|  |
| --- |
| #include <GL/glut.h> |
|  | #include <iostream> |
|  |  |
|  | #define zero 0.0 |
|  | #define one 1.0 |
|  |  |
|  | using namespace std; |
|  |  |
|  | int a, b, c, d, type; |
|  |  |
|  | void dda(int x1, int y1, int x2, int y2, int type) { |
|  | cout<<"IN"<<endl; |
|  | float step,x,y,k,Xin,Yin; |
|  | int dx=x2-x1; |
|  | int dy=y2-y1; |
|  |  |
|  | if(abs(dx)> abs(dy)) |
|  | { |
|  | step = abs(dx); |
|  | } |
|  | else |
|  | step = abs(dy); |
|  |  |
|  | Xin = dx/step; |
|  | Yin = dy/step; |
|  |  |
|  | x= x1; |
|  | y=y1; |
|  | glPointSize(1.0f); |
|  | if(type==4){ |
|  | glPointSize(10.0f); |
|  | } |
|  | glBegin(GL\_POINTS); |
|  | glVertex2i(x,y); |
|  | int j=0; |
|  | for (k=1 ;k<=step;k++) |
|  | { |
|  | x= x + Xin; |
|  | y= y + Yin; |
|  |  |
|  | if (type == 4 || type == 1) { |
|  | glVertex2i((int)x, (int)y); |
|  | } |
|  | if (j % 4 == 0 && type == 2) { |
|  | glVertex2i((int)x, (int)y); |
|  | } |
|  | if (j < 5 && type == 3) { |
|  | glVertex2i((int)x, (int)y); |
|  | } |
|  | j = (j + 1) % 10; |
|  | } |
|  | glEnd(); |
|  | } |
|  |  |
|  | void display() { |
|  | glClear(GL\_COLOR\_BUFFER\_BIT); |
|  | glColor3f(one, zero, zero); |
|  | dda(-350, 0, 350, 0, 1); |
|  | dda(0, 350, 0, -350, 1); |
|  | glFlush(); |
|  | } |
|  |  |
|  | void init() { |
|  | glClearColor(0.6, 0.6, 0.6, 0.0); |
|  | glClear(GL\_COLOR\_BUFFER\_BIT); |
|  | // glColor3f(1.0f,0.0f,0.0f); |
|  | // glPointSize(4.0); |
|  | glMatrixMode(GL\_PROJECTION); |
|  | glLoadIdentity(); |
|  | gluOrtho2D(-700 / 2, 700 / 2, -700 / 2, 700 / 2); |
|  | } |
|  |  |
|  | int cnt=0; |
|  | int oldx,oldy; |
|  | int newx,newy; |
|  | void mouse(int button, int state, int x, int y) |
|  | { |
|  | if (state == GLUT\_DOWN) |
|  | { |
|  | if (button == GLUT\_LEFT\_BUTTON) |
|  | { |
|  | int viewport[4]; |
|  | glGetIntegerv(GL\_VIEWPORT, viewport); |
|  | int winWidth = viewport[2]; |
|  | int winHeight = viewport[3]; |
|  |  |
|  | int xi = x- winWidth / 2; |
|  | int yi = winHeight/2-y; |
|  |  |
|  | cout << xi << "\t" << yi << "\n"; |
|  |  |
|  | cnt = (cnt + 1) % 2; |
|  |  |
|  | if (cnt == 1) |
|  | { |
|  | oldx = xi; |
|  | oldy = yi; |
|  | cout << "a" << endl; |
|  | } |
|  | if (cnt == 0) |
|  | { |
|  | newx = xi; |
|  | newy = yi; |
|  | cout << "b" << endl; |
|  | } |
|  |  |
|  | glPointSize(5.0f); |
|  | glColor3f(1.0, 0.0, 0.0); |
|  | glBegin(GL\_POINTS); |
|  | glVertex2i(xi, yi); |
|  |  |
|  | glEnd(); |
|  | glFlush(); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void menu(int a){ |
|  | cout<<"whatever\n"; |
|  | dda(oldx,oldy,newx,newy,a); |
|  | } |
|  |  |
|  | int main(int argc, char\*\* argv) { |
|  |  |
|  | a=200,b=-200,c=-200,d=200; |
|  | type=1; |
|  |  |
|  | glutInit(&argc, argv); |
|  | glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); |
|  | glutInitWindowSize(700, 700); |
|  | glutInitWindowPosition(50, 50); |
|  | glutCreateWindow("Line Drawing"); |
|  | init(); |
|  | glutDisplayFunc(display); |
|  | glutMouseFunc(mouse); |
|  | glutCreateMenu(menu); |
|  | glutAddMenuEntry("DDA\_SIMPLE", 1); |
|  | glutAddMenuEntry("DDA\_DOTTED", 2); |
|  | glutAddMenuEntry("DDA\_DASHED", 3); |
|  | glutAddMenuEntry("DDA\_SOLID", 4); |
|  | //glutAddMenuEntry("EXIT", 9); |
|  | glutAttachMenu(GLUT\_RIGHT\_BUTTON); |
|  | glutMainLoop(); |
|  |  |
|  | return 0; |
|  | } |