

## **ELEMENTI DI GRAFICA IN AMBIENTE DEV-PASCAL v. 9.2**

### **MODULI GRAPH, DOS E CRT**

Una volta entrati in ambiente DEV-Pascal possiamo cominciare ad editare un programma nel modo che conosciamo. Se dobbiamo far uso dell'ambiente grafico dobbiamo procedere nel seguente modo: dopo l'intestazione del programma con la linea di codice "Program nome" deve essere inserita la parola chiave "Uses" seguita dal nome del modulo grafico "Graph".

Es.

```
Program nomefile(input,output);
```

```
Uses
```

```
    Graph,Crt;
```

```
Var
```

```
    .....
```

Il modulo Crt permette di usare dei comandi per la gestione dei video in modo più diretto. Il modulo Graph, invece, ci consente di usare tutte le istruzioni grafiche del DEV-Pascal caricando in memoria le librerie grafiche.

### **DICHIARAZIONE DELLE VARIABILI PER L' AMBIENTE GRAFICO**

Per inizializzare l'ambiente grafico dobbiamo utilizzare delle variabili di tipo "Integer" che possiamo chiamare come ci pare. Le variabili inizialmente sono tre e serviranno per poter memorizzare il codice del driver grafico, il codice del modo grafico e il codice di un eventuale errore durante l' inizializzazione. Per "Driver Grafico" s'intende un programma pilota per la gestione del tipo di scheda video montata sul Personal Computer, Per "Modo Grafico" s'intende il modo di risoluzione video in numero di pixel per ogni tipo di scheda video.

```
Program nomefile(input,output);
```

```
Uses
```

```
    Graph,Crt ;
```

```
Var
```

```
    Driver, Modo : smallint;
```

```
Begin
```

```
    .....
```

```
    .....
```

```
End.
```

### **INIZIALIZZAZIONE DEL MODO GRAFICO**

In un qualsiasi punto del programma possiamo avviare la procedura di inizializzazione grafica con le seguenti istruzioni:

```
Program nomefile(input,output);
```

```
Uses
```

```
    Graph,Crt;
```

```
Var
```

```
    Driver,Modo smallint;.
```

```
Begin
```

```
    Driver:=Detect;
```

```
    initgraph(Driver,Modo,' ');
```

```
    .....
```

```
    "Istruzioni grafiche"
```

```
    etc.
```

```
    .....
```

```
End;
```

L'istruzione Detect rileva il tipo di scheda montata sul P.C. e va ad avvalorare la variabile Driver con il codice per il programma pilota da caricare. L'istruzione Initgraph carica in memoria il programma pilota interessato e attiva la risoluzione migliore per quella scheda grafica. Gli apici eventualmente, devono racchiudere il percorso che il compilatore deve seguire per trovare dove è memorizzato il programma pilota, infatti se questo programma pilota è presente nella directory di lavoro allora tra apici poniamo una stringa nulla, se il programma pilota, per esempio, è memorizzato su un floppy-disk allora dobbiamo specificarlo nel seguente modo: " Initgraph(Driver, Modo, ' A:\');

## **ISTRUZIONI PASCAL PER OPERAZIONI DI GRAFICA**

### **ClearDevice**

Pulisce lo schermo in modalità grafica.

### **Randomize**

Attiva la funzione random.

### **SETCOLOR(colore); colore:integer**

Setta il colore di primo piano. L'argomento colore può essere un numero della tabella dei colori oppure il nome del colore in lingua inglese.

Es. Setcolor(red); setta come colore di primo piano il rosso.

<b>colore</b>					
0	Black	Nero	8	Darkgray	Grigio scuro
1	Blue	Blu	9	Lightblue	Celeste
2	Green	Verde	10	Lightgreen	Verde chiaro
3	Cyan	Viola	11	Lightcyan	Viola chiaro
4	Red	Rosso	12	Lightred	Rosso chiaro
5	Magenta	Magenta	13	Lightmagenta	Magenta chiaro
6	Brown	Marrone	14	Yellow	Giallo
7	lightgray	grigio	15	white	bianco

### **SETBKCOLOR(colore);**

**colore:integer**

Setta il colore di sfondo. L'argomento "colore" può essere un numero della tabella dei colori oppure il nome del colore in lingua inglese.

Es. Setbkcolor(yellow); setta come colore di sfondo il giallo,

### **SETLINESTYLE(tipoLinea,0,tipoSpessore) tipoLinea,tipoSpessore:integer**

<b>tipoLinea</b>		
SolidLn	0	Solida
DotterLn	1	Puntini
CenterLn	2	Trattini brevi e lunghi
DashedLn	3	A trattini
UserBitLn	4	Definita dall'utente

<b>tipoSpessore</b>		
NormWidth	1	normale
ThickWidth	3	spessa

**PUTPIXEL(x,y,colore);**

Traccia un punto alle coordinate x,y di colore "colore".

Es. Putpixel(100,200,green); Traccia un puntino luminoso di colore verde alle coordinate 100,200 dello schermo grafico.

**LINE(x1,y1,x2,y2);**

Traccia una linea dal punto di coordinate x1,y1 al punto di coordinate x2,y2 il colore della linea sarà quello definito da setcolor.

Es. Line(0,0,639,479); Traccia una linea diagonale sullo schermo da vertice a vertice.

**LINE(x,y);**

Traccia una linea dall' ultimo punto attivo al punto di coordinate x,y.

Es. Line(0,0,100,100);

Lineto(200,200); Traccia la linea dal punto finale dell'istruzione precedente al punto di coordinate 200,900.

**RECTANGLE(x1,y1,x2,y2);**

Disegna un rettangolo avente vertice superiore sinistro in x1,y1 e vertice inferiore destro in x2,y2.

**CIRCLE(x,y,raggio);**

Disegna un cerchio con centro di coordinate x,y e raggio "raggio". Il colore del cerchio sarà quello definito da setcolor.

**ARC(x,y,anginiz,angfin,raggio);**

Disegna un arco di cerchio con centro del cerchio ipotetico in x,y di raggio "raggio" partendo dall'angolo di gradi "anginiz" all'angolo di gradi "angfin" in senso antiorario.

Es. Arc(200,200,0,90,50); Disegna un arco corrispondente ad un angolo di 90 gradi in senso antiorario.

**ELLIPSE(x,y,0,360,a,b)**

Disegna un'ellisse di centro x,y e semiassi a e b.

**OUTTEXTXY(x,y,'stringa');**

Permette di visualizzare una stringa o messaggio sullo schermo in modalità grafica in quanto la istruzione "writeln" non è eseguibile. Le coordinate x,y indicano il punto da dove inizia la stringa.

Es. Outtextxy(300,300,' Sto lavorando in modalità grafica ');

**STR(numero,stringa);**

In modalità grafica non è possibile visualizzare il valore di una variabile con l'istruzione outtextxy, bisogna prima convertire il valore numerico della variabile in un valore stringa.

Es. STR(numgiorno,giornostringa);

OUTTEXTXY(100,100,giornostringa);

La variabile numgiorno viene avvalorata in input dal numero del giorno che ci interessa, l'istruzione STR la converte in variabile stringa "giornostringa" dichiarata precedentemente ovviamente come stringa

Var

Giornostringa: string;

**RESTORECRTMODE;**

Ripristina il modo testo rimane però in memoria il driver per la modalità grafica, quindi per ripristinare di nuovo la modalità grafica basta scrivere l'istruzione

**SETGRAPHMODE(mod);**

## CLEARVIEWPORT;

Questa istruzione equivale all'istruzione clrscr però in modalità grafica. Infatti ripulisce lo schermo da qualsiasi figura disegnata precedentemente.

## SETTEXTSTYLE(font,direzione,grandezza);

Definisce lo stile di testo utilizzato da OUTTEXTXY per visualizzare stringhe di caratteri.

Il parametro font definisce il tipo di font per il carattere. Il parametro direzione definisce la direzione verticale o orizzontale della stringa partendo dal punto definito da x,y in OUTTEXTXY verso destra o verso l'alto a seconda del parametro. Fare delle prove in DEV-Pascal per verificarne il funzionamento.

font			direzione		
DefaultFont	0	8x8 bit mapped font	HorizDir	0	Da sinistra a destra
TriplexFont	1	grassetto	VertDir	1	Dal basso in alto (verticale)
SmallFont	2	grassetto	grandezza		
SansSerifFont	3	grassetto			
GothicFonte	4	grassetto	UserCharSize	0	Grandezza carattere

## SETFILLSTYLE(motivo,colore);      **motivo:integer**

Definisce il motivo e il colore per riempire figure circoscritte. Il motivo può essere un valore da 0 a 9 e va dalla tinta unita al tratteggio o altro, verificare con delle prove. Il parametro "colore" definisce il colore con cui si vuole ottenere il riempimento. Per ottenere il riempimento delle figure desiderate bisogna utilizzare l'istruzione Floodfill.

motivo		
EmptyFill	0	Usa il colore dello sfondo
SolidFill	1	Colore pieno
LineFill	2	--- usa righe orizzontali
LtSlashFill	3	/// linee oblique sottili
SlashFill	4	/// linee oblique spesse
BkSlashFill	5	\ linee oblique spesse
LtBkSlashFill	6	\ linee oblique sottili
HatchFill	7	quadrettino sottili
XhatchFill	8	a croce grossa
InterleaveFill	9	a croce fine
WideDotFill	10	punti spazati
CloseDotFill	11	punti fitti
UserFill	12	definito dall'utente

## FLOODFILL(x,y,colore);

Riempie una figura circoscritta di colore "colore". Le coordinate x,y devono identificare un punto interno alla figura.

## BAR(x1,y1,x2,y2);

Costruisce una barra per istogrammi dove le coordinate x1,y1 definiscono il vertice inferiore sinistro e le coordinate x2,y2 il vertice superiore destro.

## BAR3D(x1,y1,x2,y2,spessore,Top);

Costruisce una barra 3D dove le coordinate hanno la stessa funzione che in BAR, il parametro spessore definisce appunto lo spessore in profondità della barra in pixel. Il parametro Top se viene sostituito dall'identificatore TopOn chiude l'estremità superiore della barra se invece viene utilizzato TopOff l'estremità superiore rimane aperta. Verificare con delle prove in DEV-Pascal.

Es. Bar3D(400,250,300,300,25,TopOn);

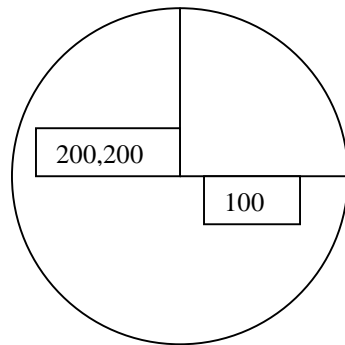
Disegna una barra 3D con vertice inferiore sin. in 400,250, vertice sup. destro in 300,300 di spessore 25 pixel con l'estremità superiore chiusa.

**PIESLICE(x,y,anginiz,angfin,raggio);**

Costruisce uno spicchio per i grafici a torta. Le coordinate x,y sono l'ipotetico centro del cerchio che delimita la torta. I parametri anginiz e angfin definiscono in senso antiorario la grandezza in gradi dell'arco dello spicchio. Il parametro raggio definisce il raggio del cerchio che delimita la torta.

Es. Pieslice(200,200,0,90,100);

Costruisce uno spicchio il cui centro del cerchio è in 200,200 con ampiezza di 90 gradi e raggio 100.



**SETVIEWPORT(x1,y1,x2,y2,Clip);**

Definisce e rende attiva una finestra in modalità grafica. X1,Y1 sono le coordinate del vertice inf. Sinistro mentre X2,Y2 sono le coordinate del vertice sup.re destro. Il parametro Clip se attivo "ClipOn", taglia tutte le figure disegnate ai bordi della finestra attiva. Se Clip non è attivato "ClipOff" le figure non vengono tagliate e trasbordano dai limiti della finestra grafica.

Una volta attivata una finestra grafica rimane attiva finché non si riattiva quella principale sempre con lo stesso comando ma con i seguenti parametri:

Es. Setviewport(0,0,GetMaxX,GetMaxY,ClipOn);

GetMaxX e GetMaxY restituiscono rispettivamente il valore massimo per l'asse X e quello per l'asse Y.