Computer Graphics Lab Manual

1. DDA Algorithm
2. Bresenham's Algorithm
3. Midpoint Circle Algorithm
4. Mid-point Ellipse Algorithm
5. Cohen Sutherland Line Clipping Algorithm
6. Liang Barsky Line Clipping Algorithm
7. 2D Transformations
8. Bezier Curves
9. Flood Filling
10. Boundary Filling
11.Introduction to OPENGL

Graphics programming in C

- For any graphics program, include **graphics.h** header file
- Initialize the graphics drivers on the computer using **initgraph** method of **graphics.h** library

void initgraph(int *graphicsDriver, int *graphicsMode, char *driverDirectoryPath);

- **graphicsDriver :** It is a pointer to an integer specifying the graphics driver to be used. DETECT macro of graphics.h library that instruct compiler for auto detection of graphics drive
- **graphicsMode :** It is a pointer to an integer that specifies the graphics mode to be used. If *gdriver is set to **DETECT**, then initgraph sets *gmode to the highest resolution available for the detected driver.
- **driverDirectoryPath :** It specifies the directory path where graphics driver files (BGI files) are located. If the directory path is not provided, then it will search for driver files in the current working directory directory. You have to change the path of the BGI directory accordingly where your Turbo C++ compiler is installed.

void closegraph();

- closegraph function closes the graphics mode, deallocates all memory allocated by graphics system and restores the screen to the mode it was in before you called initgraph

Built-in Functions

NAME	**FUNCTION**	**SYNTAX**
arc	Draws a circular arc.	***void arc(int x, int y, int stangle, int endangle, int radius);***
circle	Draws a circle.	***void circle(int x, int y, int radius);***
bar	Draws a bar.	***void bar(int left, int top, int right, int bottom);***
closegraph	Shuts down the graphics system.	***void closegraph(void);***
ellipse	Draws an elliptical arc.	***void ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius);***
floodfill	Floodfills a bound region.	***void floodfill(int x, int y, int border);***
getbkcolor	Returns the current background color.	***int getbkcolor(void);***
getgraphmode	Returns the current graphics mode.	***int getgraphmode(void);***
getmaxcolor	Returns the maximum color value.	***int getmaxcolor(void);***
getmaxx	Returns the maximum X screen coordinate.	***int getmaxx(void);***
getmaxy	Returns the maximum Y screen coordinate.	***int getmaxy(void);***

gety	Returns the current position's Y coordinate.	***int gety(void);***
getx	Returns the current position's X coordinate.	***int getx(void);***
detectgraph	Determines graphics driver and mode to use by checking the hardware.	***void detectgraph(int *graphdriver, int *graphmode);***
fillellipse	Draws and fills an ellipse.	***void fillellipse(int x, int y, int xradius, int yradius);***
getarccoords	Gets coordinates of the last call to arc.	***void getarccoords(struct arccoords *arccoords);***
getcolor	Returns the current drawing color.	***int getcolor(void);***
getfillpattern	Copies a user-defined fill pattern into memory.	***void getfillpattern(char *pattern);***
getmaxmode	Returns maximum graphics mode number for current driver.	***int getmaxmode(void);***
drawpoly	Draws the outline of a polygon.	***void drawpoly(int numpoints, int *polypoints);***
fillpoly	Draws and fills a polygon.	***void fillpoly(int numpoints, int *polypoints);***
clearviewport	Clears the current viewport.	***void clearviewport(void);***
getpixel	Gets the color of a specified pixel.	***unsigned getpixel(int x, int y);***

grapherrormsg	Returns a pointer to an error message string.	***char *grapherrormsg(int errorcode);***
lineto	Draws a line from the current position to (x, y).	***void lineto(int x, int y);***
line	Draws a line between two specified points.	***void line(int x1, int y1, int x2, int y2);***
initgraph	Initializes the graphics system.	***void initgraph(int *graphdriver, int *graphmode, char *pathtodriver);***
rectangle	Draws a rectangle.	***void rectangle(int left, int top, int right, int bottom);***
putpixel	Plots a pixel at a specified point.	***void putpixel(int x, int y, int color);***
imagesize	Returns the number of bytes required to store a bit image.	***unsigned imagesize(int left, int top, int right, int bottom);***
moveto	Moves the current position to (x, y).	***void moveto(int x, int y);***
setcolor	Sets the current drawing color.	***void setcolor(int color);***
setgraphmode	Sets the system to graphics mode, clears the screen.	***void setgraphmode(int mode);***
textwidth	Returns the width of a string in pixels.	***int textwidth(char *textstring);***
textheight	Returns the height of a string in pixels.	***int textheight(char *textstring);***