|  |
| --- |
| #include <windows.h> |
|  | #include <math.h> |
|  | #include <gl/GL.h> |
|  | #include <gl/glut.h> |
|  | int SCREEN\_HEIGHT=480; |
|  |  |
|  | int NUMPOINTS = 0; |
|  |  |
|  | class Point |
|  | { |
|  | public: |
|  | float x, y; |
|  | void setxy(float x2, float y2) |
|  | { |
|  |  |
|  | x = x2; |
|  | y = y2; |
|  | } |
|  | Point operator&(const Point &rPoint) |
|  | { |
|  |  |
|  | x = rPoint.x; |
|  |  |
|  | y = rPoint.y; |
|  | return \*this; |
|  | }}; |
|  |  |
|  | Point abc[3]; |
|  | void myInit() |
|  | { |
|  |  |
|  | glClearColor(0.0, 0.0, 0.0, 0.0); |
|  |  |
|  | glColor3f(1.0f, 0.0, 0.0); |
|  |  |
|  | glPointSize(4.0); |
|  |  |
|  | glMatrixMode(GL\_PROJECTION); |
|  | glLoadIdentity(); |
|  | gluOrtho2D(0.0, 640, 0.0, 480.0); |
|  | } |
|  |  |
|  | void drawDot(Point pt) |
|  | { |
|  |  |
|  | glBegin(GL\_POINTS); |
|  | glVertex2f(pt.x, pt.y); |
|  |  |
|  | glEnd(); |
|  | glFlush(); |
|  | } |
|  |  |
|  | void drawLine(Point p1,Point p2){ |
|  | glBegin(GL\_LINES); |
|  | glVertex2f(p1.x,p1.y); |
|  | glVertex2f(p2.x,p2.y); |
|  | glEnd(); |
|  | glFlush(); |
|  | } |
|  |  |
|  | Point drawBezier(Point A, Point B, Point C, double t) |
|  | { |
|  | Point P; |
|  | P.x = pow((1 - t), 2) \* A.x + 2 \* t \* (1 - t) \* B.x + pow(t, 2) \* C.x; |
|  | P.y = pow((1 - t), 2) \* A.y + 2 \* t \* (1 - t) \* B.y + pow(t, 2) \* C.y; |
|  | return P; |
|  | } |
|  | void myMouse(int button, int state, int x, int y) |
|  | { |
|  |  |
|  | if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN) |
|  | { |
|  | abc[NUMPOINTS].setxy((float)x, (float)(SCREEN\_HEIGHT - y)); |
|  | NUMPOINTS++; |
|  | if (NUMPOINTS == 3) |
|  | { |
|  |  |
|  | glColor3f(1.0, 0.0, 1.0); |
|  |  |
|  | drawDot(abc[0]); |
|  | drawDot(abc[1]); |
|  | drawDot(abc[2]); |
|  | glColor3f(1.0, 1.0, 0.0); |
|  | drawLine(abc[0], abc[1]); |
|  | drawLine(abc[1], abc[2]); |
|  | Point POld = abc[0]; |
|  | for (double t = 0.0; t <= 1.0; t += 0.1) |
|  | { |
|  |  |
|  | Point P = drawBezier(abc[0], abc[1], abc[2], t); |
|  | drawLine(POld, P); |
|  | POld = P; |
|  | } |
|  |  |
|  | glColor3f(1.0, 0.0, 0.0); |
|  | NUMPOINTS = 0; |
|  | } |
|  | } |
|  | } |
|  | void myDisplay() |
|  | { |
|  |  |
|  | glClear(GL\_COLOR\_BUFFER\_BIT); |
|  |  |
|  | glFlush(); |
|  | } |
|  |  |
|  | int main(int argc, char \*agrv[]) |
|  | { |
|  |  |
|  | glutInit(&argc, agrv); |
|  |  |
|  | glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); |
|  | glutInitWindowSize(640, 480); |
|  |  |
|  | glutInitWindowPosition(100, 150); |
|  | glutCreateWindow("Bezier Curve"); |
|  |  |
|  | //glutDisplayFunc(myDisplay); |
|  | glutMouseFunc(myMouse); |
|  | glutDisplayFunc(myDisplay); |
|  |  |
|  | myInit(); |
|  | glutMainLoop(); |
|  | return 0; |
|  | } |