**Assignment no: 13**

**//Aim: Draw a line using OpenGL**  
 **//DDA**  
  
**For OpenGL Packages :**

sudo apt-get update  
sudo apt-get install libglu1-mesa-dev freeglut3-dev mesa-common-dev

***$ Compile: g++ program\_name.cpp -o***  
  
***program\_name -lGL -lglut***

***$ Run: ./program\_name***

**.......................................................................................................**

#include< GL/glut.h>

#include <math.h>

#include <stdio.h>

const float PI=3.14;

void LineWithDDA(int x0,int y0,int x1,int y1)

{

glPointSize(2.0);

glBegin(GL\_POINTS);

glColor3f(0.0,0.0,0.0);

int dx,dy,steps,i;

float x,y;

float xinc,yinc;

dy=y1-y0;

dx=x1-x0;

y=y0;

x=x0;

if(abs(dx)>=abs(dy))

{

steps=dx;

}

else

{

steps=dy;

}

xinc=(float)dx/steps;

yinc=(float)dy/steps;

glVertex2d(x,y);

for(i=1;i<steps;i++)

{

x+=xinc;

y+=yinc;

x0=floor(x);

y0=floor(y);

glVertex2d(x,y);

}

glEnd();

}

void init(void)

{

glClearColor(1.0,1.0,1.0,0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0,800,0,600,0,600);

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

LineWithDDA(0,0,800,600);

LineWithDDA(0,600,800,0);

LineWithDDA(0,300,800,300);

LineWithDDA(400,0,400,600);

LineWithDDA(0,150,800,450);

LineWithDDA(0,450,800,150);

LineWithDDA(200,0,600,600);

LineWithDDA(600,0,200,600);

glutSwapBuffers();

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB);

glutInitWindowSize(800,600);

glutInitWindowPosition(100,100);

glutCreateWindow("DDA Line Drawing!");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

---------------------------------------------------------------------------------------------------------------------------------

**//Bresenham's**

#include< GL/glut.h>

#include <math.h>

#include <stdio.h>

const float PI=3.14;

int sign(float arg)

{

if(arg<0)

{

return -1;

}

else if(arg==0)

{

return 0;

}

else

{

return 1;

}

}

void LineWithBrasenham(int x1,int y1,int x2,int y2)

{

glPointSize(2.0);

glBegin(GL\_POINTS);

glColor3f(0.0,0.0,0.0);

int s1,s2,exchange,y,x,i;

float dx,dy,g,temp;

dx=abs(x2-x1);

dy=abs(y2-y1);

x=x1;

y=y1;

s1=sign(x2-x1);

s2=sign(y2-y1);

if(dy>dx)

{

temp=dx;

dx=dy;

dy=temp;

exchange=1;

}

else

{

exchange=0;

}

g=2\*dy-dx;

i=1;

while(i<=dx)

{

glVertex2d(x,y);

while(g>=0)

{

if(exchange==1)

{

x=x+s1;

}

else

{

y=y+s2;

}

g=g-2\*dx;

}

if(exchange==1)

{

y=y+s2;

}

else

{

x=x+s1;

}

g=g+2\*dy;

i++;

}

glEnd();

}

void init(void)

{

glClearColor(1.0,1.0,1.0,0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0,800,0,600,0,600);

}

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

LineWithBrasenham(100,300,700,300);

LineWithBrasenham(400,100,400,500);

glutSwapBuffers();

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB);

glutInitWindowSize(800,600);

glutInitWindowPosition(100,100);

glutCreateWindow("Brasenham Line Drawing!");

init();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}