Computer Graphics Lab File

Submitted By

Mudit Gupta

Enrollment No.

19STUJPBC0027

Program

BCA (2019 – 2022)

Batch

1 Batch

Q1. Write a program to draw a line.

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>

void main() {
        int gd = DETECT,gm;
        initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
        putpixel(20,30,4);
        line(150,150,450,150);
        getch();
        closegraph();
}
```

Q2. Write a program to draw a circle.

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>

void main() {
	int gm;
	int gd=DETECT;
	initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
	putpixel(20,30,40);
	circle(360,100,50);
	getch();
	closegraph();
}
```

Q3. Write a program to draw an rectangle and a circle.

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<doc.h>

void main() {
        int gd=DETECT,gm;
        initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
        circle(160,165,150);
        line(250,250,350,150);
        line(250,250,350,350);
        line(350,150,350,350);
        rectangle(600,260,450,140);
        for(int i=0; i<5; i++) {
                int r=100;
                int a=30;
                circle(160,175,r-a);
        }
        getch();
        closegraph();
}
```

Q4. Write a program using Digital Differential Analyzer (DDA).

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<dos.h>

void main() {
        float x,y,x1,y1,x2,y2,dx,dy,step;
        int i, gd=DETECT,gm;
        initgraph(&gd,&gm,"C:\\turboC3\\bgi");
        printf("Enter the value of x1 & y1:");
        scanf("%f%f",&x1,&y1);
        printf("Enter the value of x2 & y2:");
        scanf("%f%f",&x2,&y2);
        dx = abs(x2 - x1);
        dy = abs(y2 - y1);
        if(dx >= dy)
        {
                step = dx;
        }
        else
        {
                step = dy;
        }
        dx = dx/step;
        dy = dy/step;
        x = x1;
        y = y1;
        i = 1;
        while(i < step)
        {
                putpixel(x,y,6);
                x = x*dx;
                y = y*dy;
                i = i+1;
               delay(10);
        }
        getch();
        closegraph();
}
```

Q5. Write a program using Bresenham’s line algorithm.

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<Math.h>
#include<dos.h>

void main() {
        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,2);
                        y = y*1;
                        p = p+2*dy-2*dx;
                        delay(20);
                }
                else
                {
```

```
            putpixel(x,y,2);
            delay(20);
            p = p+2*dy-2*dx;
        }
        x = x+1;
    }
}

void main() {
    int gd=DETECT,gm,error,x0,y0,x1,y1;
    initgraph(&gd,&gm,"C:\\turboC3\\BGI");
    printf("Enter the first co-ordinates");
    scanf("%d%d",&x0,&y0);
    printf("Enter the second co-ordinates ");
    scanf("%d%d",&x1,&y1);
    drawline(x0,y0,x1,y1);
    getch();
    closegraph();
}
```

Q6. Write a program to draw a Circle Code.

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>

void drawcircle(int xc, int yc, int x, int y) {
        putpixel(xc+x,yc+y ,YELLOW);
        putpixel(xc-x,yc+y ,YELLOW);
        putpixel(xc+x,yc-y ,YELLOW);
        putpixel(xc-x,yc-y ,YELLOW);
        putpixel(xc+y,yc+x ,YELLOW);
        putpixel(xc-y,yc+x ,YELLOW);
        putpixel(xc+y,yc-x ,YELLOW);
        putpixel(xc-y,yc-x ,YELLOW);
}

void CircleBres(int xc,int yc,int r) {
        int x=0,y=r;
        int p=3-2*r;
        drawcircle(xc,yc,x,y);
        while(y >= x)
        {
                x++;
                if(p > 0)
```

```
            {
                y--;
                p =p+4*(x-y)+10;
            }
            else
            {
                p=p+4*x+6;
                drawcircle(xc,yc,x,y);
                delay(80);
            }
        }
}

void main() {
        clrscr();
        int xc=100,yc=100,r=100;
        int gd=DETECT,gm;
        initgraph(&gd,&gm,"C:\\turboC3\\bgi");
        CircleBres(xc,yc,r);
        closegraph;
        getch();
}
```

Q7. Write a program to perform transformation, rotation and scaling.

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<dos.h>

void main() {
        int gm;
        int gd=DETECT;
        int x1 , x2 , x3 , y1 , y2 , y3 , nx1 , nx2 , nx3 ,
        ny1 , ny2 , ny3 , c;
        int sx,sy,xt,yt,r;
        float t;
        initgraph(&gd,&gm,"C:\\turboC3\\bgi");
        printf("\n Program for basic transformation");
        printf("\n Enter the point of triangle:");
        setcolor(1);
        scanf("%d%d%d%d%d%d",&x1,&x2,&x3,&y1,
        &y2,&y3);
        line(x1,y1,x2,y2);
        line(x2,y2,x3,y3);
        line(x3,y3,x1,y1);
```

```
printf("Enter the choise:");
scanf("%d",&c);
switch(c)
{
        case 1:
                initgraph(&gd,&gm,"C:\\turboC3\\BG
                I");
                printf("\nEnter the translation
                factor");
                scanf("%d%d",&x1,&y1);
                nx1=x1+xt;
                ny1=y1+yt;
                nx2=x2+xt;
                ny2=y2+yt;
                nx3=x3+xt;
                ny3=y3+yt;
                line(nx1,ny1,nx2,ny2);
                line(nx2,ny2,nx3,ny3);
                line(nx3,ny3,nx1,ny1);
                getch();
                break;

        case 2:
                initgraph(&gd,&gm,"C:\\turboC3\\BG
                I");
                printf("\n Enter the angle of
                rotation" );
                scanf("%d",&r);
```

```
    t=3.14*r/180;
    nx1=abs(x1*cos(t)-y1*sin(t));
    ny1=abs(x1*sin(t)+y1*cos(t));
    nx2=abs(x2*cos(t)-y2*sin(t));
    ny2=abs(x2*sin(t)+y2*cos(t));
    nx3=abs(x3*cos(t)-y3*sin(t));
    ny3=abs(x3*sin(t)+y3*cos(t));
    line(nx1,ny1,nx2,ny2);
    line(nx2,ny2,nx3,ny3);
    line(nx3,ny3,nx1,ny1);
    getch();
    break;

case 3:
    initgraph(&gd,&gm,"C:\\turboC3\\BG
    I");
    printf("\n Enter the Scaling factor:");
    scanf("%d%d",&sx,&sy);
    nx1=x1*sx;
    ny1=y1*sy;
    nx2=x2*sx;
    ny2=y2*sy;
    nx3=x3*sx;
    ny3=y3*sy;
    line(nx1,ny1,nx2,ny2);
    line(nx2,ny2,nx3,ny3);
    line(nx3,ny3,nx1,ny1);
    getch();
```

```
                break;

            default:
                printf("Enter the correct choice");
                break;
        }
        getch();
        closegraph();
}
```

Q8. Write a program to perform 3D transformation.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
int maxx,maxy,midx,midy;

void axis() {
	getch();
	cleardevice();
	line(midx,0,midx,maxy);
	line(0,midy,maxx,midy);
}

void main() {
	int gd,gm,x,y,z,ang,x1,x2,y1,y2;
	detectgraph(&gd,&gm);
	initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
	setfillstyle(3,25);
	maxx=getmaxx();
	maxy=getmaxy();
	midx=maxx/2;
	midy=maxy/2;
	outtextxy(100,100,"ORIGINAL OBJECT");
```

```
line(midx,0,midx,maxy);
line(0,midy,maxx,midy);
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
axis();
outtextxy(100,20,"TRANSLATION");
printf("\n\nEnter the translation vector:");
scanf("%d%d",&x,&y);
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
bar3d(midx+(x+100),midy-(y+20),midx+(x+60),midy-(y+90),20,5);
axis();
outtextxy(100,20,"SCALING");
printf("\n Enter the scaling factor:");
scanf("%d%d%d",&x,&y,&z);
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
bar3d(midx+(x*100),midy-(y*20),midx+(x*60),midy-(y*90),20*z,5);
axis();
outtextxy(100,20,"ROTATION");
printf("\nEnter the Rotation Angle:");
scanf("%d",&ang);
x1=100*cos(ang*3.14/180)-20*sin(ang*3.14/180);
y1=100*sin(ang*3.14/180)+20*sin(ang*3.14/180);
```

```
    x2=60*cos(ang*3.14/180)-
    90*sin(ang*3.14/180);
    y2=60*sin(ang*3.14/180)+90*sin(ang*3.14/1
    80);
    axis();
    printf("\nAfter rotation about z-axis\n");
    bar3d(midx+100,midy-20,midx+60,midy-
    90,20,5);
    bar3d(midx+x1,midy-y1,midx+x2,midy-
    y2,20,5);
    axis();
    printf("\nAfter rotation about x-axis\n");
    bar3d(midx+100,midy-20,midx+60,midy-
    90,20,5);
    bar3d(midx+100,midy-x1,midx+60,midy-
    x2,20,5);
    axis();
    printf("\nAfter rotation about y-axis\n");
    bar3d(midx+100,midy-20,midx+60,midy-
    90,20,5);
    bar3d(midx+x1,midy-20,midx+x2,midy-
    90,20,5);
    axis();
    closegraph();
}
```