Computer Graphics LAB Manual

Practical No 2

Write a Program to draw basic graphics construction like line, circle, arc, ellipse and rectangle.

```
#include<graphics.h>
#include<conio.h>
void main()
{
   intgd=DETECT,gm;
   initgraph (&gd,&gm,"c:\\tc\\bgi");
   setbkcolor(BLUE);
   printf("\t\t\t\n\nLINE");
   line(50,40,190,40);
   printf("\t\t\n\n\n\nRECTANGLE");
   rectangle(125,115,215,165);
   printf("\t\t\t\n\n\n\n\n\n\nARC");
   arc(120,200,180,0,30);
   printf("\t\n\n\n\nCIRCLE");
   circle(120,270,30);
   printf("\t\n\n\n\nECLIPSE");
   ellipse(120,350,0,360,30,20);
   getch();
}
```

Practical No.3

Write a Program to draw animation using increasing circles filled with different colors and patterns.

```
#include<graphics.h>
#include<conio.h>
void main()
{
   intgd=DETECT, gm, i, x, y;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
```

```
    x=getmaxx()/3;
    y=getmaxx()/3;
    setbkcolor(WHITE);
    setcolor(BLUE);
    for(i=1;i<=8;i++)
         {
       setfillstyle(i,i);
       delay(20);
       circle(x, y, i*20);
       floodfill(x-2+i*20,y,BLUE);
    }
    getch();
    closegraph();
}
```

Practical No.4

Program to make screen saver in that display different size circles filled with different colors and at random places.

```
#include<stdio.h>
#include<conio.h>
#include"graphics.h"
#include"stdlib.h"
void main()
{
   intgd=DETECT,gm,i=0,x,xx,y,yy,r;
   //Initializes the graphics system
   initgraph(&gd,&gm,"c:\\tc\\bgi");
   x=getmaxx();
   y=getmaxy();
   while(!kbhit())
   {
      i++;
    //    setfillstyle(random(i),random(30));

      circle(xx=random(x),yy=random(y),random(30));
      setfillstyle(random(i),random(30));
```

```
        floodfill(xx,yy,getmaxcolor());
        delay(200);
    }
    getch();
}
```

Practical No.5

Write a program to draw a line using DDA algorithm

ALGORITHM TO DRAW A LINE USING DDA ALGORITHM.

1. Start.

2. Declare variables x,y,x1,y1,x2,y2,k,dx,dy,s,xi,yi and also declare

gdriver=DETECT,gmode.

3. Initialise the graphic mode with the path location in TC folder.

4. Input the two line end-points and store the left end-points in (x1,y1).

5. Load (x1,y1) into the frame buffer;that is,plot the first point.put x=x1,y=y1.

6. Calculate dx=x2-x1 and dy=y2-y1.

7. If abs(dx) > abs(dy), do s=abs(dx).

8. Otherwise s= abs(dy).

9. Then xi=dx/s and yi=dy/s.

10. Start from k=0 and continuing till k<s,the points will be

i. x=x+xi.

ii. y=y+yi.

11. Place pixels using putpixel at points (x,y) in specified colour.

12. Close Graph.

13. Stop.

WAP TO DRAW A LINE USING DDA ALGORITHM.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int x,y,x1,x2,y1,y2,k,dx,dy,s,xi,yi;
int gdriver=DETECT,gmode;
initgraph(&gdriver,&gmode,"C:\\tc\\bgi:");
printf("enter first point");
scanf("%d%d",&x1,&y1);
printf("enter second point");
scanf("%d%d",&x2,&y2);
x=x1;
y=y1;
putpixel(x,y,7);
dx=x2-x1;
dy=y2-y1;
if(abs(dx)>abs(dy))
s=abs(dx);
else
s=abs(dy);
```

```
xi=dx/s;

yi=dy/s;

x=x1;

y=y1;

putpixel(x,y,7);

for(k=0;k<s;k++)

{

x=x+xi;

y=y+yi;

putpixel(x,y,7);

}

getch();

closegraph();

}
```

Practical No.6

Write a program to draw a line using Bresenham's Algorithm

BRESENHEM'S ALGORITHM FOR LINE DRAWING.

```
#include<stdio.h>
#include<graphics.h>
void drawline(int x0, int y0, int x1, int y1)
{
   int dx, dy, p, x, y;
   dx=x1-x0;
   dy=y1-y0;
   x=x0;
   y=y0;
   p=2*dy-dx;
   while(x<x1)
```

```
    {
      if(p>=0)
      {
        putpixel(x,y,7);
        y=y+1;
        p=p+2*dy-2*dx;
      }
      else
      {
        putpixel(x,y,7);
        p=p+2*dy;}
        x=x+1;
      }
}
int main()
{
   int gdriver=DETECT, gmode, error, x0, y0, x1, y1;
   initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
   printf("Enter co-ordinates of first point: ");
   scanf("%d%d", &x0, &y0);
   printf("Enter co-ordinates of second point: ");
   scanf("%d%d", &x1, &y1);
   drawline(x0, y0, x1, y1);
   return 0;
}
```

Practical No.7

Write a program to draw a circle using mid-point algorithm.

```
#include<graphics.h>

#include<conio.h>

#include<stdio.h>

void main()

{

int x,y,x_mid,y_mid,radius,dp;
```

```
int g_mode,g_driver=DETECT;
clrscr();
initgraph(&g_driver,&g_mode,"C:\\TURBOC3\\BGI");
printf("********** MID POINT Circle drawing algorithm ********\n\n");
printf("\nenter the coordinates= ");
scanf("%d %d",&x_mid,&y_mid);
printf("\n now enter the radius =");
scanf("%d",&radius);
x=0;
y=radius;
dp=1-radius;
do
{
putpixel(x_mid+x,y_mid+y,YELLOW);
putpixel(x_mid+y,y_mid+x,YELLOW);
putpixel(x_mid-y,y_mid+x,YELLOW);
putpixel(x_mid-x,y_mid+y,YELLOW);
putpixel(x_mid-x,y_mid-y,YELLOW);
putpixel(x_mid-y,y_mid-x,YELLOW);
putpixel(x_mid+y,y_mid-x,YELLOW);
putpixel(x_mid+x,y_mid-y,YELLOW);
if(dp<0) {
dp+=(2*x)+1;
}
else{
```

```
y=y-1;
dp+=(2*x)-(2*y)+1;
}
x=x+1;
}while(y>x);
getch();}
```

Practical No.8

Write a program to implement 2-D transformation by showing scaling of a triangle

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int x,y,x1,y1,x2,y2;
int scl_fctr_x,scl_fctr_y;
int gd=DETECT,gm;
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
printf("\t\t\t********** Scaling ***********\n");
printf("\n\t\t\t Please enter first coordinate of Triangle = ");
scanf("%d %d",&x,&y);
printf("\n\t\t\t Please enter second coordinate of Triangle = ");
scanf("%d %d",&x1,&y1);
printf("\n\t\t\t Please enter third coordinate of Triangle = ");
scanf("%d %d",&x2,&y2);
```

```
line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x,y);
printf("\n\t\t\t Now Enter Scaling factor x and y = ");
scanf("%d %d",&scl_fctr_x,&scl_fctr_y);
x = x* scl_fctr_x;
x1 = x1* scl_fctr_x;
x2 = x2* scl_fctr_x;
y = y* scl_fctr_y;
y1 = y1* scl_fctr_y;
y2= y2 * scl_fctr_y ;
line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x,y);
getch();
closegraph();
}
```