric curves and surfaces. It's particularly useful for creating smooth curves and surfaces, often used in applications like 2D and 3D graphics, CAD (Computer-Aided Design).

**Source code:**

| #include <iostream>  #include <graphics.h>  using namespace std;  struct Point2D  {  int x, y;  };  double Bernstein(int n, int i, double t)  {  double coeff = 1.0;  for (int j = 0; j < n - i; ++j)  {  coeff \*= (1 - t);  }  for (int j = 0; j < i; ++j)  {  coeff \*= t;  }  return coeff;  }  void BezierCurve(Point2D\* controlPoints, int numPoints, double t, Point2D& result)  {  result.x = 0;  result.y = 0; | for (int i = 0; i < numPoints; ++i)  {  double coeff = Bernstein(numPoints - 1, i, t);  result.x += coeff \* controlPoints[i].x;  result.y += coeff \* controlPoints[i].y;  }  }  int main()  {  int gd = DETECT, gm;  initgraph(&gd, &gm, "C:\\Turboc3\\BGI");  int numPoints = 5;  Point2D controlPoints[numPoints] = {{100,100}, {100,200}, {200,100}, {300,-100},{400,100}};  for (double t = 0; t <= 1; t += 0.01)  {  Point2D curvePoint;  BezierCurve(controlPoints, numPoints, t, curvePoint);  putpixel(curvePoint.x, curvePoint.y, WHITE);  }  getch();  closegraph();  return 0;  } |
| --- | --- |

**Output:**

****

**Result and Discussion:**

The result of the Bezier-Bernstein algorithm is the position of a point on a curve or surface defined by control points and parameter values. This point is calculated using mathematical formulas that ensure smoothness

andinterpolation/approximation properties based on the control points' arrangement.