DATE 20/8/20
PAGE ①

Experiment - ① @

Aim:- To study and prepare a list of graphic functio

THEORY

C graphics using graphic.h function or WinBGIM (Windows 7) can be used to draw different shape display text in different fonts, change colors and many more. Using functions of graphics.h in Turbo C compiler, you can make graphic programs, animations, projects and games. You can draw lines, circles, rectangles, bars and many other geometrical shapes. You can change their colors using the available functions and fill them. Following is a list of functions of graphics.h header file.

initgraph(): initgraph() initializes the graphic system by loading a graphics driver from the disk then putting the system into graphics mode. It also resets graphic settings and the graph. It has 2 parameters "gd" which is the graphics driver and "gm" to be used and "gm" in what graphics mode is should the graph be initialised in.

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

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

Syntax: putpixel (int x, int y, int color);

setbkcolor(): setbkcolor() function is used to set the background color in graphics mode. The default background color is black.

Syntax: void setbkcolor (int color);

setlinestyle(): This function sets the style for all lines drawn by line, line to, rectangle, draw poly and so on.

Syntax: void setlinestyle (int line style, unsigned pattern, int thickness);

Setcolor(): setcolor() function is used to set the current drawing color to new color.

Syntax: void setcolor (int color);

rectangle(): rectangle() is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle.

Syntax: void rectangle (int left, int top, int right, int bottom);

getx(): getx() function returns the x coordinate of the current position.

Syntax: int getx();

gety(): gety() function returns the y coordinate of the current position.

Syntax: int gety();

getmaxx(): This function returns the maximum x coordinate for current graphics mode and driver.

Syntax: int getmaxx();

getmaxy(): This function returns the maximum y coordinate for current graphics mode and driver.

Syntax: int getmaxy();

moveto(): moveto() function changes the current position to (x,y).

Syntax: void moveto(int x, int y);

setfillstyle(): This function sets the current fill patter and fill color.

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

floodfill(): It is used to fill an enclosed area. Current fill pattern and fill color is used to fill the area.

Syntax: void floodfill (int x, int y, int border_color);

getcolor(): getcolor() function returns the current drawing color and default drawing color is white.

Syntax: int getcolor();

getpixel(): getpixel() function returns the color of pixel present at location (x, y).

Syntax: int getpixel (int x, int y)

closegraph(): This 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.

Syntax: void closegraph();

- line(): 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 the end points of the line.

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

- circle(): circle() function draws a circle with center at (x, y) and given radius.

  Syntax: void circle(int x, int y, int radius);

DATE 20/8/20
PAGE (4)

Experiment - 1 (b)

Aim:- Write a program to make a hut and car using built in graphics

DESCRIPTION

The following program is using <graphics.h> library of Turbo C++ (v 3.2.2.0). The program has 2 defined functions

- hut (int x, int y) : Draws the hut. x and y are the midpoint of the graph.
- car (int x, int y) : Draws the car. x and y are the midpoint of graph

PROGRAM

```
# include <iostream.h>
# include <conio.h>
# include <graphics.h>
# include <maths.h>
# include <dos.h>

# define PI 3.1415926
# define rad(x) (2* PI* x)/360
```

```
void hut (int x, int y) {

// Ground
Setcolor ( BROWN);
Setfillstyle ( INTERLEAVE_FILL , BROWN);
rectangle (0, 2*y - 100, 2*x, 2*y);
Floodfill (1, 2*y - 99, BROWN);

// Background
Setfillstyle ( INTERLEAVE_FILL , LIGHTBLUE );
Floodfill (1, 1, BROWN);

// Sun
Setcolor (YELLOW);
Setfillstyle ( SOLID_FILL , YELLOW);
circle ( 2*x - 100, 100, 300);
Floodfill ( 2*x - 100, 100, YELLOW);

// BODY
Setcolor (LIGHT GREEN);
Setfill style ( SOLID_FILL, LIGHT GREEN);
rectangle ( x - 140, y, x+140, 2*y - 101);
floodfill ( x - 139, y+1, LIGHT GREEN);

// TERRACE
for (int i = 0; i < 41; ++i) {

        line (x-140+i, y-i, x-60-i, y-i);
}
```

```
for(int j=0; j<46; ++j){
    setcolor(RED);
    if(j%9 == 0){
        setcolor(BLACK);
    }
    line(x-150+j, y-j, x-140+j, y-j);
    line(x-105+j, y-45+j, x+105+j, y-45+j);
}
setcolor(BLACK);
line(x-106, y-46, x-60, y);
line(x+104, y-46, x+150, y);
line(x-105, y-46, x-151, y);
line(x-99, y-41, x-140, y);

// Interior DOOR
rectangle(x-120, 2*y-180, x-80, 2*y-101);
line(x-60, y, x-60, 2*y-101);
circle(x-113, 2*y-140, 3);
setfillstyle(SOLID_FILL, BLACK);
Floodfill(x-119, 2^y-169, BLACK);

// Interior Circle Window
circle(x-100, y+10, 15);
circle(x-100, y+10, 16);
setfillstyle(SOLID_FILL, WHITE);
Floodfill(x-100, y+10, BLACK);
line(x-100, y-5, x-100, y+25);
```

```
line (x-115 , y+10, x-85, y+10);
// Interior Rectangle Window.
rectangle (x+10, y+100, x+70, 2*y-201);
rectangle(x+9, y+99, x+71, 2*y-200);
floodfill (x+13, 2*y-198, BLACK);
line ( x+40, y+100, x+40, 2*y-201);
line ( x+10, 1.5*y-50 , x+70, 1.5*y-50);
}

void makearc ( int x, int y, int r, int color=0,
int sangle=0, int eangle=360, int arcmode=0){

    int xp , yp;
    for (int i= sangle; i < eangle ; ++i){

        xp = x + r * sin (rad(i));
        yp = y + r * cos (rad(i));
        if (arcmode ==1 && ( i== sangle || i== eangle))
        {
            setcolor (color);
            line (x, y, xp, yp);
        }
        putpixel( xp, yp, color);

    }

}
```

```
void car (int x, int y) {
    setbkcolor (BLACK);
    setcolor (WHITE);

    // Exterior
    line (x+220, y-20, x+225, y-20);
    line (x+225, y-20, x+225, y-45);
    line (x+220, y-20, x+222, y-45);
    line (x+225, y-45, x+125, y-50);
    line (x+125, y-50, x+75, y-95);
    line (x+75, y-95, x+75, y-95);
    line (x-75, y-95, x-140, y-50);
    line (x-140, y-50, x-180, y-50);
    line (x-180, y-50, x-180, y-20);
    line (x-73, y+12, x+131, y+12);
    line (x-73, y+12, x-75, y+8)
    line (x+131, y+12, x+134, y+8);
    line (x-129, y+12, x-170, y+12);
    line (x-127, y+8, x-129, y+12);
    line (x+186, y+8, x+189, y+12);
    line (x-180, y-20, x-114, y-20);
    line (x+220, y-20, x+173, y-20);
    line (x+189, y+12, x+25, y+12);
    line (x+213, y+12, x+225, y);
    line (x+225, y, x+210, y);
    line (x+210, y, x+210, y-10);
    line (x+210, y-10, x+230, y-10);
    line (x+230, y-10, x+230, y-20);
```

```
line (x+230, y-20, x+225, y-20);
line ( x+222, y-10, x+222, y);
line (x-170 , y+12, x-180, y);
line (x-180, y , x-185, y-10);
line ( x-185, y-10, x-185, y-20);
line ( x-185, y-20, x-180, y-20);
line ( x-185, y-10, x-165, y-10);
line (x-177, y+4, x-179, y+4);
line (x-179 , y+4, x-179, y+10);
line (x-172, y+10, x-179, y+10);

// Interior
line ( x+79, y-90, x+115, y-50);
line ( x+115, y-50, x+125, y-50);
line ( x+105, y-50, x+70, y-90);
line (x-70, y-90, x+70, y-90);
line ( x-70, y-90, x-110, y-50);
line ( x-110, y-50 , x+105, y-50);
line (x-70, y-50, x-70, y-90);
line (x-68, y-50, x-68, y-90);
line (x, y-90, x , y-50);
line (x+5, y-90, x+5, y-50);
line (x+10, y-90, x+10, y-50);
line (x-110, y-50, x-110, y-30);
line ( x-72 , y+5 , x+113, y+5);
line (x+5, y+5, x+5, y-30);
makearc (x-100, y+5, 35, WHITE, 90, 195);
line ( x+105 , y-50 , x+105, y+5);
```

```
//Wheel
makearc(x-100, y+5, 29, WHITE, 75, 285);
circle(x-100, y+5, 25);
circle(x-100, y+5, 17);
circle(x-100, y+5, 6);
circle(x-100, y+5, 3);
line(x-101, y-1, x-102, y-12);
line(x-99, y-1, x-97, y-12);
line(x-101, y+11, x-102, y+22);
line(x-99, y+11, x-97, y+22);
line(x-106, y+4, x-117, y+2);
line(x-106, y+4, x-117, y+6)
line(x-94, y+4, x-83, y+2);
line(x-94, y+4, x-83, y+6);
makearc(x+160, y+5, 29, WHITE, 75, 285)
circle(x+160, y+5, 25);
circle(x+160, y+5, 17);
circle(x+160, y+5, 6);
circle(x+160, y+5, 3);
line(x+161, y-1, x+162, y-12);
line(x+159, y-1, x+158, y-12);
line(x+161, y+11, x+162, y+22);
line(x+159, y+11, x+158, y+22);
line(x+154, y+4, x+143, y+2);
line(x+154, y+4, x+143, y+6);
line(x+166, y+4, x+117, y+2);
line(x+166, y+4, x+117, y+6);
```

```
// Lights and Accessories
rectangle (x+198, y-38, x+215, y-26);
rectangle (x+170, y-40, x-180, y-25);
rectangle (x+12, y-42, x+25, y-38);
rectangle (x-85, y-42, x-72, y-38);
rectangle (x+96, y-52, x+85, y-60);
line (x+100, y-50, x+96, y-54);
line (x+105, y-50, x+96, y-57);
makearc (x+102, y-50, 3, WHITE, 270);
makearc (x+102, y-50, 3, WHITE, 0, 90);
}

int main() {

    int ch;
    int gd = DETECT, gm, x, y;
    clrscr();
    cout<<"What do you want to draw? \n 1. Hut \n 2. Car \n Enter your choice (1/2):";

    cin>>ch;

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

    x = getmaxx()/2;
    y = getmaxy()/2;
```

```
switch(ch) {

case 1: hut(x, y);
        break;

case 2: car(x, y);
        break;

default: cout << "Invalid Input";

}
 getch();
closegraph();
return 0;

}
```

VIVA Questions

Q① What is the use of initgraph() and closegraph() function?

initgraph() initializes the graphic system by loading a graphics driver from disk then putting the system into graphics mode. It also resets graphic settings and the graph.

closegraph() closes the graphic mode, deallocates all memory allocated by graphics system and restores the screen to the mode it was in before

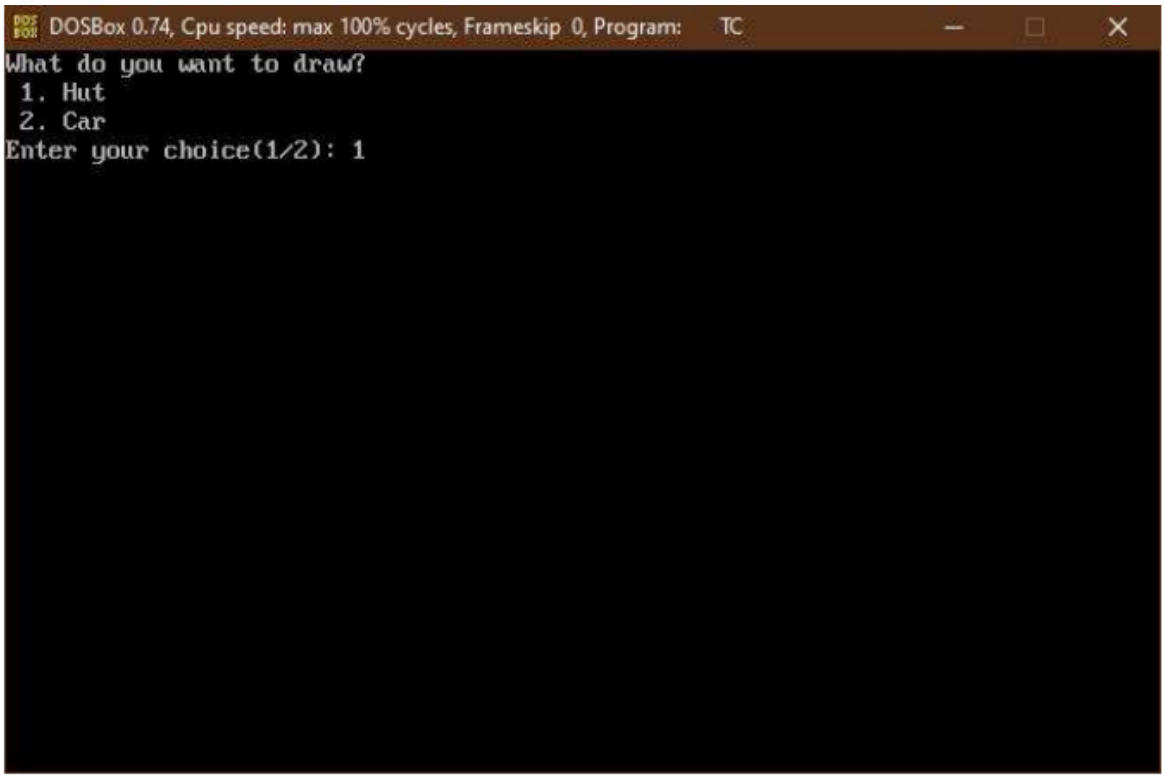
you called initgraph

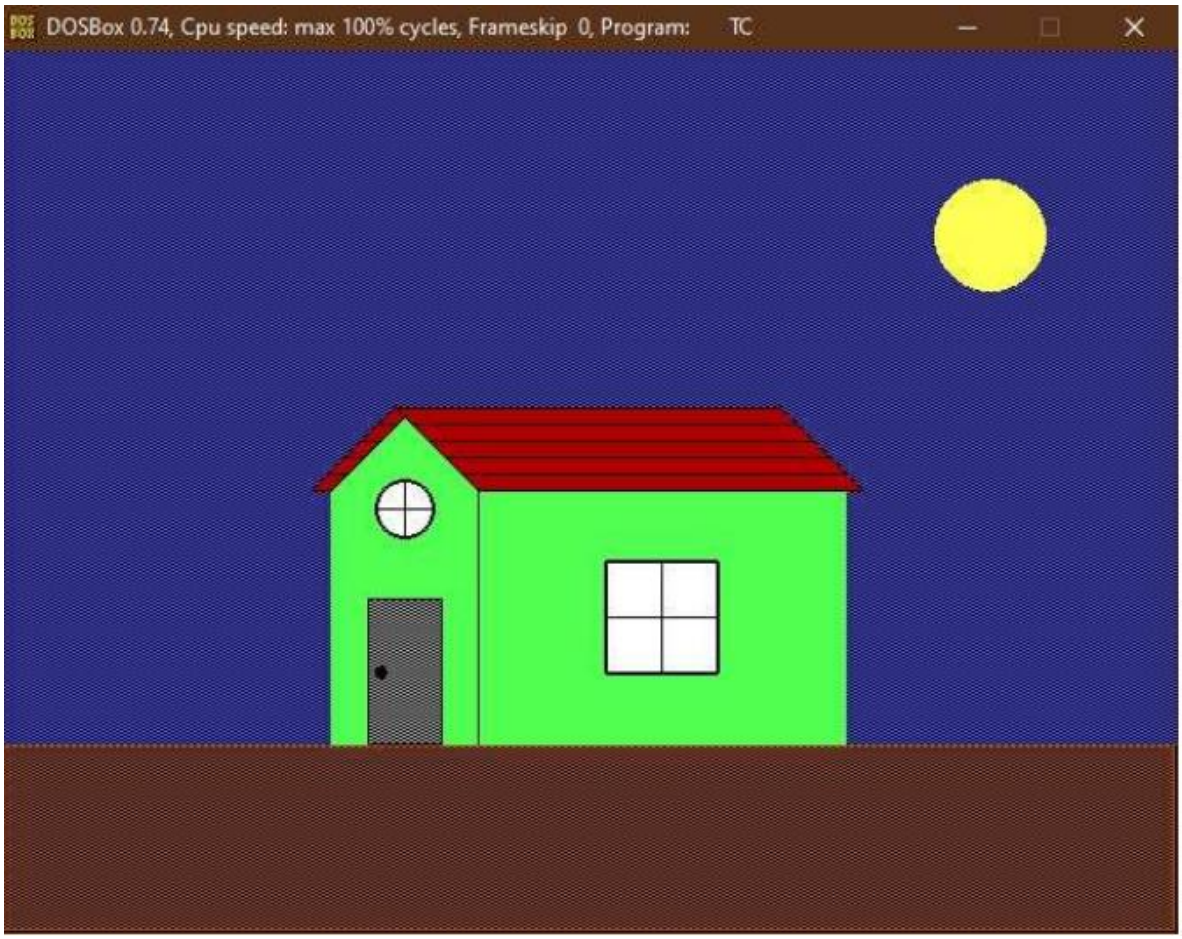
② Why do we need to use closegraph() function after getch()?
Generally, getch() is used at the end of the program so that the program terminates after user confirmation. closegraph() is used after this point so that all the memory that was held up by graphical elements is deallocated and thus preventing possible memory errors

③ Which parameters are used to find resolution of the screen?
Graphics driver along with graph mode are used to determine the resolution, palette and pages of display. detectgraph() detects your system's graphics adapter and chooses the mode that provides highest resolution for that adapter.

④ How is putpixel() different from getpixel()?
putpixel() is used to draw a pixel of given specified point (x, y) whereas getpixel() is used to get the colour of a specified point (x, y)

OUTPUT


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
What do you want to draw?
1. Hut
2. Car
Enter your choice(1/2): 1


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
What do you want to draw?
 1. Hut
 2. Car
Enter your choice(1/2): 2
```