Computer Graphics

Lab –

Bezier curve drawing algorithm

Supervised By: Submitted By:

Dr Ajay Mittal Aniket Singh

UIET CSE 3nd Year(Section-1)

Panjab University UE183010

Objective:

To draw a curve using Bezier curve drawing algorithm.

### Procedure:

**FOR BEZIER CURVE DRAWING ALGORITHM:**

curve

{

double step\_size = 0.005, multi, t = 0.0;

double step = (1/step\_size);

coeff\_calc();

for(int j = 0; j<=step; j++){

vector<double> v1;

t = ((double)j\*step\_size);

double sum\_x = 0.0, sum\_y = 0.0;

for(int i = 0;i<=coun-1;i++){

multi = ((ber\_coeff[i]\*power(t, i))\*(power(1.0f-t, (coun-1-i))));

sum\_x += multi\*point\_x[i];

sum\_y += multi\*point\_y[i];

}

v1.push\_back(sum\_x);

v1.push\_back(sum\_y);

v.push\_back(v1);

}

}

**FOR COEFFICIENT CALCULATION:**

coeff\_calc(){

for(int i = 0; i<coun; i++){

ber\_coeff[i] = (fact(coun-1)/(fact(i)\*fact(coun-1-i))); } }

Program Design

**FOR MOUSE INPUTS:**

mouse(int button, int state, int mousex, int mousey){

but\_x = mousex ;

but\_y = (screen\_breadth) - mousey;

point\_x[coun] = but\_x;

point\_y[coun] = but\_y;

if(button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN){

check = true;

touch++;

plot\_point(but\_x, but\_y, colour, 3);

}

else if(button==GLUT\_RIGHT\_BUTTON && state==GLUT\_DOWN)

{

check = false;

touch = 0;

}

glutPostRedisplay();

}

**FOR RENDER FUNCTION:**

void render(void){

if((but\_x != temp\_x && but\_y != temp\_y) || (but\_x == temp\_x && but\_y != temp\_y) || (but\_x != temp\_x && but\_y == temp\_y)){

coun++;

if(check == false && touch == 0){

double a, b;

glClear(GL\_COLOR\_BUFFER\_BIT);

for(int i = 0;i<coun; i++){

plot\_point(point\_x[i], point\_y[i], colour\_after, 9);

}

rough\_line();

curve();

glColor3f(0.2, 0.7, 0.4);

for (int i = 0; i < v.size(); i++) {

if(i==0){

a = v[i][0];

b = v[i][1];

}

else{

glLineWidth(2);

glBegin(GL\_LINES);

glVertex2f(a, b);

glVertex2f(v[i][0], v[i][1]);

glEnd();

a = v[i][0];

b = v[i][1];

}

}

glFlush();

}

temp\_x = but\_x;

temp\_y = but\_y;

}

}

**FOR INIT FUNCTION:**

init(float r, float g, float b){

glViewport(0, 0, screen\_breadth, screen\_height);

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(r, g, b, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,(GLdouble)screen\_height,0.0,(GLdouble)screen\_breadth);

}

**FOR MAIN FUNCTION:**

main(int argc, char\*\* argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(100, 100);

glutInitWindowSize(screen\_height, screen\_breadth);

glutCreateWindow("Bezier Curve");

init(0.0, 0.0, 0.0);

glutDisplayFunc(render);

glutMouseFunc(mouse);

glutMainLoop();

}

The Program consists of a function and main function. The function named as curve is used finding the coordinates which will be used for Curve drawing, given the points.

**MOUSE FUNCTION:** Inputs is taken through mouse click (left click) and to take the last point input, right click was used.

**RENDER FUNCTION:** In render function, which runs after inputs are taken. It draws the curve after the last point was taken through right click.

**INIT FUNCTION:** Init is used to initialize the background in black colour and to reset some buffers

**MAIN FUNCTION:** Main function is used to set display window’s size and position.

**COEFFICIENT CALCULATION FUNCTION:** Coefficients are calculated and stored in a global array.

**BEZIER CURVE FUNCTION:** Given the Points, all the points are calculated in the curve by this formula.



These values are stored in a vector which are later plotted in render.

## Output:

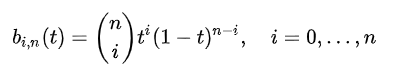
All the input points are displayed by square box. Pink line is the line after joining them and green line is Bezier curve.

Stepsize = 0.05



## Analysis of the Program

The program is written to draw a curve using Bezier algorithm. In this we are finding Bezier coefficient which are



These are multiplied with the x and y coordinated which various t values ranging from 0 to 1. According to the step-size and number of points plotted we get different number of coefficients. Generally degree of the polynomial is number of point plotted minus one.

Result:

As shown in the outputs, we can draw any curve using Bezier curve drawing algorithm with, in the 2-D plane.