

Computer Graphics Lab

Day-1

##Graphics configure for code::blocks

1. First download WinBGIm GCC47 file from.
2. unzip the WinBGIm_GCC47.zip
3. open "graphics.h" file. and replace 302 line with "int left=0, int top=0, int right=INT_MAX, int bottom=INT_MAX," line.
4. copy & paste "graphics.h" and "winbgim.h" into compiler include folder.
5. copy & paste libbgi.a file into compiler lib folder.
6. Now open code::blocks.
7. Go to settings -> Compiler -> Linker settings
8. Left side Linker libraries click add, and browse the file "libbgi.a" and add. In my pc this is "C:\Program Files (x86)\CodeBlocks\MinGW\lib\libbgi.a"
9. In Right side Other linker options text field copy paste this, "-lbgi -lgdi32 -lcomdlg32 -luuid -oleaut32 -ole32" (without quotes)
10. Press ok.

Computer Graphics Lab

Day-1

What are initgraph, gd and gm?

- gd = graphdriver;
- gm = graphmode;

Syntax for initgraph:

```
void initgraph (int *graphdriver, int *graphmode, char *pathtodriver) ;
```

Description for initgraph:

initgraph

initgraph is used to initialize the graphics system by loading a graphics driver from disk and thereby putting the system into graphics mode.

To start the graphics system, we first call the initgraph function. initgraph may use a particular graphics driver and mode, or it may auto-detect and pick the corresponding driver at runtime, according to our needs.

If we tell initgraph to autodetect, it calls detectgraph to select a graphics driver and mode. It also resets all graphics settings to their defaults values like current position, color, viewport and so on and also resets graph result to 0.

Normally, memory is allocated by initgraph to load a particular graphics driver through _graphgetmem, then it loads the appropriate BGI file from disk.

pathtodriver

pathtodriver denotes the directory path where initgraph must look for graphic drivers. initgraph first goes through the directed path to look for the files and if they are not found there, it goes to the current directory. The graphic driver must files must be present in the current directory if the pathtodriver is null.

graphdriver

*graphdriver is the integer that specifies which graphics driver is to be used. We can give it a value using a constant of the graphics_drivers enum type, which is defined in graphics.h and listed below.

graphics_drivers constant	Numeric value
DETECT	0 (requests autodetect)
CGA	1
MCGA	2
EGA	3
EGA64	4
EGAMONO	5
IBM8514	6
HERCMONO	7

Computer Graphics Lab

Day-1

ATT400	8
VGA	9
PC3270	10

graphmode

*graphmode is also an integer that specifies the initial graphics mode.

graphdriver and graphmode must be given proper values from the tables or we will get absurd and unexpected results. The exception here is when graphdriver = DETECT. In this case, initgraph sets *graphmode to the highest resolution available for the detected driver.

Draw a line in C++ graphics

graphics.h library is used to include and facilitate graphical operations in program. graphics.h functions can be used to draw different shapes, display text in different fonts, change colors and many more. Using functions of graphics.h you can make graphics programs, animations, projects and games. You can draw circles, lines, rectangles, bars and many other geometrical figures. You can change their colors using the available functions and fill them.

Explanation : The header file graphics.h contains line() function which is described below :

Declaration : **void line(int x1, int y1, int x2, int y2);**

line function is used to draw a line from a point(x1,y1) to point(x2,y2) i.e. (x1,y1) and (x2,y2) are end points of the line.

Draw a line from current point to point(x,y) using lineto() function

lineto() is a library function of graphics.h and it can be used to draw a line from current point to specified point.

For example, if you want to draw a line from current point to point(x,y), you have to use lineto() function like lineto(x,y)

Syntax (Declaration of line() function in C)

```
lineto(int x, int y);
```

putpixel function in c

putpixel function plots a pixel at location (x, y) of specified color.

Declaration: void putpixel(int x, int y, int color);

For example, if we want to draw a GREEN color pixel at (35, 45) then we will write putpixel(35, 35, GREEN);

Draw Rectangle in C graphics

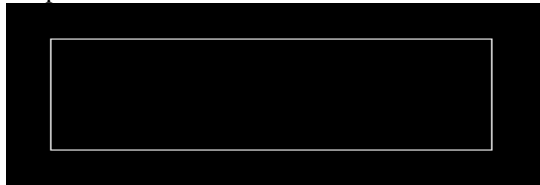
rectangle() is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle. left specifies the X-coordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner.

Syntax :

```
rectangle(int left, int top, int right, int bottom);
```

Examples: Input : left = 150, top = 250, right = 450, bottom = 350;

Output :



Input : left = 150, top = 150, right = 450, bottom = 450;

Output :



Draw circle in C graphics

The header file `graphics.h` contains **circle()** function which draws a circle with center at (x, y) and given radius.

Syntax :

```
circle(x, y, radius);
```

where,

(x, y) is center of the circle.

'radius' is the Radius of the circle.

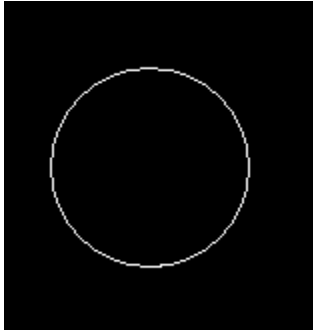
Examples :

Computer Graphics Lab

Day-1

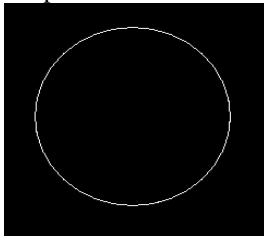
Input : $x = 250$, $y = 200$, radius = 50

Output :



Input : $x = 300$, $y = 150$, radius = 90

Output :



Draw ellipse in C graphics

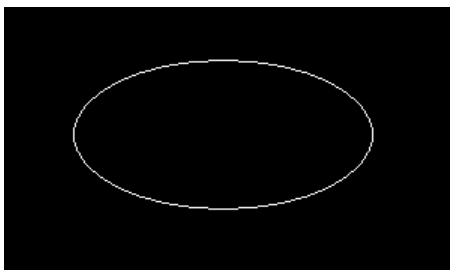
Program to draw ellipse in C using graphics.h header file.

graphics.h library is used to include and facilitate graphical operations in program. C graphics using graphics.h functions can be used to draw different shapes, display text in different fonts, change colors and many more. Using functions of graphics.h you can make graphics programs, animations, projects and games. You can draw circles, lines, rectangles, bars and many other geometrical figures. You can change their colors using the available functions and fill them.

Examples:

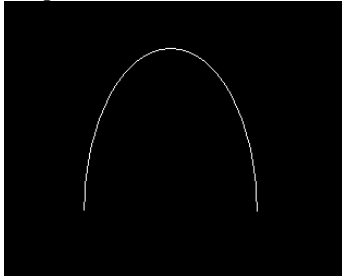
Input : $x=250$, $y=200$, start_angle = 0,
end_angle = 360, x_rad = 100, y_rad = 50

Output :



Input : $x=250$, $y=200$, start_angle = 0,
end_angle = 180, x_rad = 80, y_rad = 150

Output :



Explanation : The header file `graphics.h` contains `ellipse()` function which is described below :

`void ellipse(int x, int y, int start_angle, int end_angle, int x_radius, int y_radius)`

In this function `x`, `y` is the location of the ellipse. `x_radius` and `y_radius` decide the radius of form `x` and `y`.

`start_angle` is the starting point of angle and `end_angle` is the ending point of angle. The value of angle can vary from 0 to 360 degree

setcolor function in c

Declaration: `void setcolor(int color);`

In Turbo Graphics each color is assigned a number. Total 16 colors are available. Strictly speaking number of available colors depends on current graphics mode and driver.

For Example :- BLACK is assigned 0, RED is assigned 4 etc.

Remember that default drawing color is WHITE.

Floodfill function

Declaration: `void floodfill(int x, int y, int border);`

floodfill function is used to fill an enclosed area. Current fill pattern and fill color is used to fill the area. `(x, y)` is any point on the screen if `(x,y)` lies inside the area then inside will be filled otherwise outside will be filled, `border` specifies the color of boundary of area. To change fill pattern and fill color use `setfillstyle`. Code given below draws a circle and then fills it.

setfillstyle function in c

Computer Graphics Lab

Day-1

setfillstyle function sets the current fill pattern and fill color.

Declaration: void setfillstyle(int pattern, int color);

Below is the table showing INT VALUES corresponding to Colors :

COLOR	INT VALUES
BLACK	0
BLUE	1
GREEN	2
CYAN	3
RED	4
MAGENTA	5
BROWN	6
LIGHTGRAY	7
DARKGRAY	8
LIGHTBLUE	9
LIGHTGREEN	10
LIGHTCYAN	11
LIGHTRED	12
LIGHTMAGENTA	13
YELLOW	14
WHITE	15

Below is the table showing INT VALUES corresponding to Patterns :

PATTERN	INT VALUES
EMPTY_FILL	0
SOLID_FILL	1
LINE_FILL	2
LTSLASH_FILL	3
SLASH_FILL	4
BKSLASH_FILL	5
LTBKSLASH_FILL	6
HATCH_FILL	7
XHATCH_FILL	8
INTERLEAVE_FILL	9
WIDE_DOT_FILL	10
CLOSE_DOT_FILL	11
USER_FILL	12

Functions of graphics.h

arc

bar

bar3d

circle

cleardevice

closegraph

Computer Graphics Lab

Day-1

drawpoly
ellipse
fillellipse
fillpoly
floodfill
getarccords
getbkcolor
getcolor
getdrivername
getimage
getmaxcolor
getmaxx
getmaxy
getpixel
getx
gety
graphdefaults
grapherrormsg
imagesize
line
lineto
linerel
moveto
moverel
outtext
outtextxy
pieslice
putimage
putpixel
rectangle
sector
setbkcolor
setcolor
setfillstyle
setlinestyle
settextstyle
setviewport
textheight
textwidth