Laboratory Practical Report

of

**Computer Graphics**

**(ICT ED 466)**

Submitted To

**TRIBHUVAN UNIVERSITY**

In Partial Fulfillment of the Requirements of the course

**B.Ed. ICTE 6th Semester**

Submitted By

Sanam Tamang

Symbol No.: 76214020

T.U. Regd. No.: 9-2-214-54-2019

Under the guidance of

**Mr. Atul Bhattarai**

Lecturer

Sukuna Multiple Campus

**SUKUNA MULTIPLE CAMPUS**

Sundarharaincha-12, Morang, Nepal

2080

**CERTIFICATE**

This is to certify that the Laboratory Practical Report

of

**Computer Graphics**

**(ICT ED 466)**

In Partial Fulfillment of the Requirements of the course

**B.Ed. ICTE 6th Semester**

Submitted By

Sanam Tamang

Symbol No.: 76214020

T.U. Regd. No.: 9-2-214-54-2019

is a bonafide record of experiments carried out by him/her under the guidance of

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Atul Bhattarai

Lecturer

Sukuna Multiple Campus

Sundarharaincha-12, Morang

(Internal Examiner)

Submitted for the Final Examination on: 2080/02/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lecturer

(External Examiner)

Contents

[**1. Write a program to plot a pixel.** 1](#_Toc152649142)

[**Output:** 1](#_Toc152649143)

[**2. Write a program to draw a rectangle using line function.** 2](#_Toc152649145)

[**Output:** 2](#_Toc152649146)

[**3. Write a program to draw a line using DDA line drawing algorithm.** 3](#_Toc152649147)

[**Output:** 3](#_Toc152649148)

[**4. Write a program to draw a line using Bresenham’s line drawing algorithm.** 4](#_Toc152649149)

[**Output:** 4](#_Toc152649150)

[**5. Write a program to draw a circle using Midpoint circle drawing algorithm.** 5](#_Toc152649151)

[**Output:** 5](#_Toc152649152)

[**6. Write a program to draw a circle using Bresenham’s circle drawing** 6](#_Toc152649153)

[**algorithm.** 6](#_Toc152649154)

[**Output:** 7](#_Toc152649155)

[**7. Write a program to perform 2-D transformation on a line.** 8](#_Toc152649156)

[**Output:** 10](#_Toc152649157)

[**8. Write a program to perform 3-D transformation on a line.** 11](#_Toc152649158)

[**Output:** 12](#_Toc152649159)

# **1. Write a program to plot a pixel.**

#include <graphics.h>

#include <stdio.h>

int main()

{

    int gd = DETECT, gm, color;

    initgraph(&gd, &gm, "");

    putpixel(85, 35, GREEN);

    putpixel(30, 40, RED);

    putpixel(115, 50, YELLOW);

    putpixel(135, 50, CYAN);

    putpixel(45, 60, BLUE);

    putpixel(20, 100, WHITE);

    putpixel(200, 100, LIGHTBLUE);

    putpixel(150, 100, LIGHTGREEN);

    putpixel(200, 50, YELLOW);

    putpixel(120, 70, RED);

    getch();

    closegraph();

    return 0;

}

### **Output:**

### 

# **2. Write a program to draw a rectangle using line function.**

 #include<stdio.h>

 #include<conio.h>

 #include<graphics.h>

 void main()

 {

 int gd=DETECT, gm;

 initgraph(&gd, &gm, "C:\\TURNOC3\\BGI");

 printf("RECTANGLE");

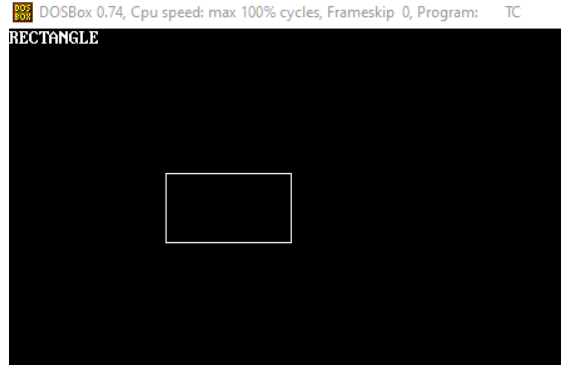
 rectangle(125,115,225,170);

 getch();

 closegraph();

 }

### **Output:**



# **3. Write a program to draw a line using DDA line drawing algorithm.**

 #include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT, gm;

initgraph(&gd, &gm, "C:\\TURNOC3\\BGI");

printf("RECTANGLE");

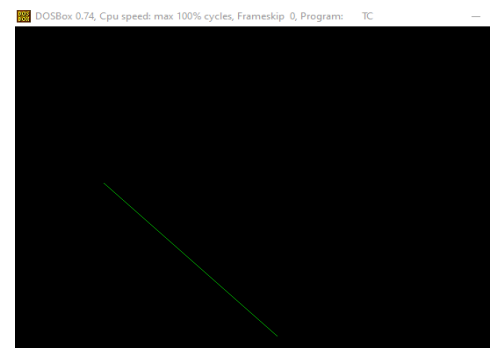
rectangle(125,115,225,170);

getch();

closegraph();

}

### **Output:**



# **4. Write a program to draw a line using Bresenham’s line drawing algorithm.**

#include<stdio.h>

#include<graphics.h>

#include<math.h>

void eight(int xc, int yc, int x, int y){

putpixel(x+xc,y+yc,RED);

putpixel(x+xc,-y+yc,YELLOW);

putpixel(x+xc,-y+yc,GREEN);

putpixel(-x+xc,y+yc,YELLOW);

putpixel(y+xc,x+yc,12);

putpixel(y+xc,-x+yc,14);

putpixel(-y+xc,-x+yc,15);

putpixel(-y+xc,x+yc,6);}

void breshnamcircle(int xc, int yc, int r){

int x=0,y=r,d=3-(2\*r);

eight(xc,yc,x,y);

while(x<=y){

if(d<=0){

d=d+(4\*x)+6;}

Else{

d=d+(4\*x)-(4\*y)+10;

y=y-1;}

x=x+1;

eight(xc,yc,x,y); }}

int main(void){

int xc,yc,r,gd=DETECT,gm,errorcode;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

errorcode = graphresult();

if(errorcode != grOk){

printf("GRAPHICS error: %s\n", grapherrormsg(errorcode));

printf("press any key to halt:");

getch();

exit(1);}

printf("enter the values of xc antd yc:");

scanf("%d%d", &xc,&yc);

printf("Enter the radius:");

scanf("%d",&r);

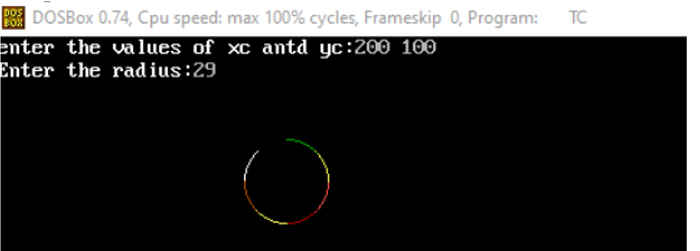
breshnamcircle(xc,yc,r);

getch();

closegraph();

return 0;}

### **Output:**



# **5. Write a program to draw a circle using Midpoint circle drawing algorithm.**

#include<graphics.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();}

### **Output:**

# **6. Write a program to draw a circle using Bresenham’s circle drawing**

# **algorithm.**

 #include<stdio.h>

 #include<conio.h>

 #include<graphics.h>

 #include<math.h>

 void eight(int xc, int yc, int x, int y)

 {

 putpixel(x+xc,y+yc,RED);

 putpixel(x+xc,-y+yc,YELLOW);

 putpixel(x+xc,-y+yc,GREEN);

 putpixel(-x+xc,y+yc,YELLOW);

 putpixel(y+xc,x+yc,12);

 putpixel(y+xc,-x+yc,14);

 putpixel(-y+xc,-x+yc,15);

 putpixel(-y+xc,x+yc,6);

 }

 void breshnamcircle(int xc, int yc, int r)

 {

 int x=0,y=r,d=3-(2\*r);

 eight(xc,yc,x,y);

 while(x<=y)

 {

 if(d<=0)

 {

 d=d+(4\*x)+6;

 }

 else

 {

 d=d+(4\*x)-(4\*y)+10;

 y=y-1;

 }

 x=x+1;

 eight(xc,yc,x,y);

 }

 }

 int main(void)

 {

 int xc,yc,r,gd=DETECT,gm,errorcode;

 initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

9

 errorcode = graphresult();

 if(errorcode != grOk)

 {

 printf("GRAPHICS error: %s\n", grapherrormsg(errorcode));

 printf("press any key to halt:");

 getch();

 exit(1);

 }

 printf("enter the values of xc antd yc:");

 scanf("%d%d", &xc,&yc);

 printf("Enter the radius:");

 scanf("%d",&r);

 breshnamcircle(xc,yc,r);

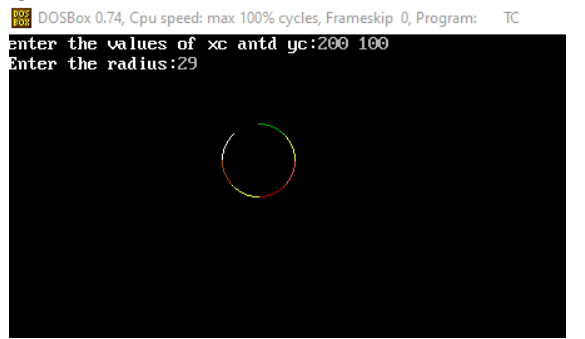
 getch();

 closegraph();

 return 0;

 }

### **Output:**



# **7. Write a program to perform 2-D transformation on a line.**

#include<stdio.h>

#include<graphics.h>

#include<stdlib.h>

#include<math.h>

#include<conio.h>

int x1,y1,x2,y2;

void translation()

{

int tx,ty,xn1,xn2,yn1,yn2;

printf("\n Enter the translation\n");

scanf("%d%d",&tx,&ty);

cleardevice();

outtextxy(400,100,"TRANSLATION");

xn1=x1+tx;

yn1=y1+ty;

xn2=x2+tx;

yn2=y2+ty;

line(x1,y1,x2,y2);

line(xn1,yn1,xn2,yn2);

getch();

}

void scaling()

{

int xn1,xn2,yn1,yn2;

float sx,sy;

printf("Enter the scaling factor");

scanf("%f%f",&sx,&sy);

cleardevice();

outtextxy(300,200,"SCALING");

xn1=x1\*sx;

yn1=y1\*sy;

xn2=x2\*sx;

yn2=y2\*sy;

line(x1,y1,x2,y2);

line(xn1,yn1,xn2,yn2);

getch();

11

}

void rotation()

{

int r;

float rx,xn1,xn2,yn1,yn2;

printf("\n enter the angle for rotation");

scanf("%d",&r);

cleardevice();

outtextxy(500,200,"ROTATION");

rx=(r\*3.14)/180;

xn1=x1\*cos(rx)-y1\*sin(rx);

yn1=y1\*cos(rx)+x1\*sin(rx);

xn2=x2\*cos(rx)-y2\*sin(rx);

yn2=y2\*cos(rx)+x2\*sin(rx);

line(x1,y1,x2,y2);

line(xn1,yn1,xn2,yn2);

getch();

}

void shearing()

{

float sh;

float xn1,xn2,yn1,yn2;

printf("\n Enter the value for shearing");

scanf("%f",&sh);

cleardevice();

outtextxy(500,100,"SHEARING");

xn1=x1+sh\*y1;

yn1=y1;

xn2=x2+sh\*y2;

yn2=y2;

line(x1,y1,x2,y2);

line(xn1,yn1,xn2,yn2);

getch();

}

void reflection()

{

int xn1,xn2,yn1,yn2;

12

cleardevice();

outtextxy(300,100,"REFLECTION");

if((x1<y1)^(x2<y2))

{

xn1=x1+50;

xn2=x2+50;

yn1=y1;

yn2=y2;

}

else

{

xn1=x1;

xn2=x2;

yn1=y1+50;

yn2=y2+50;

}

line(x1,y1,x2,y2);

line(xn1,yn1,xn2,yn2);

getch();

}

void get()

{

printf("\n Enter the coordinates x1,y1,x2,y2");

scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

outtextxy(200,100,"ORIGINAL OBJECT");

line(x1,y1,x2,y2);

getch();

}

void main()

{

int ch,gd=DETECT,gm;

initgraph(&gd,&gm,"c:\\tc\\bgi");

get();

do

{

cleardevice();

outtextxy(10,10,"1)TRANSLATION");

outtextxy(10,20,"2)SCALING");

13

outtextxy(10,30,"3)ROTATION");

outtextxy(10,40,"4)SHEARING");

outtextxy(10,50,"5)REFLECTION");

outtextxy(10,60,"6)EXIT");

outtextxy(10,70,"ENTER UR CHOICE:");

scanf("%d",&ch);

switch(ch)

{

case 1:

translation();

break;

case 2:

scaling();

break;

case 3:

rotation();

break;

case 4:

shearing();

break;

case 5:

reflection();

break;

case 6:

exit(0);

}

}while(ch<6);

}

### **Output:**

# **8. Write a program to perform 3-D transformation on a line.**

 #include<stdio.h>

#include<conio.h>

#include<graphics.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,o,x1,x2,y1,y2;

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"e:\tc\bgi");

setfillstyle(0,getmaxcolor());

maxx=getmaxx();

maxy=getmaxy();

midx=maxx/2;

midy=maxy/2;

axis();

bar3d(midx+50,midy-100,midx+60,midy-90,5,1);

printf("Enter translation factor");

scanf("%d%d",&x,&y);

axis();

printf("After translation:");

bar3d(midx+x+50,midy-(y+100),midx+x+60,midy-(y+90),5,1);

axis();

bar3d(midx+50,midy-100,midx+60,midy-90,5,1);

printf("Enter scaling factors");

scanf("%d%d%d", &x,&y,&z);

axis();

printf("After scaling");

bar3d(midx+(x\*50),midy-(y\*100),midx+(x\*60),midy-(y\*90),5\*z,1);

axis();

bar3d(midx+50,midy-100,midx+60,midy-90,5,1);

printf("Enter rotating angle");

scanf("%d",&o);

x1=50\*cos(o\*3.14/180)-100\*sin(o\*3.14/180);

y1=50\*sin(o\*3.14/180)+100\*cos(o\*3.14/180);

16

x2=60\*cos(o\*3.14/180)-90\*sin(o\*3.14/180);

y2=60\*sin(o\*3.14/180)+90\*cos(o\*3.14/180);

axis();

printf("After rotation about z axis");

bar3d(midx+x1,midy-y1,midx+x2,midy-y2,5,1);

axis();

printf("After rotation about x axis");

bar3d(midx+50,midy-x1,midx+60,midy-x2,5,1);

axis();

printf("After rotation about yaxis");

bar3d(midx+x1,midy-100,midx+x2,midy-90,5,1);

getch();

closegraph();

}

### **Output:**

