

**Arab American University**

**Faculty of Engineering and IT**

**Assembly LAB**

## (Graphics INT10 Part I)

**Suha Alardah 201811033**

To: Dr.Tariq Zanoon

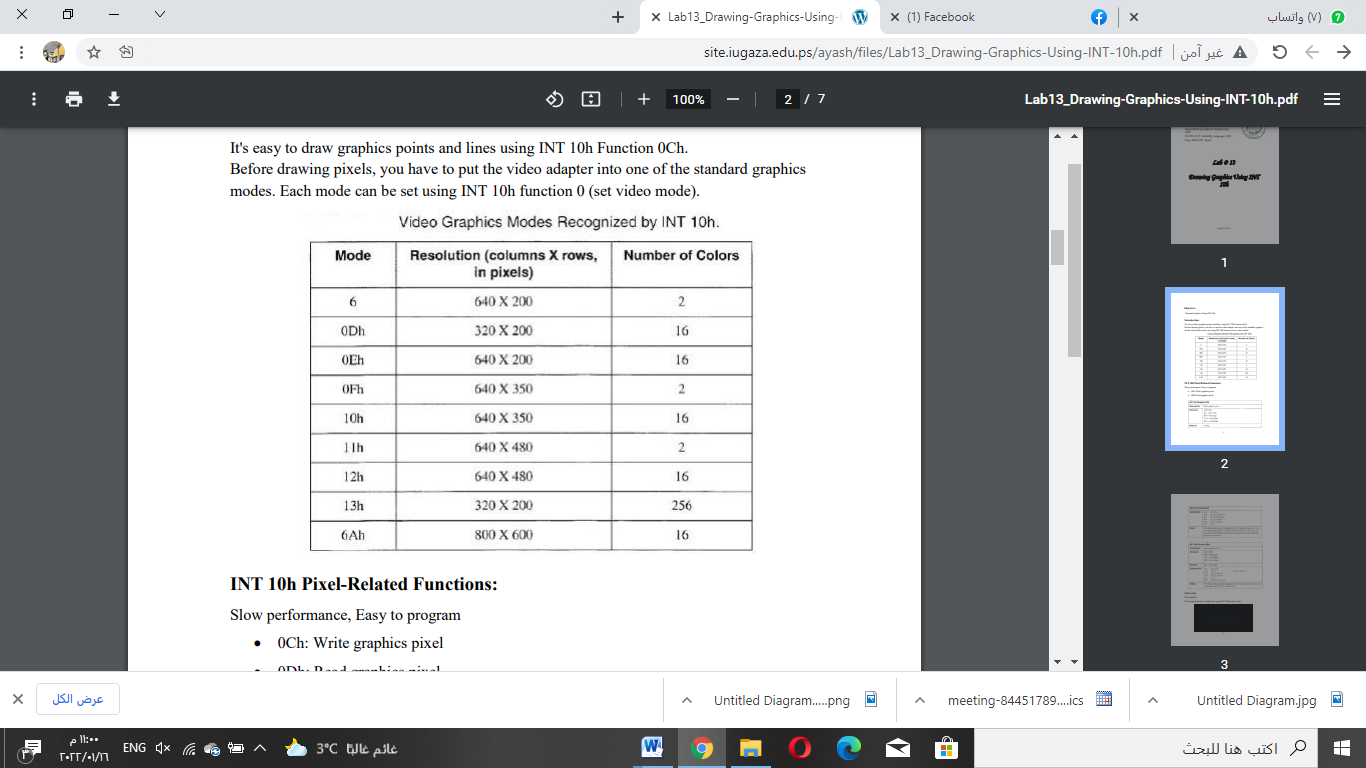
To: Mrs.Noor Kmail

Objective:

Drawing Graphics Using INT 10h.

Introduction:

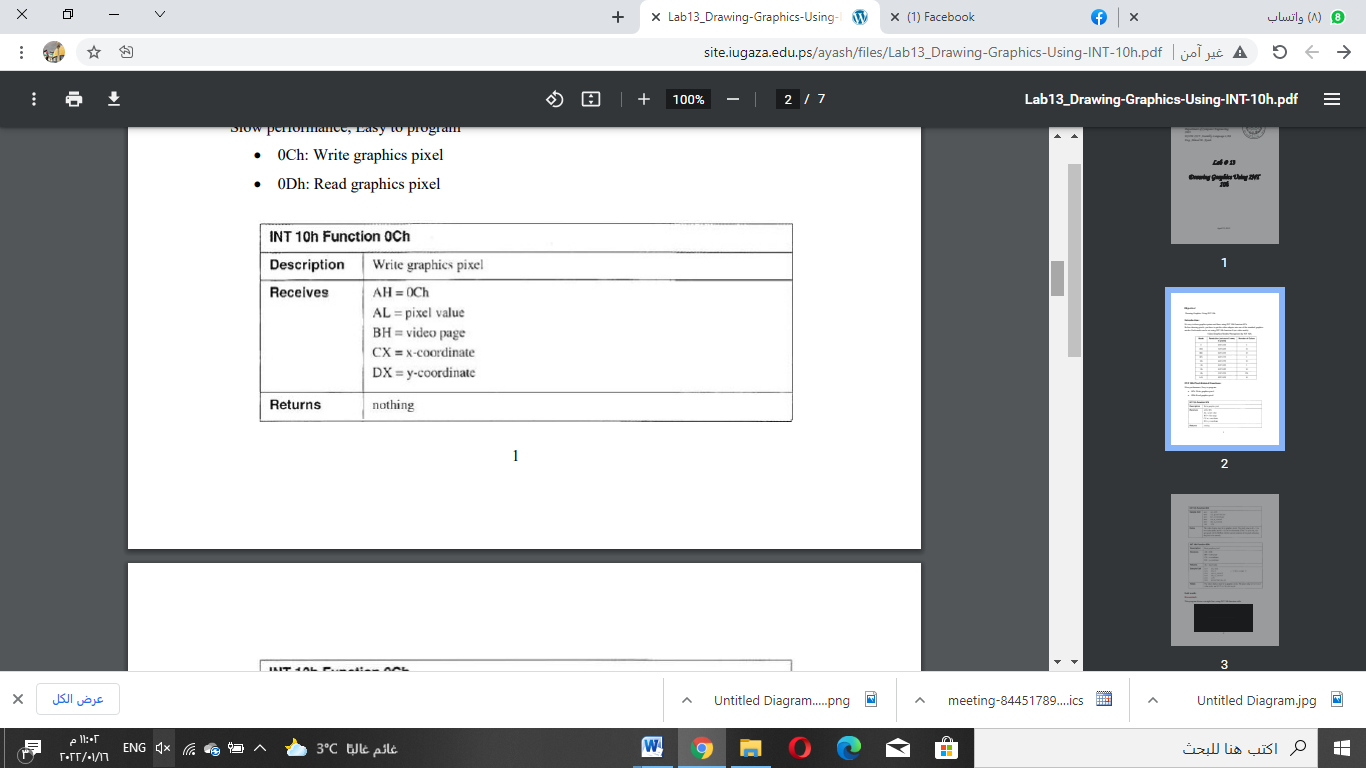
It's easy to draw graphics points and lines using INT 10h Function 0Ch. Before drawing pixels, you have to put the video adapter into one of the standard graphics modes. Each mode can be set using INT 10h function 0 (set video mode).

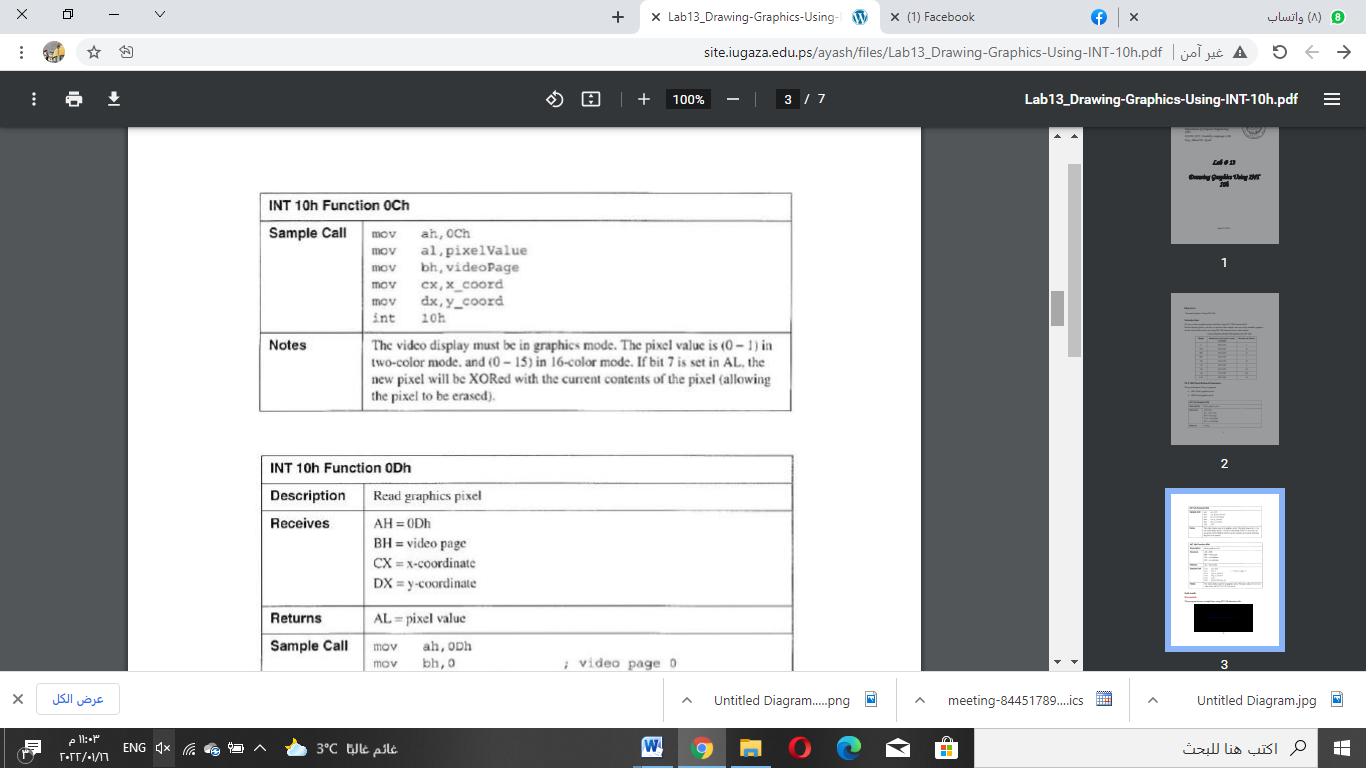


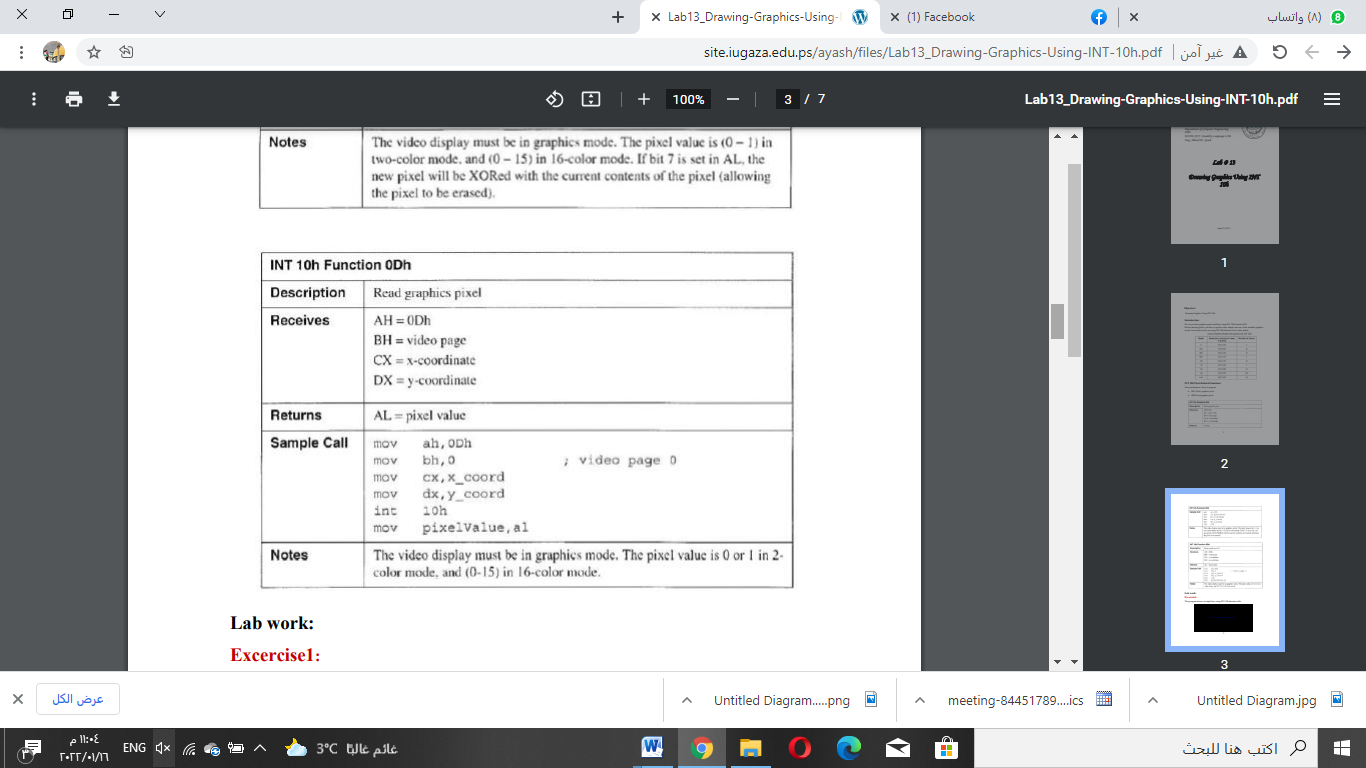
INT 10h Pixel-Related Functions: Slow performance, Easy to program

• 0Ch: Write graphics pixel

• 0Dh: Read graphics pixel







**TASK1. Drawing a Pixel**

The following program draws a pixel on the screen at location (320, 240) using the “write pixel”

function (AH=0Ch) of INT 10h. Run the program after assembling and linking it.

TITLE "Program to draw a pixel on the screen"

.MODEL SMALL ; this defines the memory model

.STACK 100 ; define a stack segment of 100 bytes

.DATA ; this is the data segment

.CODE ; this is the code segment

MOV AX,@DATA ; get the address of the data segment

MOV DS, AX ; and store it in DS register

MOV AH, 0Fh ; get current video mode

INT 10h

PUSH AX ; save current video mode

MOV AH, 00h ; set video mode

MOV AL, 12h ; graphics 640x480

INT 10h

; draw a green color pixel at location (320, 240)

MOV AH, 0Ch ; Function 0Ch: Write pixel dot

MOV AL, 02 ; specify green color

MOV CX, 320 ; column 320

MOV DX, 240 ; row 240

MOV BH, 0 ; page 0

INT 10h

MOV AH, 07h ; wait for key press to exit program

INT 21h

POP AX ; retrieve original video mode

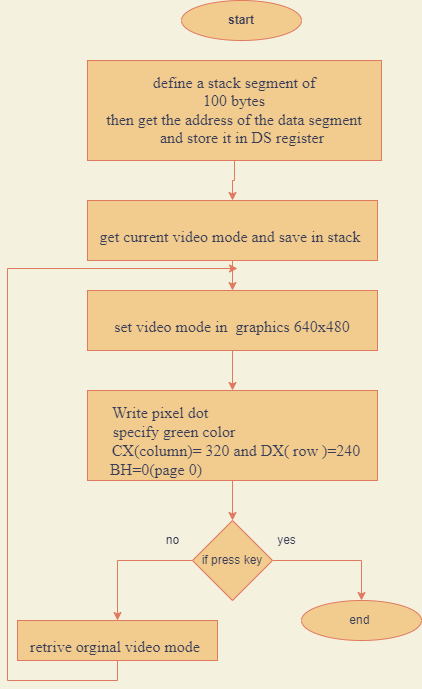
MOV AH, 00h

INT 10h ; restore original video mode

MOV AX, 4C00H ; Exit to DOS function

INT 21H

END ; end of the program



**TASK2. Drawing a horizontal line**

The following program draws a horizontal line on the screen from location (170, 240) to (470,

240)

by writing pixels on the screen using function (AH=0Ch) of INT 10h. Run the program after

assembling and linking it.

TITLE "Program to draw a horizontal line on the screen"

.MODEL SMALL ; this defines the memory model

.STACK 100 ; define a stack segment of 100 bytes

.DATA ; this is the data segment

.CODE ; this is the code segment

MOV AX,@DATA ; get the address of the data segment

MOV DS, AX ; and store it in DS register

MOV AH, 0Fh ; get current video mode

INT 10h

PUSH AX ; save current video mode

MOV AH, 00h ; set video mode

MOV AL, 12h ; graphics 640x480

INT 10h

; draw a green color line from (170, 240) to (470, 240)

MOV CX, 170 ; start from column 170

MOV DX, 240 ; and row 240

MOV AX, 0C02h ; AH=0Ch and AL = pixel color (green)

BACK: INT 10h ; draw pixel

INC CX ; go to next column

CMP CX, 470 ; check if column=470

JB BACK ; if not reached column=470, then continue

MOV AH, 07h ; wait for key press to exit program

INT 21h

POP AX ; retrieve original video mode

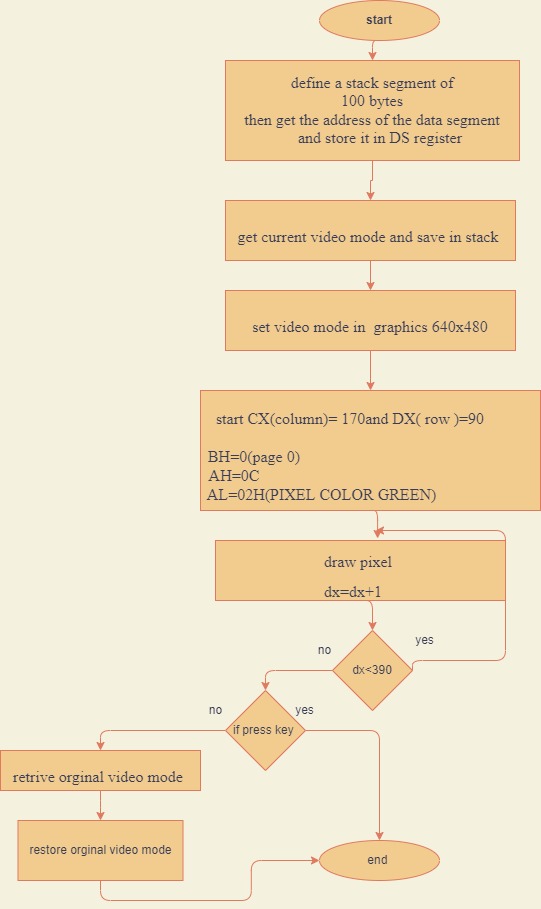
MOV AH, 00h

INT 10h ; restore original video mode

MOV AX, 4C00H ; Exit to DOS function

INT 21H

END ; end of the program



**3. Drawing a vertical line**

Using the procedure followed in part 2 (drawing a horizontal line), draw a vertical line on the

screen from location (320, 90) to (320, 390). Run the program after assembling and linking it.

**4. Drawing a plus (+) sign in the middle of the screen**

Combine the programs written for parts 2 and 3 above to draw a plus sign. All you have to do is

to insert the code for drawing the vertical line [from location (320, 90) to (320, 390)] right after

the code for drawing the horizontal line [from location (170, 240) to (470, 240)]. Run the

program after assembling and linking it.

**5.** Draw the following figure on the screen using function 0Ch of INT 10h. Assemble, link, and run

it and show it to your lab instructor for credit.

dosseg

.model small

.stack

.data

.code

mov ax,@data

mov ds,ax

mov ah,0fh

int 10h

push ax

mov ah,00h

mov al,12h

int 10h

mov cx,170

mov dx,90

mov ax,0c02h

back:int 10h

inc dx

cmp dx,390

jb back

mov cx,170

mov dx,240

mov ax,0c02h

back1:int 10h

inc cx

cmp cx,470

jb back1

mov cx,320

mov dx,90

mov ax,0c02h

back2:int 10h

inc dx

cmp dx,390

jb back2

mov cx,470

mov dx,90

mov ax,0c02h

back3:int 10h

inc cx

cmp cx,390

jb back3

mov cx,170

mov dx,90

mov ax,0c02h

back4:int 10h

inc cx

cmp cx,470

jb back4

mov cx,170

mov dx,90

mov ax,0c02h

back5:int 10h

inc cx

cmp cx,470

jb back5

mov cx,470

mov dx,90

mov ax,0c02h

back6:int 10h

inc dx

cmp dx,390

jb back6

mov cx,170

mov dx,390

mov ax,0c02h

back7:int 10h

inc cx

cmp cx,470

jb back7

mov ah,07h

int 21h

pop ax

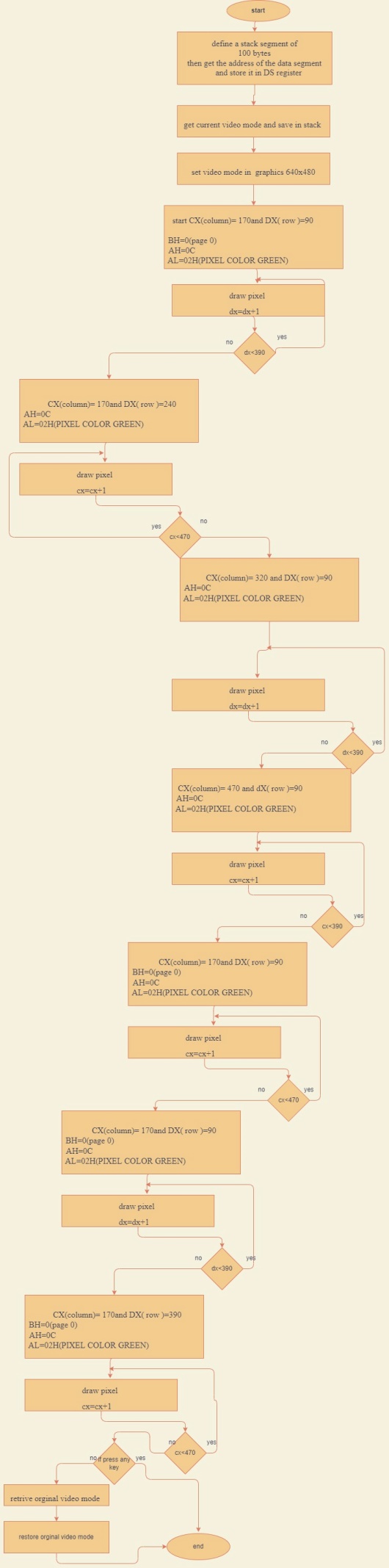
mov ah,00h

int 10h

mov ax,4c00h

int 21h

end

****

## Conclusion:

In this experiment we learn how to drawing in the screen using pixel an video set and get we have funny experiment,

In this experiment, we use BIOS and DOS services to write programs that can do Write a pixel on the screen

 Draw a line on the screen

 Draw a rectangle and plus inside on the screen.