C graphics.h 函數

C graphics.h 函數

(一) 圖元函數

56. putpiel() 畫圖元點函數
57. getpixel()返回圖元色函數

(二) 直線和線型函數

58. line() 畫線函數
59. lineto() 畫線函數
60. linerel() 相對畫線函數
61. setlinestyle() 設置線型函數
62. getlinesettings() 獲取線型設置函數
63. setwritemode() 設置畫線模式函數

(三)、多邊形函數

64. rectangle() 畫矩形函數
65. bar() 畫條函數
66. bar3d() 畫條塊函數
67. drawpoly() 畫多邊形函數

(四)、 圓、弧和曲線函數

68. getaspectratio()獲取縱橫比函數
69. circle()畫圓函數
70. arc() 畫圓弧函數
71. ellipse()畫橢圓弧函數
72. fillellipse() 畫橢圓區函數
73. pieslice() 畫磁區函數
74. sector() 畫橢圓磁區函數
75. getarccoords()獲取圓弧座標函數

(五)、 填充函數

76. setfillstyle() 設置填充圖樣和顏色函數
77. setfillpattern() 設置用戶圖樣函數
78. floodfill() 填充閉域函數
79. fillpoly() 填充多邊形函數
80. getfillsettings() 獲取填充設置函數
81. getfillpattern() 獲取用戶圖樣設置函數

(六)、圖像函數

82. imagesize() 圖像存儲大小函數
83. getimage() 保存圖像函數
84. putimage() 輸出圖像函數

四、圖形和圖像函數

對許多圖形應用程式，直線和曲線是非常有用的。但對有些圖形只能靠操作單個圖元才能畫出。當然如果沒有畫圖元的功能，就無法操作直線和曲線的函數。而且通過大規模使用圖元功能，整個圖形就可以保存、寫、擦除和與螢幕上的原有圖形進行疊加。

(一) 圖元函數

56. putpixel() 畫圖元點函數

功能： 函數 putpixel() 在圖形模式下螢幕上畫一個圖元點。

用法： 函數調用方式爲 void putpixel(int x,int y,int color);

說明： 參數 x,y 爲圖元點的座標，color 是該圖元點的顏色，它可以是顏色符號名，也可以是整型色彩值。

此函數相應的頭檔是 graphics.h

返回值： 無

例： 在螢幕上(6,8)處畫一個紅色圖元點：

putpixel(6,8,RED);

57. getpixel()返回圖元色函數

功能： 函數 getpixel()返回圖元點顏色值。

用法： 該函數調用方式爲 int getpixel(int x,int y);

說明： 參數 x,y 爲圖元點座標。

函數的返回值可以不反映實際彩色值，這取決於調色板的設置情況(參見 setpalette()函數)。

這個函數相應的頭檔爲 graphics.h

返回值： 返回一個圖元點色彩值。

例： 把螢幕上(8,6)點的圖元顏色值賦給變數 color。

color=getpixel(8,6);

(二) 直線和線型函數

有三個畫直線的函數，即 line(),lineto(),linerel()。這些直線使用整型座標，並相對於當前圖形視口，但不一定受視口限制，如果視口裁剪標誌 clip 爲真，那麼直線將受到視口邊緣截斷；如果 clip 爲假，即使終點座標或新的當前位置在圖形視口或螢幕極限之外，直線截斷到螢幕極限。

有兩種線寬及幾種線型可供選擇，也可以自己定義線圖樣。下面分別介紹直線和線型函數。

58. line() 畫線函數

功能： 函數 line()使用當前繪圖色、線型及線寬，在給定的兩點間畫一直線。

用法： 該函數調用方式爲 void line(int startx,int starty,int endx,int endy);

說明： 參數 startx,starty 爲起點座標,endx,endy 爲終點座標，函數調用前後，圖形狀態下螢幕游標(一般不可見)當前位置不改變。

此函數相應的頭檔爲 graphics.h
返回值： 無
例： 見函數 60.linerel()中的實例。

59. lineto() 畫線函數
功能： 函數 lineto()使用當前繪圖色、線型及線寬，從當前位置畫一直線到指定位置。
用法： 此函數調用方式爲 void lineto(int x,int y);
說明： 參數 x,y 爲指定點的座標，函數調用後，當前位置改變到指定點(x,y)。
該函數對應的頭檔爲 graphics.h
返回值： 無
例： 見函數 60.linerel()中的實例。

60.linerel() 相對畫線函數
功能： 函數 linerel() 使用當前繪圖色、線型及線寬，從當前位置開始，按指定的水準和垂直偏移距離畫一直線。
用法： 這個函數調用方式爲 void linerel(int dx,int dy);
說明： 參數 dx,dy 分別是水準偏移距離和垂直偏移距離。
函數調用後，當前位置變爲增加偏移距離後的位置，例如，原來的位置是(8,6)，調用函數 linerel(10,18)後，當前位置爲(18,24)。
返回值：無
例： 下面的程式爲畫線函數調用實例：

```
＃i nclude<graphics.h>
void main()
{
int driver,mode;
driver=DETECT;
mode=0;
initgraph(&driver,&mode,"");
setcolor(15);
line(66,66,88,88);
lineto(100,100);
linerel(36,64);
getch();
restorecrtmode();
}
```

61. setlinestyle() 設置線型函數
功能： setlinestyle() 爲畫線函數設置當前線型，包括線型、線圖樣和線寬。
用法： setlinestyle() 函數調用方式爲 void setlinestyle(int stly,unsigned pattern,int width);
說明： 參數 style 爲線型取值，也可以用相應名稱表示，如表 1-10 中所示。

參數 pattern 用於自定義線圖樣，它是 16 位(bit)字，只有當 style=USERBIT_LINE(值爲 1)時，pattern 的值才有意義，使用用戶自定義線圖樣，與圖樣中“1”位元對應的圖元顯示，因此，pattern=0xFFFF，則畫實線；pattern=0x9999，則畫每隔兩個圖元交替顯示的虛線，如果要畫長虛線，那麼 pattern 的值可爲 0xFF00 和 0xF00F，當 style 不爲 USERBIT_LINE 值時，雖然 pattern 的值不起作用，但扔須爲它提供一個值，一般取爲 0。
參數 wigth 用來設定線寬，其取值見表 1-11，表中給出了兩個值，即 1 和 3，實際上，線寬取值爲 2 也是可以接受的。
若用非法參數調用 setlinestyle()函數，那麼 graphresult()會返回錯誤代碼，並且當前線型繼續有效。
Turbo C 提供的線型與線寬定義在頭檔 graphics.h 中，表 1-10 和 1-11 分別列出了參數的取值與含義。

表 1-10 線型

名 稱	取 值	含 義
SOLID_LINE	0	實線
DOTTED_LINE	1	點線
CNTER_LINE	2	中心線
DASHED_LINE	3	虛線
USERBIT_LINE	4	用戶自定義線型

表 1-11 線寬

名 稱	取 值	說 明
NORM_WIDTH(常寬)	1	一個圖元寬(缺省值)
THICK_WIDTH(加寬)	3	三個圖元寬

這個函數的頭檔是 graphics.h
返回值： 無
例： 下面的程式顯示了 BC 中所提供的線型圖樣：

```
＃i nclude<graphics.h>
void main()
{
int driver,mode;
driver=DETECT;
mode=0;
```

```
initgraph(&driver,&mode,"");
for(i=0;i<4;i++)
{
setlinestyle(i,0,1);
line(i*50,200,i*50+60,200)
}
getch();
restorecrtmode();
}
```

62. getlinesettings() 獲取線型設置函數

功能： 函數 getlinesettings() 用當前設置的線型、線圖樣和線寬填 寫 linesettingstype 型結構。

用法： 函數調用方式爲 void getlinesettings(struct linesettingstype *info);

說明： 此函數調用執行後，當前的線型、線圖樣和線寬值被裝入 info 指向的結構裏，從而可從該結構中獲得線型設置。

linesettingstype 型結構定義如下：

```
struct linesettingstype {
int linestyle;
unsigned upattern;
int thickness;
};
```

其中 linestyle 用於存放線型，線型值爲表 1-10 中的各值之一。

upattern 用爲裝入用戶自定義線圖樣，這是 16 位字，每一位元等於一個圖元，如果哪個位元被設置，那麼該圖元打開，否則關閉。

thickness 爲線寬值存放的變數，可參見表 1-11。

getlinesettings()函數對應的頭檔爲 graphics.h

返回值： 返回的線型設置存放在 info 指向的結構中。

例： 把當前線型的設置寫入 info 結構：

```
struct linesettingstype info;
getlinesettings(&info);
```

63.setwritemode() 設置畫線模式函數

功能： 函數 setwritemode() 設置畫線模式

用法： 函數調用方式爲 void setwritemode()(int mode);

說明： 參數 mode 只有兩個取值 0 和 1，若 mode 爲 0，則新畫的線將複蓋螢幕上原有的圖形，此爲缺省畫線輸出模式。如果 mode 爲 1，那麼新畫的圖元點與原有圖形的圖元點先進行異或(XOR)運算，然後輸出到螢幕上，使用這種畫線輸出模式，第二次畫同一圖形時，將擦除該圖形。調用 setwritemode()設置的畫線輸出模式只影響函數

line(),lineto(),linerel(),recangle()和 drawpoly()。

setwritemode()函數對應的頭檔是 graphics.h

返回值： 無

例： 設置畫線輸出模式爲 0：

```
setwritemode(0);
```

(三)、多邊形函數

對多邊形，無疑可用畫直線函數來畫出它，但直接提供畫多邊形的函數會給用戶很大方便。最常見的多邊形有矩形、矩形塊(或稱條形)、多邊形和多邊形塊，我們還把長方形條塊也放到這裏一起考慮，雖然它不是多邊形，但它的特例就是矩形(塊)。下面直接介紹畫多邊形的函數。

64. rectangle() 畫矩形函數

功能： 函數 rectangle() 用當前繪圖色、線型及線寬，畫一個給定左上角與右下角的矩形(正方形或長方形)。

用法： 此函數調用方式爲 void rectangle(int left,int top,int right,int bottom);

說明： 參數 left,top 是左上角點座標，right,bottom 是右下角點座標。如果有一個以上角點不在當前圖形視口內，且裁剪標誌 clip 設置的是真(1)，那麼調用該函數後，只有在圖形視口內的矩形部分才被畫出。

這個函數對應的頭檔爲 graphics.h

返回值： 無

例： 下面的程式畫一些矩形實例：

```
＃i nclude<graphics.h>
void main()
{
int driver,mode;
driver=DETECT;
mode=0;
initgrpah(&driver,&mode,"");
rectangle(80,80,220,200);
rectangle(140,99,180,300);
rectangle(6,6,88,88);
rectangle(168,72,260,360);
getch();
restorecrtmode();
}
```

65. bar() 畫條函數

功能： 函數 bar()用當前填充圖樣和填充色(注意不是給圖色)畫出一個指定上左上角與右下角的實心長條形(長方塊或正方塊)，但沒有四條邊線)。

用法： bar()函數調用方式爲 void bar(int left,int top,int right,int bottom);

說明： 參數 left,topright,bottom 分別爲左上角座標與右下角座標，它們和調用函數 rectangle()的情形相同，調用此函數前，可用 setfillstyle()或 setfillpattern()設置當前填充圖樣和填充色。

注意此函數只畫沒有邊線的條形，如果要畫有邊線的的條形，可調用下面的函數 bar3d()來畫，並將深度參數設爲 0，同時 topflag 參數要設置爲真，否則該條形無頂邊線。

這 應的頭文件爲 graphics.h

返回值： 無

例： 見函數 bar3d()中的實例。

66.bar3d() 畫條塊函數

功能： 函數 bar3d() 使用當前繪圖色、線型及線寬畫出三維長方形條塊，並用當前填充圖樣和填 充色填充該三維條塊的表面。

用法： 此函數調用方式爲 void bar3d(int left,int top,int right,int bottom,int depth,int topflag);

說明： 參數 left,top,right,bottom 分另爲左上角與右下角座標，這與 bar()函數中的一樣。參數 depth 爲條塊的深度，以圖元爲單位，通常按寬度的四分之一計算。深度方向通過屏顯縱橫比調節爲約 45 度(即這時 x/y 比設置爲 1：1)。

參數 topflag 相當於一個布林參數，如果設置爲 1(真)那麼條塊上放一頂面；若設置爲 0(假)，則三維條形就沒有頂面，這樣可使多個三維條形疊加在一起。

要使圖形更加美觀，可利用函數 floodfill()或 setfillpattern()來選擇填充圖樣和填充色(參見本小節(五)填充函數)。

bar3d()函數對應的頭檔爲 graphics.h

返回值： 無

例： 下面的程式畫一個條形和條塊：

```
＃i nclude<graphics.h>
void main()
{
int driver,mode;
driver=DETECT;
mode=0;
initgraph(&driver,&mode,"");
setfillstyle(SOLID-FILL,GREEN);
bar(60,80,220,160);
setfillstyle(SOLID-FILL,RED);
bar3d(260,180,360,240,20,1);
getch();
restorecrtmode();
}
```

67. drawpoly() 畫多邊形函數

功能： 函數 drawpoly() 用當前繪圖色、線型及線寬，畫一個給定若干點所定義的多邊形。

用法： 此函數調用方式爲 void drawpoly(int pnumber,int *points);

說明： 參數 pnumber 爲多邊形的頂點數；參數 points 指向整型陣列，該陣列中是多邊形所有頂點(x,y)座標值，即一系列整數對，x 座標值在前。顯然整型陣列的維數至少爲頂點數的 2 倍，在定義了多邊形所有頂點的陣列 polypoints 時，頂點數目可通過計算 sizeof(polypoints)除以 2 倍的 sizeof(int)得到，這裏除以 2 倍的原因是每個頂點有兩個整數座標值。另外有一點要注意，畫一個 n 個頂點的閉合圖形，頂點數必須等於 n+1，並且最後一點(第 n+1)點座標必須等於第一點的座標。

drawpoly()函數對應的頭檔爲 grpahics.h

返回值： 無

例： 下面的程式畫一個封閉星形圖與一個不封閉星形圖：

```
＃i nclude<graphics.h>
void main()
{
int driver,mode;
static int polypoints1[18]={100,100,110,120,100,130,120,125,140,140,130,120,
140,110,120,115,100,100};
static int polypoints2[18]={180,100,210,120,200,130,220,125,240,140,230,120,
240,110,220,115,220,110};
driver=DETECT;
mode=0;
initgraph(&driver,&mode,"");
drawpoly(9,polypoints1);
drawpoly(9,polypoints2);
getch();
restorecrtmode();
}
```

(四)、 圓、弧和曲線函數

在一個螢幕上畫得很圓的圖形到另一個螢幕上可能被壓扁或拉長，這是因爲每一種顯示卡與之相應的顯示模式都有一個縱橫比。縱橫比是指圖元的水準方向大小與垂直方向大小的比值。如 VGA 顯示卡由於偈素基本上是正方形，所以縱橫比爲 1.000。

爲了保證幾何圖形基本按預計情況顯示在螢幕上，用屏顯的縱橫比來計算和糾正不同硬體及顯示卡產生的畸變。計算縱橫比所需要的水準方向和垂直方向的比例係數可調用函數 getaspectratio()獲得。

68. getaspectratio()獲取縱橫比函數

功能： 函數 getaspectratio()返回 x 方向和 y 方向的比例係數，用這兩個整型值可計算某一特定屏顯的縱橫比。

用法： 此函數調用方式爲 void getaspectratio(int xasp,int yasp);

說明： 參數 xasp 指向的變數存放返回的 x 方向比例係數；參數 yasp 指向的變數存放返回的 y 方向比例係數。通常 y 方向比例係數爲 10 000， x 方向比例係數不大於 10 000(這是因爲大多數螢幕圖元高比寬長)。

注意縱橫比自動用作下面函數 arc(),circle()和 pieslice()中的尺規因數，使螢幕上圓或弧正常顯示。但用 ellipse()函數畫橢圓必須調用本函數獲取縱橫比作爲尺規因數，否則不予調整。縱橫比可用於其他幾何圖形，目的是校正和顯示圖形。

getaspectratio()函數對應的頭檔爲 graphics.h

返回值： 返回 x 與 y 方向比例係數分別存放在 xasp 和 yasp 所指向的變數中。

例： 下面的程式顯示縱橫比：

```
int xasp,yasp;
float aspectratio;
getaspectratio(&xasp,&yasp);
aspectratio=xasp/yasp;
printf("aspect ratio: %f",aspectratio);
```

69. circle()畫圓函數

功能： 函數 circle()使用當前繪圖色並以實線畫一個完整的圓。

用法：該函數調用方式爲 void circle(int x,int y,int radius);

說明： 參數 x,y 爲圓心座標，radius 爲圓半徑，用圖元個素表示。注意，調用 circle()函數畫圓時不用當前線型。

不同於 ellipse()函數，只用單個半徑 radius 參數調用 circle()函數，故屏顯縱橫比可以自動調節，以產生正確的顯示圖。

此函數對應的頭檔爲 graphics.h

返回值： 無

例： 畫六個同心圓，圓心在（100,100）。

```
＃i nclude<graphics.h>
void main()
{
int driver,mode;
driver=DETECT;
mode=0;
initgraph(&driver,&mode,"");
circle(100,100,10);
circle(100,100,20);
circle(100,100,30);
circle(100,100,40);
circle(100,100,50);
circle(100,100,60);
getch();
```

```
restorecrtmode();
}
```

70. arc() 畫圓弧函數

功能： 函數 arc()使用當前繪圖色並以實線畫一圓弧。

用法： 函數調用方式爲 void arc(int x,int y,int startangle,int endangle,int radius);

說明： 參數 x,y 爲圓心座標，startangle 與 endangle 分別爲起始角與終止角，radius 爲半徑。圓心座標和半徑以圖元個數給出，起始角和終止角以度爲單位，0 度位於右邊，90 度位於頂部，180 度位於左邊，底部是 270 度。同往常一樣，360 度與 0 度重合。角度按逆時針方向增加，但並不要求終止角一定比起始角大。例如指定 300 度和 90 度分別爲起始角和終止角，與指定 300 度和 450 度分別爲起始角和終止角可畫出相同的弧。大於 360 度可作爲參數，它將被化到 0 度¯360 度範圍裏。函數 arc()能畫封閉圓，只要取起始角爲 0 度，終止角爲 360 度即可。此函數中，屏顯縱橫比可自動調節。

arc()函數對應的頭檔爲 graphics.h

返回值： 無

例： 以（200,200）爲圓心，100 爲半徑，從 0 度到 120 度畫圓弧：

```
＃i nclude<graphics.h
void main()
{
int driver,mode;
driver=DETECT;
mode=0;
initgraph(&drivwer,&mode,"");
setcolor(WHITE);
arc(200,200,0,120,100);
getch();
restorecrtmode();
}
```