هذه النسخة بتاريخ: 2007/11/12

برمجة الرسم بلغة سي المحسنة

الجزء الثاني

باستخدام TURBO C PLUS PLUS 3.0

برمجة: البراء عبد الرؤوف الرملي

طرابلس / ليبيا

نسخة © 2007 , حقوق الطبع محفوظة SBR

Software Bara Ramli (SBR)

لا يسمح بإعادة طبع هذا الكتاب إلا بإذن خطي مسبق من المؤلف.

بينما يسمح بنسخه و تصويره في نطاق الاستعمال الشخصي (الغير تجاري) , ولكن لا يمكنك الإدعاء بأنك من قام بهذا العمل وعليك الإشارة لمؤلفه الأصلي.

ملاحظة: يقدم هذا الكتاب كما هو من دون أي كفالة أو ضمان لمحتوياته.

All programs in this book is free software:
you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

هذا الكتاب مجاني

مقدمة

أقدم إليكم كتاب برمجة الرسم الجزء الثاني وهو تتمة لأساسيات الرسم على الشاشة مع شرح لدوال مكتبة الرسم graphics.h

أرجو الله أن ينفع به وأن يكون مساهمة منا في إثراء المكتبة العربية والمبرمج العربي.

ملاحظة: المكتبات والبرامج المعروضة , مرفقة مع الكتاب في مجلد (المرفقات).

وأريد أن أنبه على أن البرامج والمكتبات المعروضة في هذه السلسلة , لا زالت تحتاج إلى تطوير وإضافات, وهذا يقع على عاتقنا جميعا حتى نصل بها إلى المستوى المطلوب , لذا فهي الآن بين يديك لتضيف إليها ما تظن أنه يرقى بها إلى الأفضل ومن ثم تقوم بنشرها لتعم الفائدة لنا جميعا , لأنه ما لم نتشارك بأفكارنا , فلن نتقدم خطوة إلى الأمام.

البراء عبد الرؤوف الرملي
opencpp@yahoo.com
طرابس/ليبيا

يمكنك زيارة موقعي: www.khayma.com/opencpp

المحتويات

الفصل الرابع/

النقطة المرجعية

النقطة المرجعية

النقطة المرجعية هي نقطة معلومة ولكن لا تظهر على الشاشة ويمكن
معرفة موقعها أثناء البرنامج كما يمكن تغيير مكانها.

دالة لتغيير مكان النقطة المرجعية/ moveto

لتغيير مكان النقطة المرجعية إلى الإحداثي (x,y):

```
moveto(x,y);
```

دالة لإزاحة مكان النقطة المرجعية/ moveto

لإزاحة النقطة المرجعية (مسافة قدرها dx أفقيا ومسافة قدرها dy رأسيا) وذلك نسبة إلى الموقع الأصلي للنقطة المركزية.

```
moverel(dx,dy);
```

دالة للحصول على الإحداثي السيني للنقطة المرجعية/ getx

تمكنك من الحصول على قيمة عددية.

مثلا: لو كان الإحداثي السيني=100 فسوف ترجع الدالة رقم (100) وتخصصه للمتغير m مثلا.

```
m=getx();
```

دالة للحصول على الإحداثي الصادي للنقطة المرجعية/ gety

تمكنك من الحصول على قيمة عددية .

مثلا: لو كان الإحداثي الصادي=100 فسوف ترجع الدالة رقم (100) وتخصصه للمتغير m مثلا.

```
m=gety();
```

دالة لرسم مستقيم تتابعي/ lineto

لرسم مستقيم من النقطة المرجعية إلى النقطة (x,y) ثم تحويل النقطة الجديدة إلى نقطة مرجعية.

```
lineto(x,y);
```

دالة لرسم مستقيم تتابعي آخر/ linerel

لرسم مستقيم من النقطة المركزية إلى نقطة (تبعد أفقيا مسافة قدرها dx وتبعد مسافة قدرها dy رأسيا) وذلك نسبة إلى النقطة المرجعية, ثم تحويل النقطة الجديدة إلى نقطة مرجعية.

```
linerel(dx,dy);
```

الفصل الخامس/

كتابة النصوص

دالة لإظهار نص عند النقطة المرجعية/ outtext

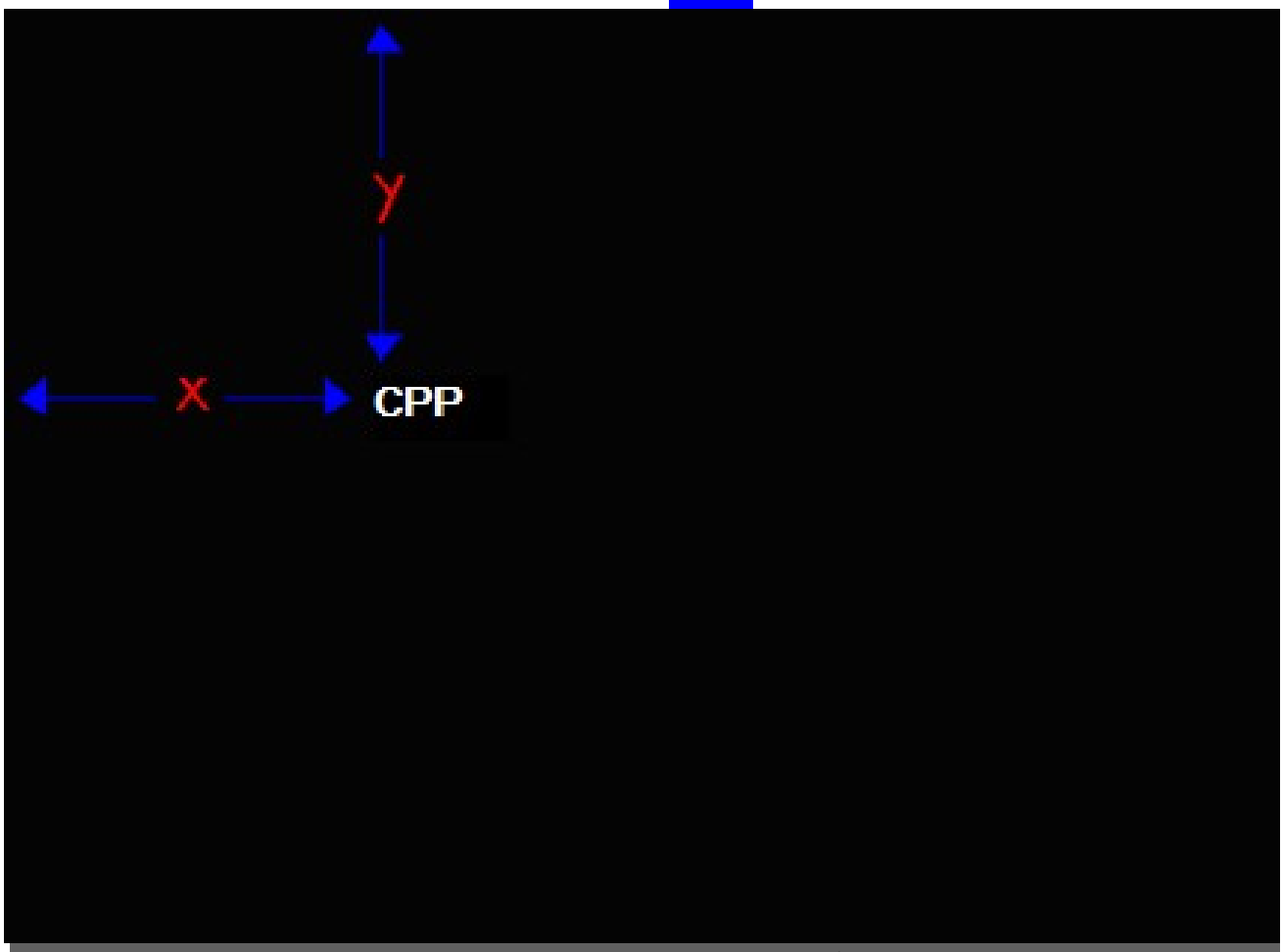
ملاحظة: ضع النص المراد إظهاره بين علامتي التنصيص في المكان المظلل.

فيظهر النص عند إحداثيات النقطة المركزية (x,y).

ملاحظة: الموقع الإبتدائي للنقطة المركزية هو (0,0) ما لم يتم تغيير مكانها.

ملاحظة: هذه الدالة تطبع النصوص ولا تطبع المتغيرات العددية.

```
outtext("CPP");
```



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
int x=45,y=60;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode,
"c:\\tc\\bgi");
moveto(x,y);
outtext("CPP");
getch();
}
```

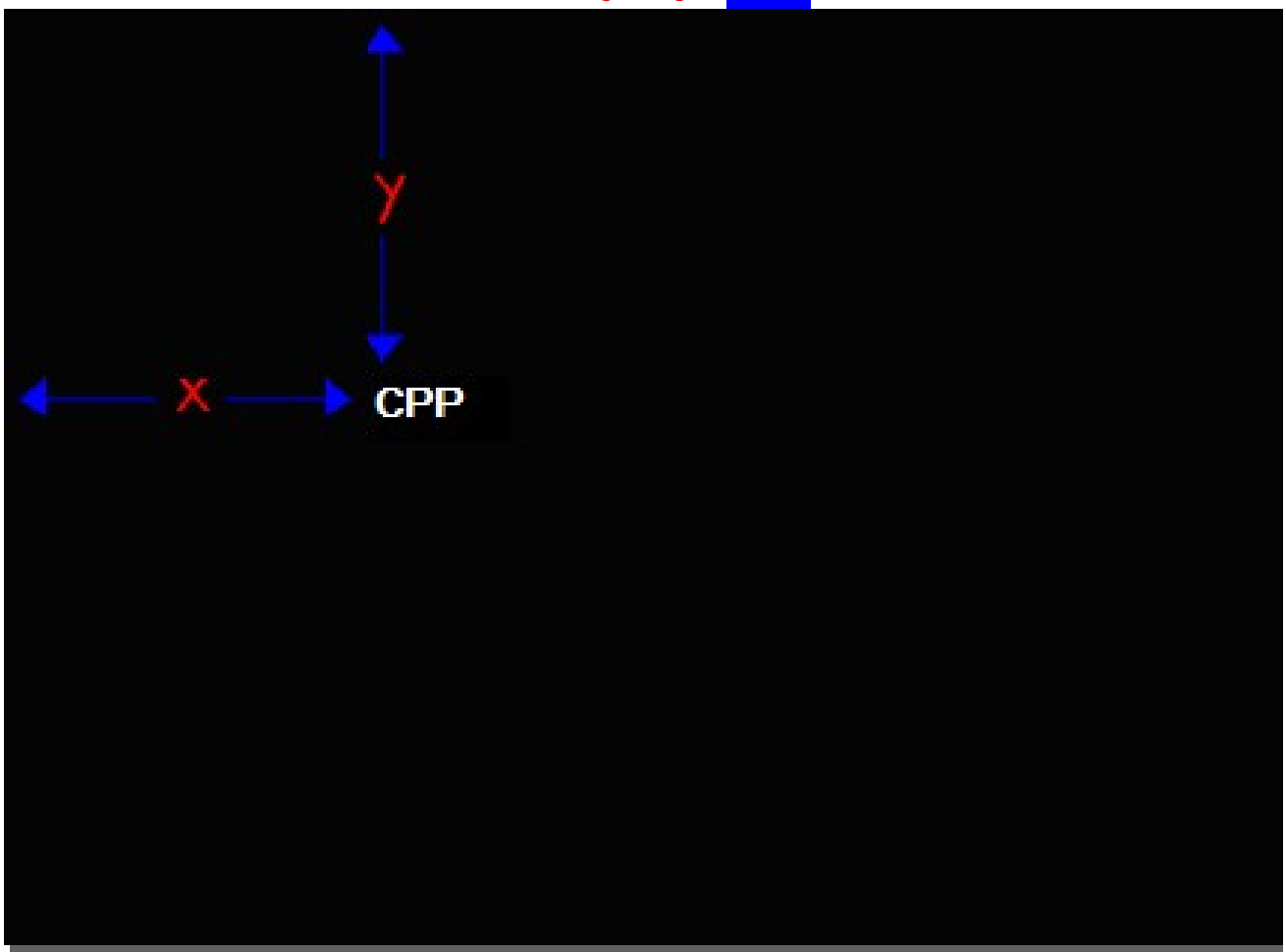
دالة إظهار نص عند نقطة معينة/ outtextxy

تحتوي على 2 متغيرات عددية هما: إحداثيات النقطة (x,y).

ملاحظة: ضع النص المراد إظهاره بين علامتي التنصيص في المكان المظلل.

ملاحظة: هذه الدالة تطبع النصوص ولا تطبع المتغيرات العددية.

outtextxy(x,y,"CPP");



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

#include <stdio.h>	
#include <conio.h>	
#include <graphics.h>	
void main()	
{	
int x=45,y=60;	
int gdriver = DETECT, gmode, errorcode;	
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	
outtextxy(x,y,"CPP");	
getch();	
}	

دالة تغيير حجم ونوع النص/ settextstyle

تحتوي على 3 متغيرات عددية:
A قيمة عددية للحجم (الأحجام مرتبة تصاعديا من 1 إلى 11)
B قيمة عددية لاتجاه النص (0 للاتجاه الأفقي بينما 1 للاتجاه العمودي), ملاحظة: "النص باللغة الإنجليزية"
C قيمة عددية لنوع الخط (نوع الخط مرتب تصاعديا من 1 إلى 10 تقريبا)

```
settextstyle(A,B,C);
```

تنبيه: تكتب (هذه الدالة) قبل دالة (كتابة النص).

دالة لتخزين القيم العددية في مصفوفة نصية/ sprintf

تستخدم هذه الدالة لتحويل الأعداد إلى مصفوفة نصية, حتى نستطيع طباعة الأعداد باستخدام دالة outtext

```
sprintf(msg,"%d %d",a,b);
```

a متغير صحيح.
b متغير صحيح.
msg مصفوفة من نوع char

#include <stdio.h>	
#include <conio.h>	
#include <graphics.h>	
void main()	
{	
char msg[10]; int a=12,b=10,x=50,y=50;	
int gdriver = DETECT, gmode, errorcode;	
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	
sprintf(msg,"%d %d",a,b); outtextxy(x,y,msg);	
getch();	
}	

الفصل السادس/

الأشكال الهندسية المسطحة

دالة رسم قطع ناقص مصمت/ fillellipse

.تستخدم لرسم قطع ناقص مع طلائه من الداخل باللون المطلوب

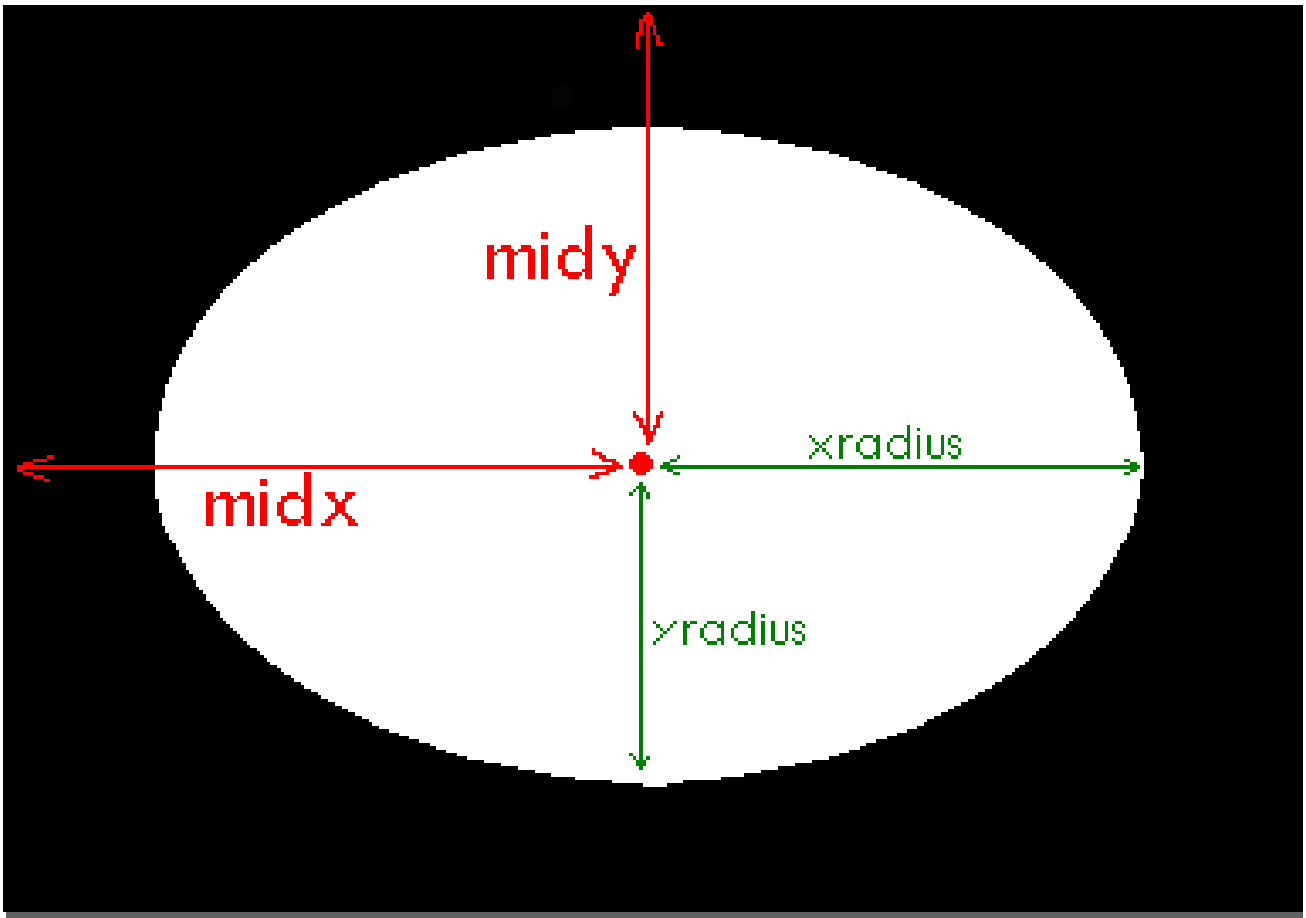
احداثي المركز=(midx, midy)

نصف قطر الإحداثي x =Xradius

نصف قطر الإحداثي y =Yradius

.ملاحظة: تشبه دالة رسم القطع الناقص فيما عدا أنها لا تتضمن زاويتي البدء والنهاية, لأنها ترسم قطع ناقص

Fillellipse(midx, midy, xradius, yradius);



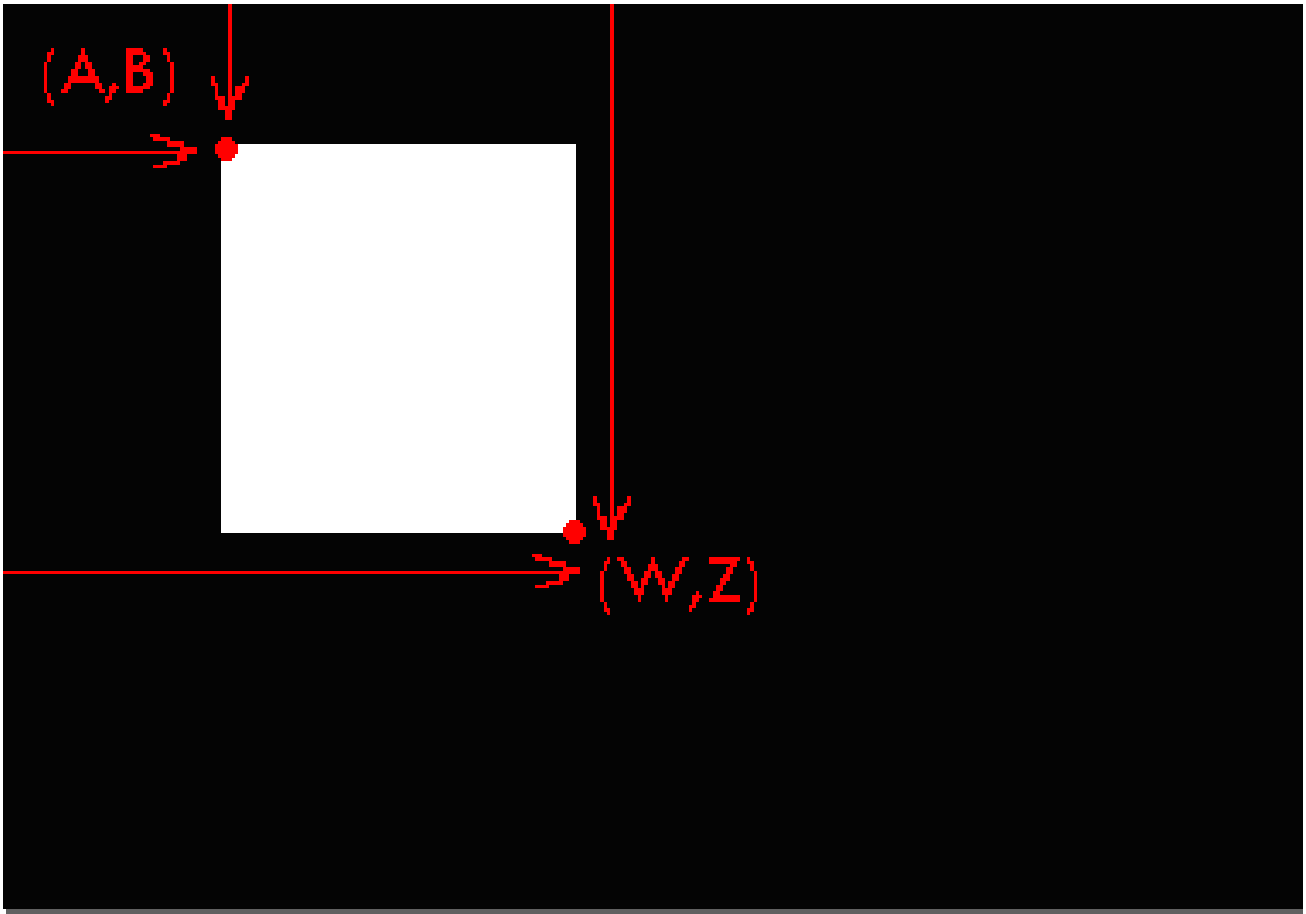
المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
int midx=320, midy=240, stangle = -45;
int endangle = 135, xradius = 70, yradius = 30;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
fillellipse (midx, midy, xradius,yradius);
getch();
}
```

دالة رسم مستطيل مصمت/ bar

إحداثي أقصى اليسار=(A,B)
إحداثي أقصى اليمين= (W,Z)

bar(A,B,W,Z);



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

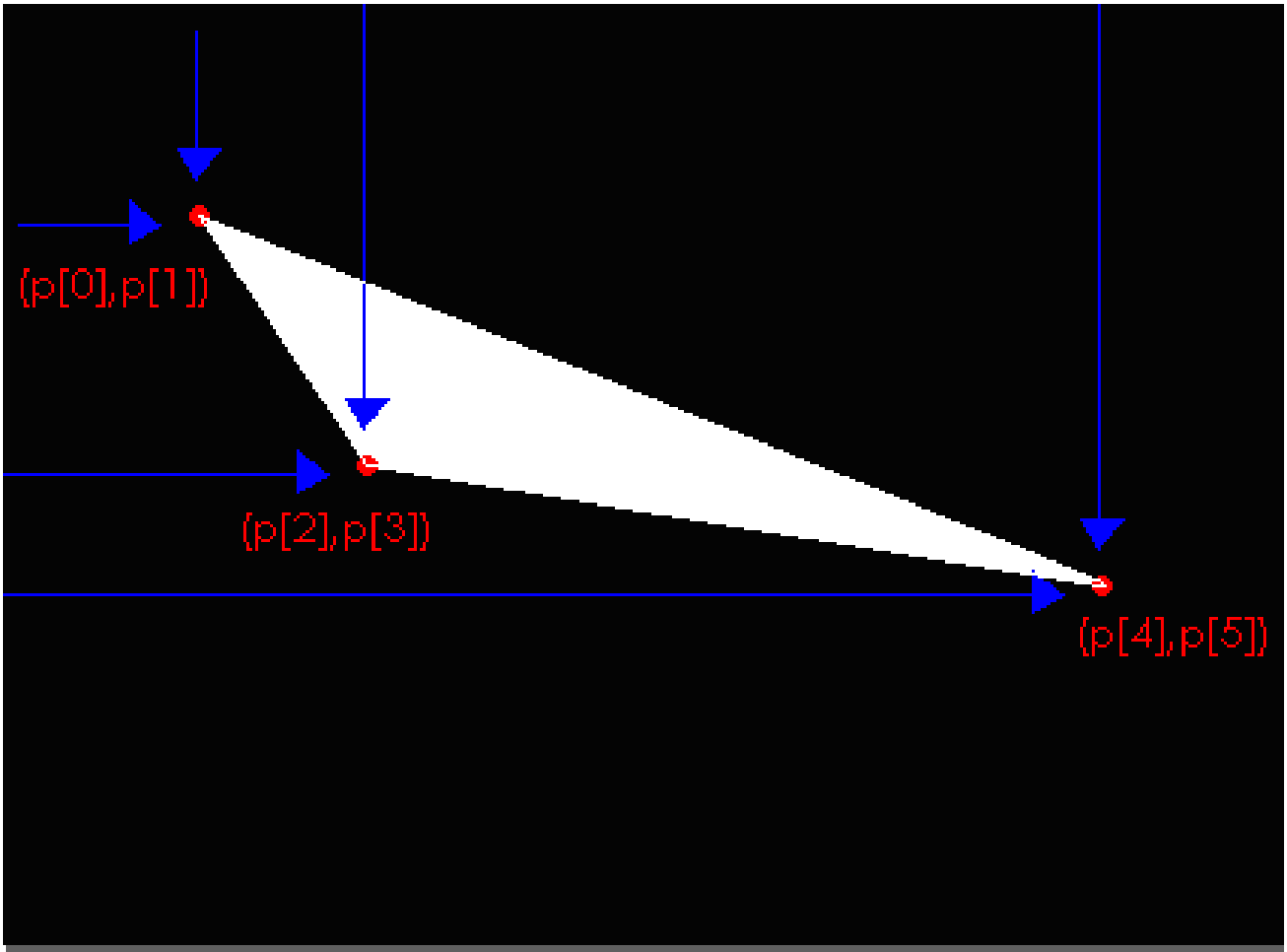
```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
int a=10,b=20,w=150,z=200;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
bar(a,b,w,z);
getch();
}
```

دالة رسم أشكال مضلعة مصمته/ fillpoly

تستخدم لرسم شكل مضلع مصمت بالإضافة لتلوينه.

اسم مصفوفة النقاط هي p حيث n هو عدد النقاط.

Fillpoly(n,p);



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

#include <stdio.h>	
#include <conio.h>	
#include <graphics.h>	
void main()	
{	
int p[6];	
int gdriver = DETECT, gmode, errorcode;	
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	

p[0]=10;p[1]=15;
p[2]=100;p[3]=200;
p[4]=200;p[5]=250;
fillpoly(3,p);	

getch();	
}	

الفصل السابع/

تلوين الأشكال المسطحة

دالة تغيير نوع ولون السطح المصمت/ setfillstyle

تستخدم لملء المساحات باللون المطلوب.

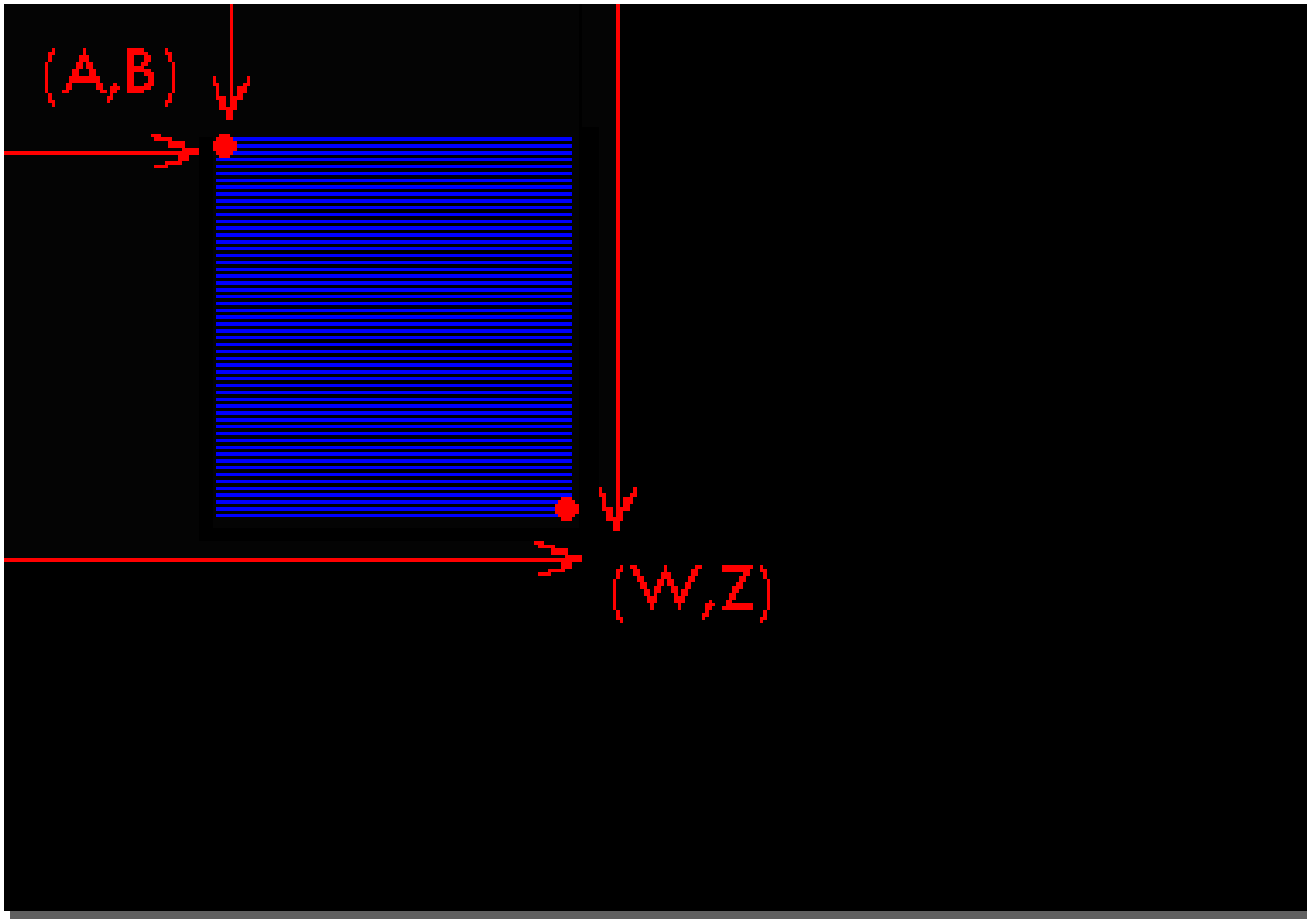
يمكنك تغيير لون ونوع السطح للأشكال المرسومة باستخدام هذه الدالة:

K عدد صحيح من 0 إلى 12 يرمز لنوع السطح

C عدد صحيح من 0 إلى 15 يرمز للون السطح

setfillstyle(k,c);

ملاحظة: يجب أن تكتب (هذه الدالة) قبل (دالة الرسم), وإذا لم تستعمل دالة تغير اللون فإن لون السطح سيكون أبيض تلقائيا.



المرسوم باللون الأبيض هو ما سيظهر في الشاشة عند تنفيذ البرنامج

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
int a=10,b=20,w=150,z=200;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
setfillstyle(2,1);
bar(a,b,w,z);
getch();
}
```

الفصل الثامن/

طرق الإزاحة

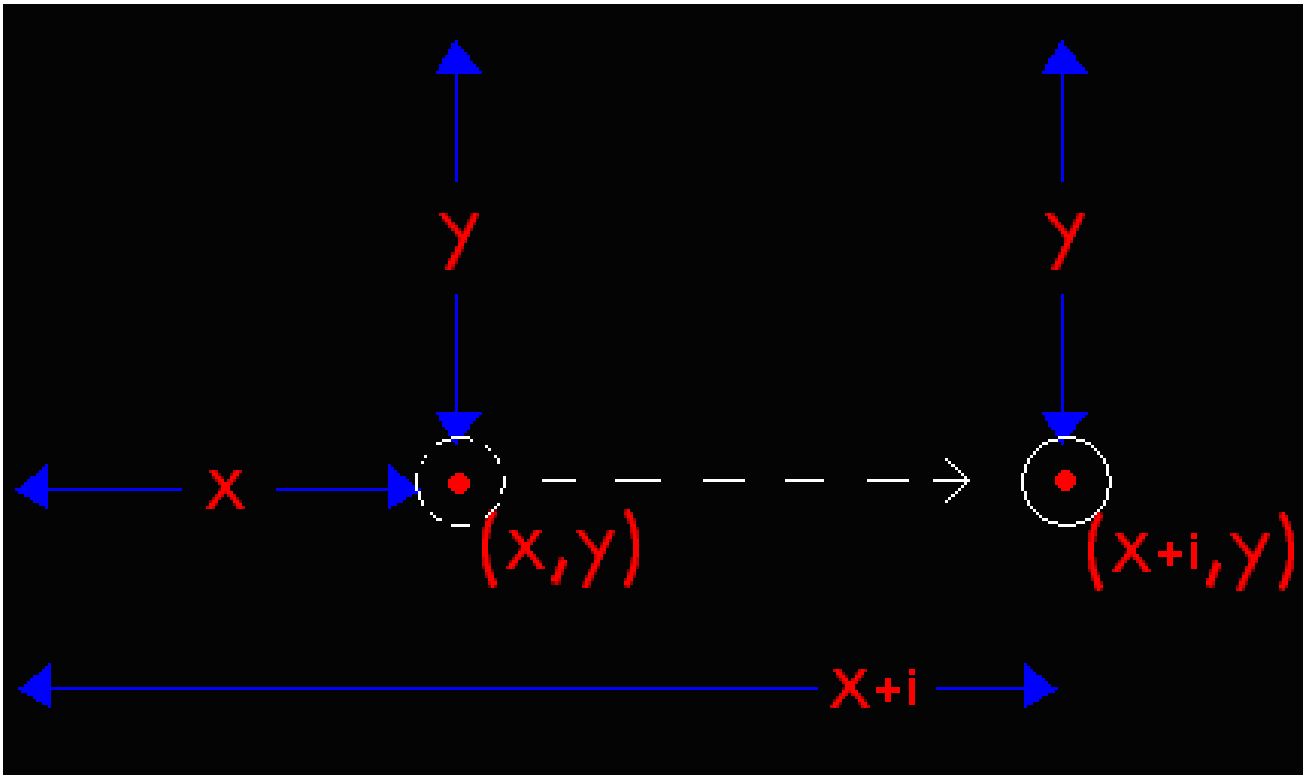
تصميم برنامج لتحريك دائرة أفقيا

إذا أردنا تحريك نقطة أفقيا جهة اليمين من (x1,y) إلى (x2,y).

1. فإننا نرسم الدائرة في (x1,y), ونصف قطرها h
2. ثم نثبتها على الشاشة لحظة, وذلك باستخدام دالة:

```
delay(50);
```

3. وهي تقوم بتثبيت الشاشة لمدة 50ms حيث (الثانية الواحدة= 1000ms), ويكتب مقدار الزمن بين قوسي الدالة في المكان المظلل.
4. ثم نقوم بمسحها وذلك بأن نرسم نفس الدائرة و لكن باللون الأسود (0)
5. ثم نقوم برسم الدائرة مرة أخرى ولكن بإزاحة قدرها x=x+1;
6. ونكرر هذه العمليات عدة مرات حتى نصل لمقدار الإزاحة المطلوبة.



#include <stdio.h> #include <conio.h> #include <dos.h> #include <graphics.h>	مكتبة الدالة delay

void main()
{
int x=45,y=60,h=3;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	

for(int i=0;i<100;i++)
{
setcolor(0); circle(x,y,h);
x=x+1;
setcolor(15); circle(x,y,h);
delay(50);
}
	مقدار الإزاحة من 0 إلى 99 نقطة.
	
	لمسح الدائرة (نرسمها باللون الأسود)
	إزاحة أفقية 1
	لرسم الدائرة (نرسمها باللون الأبيض)
	دالة تثبيت الشاشة لمدة 50 ملي ثانية

getch(); }	

طريقة أخرى لتحريك دائرة أفقيا

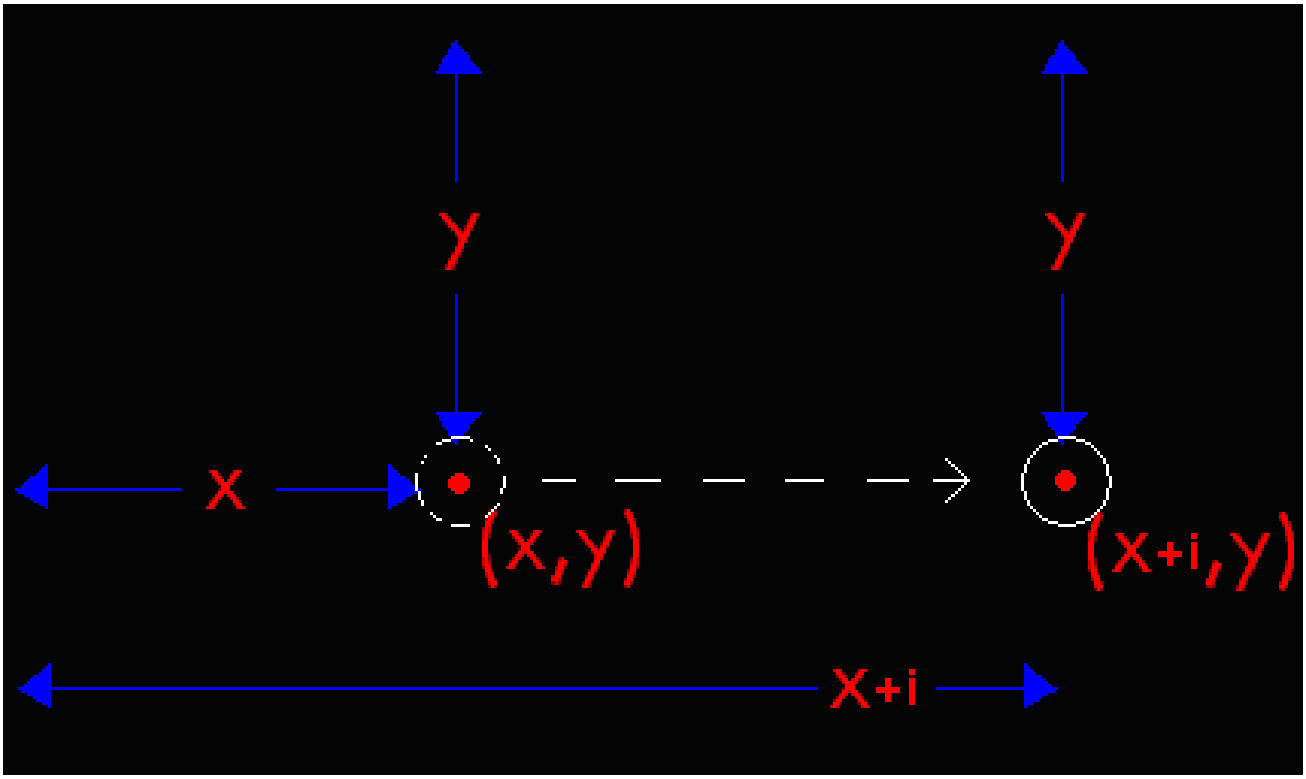
إذا أردنا تحريك نقطة أفقيا جهة اليمين من (x1,y) إلى (x2,y).

1. نرسم الدائرة وليكن مركزها هو (x1,y) ونصف قطرها h
2. ثم نثبتها على الشاشة لحظة, وذلك باستخدام دالة:

```
delay(50);
```

3. وهي تقوم بتثبيت الشاشة لمدة 50ms حيث (الثانية الواحدة= 1000ms), ويكتب مقدار الزمن بين قوسي الدالة في المكان المظلل.
4. ثم نقوم بمسح الشاشة كلها باستخدام دالة مسح الشاشة/ cleardevice
5. ثم نقوم برسم الدائرة مرة أخرى ولكن بإزاحة قدرها x=x+1

ونكرر هذه العمليات عدة مرات حتى نصل لمقدار الإزاحة المطلوبة.



#include <stdio.h> #include <conio.h> #include <dos.h> #include <graphics.h>	مكتبة الدالة delay

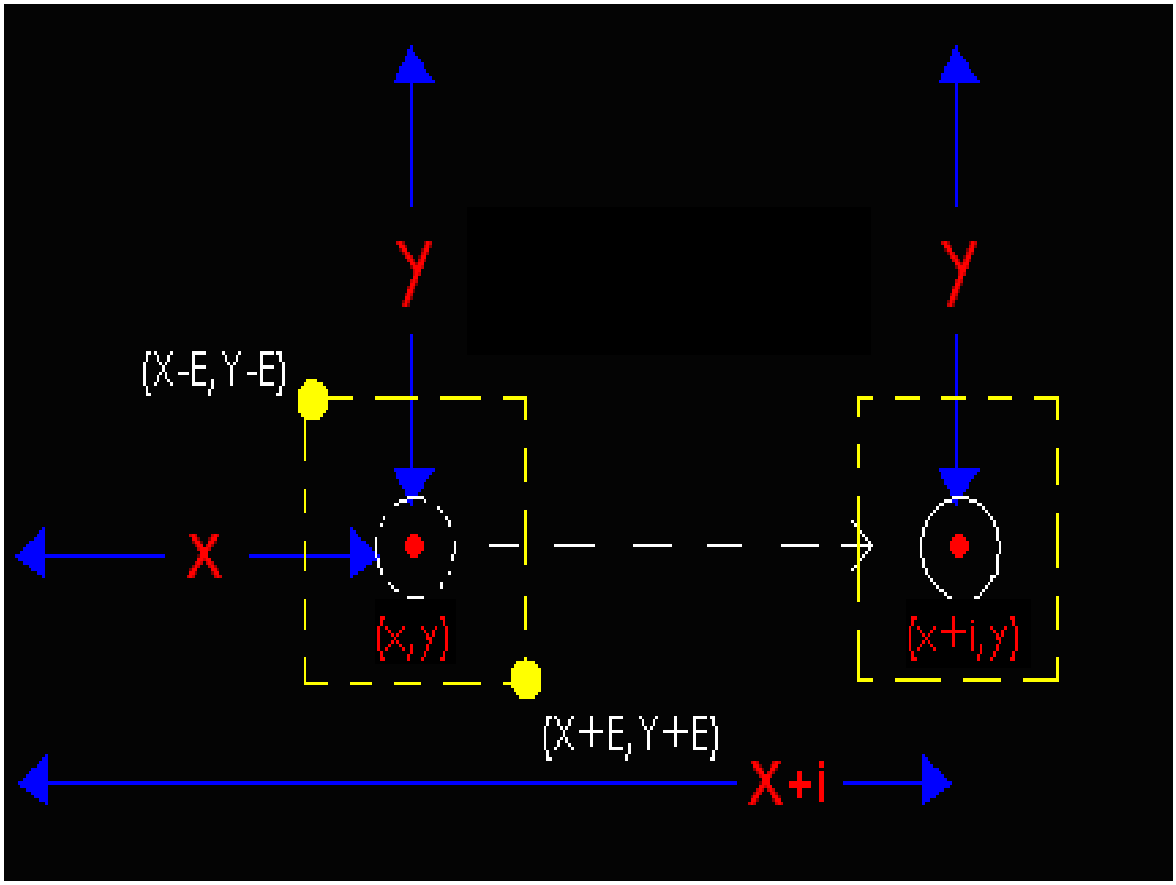
void main()
{
int x=45,y=60,h=3;
int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");	

for(int i=0;i<100;i++)
{
cleardevice ();
x+=1;
circle(x,y,h);
delay(50);
}
	مقدار الإزاحة من 0 إلى 99 نقطة.
	
	دالة مسح الشاشة
	إزاحة أفقية 1
	
	دالة تثبيت الشاشة لمدة 50 ملي ثانية

getch(); }	

تصميم برنامج لتحريك دائرة أفقيا (وإيقاف الحركة عند أي لحظة)

يقوم البرنامج بعمل نسخة من الشكل في الذاكرة ثم يقوم بمسح الشكل من الشاشة ثم يرسم الشكل مرة أخرى ولكن بإزاحة قدرها "واحد صحيح=نقطة واحدة", ويتوقف البرنامج عند الضغط على أي زر باستخدام دالة kbhit




```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>
void draw_arrow(int x, int y);
```

```
int main()
{
int gdriver = DETECT, gmode, errorcode;
void *arrow;
int x, y, E=10, i=0;
unsigned int size;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
x = 45;
y = 60;
draw_arrow(x, y);
size = imagesize(x-E, y- E, x+ E, y+ E);
arrow = malloc(size);
```

```
getimage(x-E, y- E, x+ E, y+ E, arrow);
```

```
while (!kbhit())
{
putimage(x-E, y- E, arrow, XOR_PUT);
x += 1;
if (i==100) x = 45;
putimage(x-E, y- E, arrow, XOR_PUT);
delay(10);
i+=1;
}
free(arrow);
closegraph();
return 0;
}
```

```
void draw_arrow(int x, int y)
{
int h=3;
circle(x,y,h);
}
```