**TITLE: - BASIC GRAPHICS FUNCTION.**

**OBJECTIVE: -**

* To understand the graphics mode and graphics function in C++
* To draw line, curve, point, arc, square, rectangle, triangle, ellipse, etc.

**THEORY: -**

Computer graphics is a field focused on creating and rendering visual content using computers. It has significant applications in entertainment, design, education, and scientific research. This technology enables the creation of realistic simulations, immersive virtual environments, and interactive interfaces. In industries like gaming and film, computer graphics plays a key role in generating captivating visual experiences. It also facilitates architectural and product design by allowing for visualization and iteration of concepts. Moreover, computer graphics aids in data visualization and scientific research, enabling the representation and understanding of complex phenomena. Overall, computer graphics is a powerful tool that enhances creativity, communication, and innovation in various domains.

**There are two main areas within computer graphics:**  
**2D Graphics:** This involves working with two-dimensional shapes, images, and text. It includes tasks such as creating and editing images using software like Adobe Photoshop, designing user interfaces, producing digital illustrations, and typesetting.

**3D Graphics:** This focuses on creating and manipulating three-dimensional objects and environments. It involves modelling objects using geometric primitives or capturing real-world objects with 3D scanners, applying textures and materials, setting up virtual cameras, and rendering realistic or stylized images or animations using software like Autodesk Maya, Blender, or Cinema 4D.

The `graphics.h` library is a graphics library commonly used in older versions of the C and C++ programming languages. It provides functions for drawing basic shapes and creating graphical applications on the screen.

Here is an overview of some common shapes and functions available in `graphics.h`:

1. Line: You can draw a line using the `line()` function. It takes four arguments: the starting coordinates (x1, y1) and the ending coordinates (x2, y2) of the line.

2. Rectangle: The `rectangle()` function is used to draw a rectangle. It takes four arguments: the coordinates of the top-left corner (left, top) and the coordinates of the bottom-right corner (right, bottom) of the rectangle.

3. Circle: To draw a circle, you can use the `circle()` function. It takes three arguments: the center coordinates (x, y) of the circle and the radius (r).

4. Ellipse: The `ellipse()` function is used to draw an ellipse. It takes five arguments: the coordinates of the bounding rectangle (left, top, right, bottom) and the angle of rotation (start\_angle, end\_angle).

5. Text: You can display text on the screen using the `outtextxy()` function. It takes two arguments: the x and y coordinates of the starting point and a string of characters to be displayed.

These are just a few examples of the functions available in `graphics.h` for drawing shapes. However, please note that the `graphics.h` library is specific to certain compilers and operating systems, and it may not be supported in newer versions or on all platforms. For modern graphics programming, other libraries like OpenGL, DirectX, or more advanced graphics frameworks are typically used.

# WAP to draw point, line, circle, ellipse, triangle, rectangle, square, arc using C and C++.

#include<iostream.h>

#include<graphics.h>

#include<conio.h>

#include<process.h>

#include<math.h>

void call();

void main()

{

clrscr();

int choice;

int gm=DETECT, gd=0;

initgraph(&gm,&gd,"..\\BGI");

int xc=getmaxx()/2;

int yc=getmaxy()/2;

cout<<"\n\n\n";

cout<<"[1]. Point"<<endl;

cout<<"[2]. Line"<<endl;

cout<<"[3]. Circle"<<endl;

cout<<"[4]. Ellipse"<<endl;

cout<<"[5]. Rectangle"<<endl;

cout<<"[6]. Square"<<endl;

cout<<"[7]. Arc"<<endl;

cout<<"[8]. Triangle"<<endl;

cout<<"[0]. Exit"<<endl;

cout<<endl<<"Enter Choice:";

cin>>choice;

switch(choice){

case 1: cleardevice();

outtextxy(xc-60,yc+20,"Point at centre");

putpixel(xc,yc,WHITE);

call();

case 2: cleardevice();

outtextxy(xc-60,yc+20,"Line at centre");

line(xc-50,yc,xc+50,yc);

call();

case 3: cleardevice();

int radius;

cout<<"Enter the radius";

cin>>radius;

outtextxy(xc-60,yc+80,"Circle");

circle(xc,yc,radius);

call();

case 4: cleardevice();

int xr,yr;

cout<<"Radius in X-axis"; cin>>xr;

cout<<"Radius in Y-zxis"; cin>>yr;

ellipse(xc,yc,0,360,xr,yr);

call();

case 5: cleardevice();

cout<<"Length"; cin>>xr;

cout<<"Breadth"; cin>>yr;

rectangle(xc-xr,yc-yr,xc+xr,yc+yr);

call();

case 6: cleardevice();

cout<<"Length:"; cin>>xr;

rectangle(xc-xr,yc-xr,xc+xr,yc+xr);

call();

case 7: cleardevice();

int r;

cout<<"Starting Angle: "; cin>>xr;

cout<<"Ending Angle: "; cin>>yr;

cout<<"Radius: "; cin>>r;

arc(xc,yc,xr,yr,r);

call();

case 8:{ cleardevice();

int a,b,c;

cout<<"Height of triangle: "; cin>>a;

cout<<"Length of base: "; cin>>b;

line(xc-b/2,yc+a/2,xc+b/2,yc+a/2);

line(xc-b/2,yc+a/2,xc,yc-a);

line(xc+b/2,yc+a/2,xc,yc-a);

outtextxy(xc-30,yc+100,"Triangle");

call(); }

case 0:

exit(0);

default:

outtextxy(xc-50,yc,"Invalid Option");

call();

}}

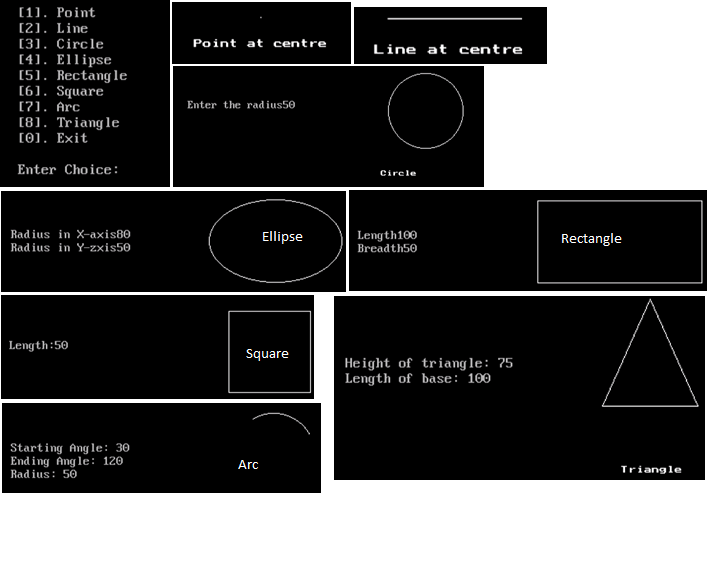
void call(){

getch();

cleardevice();

main();}

OUTPUT: -



CONCLUSION: -

In conclusion, In summary, the graphics.h library is a graphics library used in older versions of C and C++. It provides functions for drawing basic shapes such as lines, rectangles, circles, and ellipses. Additionally, it allows for displaying text on the screen. However, it's important to note that graphics.h is specific to certain compilers and operating systems, and newer graphics libraries are typically used for modern graphics programming.

