AMITY UNIVERSITY MAHARASHTRA

AMITY INSTITUTE OF INFORMATION TECHNOLOGY

COMPUTER GRAPHICS

LAB FILE

BCA

2019-2022



Prathmesh Patil

BCA SEM6

A71004819033

|  |  |  |
| --- | --- | --- |
| Sr.No. | Experiment | Date |
| 1 | Write a program to setup the graphics mode,plot a pixel.  Write a program to draw a line using DDA algorithm | 1/2/22 |
| 2 | Write a program to draw a line using Bressenham’s algorithm where the coordinates are provided by the user. | 8/2/22 |
| 3 | Write a C program to draw a circle using Bressenham’s Circle drawing algorithm. | 15/2/22 |
| 4 | Write a program to draw a circle using mid point circle algorithm.  Write a program to draw an emoji | 26/2/22 |
| 5 | Write a program to color a circle with boundary fill algorithm using 4 connected approach.  Write a rectangle to color a rectangle with boundary fill using 8 connected approach. | 1/3/22 |
| 6 | Write a C program to perform shearing in x direction by factor 4.  Write a C program to perform reflection about x=y for any given object | 20/4/22 |
| 7 | Write a program to perform rotation of a triangle.  Write a program to create an animation of a moving car | 28/4/22 |

**EXPERIMENT 1**

Write a program and setup the graphics mode, plot a pixel.

Code:-

//2-2-22 CG lab code 1

//printing a pixel

#include<iostream.h>

#include<stdio.h>

#include<graphics.h>

#include<stdlib.h>

#include<ctype.h>

#include<math.h>

#include<conio.h>

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "c:\\turboc3\\bgi");

putpixel(100, 100, WHITE);

putpixel(101,100,WHITE);

putpixel(101,101,WHITE);

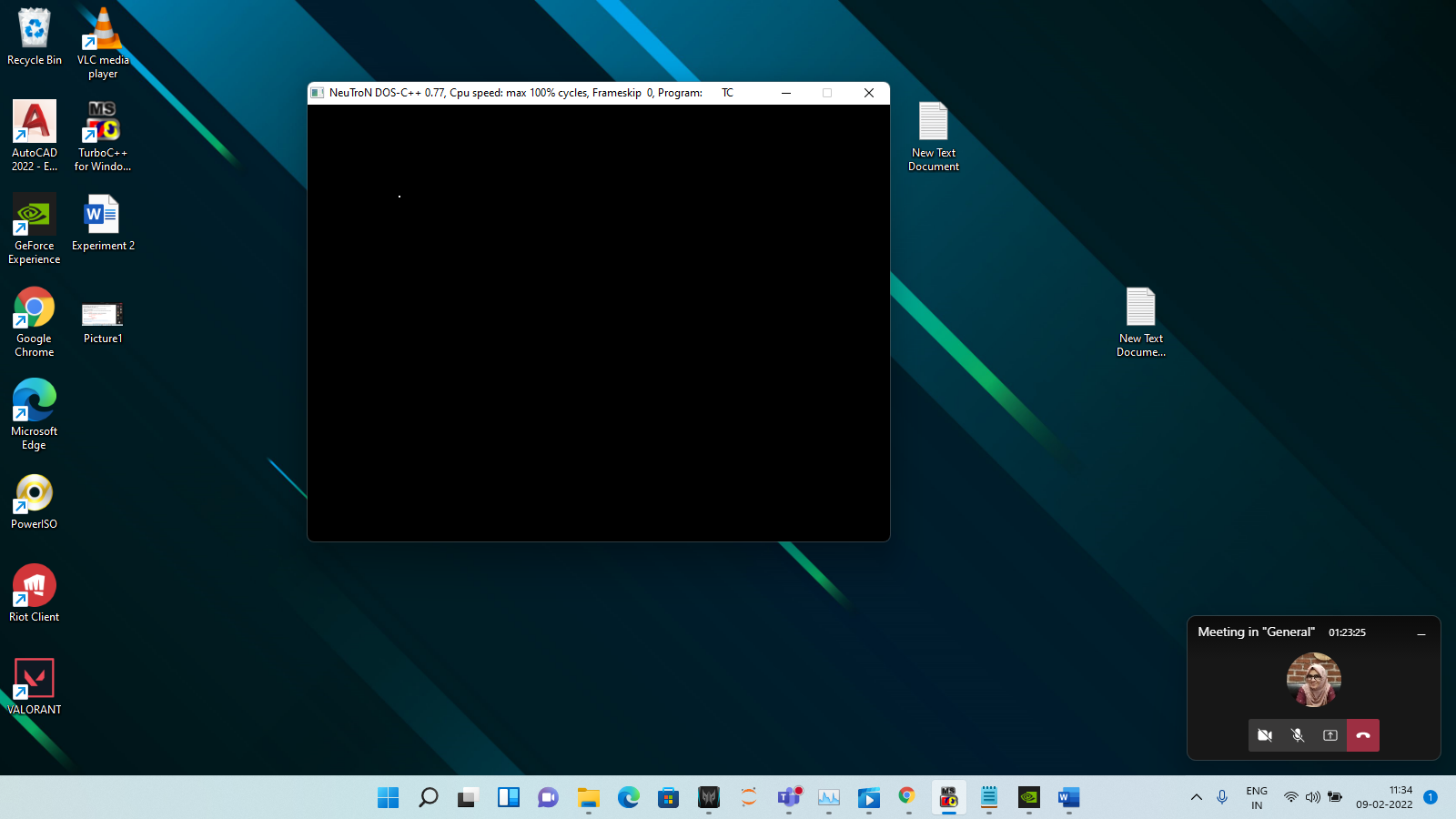
putpixel(100,101,WHITE);

getch();

return 0;

}

Output:-



Write a program to draw a line using DDA algorithm.

Code:-

//2-2-22 CG LAB CODE 2

//DDA ALGORITHM

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

#include<stdio.h>

#include<math.h>

#include<ctype.h>

#include<graphics.h>

#include<dos.h>

void DDA(int,int,int,int);

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "c:\\turboc3\\bgi");

DDA(150,150,300,300);

getch();

return 0;

}

int abs ()

{

int n;

return ( (n>0) ? n: (n\* (-1)));

}

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

{

int i,dx,dy,steps;

float X,Y,Xinc,Yinc;

dx = x1-x0;

dy = y1-y0;

steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

Xinc = dx / steps;

Yinc = dy / steps;

X=x0;

Y=y0;

for(i=0;i<=steps;i++)

{

putpixel (X,Y,GREEN);

X += Xinc;

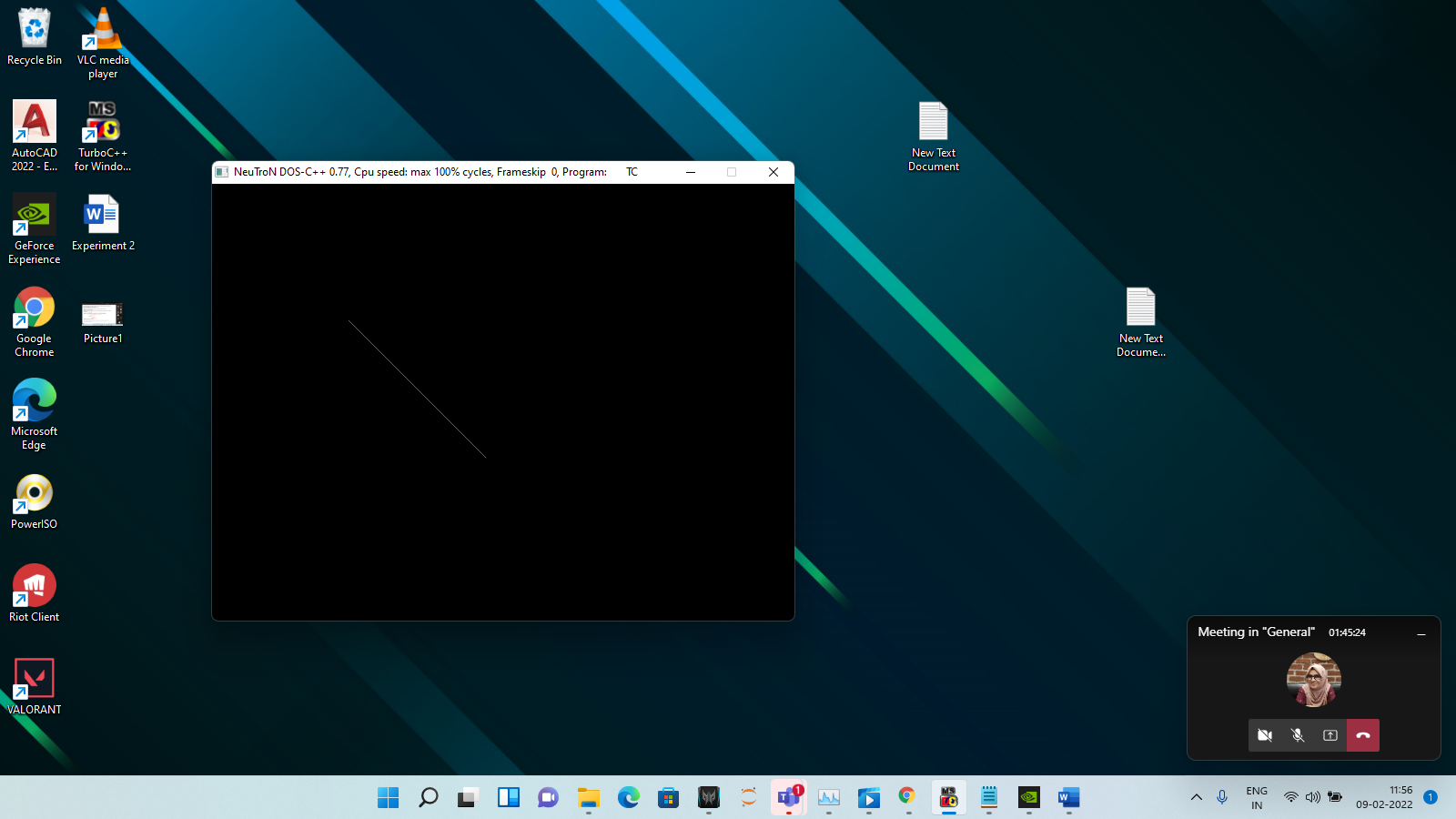
Y += Yinc;

delay(20);

}

}

Output:-



**EXPERIMENT 2**

Write a program to draw a line using Bressenham's algorithm where the coordinates are provided by user.

Code:-

//9-2-22

//CG LAB 2 CODE 3

//BRESENHAMS LINE ALGO

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<dos.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,YELLOW);

y=y+1;

p=p+2\*dy-2\*dx;

}

else

{

putpixel(x,y,YELLOW);

p=p+2\*dy;

}

x=x+1;

delay(50);

}

getch();

}

int main()

{

int gdriver = DETECT, gmode, error, x0, y0, x1, y1;

initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");

printf("Enter the first point Co-ordinates: ");

scanf("%d%d", &x0, &y0);

printf("Enter the second point Co-ordinates: ");

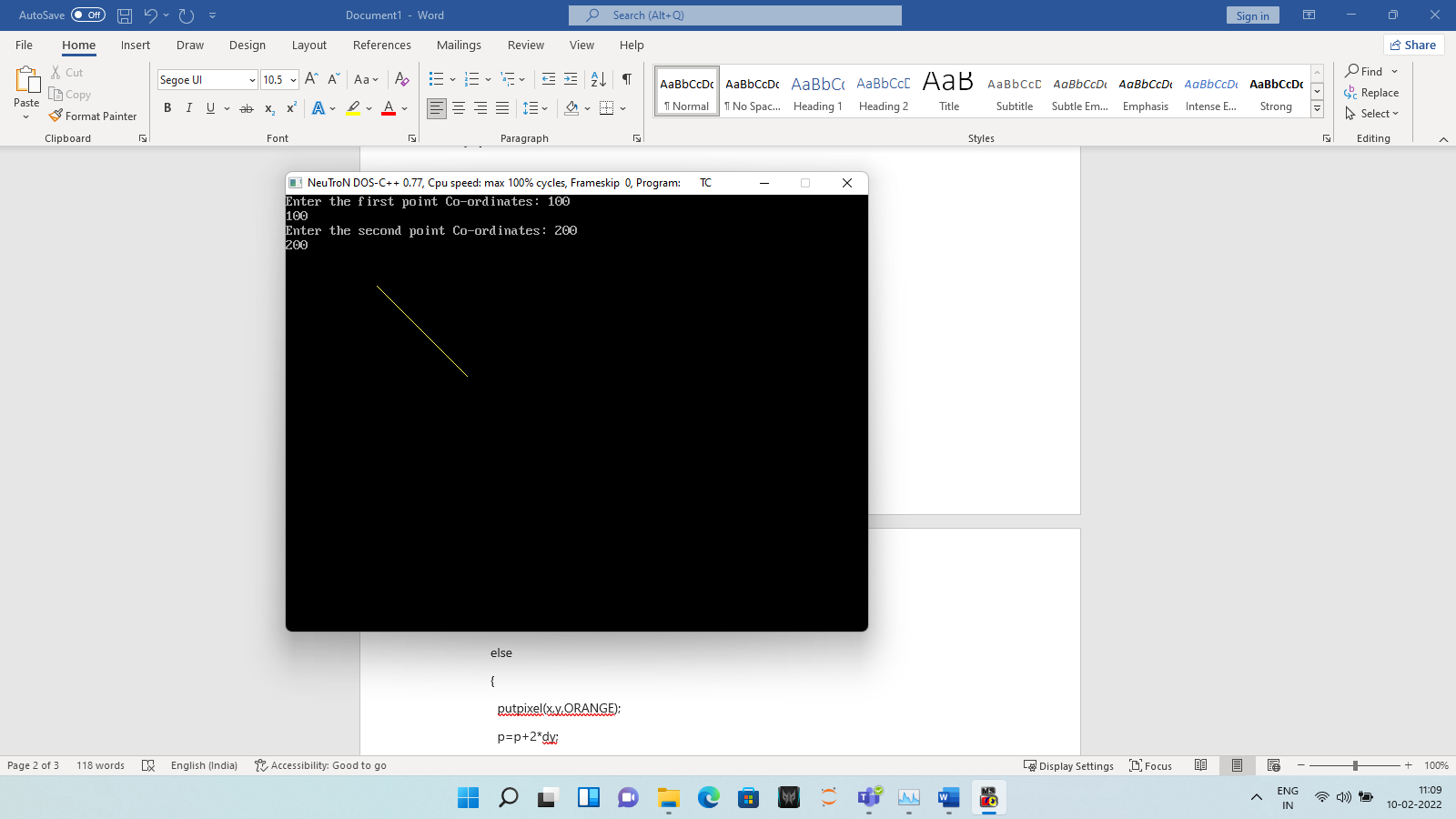
scanf("%d%d", &x1, &y1);

drawline(x0, y0, x1, y1);

return 0;

}

Output:-



**EXPERIMENT 3**

Write a C program to draw a circle using Bressenham's Circle drawing algorithm.(take the radius and center coordinates from user )

Code:-

// CG LAB 3

// BRESENHAMS CIRCLE ALGORITHM

#include<conio.h>

#include<graphics.h>

#include<dos.h>

#include<stdio.h>

void main()

{

int gd=DETECT,gm;

int xc,yc,x,y,r,p;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

printf("Enter the radius of the circle: ");

scanf("%d", &r);

printf("Enter the co-ordinates of center: ");

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

x=0;

y=r;

p=3-2\*4;

while(x<=y)

{

putpixel(xc+x,yc+y,YELLOW);delay(30);

putpixel(xc-x,yc-y,RED);delay(30);

putpixel(xc+x,yc-y,BLUE);delay(30);

putpixel(xc-x,yc+y,GREEN);delay(30);

putpixel(xc+y,yc+x,WHITE);delay(30);

putpixel(xc-y,yc-x,CYAN);delay(30);

putpixel(xc+y,yc-x,5);delay(30);

putpixel(xc-y,yc+x,6);delay(30);

if(p<=0)

{

p=p+4\*x+6;

}

else

{

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

y=y-1;

}

x=x+1;

}

getch();

closegraph();

}

Output:-

A screenshot of a computer

Description automatically generated with medium confidence

**EXPERIMENT 4**

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

Code:-

//CG LAB 4

//MID POINT CIRCLE ALGORITHM

#include<conio.h>

#include<graphics.h>

#include<dos.h>

#include<stdio.h>

void main()

{

int gd=DETECT,gm;

int xc,yc,x,y,r,p;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

printf("Enter the radius of the circle: ");

scanf("%d", &r);

printf("Enter the co-ordinates of center: ");

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

x=0;

y=r;

p=1-r;

while(x<=y)

{

putpixel(xc+x,yc+y,YELLOW);delay(30);

putpixel(xc-x,yc-y,RED);delay(30);

putpixel(xc+x,yc-y,BLUE);delay(30);

putpixel(xc-x,yc+y,GREEN);delay(30);

putpixel(xc+y,yc+x,WHITE);delay(30);

putpixel(xc-y,yc-x,CYAN);delay(30);

putpixel(xc+y,yc-x,5);delay(30);

putpixel(xc-y,yc+x,6);delay(30);

if(p<=0)

{

p=p+2\*x+1;

}

else

{

p=p+2\*(x-y)+5;

y=y-1;

}

x=x+1;

}

getch();

closegraph();

}

Output:-

Graphical user interface, application

Description automatically generated

1. Write a program to draw an emoji.

Code:-

//CG LAB 4

//EMOJI ALGORITHM

#include<conio.h>

#include<graphics.h>

#include<dos.h>

#include<stdio.h>

int main()

{

int gd=DETECT,gm;

int xc,yc,x,y,r,p;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

setcolor(YELLOW);

circle(200,200,100);

circle(160,170,20);

delay(1000);

circle(240,170,20);

delay(1000);

line(150,230,250,230);

delay(1000);

arc(200,230,180,360,50);

delay(1000);

getch();

closegraph();

return 0;

}

Output:-

Graphical user interface, application, PowerPoint

Description automatically generated

**EXPERIMENT 5**

1. Write a program to color a circle with Boundary fill algorithm using 4 connected approach.

Code:-

//2-3-2022

//CG LAB 5

//BOUNDARY FILL ALGORITHM USING 4 CONNECTED APPROACH

#include<stdio.h>

#include<graphics.h>

#include<dos.h>

void boundaryfill(int x, int y, int f, int b)

{

if(getpixel(x,y)!=f && getpixel(x,y)!=b)

{

putpixel(x,y,f);

boundaryfill(x,y+1,f,b);

delay(5);

boundaryfill(x+1,y,f,b);

delay(5);

boundaryfill(x-1,y,f,b);

delay(5);

boundaryfill(x,y-1,f,b);

delay(5);

}

}

int main()

{

int gd=DETECT,gm;

int x=150, y=200 , r=30;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

circle(x,y,r);

boundaryfill(x+1,y+1,14,15);

closegraph();

return 0;

}

Output:-

Graphical user interface, application

Description automatically generated

1. Write a a program to color a rectangle with Boundary fill algorithm using 8 connected approach.

Code:-

//2-3-22

//CG LAB 5

//8 CONNECTED APPROACH USING BOUNDARY FILL ALGORITHM

#include<stdio.h>

#include<graphics.h>

#include<dos.h>

void boundaryfill(int x, int y, int f, int b)

{

if(getpixel(x,y)!=f && getpixel(x,y)!=b)

{

putpixel(x,y,f);

boundaryfill(x,y+1,f,b);

delay(5);

boundaryfill(x+1,y,f,b);

delay(5);

boundaryfill(x-1,y,f,b);

delay(5);

boundaryfill(x,y-1,f,b);

delay(5);

boundaryfill(x+1,y+1,f,b);

delay(5);

boundaryfill(x-1,y+1,f,b);

delay(5);

boundaryfill(x+1,y-1,f,b);

delay(5);

boundaryfill(x-1,y-1,f,b);

delay(5);

}

}

int main()

{

int gd=DETECT,gm;

int l=200, t=200, r=250, b=250;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

rectangle(l,t,r,b);

boundaryfill(202,202,14,15);

closegraph();

return 0;

}

Output:-

Graphical user interface, application

Description automatically generated

**EXPERIMENT 6**

Write a C program to perform shearing in x direction by factor 4.

Code:-

#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

float shx,shy;

initgraph(&gd,&gm,"C:\\TurboC3\\BGI");

printf("Enter shearing factor along x-axis :");

scanf("%f",&shx);

line(50,100,70,100);

line(70,100,70,70);

line(70,70,50,70);

line(50,70,50,100);

setcolor(12);

line((50+(100\*shx)),100,(70+(100\*shx)),100);

line((70+(100\*shx)),100,(70+(70\*shx)),70);

line((70+(70\*shx)),70,(50+(70\*shx)),70);

line((50+(70\*shx)),70,(50+(100\*shx)),100);

getch();

}

Output:-

A screenshot of a computer

Description automatically generated

Write a C program to perform Reflection about x=-y for any given object.

(take the object as per your choice)

Code:-

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

#include<process.h>

void disp(float v[][3])

{

float xmax, ymax ;

xmax = getmaxx()/2 ;

ymax = getmaxy()/2 ;

int i = 0;

while(i<2)

{

line(xmax+v[i][0], ymax-v[i][1], xmax+v[i+1][0], ymax-v[i+1][1]);

i++ ;

}

i = 2 ;

line(xmax+v[i][0],ymax-v[i][1],xmax+v[0][0],ymax-v[0][1]);

setcolor(RED);

line(0,ymax,xmax\*2,ymax); // horizontal x axis ;

line(xmax,0,xmax,ymax\*2); // vertical y axis

setcolor(WHITE);

}

void multiply (float b[][3],float v[][3],float a[][3])

{

int i,j,k;

for(i=0; i<3; i++)

for(j=0; j<3; j++)

a[i][j]=0;

for(i=0; i<3; i++)

for(j=0; j<3; j++)

for(k=0; k<3; k++)

{

a[i][j] = a[i][j] + (v[i][k] \* b[k][j]);

}

}

void reflect(float v[][3])

{

float b[10][3],a[10][3];

int i=0, j;

clrscr();

cleardevice();

disp(v);

for(i=0;i<3;i++)

for(j=0;j<3;j++)

{

b[i][j]=0;

if(i==j)

b[i][j]=1;

}

b[1][1]=-1; //About X-axis ;

multiply(b,v,a);

setcolor(YELLOW);

disp(a);

}

void main()

{

clrscr();

int gd=DETECT, gm ;

float v[10][3] ;

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

printf("Enter the vertex coordinate of triangle : \n");

for (int i = 0; i < 3; i++)

{

printf("Enter the coordinate v%d :\n", i+1);

scanf("%f%f", &v[i][0], &v[i][1]);

v[i][2] = 1;

}

clrscr();

cleardevice();

setcolor(BLACK);

disp(v);

reflect(v);

getch();

closegraph();

}

Output:-

Graphical user interface, application, PowerPoint

Description automatically generated

**EXPERIMENT 7**

1. Write a program to perform rotation of a triangle.

Code:-

#include<conio.h>

#include<stdio.h>

#include<math.h>

#include<graphics.h>

void main(){

int gd = DETECT,gm;

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

int x1=100,y1=100,x2=150,y2=150,x3=200,y3=100;

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

float r;

printf("Enter the rotation angle: ");

scanf("%f",&r);

r=(3.14/180)\*r;

int rx1 = x1\*cos(r)+y1\*sin(r);

int rx2= x2\*cos(r)+y2\*sin(r);

int rx3= x3\*cos(r)+y3\*sin(r);

int ry1= x1\*sin(r)-y1\*cos(r);

int ry2= x2\*sin(r)-y2\*cos(r);

int ry3= x3\*sin(r)-y3\*cos(r);

line(rx1,ry1,rx2,ry2);

line(rx2,ry2,rx3,ry3);

line(rx3,ry3,rx1,ry1);

getch();

closegraph();

}

Output:-

A screenshot of a computer

Description automatically generated with medium confidence

1. Write a program to create an animation of moving car.

Code:-

#include<conio.h>

#include<graphics.h>

#include<dos.h>

#include<stdio.h>

#include<conio.h>

void main()

{

int gd=DETECT,gm;

int x,y,r,p;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

int i=0;

while(i<=800){

arc(65+i,200,0,180,50);

circle(30+i,250,10);

circle(100+i,250,10);

rectangle(0+i,200,140+i,240);

line(10,260,800,260);

if(i>=600){

break;

}

i=i+2;

clearviewport();

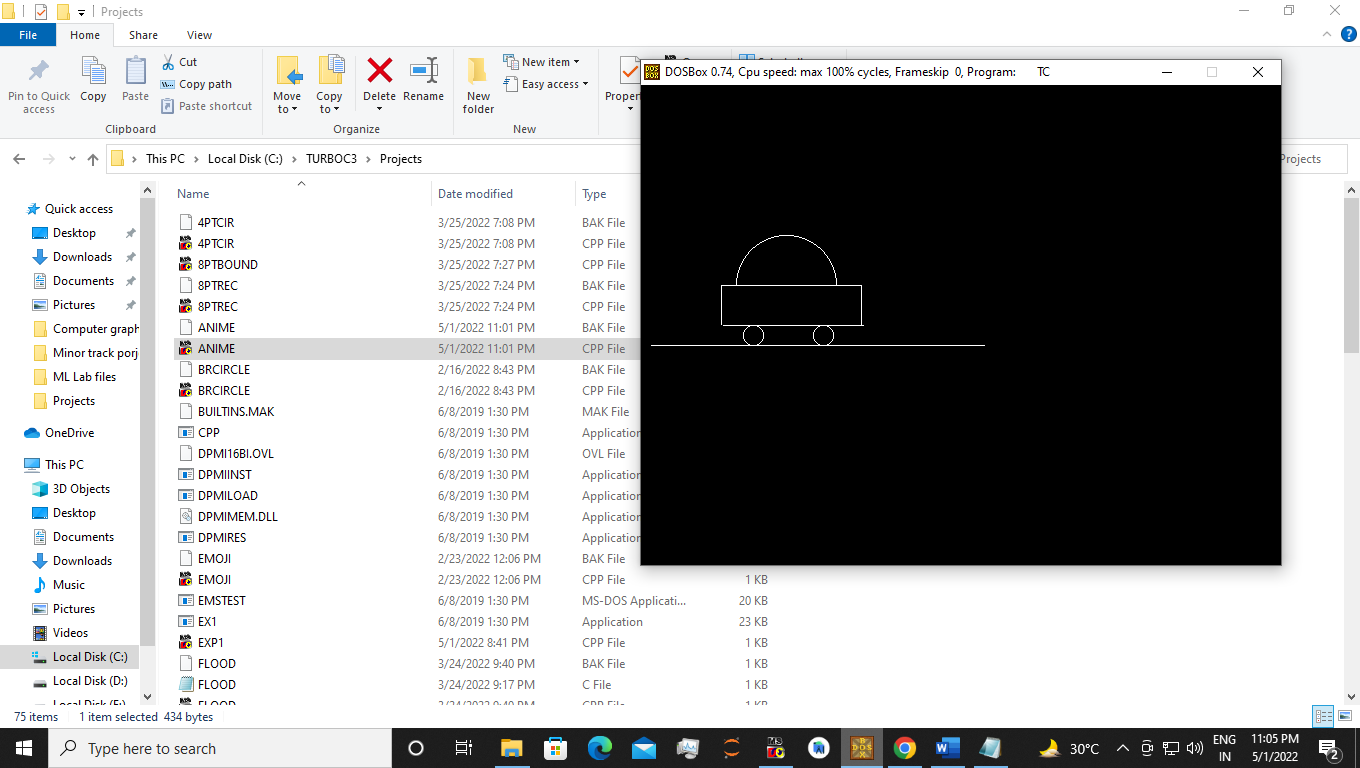
}

getch();

closegraph();

}

Output:-



A screenshot of a computer

Description automatically generated with medium confidence