

## الفصل الرابع

---

### الفأرة

سننتعلم في هذا الفصل كيفية اكتشاف واستخدام الحوادث التي تتجم عن استخدام الفأرة، والتعامل مع النقر على أحد أزرارها أو مع الدمج بين النقر على أحد أزرارها والضغط على لوحة المفاتيح. كما سنتعلم أيضاً ما يتعلق بسحب وإسقاط الكائنات بواسطة الفأرة.

#### برنامج التحريك

يوضح برنامج التحريك كيفية تصميم برامج تمكن المستخدم من تحريك الكائنات بواسطة الفأرة ضمن إطار البرنامج.

#### التمثيل المرئي لبرنامج التحريك

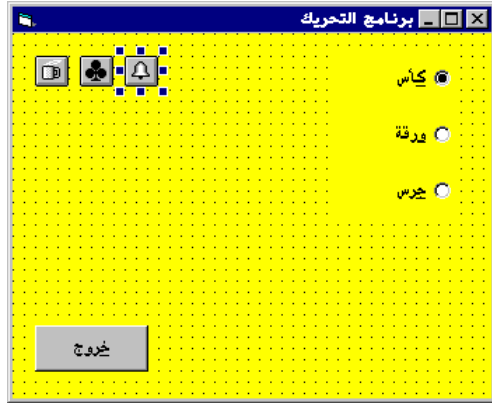
سنبدأ بطور التمثيل المرئي للبرنامج:

□ أنشئ الدليل C:\VB5Prg\Ch04، سنستخدم هذا الدليل لحفظ العمل المنجز فيه.

- ❑ أنشئ مشروعاً جديداً من نوع Standard EXE.
  - ❑ احفظ نموذج المشروع بالاسم Move.Frm في الدليل C:\VB5Prg\Ch04، واحفظ ملف المشروع بالاسم Move.Vbp في نفس الدليل.
  - ❑ أنشئ النموذج frmMove لبرنامج التحريك طبقاً للجدول ١-٤.
- يفترض أن يبدو النموذج عند اكتماله كما في الشكل ١-٤.

الشكل ١-٤

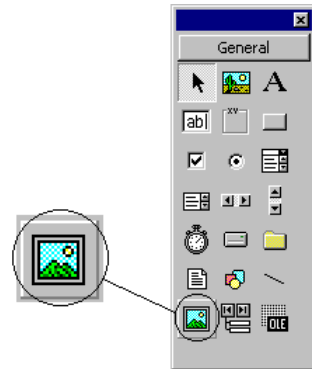
النموذج frmMove  
في مرحلة التصميم.



يبين الشكل ٢-٤ رمز عنصر التحكم المدعو Image، ولقراءة اسم عنصر التحكم Image ضع مؤشر الفأرة فوق رمزه بدون النقر عليه، فيظهر إطار صغير أصفر يحمل النص Image داخله.

الشكل ٢-٤

رمز عنصر التحكم Image في إطار  
مربع الأدوات.



الجدول ٤-١. جدول خصائص برنامج التحريك.

القيمة	الخاصية	الكائن
<b>frmMove</b>	<b>Name</b>	<b>Form</b>
برنامج التحريك	Caption	
True	RightToLeft	
Yellow	BackColor	
<b>cmdExit</b>	<b>Name</b>	<b>CommandButton</b>
&خروج	Caption	
True	RightToLeft	
<b>optCub</b>	<b>Name</b>	<b>Option Button</b>
&كأس	Caption	
Yellow	BackColor	
True	Value	
True	RightToLeft	
<b>optClub</b>	<b>Name</b>	<b>Option Button</b>
&ورقة	Caption	
Yellow	BackColor	
True	RightToLeft	
<b>optBell</b>	<b>Name</b>	<b>Option Button</b>
&جرس	Caption	
Yellow	BackColor	
True	RightToLeft	
<b>imgCup</b>	<b>Name</b>	<b>Image</b>
Cup.Bmp	Picture	
<b>imgClub</b>	<b>Name</b>	<b>Image</b>
Club.Bmp	Picture	
<b>imgBell</b>	<b>Name</b>	<b>Image</b>
Bell.Bmp	Picture	

يطالبك الجدول ٤-١ بوضع ثلاثة عناصر تحكم من نوع Image في النموذج frmMove.

تذكر ما يلي لدى استخدام الجدول ٤-١:

■ لديك حرية تحديد خصائص الخط Font لعناصر التحكم التي تضعها في النموذج.

■ يطالبك الجدول بإسناد ملفات متنوعة من نوع BMP إلى الخصائص Pictures لعناصر التحكم Image. المشكلة أن هذه الملفات قد تكون متوفرة أو غير متوفرة لديك تبعاً لإصدار فيجول بيسك الذي تستخدمه. وهذه الملفات ستجدها في أحد الأدلة الفرعية تحت الدليل الذي تم تنصيب فيجول بيسك فيه. بل ستجدها غالباً تحت الدليل Bitmaps. فإذا لم تتمكن من العثور عليها، (بسبب أن رزمة فيجول بيسك التي تستخدمها لا توفرها). فبإمكانك استخدام ملفات BMP من أي مصدر آخر. تستطيع مثلاً استخدام برنامج الرسم Paint الذي يأتي مع ويندوز لرسم صور خاصة بك وحفظها كملف من نوع BMP.

### إدخال نص برنامج التحريك

سندخل الآن نص برنامج التحريك. ونبدأ كما قمنا بقسم التصاريح العامة General Delarations:

□ تحقق من أن قسم التصاريح العامة General Delarations للنموذج frmMove يحوي العبارة التالية:

يجب التصريح عن كل المتحولات'  
Option Explicit

□ أدخل النص التالي ضمن الإجراء Form\_MouseDown():

```
Private Sub Form_MouseDown(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If optBell.Value = True Then
        imgBell.Move X, Y
    ElseIf optClub.Value = True Then
        imgClub.Move X, Y
    Else
        imgCup.Move X, Y
    End If
End Sub
```

```
End If
End Sub
```

□ أدخل النص التالي ضمن الإجراء cmdExit\_Click():

```
Private Sub cmdExit_Click()
End
End Sub
```

□ احفظ المشروع باختيار **Save Project** من القائمة **File**.

### تنفيذ برنامج التحريك

قبل أن نباشر بشرح نص البرنامج الذي كتبناه، دعنا نرى كيف يعمل أولاً:

□ نفذ برنامج التحريك.

حسب ما يظهره الشكل ٤-٣، يظهر البرنامج ثلاث صور، وثلاثة أزرار خيارات معنونة كالتالي كأس و ورقة و جرس.

يؤدي نقر زر الفأرة على نافذة البرنامج، إلى انتقال أحد الصور إلى الموقع الذي تم النقر بالفأرة عنده. تحدّد الصورة المطلوب تحريكها باختيار أحد أزرار الخيارات الثلاثة.

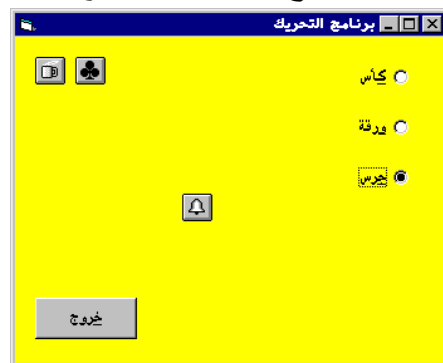
لنفترض مثلاً، أن الخيار يقع عند زر الخيار الجرس، يؤدي النقر بالفأرة في أي مكان نافذة البرنامج، إلى انتقال صورة الجرس إلى ذلك المكان (الشكل ٤-٣).

□ أنه البرنامج بنقر الزر خروج.

الشكل ٤-٣

برنامج التحريك

بعد تحريك الجرس.



## كيف يعمل برنامج التحريك

ستصادف لدى مناقشتنا برنامج التحريك، مصطلح إحداثيات النموذج Form Coordinates . لهذا ستبدأ مناقشتنا لبرنامج التحريك بعد هذا الموضوع.

## إحداثيات النموذج Form Coordinates

يمكن تحديد إحداثيات نموذج وفق عدة وحدات قياس بواسطة الخاصية ScaleMode. علماً بأن فيجول بيسك يستخدم وحدة قياس تدعى twip، في الحالة الافتراضية (يوجد 1440 twips في البوصة الواحدة، 1 inch = 1440 twips).

### ملاحظة

- يمكن إسناد أي من الوحدات التالية إلى الخاصية ScaleMode:
- Twips كل بوصة واحدة تساوي 1440 twips.
- Points كل بوصة واحدة تساوي 72 Points (أي ٧٢ نقطة).
- Pixels يتحدد عدد النقاط الضوئية في كل بوصة تبعاً لدقة تحديد الشاشة Resolution.
- Character يعرف الرمز Character كمستطيل بعرض 128 Twips وارتفاع 240 twips.
- Inches بوصة.
- Millimeters ميليمترات.
- Centimeters سنتيمترات.

يتحدد قيمة مبدأ نظام إحداثيات النموذج بالخاصيتين ScaleLeft و ScaleTop للنموذج. علماً بأن القيمة الافتراضية لهاتين الخاصيتين هي صفر. مما يعني أن الزاوية العليا اليسارية، لمنطقة النموذج تمثل مبدأ الإحداثيات وتتحدد بالإحداثيين (٠,٠). ويقصد بمنطقة النموذج المنطقة القابلة للاستخدام والتي لا تتضمن الحواف ولا شريط العنوان. (المنطقة التي يمكن وضع عناصر التحكم عليها).

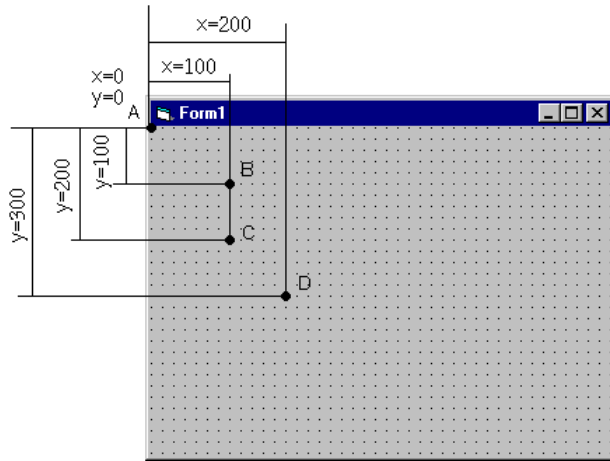
إذا باختصار، الخاصية ScaleTop للنموذج تساوي صفر، والخاصية ScaleLeft للنموذج تساوي صفر، والزاوية العليا اليسارية للنموذج، تقع عند الإحداثيين ( $x = 0$  و  $y = 0$ ). تزداد قيمة  $x$  كلما انتقلت باتجاه اليمين، وتزداد قيمة  $y$  كلما اتجهت إلى الأسفل (انظر الشكل ٤-٤).

الشكل ٤-٤

إحداثيات النموذج عندما

ScaleTop = 0

و ScaleLeft = 0



النقطة A حسب الشكل ٤-٤، تقع عند الإحداثيين ( $x = 0$  و  $y = 0$ ). وتقع النقطة B عند الإحداثيين ( $x = 100$  و  $y = 100$ )، وتقع C عند ( $x = 100$  و  $y = 200$ )، بينما تقع D عند ( $x = 200$  و  $y = 300$ ).

#### ملاحظة

يتألف النموذج من شريط عنوان الإطار، وحواف الإطار، والمنطقة المحاطة من شريط العنوان والحواف، والتي تدعى منطقة النموذج.

#### نص الإجراء cmdExit\_Click()

ينفذ هذا الإجراء عند نقر الزر خروج. تتسبب عبارة End الموجودة ضمن الإجراء بإنهاء الإجراء:

```
Private Sub cmdExit_Click()  
    End  
End Sub
```

### نص الإجراء Form\_MouseDown()

يمكن للفأرة أن تمتلك من زر واحد، وحتى ثلاثة أزرار. يؤدي الضغط على أي من أزرار الفأرة إلى الأسفل، ضمن منطقة النموذج، إلى تنفيذ الإجراء Form\_MouseDown() آلياً.

#### ملاحظة

يقصد بضغط أزرار الفأرة، الضغط عليها إلى الأسفل Down، أما نقر Clicking الفأرة، فيقصد به ضغط أحد الأزرار ثم تحريره. ينفذ الإجراء Form\_MouseDown() عند ضغط أحد أزرار الفأرة، (عندما تكون مشيرة الفأرة ضمن منطقة العمل للنموذج). وهكذا فالإجراء ينفذ، حتى قبل تحرير زر الفأرة.

لا ينفذ هذا الإجراء عند ضغط أحد أزرار الفأرة، فوق شريط عنوان النموذج، باعتبار أن منطقة النموذج لا تشمل شريط العنوان ولا الحواف Border. ينفذ الإجراء فقط عند نقر (أو ضغط) زر الفأرة في منطقة خالية (من عناصر التحكم) ضمن منطقة النموذج. فالإجراء Form\_MouseDown() مثلاً، لا ينفذ عند ضغط زر الفأرة فوق الزر خروج، أو فوق أحد أزرار الخيارات.

### وسائط الإجراء Form\_MouseDown()

يملك الإجراء أربعة وسائط: Button و Shift و X و Y تحتوي هذه الوسائط على معلومات عن حالة الفأرة أثناء النقر عليها:

```
Private Sub Form_MouseDown(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If optBell.Value = True Then
        imgBell.Move X, Y
    ElseIf optClub.Value = True Then
        imgClub.Move X, Y
    Else
        imgCup.Move X, Y
    End If
End Sub
```

ستجد أن السطر الأول من الإجراء `Form_MouseDown()` يمتد على سطر واحد في نافذة نص البرنامج، بدلاً من سطرين، كما هي الحال هنا، وذلك طبعاً لأن عرض صفحة الكتاب لا يتسع لكامل السطر.

الوسيط الأول من الإجراء `Form_MouseDown()` عبارة عن متحول صحيح Integer يدعى `Button`. تُساعد قيمة هذا الوسيط، بتحديد الزر المضغوط (الأيسر، الأوسط، الأيمن). لا يكثرث برنامج التحريك لهذا الوسيط لأنه لا يهتم من هو الزر المضغوط. الوسيط الثاني من الإجراء أيضاً من النوع الصحيح ويدعى `Shift`، تشير قيمة هذا الوسيط، إلى حالة مفاتيح التحكم `Shift` و `Ctrl` و `Alt` لحظة ضغط أحد أزرار الفأرة. لعل أفضل اسم لهذا الوسيط هو `ShiftCtrlAlt`، لكن لا بد من العمل مع الاسم الذي فرضه مصممو فيجول بيسك. لا يكثرث برنامج التحريك أيضاً لهذا الوسيط (لا يستخدمه). لأنه لا يبدي أي تصرف خاص تجاه الضغط على أحد هذه المفاتيح أثناء النقر على الفأرة.

الوسيطان الثالث والرابع هما `X` و `Y`، ويشيران إلى إحداثيي مؤشر الفأرة لحظة الضغط على زر الفأرة. طبعاً الإحداثيات تؤخذ بدلالة النموذج، لأن الإجراء ينفذ فقط عندما يكون مؤشر الفأرة ضمن منطقة النموذج. لنفترض مثلاً أن الخاصية `ScaleMode` للنموذج `frmMove` على الوضعية `Twips`، وأن قيمة كل من `ScaleTop` و `ScaleLeft` تساوي الصفر. إذا أعطى الإجراء `Form_MouseDown()` تقريراً بأن `x = 0` و `y = 0` عند النقر على الفأرة في منطقة النموذج فهذا معناه أن الضغط على الفأرة كان عند الزاوية العليا اليسارية. أما إذا كانت `x = 10` و `y = 20` فهذا يعني أن الضغط كان على بعد 10 twips من يسار النموذج و 20 twips تحت قمة النموذج، وهكذا.

### عبارة `If.Else` الواردة ضمن الإجراء `Form_MouseDown()`

تسهم عبارة `If.Else` ضمن الإجراء، بتحديد زر الخيار الذي انتقاه المستخدم. فمثلاً إذا كان زر الخيار الحالي هو الزر جرس فسوف يتحقق أول جزء من الشرط:

```
If optBell.Value = True Then
```

وستنفذ العبارة التالية:

```
imgBell.Move X, Y
```

التي تستخدم طريقة Move لنقل الصورة إلى الإحداثيين x و y. طبعاً x و y هما إحداثيي مؤشر الفأرة عندما تم الضغط على زر الفأرة. إذا باختصار، سنتقل صورة الجرس إلى النقطة ذات الإحداثيين x و y. (تصور أن عنصر التحكم Image محاط بمستطيل، تنقل الطريقة Move عنصر التحكم Image (والصورة المرافقة) بحيث تتوضع الزاوية العليا اليسارية للمستطيل عند الإحداثيين x و y بعد التحريك).

### الطريقة Move

تستخدم طريقة Move لتحريك الكائنات المختلفة، مثل النماذج وعناصر التحكم. استخدمنا العبارة `imgCup.Move x,y` مثلاً لنقل الكائن المدعو `imgCup` من موقعه الراهن إلى الموقع الجديد المحدد بالإحداثيين x و y:

```
imgCup.Move X, Y
```

#### ملاحظة

بعد تنفيذ العبارة التالية:

```
imgCup.Move x,y
```

يتحدد الموقع الجديد للزاوية العليا اليسارية للصورة بالإحداثيين x و y.

يؤدي استخدام العبارة التالية إلى توضع مركز الصورة عند الإحداثيين x و y:

```
imgCup.Move (X - imgCup.Width / 2),(Y - imgCup.Hight / 2)
```

### معلومة هامة عن أزرار الخيارات Option Buttons

أنهينا برنامج التحريك. لكن بقيت نقطة واحدة بخصوص أزرار الخيارات Option Buttons لا بد لك من الإطلاع عليها.

يمكن اختيار زر واحد من أزرار الخيارات التي تنتمي إلى مجموعة واحدة.

ففي مثالنا لدينا ثلاثة أزرار خيارات Option Buttons، ولكن يمكن اختيار واحد فقط من الأزرار الثلاثة، فنستطيع اختيار الجرس أو الكأس أو الورقة ولكن لا نستطيع اختيار الجرس والكأس سوياً.

قد تحتاج إلى استخدام أكثر من مجموعة أزرار خيارات واحدة ضمن نفس النموذج. تستطيع إضافة مجموعة ثانية من أزرار الخيارات إلى النموذج وفق ما يلي:

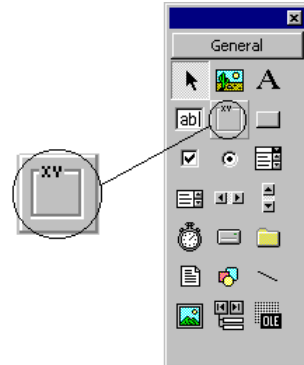
□ انقر نقرة مزدوجة على عنصر التحكم Frame في إطار مربع الأدوات (انظر الشكل ٤-٥ لمشاهدة شكل عنصر التحكم Frame ضمن مربع الأدوات). يؤدي وضع مؤشر الفأرة فوق أي عنصر تحكم ضمن مربع أدوات بدون ضغط أحد أزرار الفأرة، إلى ظهور مستطيل أصفر يحمل اسم عنصر التحكم المرافق. وبهذه الطريقة تستطيع التأكد من اسم عنصر التحكم الصحيح.

يستجيب فيجول بيسك بوضع عنصر التحكم Frame في النموذج frmMove.

الشكل ٤-٥

رمز عنصر التحكم Frame

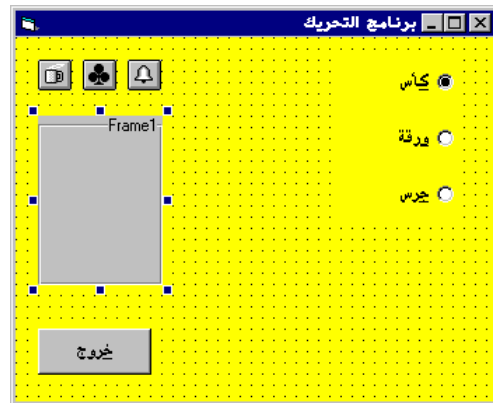
ضمن إطار مربع الأدوات.



الشكل ٤-٦

النموذج frmMove بعد إضافة عنصر

التحكم Frame إليه (طور التصميم).



□ انقر نقرة مزدوجة على رمز عنصر التحكم Frame مجدداً ضمن إطار مربع

الأدوات.

يستجيب فيجول بيسك بوضع عنصر تحكم *Frame* آخر في النموذج *frmMove*.

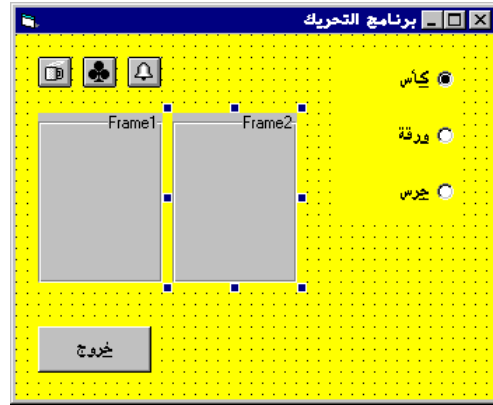
□ انقل وكبر عنصر التحكم *Frame* الثاني الذي وضعناه للتو في النموذج *frmMove*.

يفترض أن يبدو النموذج الآن حسب ما هو مبين في الشكل ٧-٤.

الشكل ٧-٤

النموذج *frmMove* بعد إضافة

عنصري تحكم *Frame* إليه.



□ ضع الآن زر خيار في أول إطار *Frame* أضفناه إلى النموذج:

□ انقر (لا تنقر نقرة مزدوجة) على رمز زر الخيار في إطار مربع الأدوات.

□ ضع مؤشر الفأرة ضمن عنصر التحكم *Frame* الأول واسحب الفأرة.

يستجيب فيجول بيسك بوضع زر خيار في عنصر التحكم *Frame*.

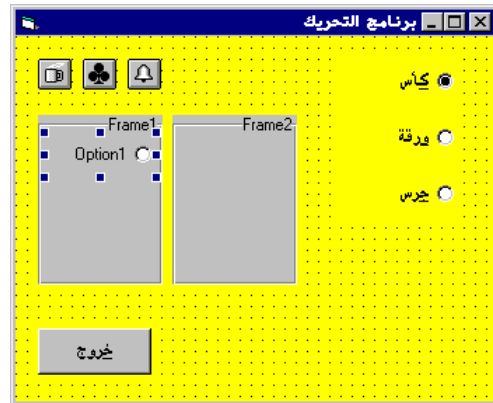
يفترض أن يبدو النموذج الآن كما في الشكل ٨-٤.

الشكل ٨-٤

النموذج *frmMove* بعد إضافة

زر خيار إلى عنصر التحكم

*Frame* الأول.



□ كرر خطوات إضافة زر خيار إلى الإطارين. تذكر عدم النقر المزدوج على رمز

زر الخيار في إطار مربع الأدوات. ضع زر خيار ثاني في الإطار الأول، وضع

ثلاثة أزرار خيارات في الإطار الثاني.

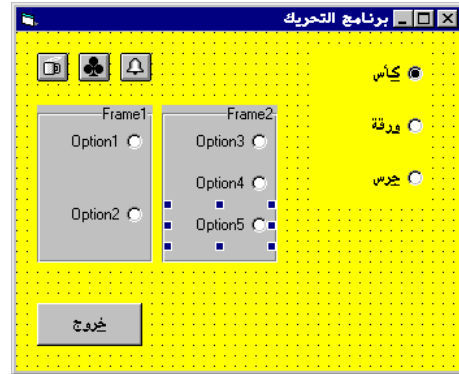
يفترض أن يبدو النموذج كما في الشكل ٩-٤.

الشكل ٩-٤

النموذج frmMove بعد إضافة

مجموعتي أزرار خيارات

إليه ضمن الإطارين.



□ احفظ المشروع باختيار **Save Project** من القائمة **File**.

□ نفذ برنامج التحريك.

النقطة الجديدة بالملاحظة أن النموذج frmMove يمتلك الآن ثلاثة مجموعات مستقلة من أزرار الخيارات Option Button. ولا يؤثر اختيار زر من مجموعة أولى على أي من المجموعات المتبقية، فاختيار كأس مثلاً لا يؤثر على المجموعتين الأخريتين. كذلك اختيار Option1 و Option3 لا يؤثر على المجموعة الأولى. تستطيع التحقق من ذلك وفق ما يلي:

□ انتق Option1 و Option3 وكأس ثم انظر إلى الشكل ١٠-٤.

الشكل ١٠-٤

اختيار أزرار خيارات

من ثلاث مجموعات مستقلة.



□ بدّل الآن الخيارات في كل مجموعة من مجموعات أزرار الخيارات وراقب النتائج ؟

□ أنه البرنامج الآن بنقر الزر خروج.

انتهينا الآن من التمرن على برنامج التحريك. تستطيع إن شئت حذف الإطارين Frame1 و Frame2 وأزرار الخيارات Option1 و Option2 و Option3 و Option4 و Option5. ولحذف أي من عناصر التحكم هذه، اختر العنصر بنقره مرة واحدة ثم ضغط المفتاح Delete على لوحة المفاتيح. وتستطيع إن شئت حذف كل زر من أزرار الخيارات ضمن الإطار ثم حذف الإطار، أو تستطيع حذف الإطار برمته مما سيتسبب بحذف كل عناصر التحكم الموجودة فيه.

### برنامج الرسم

يوضح برنامج الرسم، كيف يمكن لبرنامج ما، استخدام حوادث الفأرة لإنشاء برنامج رسم بسيط.

### التمثيل المرئي لبرنامج الرسم

سنبدأ كالعادة بطور التمثيل المرئي للبرنامج:

□ أنشئ مشروعاً جديداً من نوع Standard EXE.

□ احفظ نموذج المشروع بالاسم Draw.Frm في الدليل C:\VB5Prg\Ch04،

واحفظ ملف المشروع بالاسم Draw.Vbp في نفس الدليل.

□ أنشئ النموذج frmDraw لبرنامج الرسم طبقاً للجدول ٤ - ٢.

يفترض أن يبدو النموذج عند اكتماله كما في الشكل ٤ - ١١.

الشكل ٤ - ١١

النموذج frmDraw

(طور التصميم).



## الجدول ٤-٢. جدول خصائص النموذج frmDraw.

القيمة	الخاصية	الكائن
<b>FrmDraw</b>	<b>Name</b>	<b>Form</b>
برنامج الرسم	Caption	
True	RightToLeft	
<b>cmdExit</b>	<b>Name</b>	<b>CommandButton</b>
&خروج	Caption	
True	RightToLeft	

## إدخال نص برنامج الرسم

□ تحقق من وجود العبارة Option Explicit ضمن قسم التصاريح العامة.

```
Option Explicit
```

□ أدخل النص التالي ضمن الإجراء :Form\_MouseDown()

```
Private Sub Form_MouseDown(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    frmDraw.CurrentX = X
    frmDraw.CurrentY = Y
End Sub
```

□ أدخل النص التالي ضمن الإجراء :Form\_MouseMove()

```
Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then
        Line (frmDraw.CurrentX, frmDraw.CurrentY)- _
            (X, Y), QBColor(0)
    End If
End Sub
```

□ أدخل النص التالي ضمن الإجراء :cmdExit\_Click()

```
Private Sub cmdExit_Click()
    End
End Sub
```

□ احفظ العمل المنجز باختيار **Save Project** من القائمة **File**.

## تنفيذ برنامج الرسم

□ نفذ برنامج الرسم.

يظهر الإطار الفارغ المبين في الشكل ١٢-٤.

□ اضغط زر الفأرة وحركها أثناء ذلك. لترسم خطوط تبعاً لحركة الفأرة.

بسبب تحرير زر الفأرة بإيقاف الرسم. باختصار يمكنك الرسم بنفس الطريقة المتبعة في برنامج الرسام Paint الذي يأتي مع ويندوز. يبين الشكل ١٣-٤ شكلاً يمكن رسمه ببرنامجنا هذا.

الشكل ١٢-٤

إظهار برنامج الرسم حال  
تشغيل البرنامج.



الشكل ١٣-٤

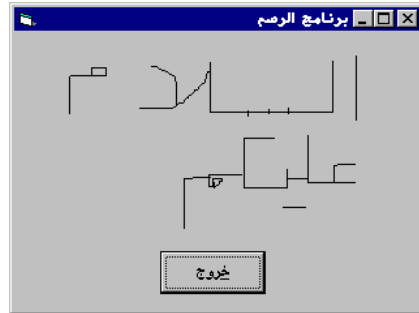
الرسم بواسطة برنامج الرسم.



كما يوضح الشكل ١٤-٤ كيف تستطيع الكتابة بواسطة برنامج الرسم.

الشكل ١٤-٤

الكتابة بواسطة برنامج الرسم.



□ أنه البرنامج بعد انتهائك من الرسم بنقر الزر خروج.

## كيف يعمل برنامج الرسم

يستخدم برنامج الرسم مضمونين متعلقين بالرسم في فيجول بيسك: طريقة Line والخاصيتين CurrentX و CurrentY. ولفهم برنامج الرسم يلزمنا أولاً فهم هذين المضمونين.

## الطريقة Line

تستخدم طريقة Line لرسم خط مستقيم، من نقطة إلى نقطة، ضمن نموذج. فمثلاً، تستخدم العبارة التالية:

```
Line (2000, 1500) - (5000, 6000)
```

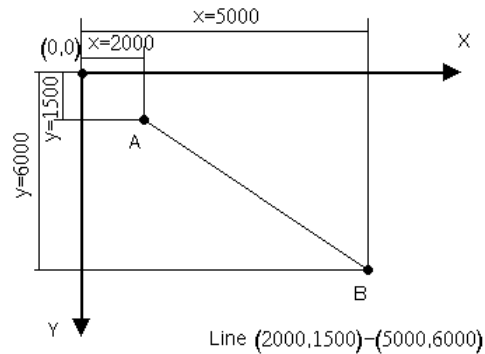
لرسم خط من  $x=2000$  ،  $y=1500$  إلى  $x=5000$  ،  $y=6000$ .

يبين الشكل ١٥-٤ الخط المرسوم نتيجة تنفيذ هذه العبارة. يبدأ الخط على بعد 2000 twips من الجانب الأيسر للنموذج و 1500 twips من قمة النموذج (أي من النقطة A في الشكل ١٥-٤)، وينتهي على بعد 5000 twips من الجانب الأيسر للنموذج 6000 twips من قمة النموذج (النقطة B في الشكل ١٥-٤).

الشكل ١٤-٥

رسم خط بواسطة

الطريقة Line.



يمكن استخدام التابع الوظيفي QBColor() لتحديد اللون الذي سيرسم الخط به. يبين الجدول ٣-٤ الأرقام التي يستخدمها QBColor().

الجدول ٣-٤. الأرقام المستخدمة في التابع الوظيفي QBColor().

Color	Number
-------	--------

Black	0
Blue	1
Green	2
Cyan	3
Red	4
Magenta	5
Yellow	6
White	7
Gray	8
Light Blue	9
Light Green	10
Light cyan	11
Light Red	12
Light Magenta	13
Light Yellow	14
Bright White	15

فمثلاً لرسم نفس الخط المبين بالشكل ٤-١٥ بلون أسود، استخدم العبارة:

```
Line (2000, 1500) - (5000, 6000),QBColor(0)
```

ولرسم نفس الخط بلون أحمر، استخدم العبارة التالية:

```
Line (2000, 1500) - (5000, 6000),QBColor(4)
```

### الخاصيتين CurrentX و CurrentY للنموذج

لن تجد لدى تفحصك خصائص النموذج، الخاصتين CurrentX و CurrentY لأن فيجول بيسك لا يمكنك من إسناد قيمة إليهما في مرحلة التصميم. فقيمة هاتين الخاصتين يمكن تغييرها فقط أثناء زمن التنفيذ (أي ضمن نص البرنامج).

تُحدَّث قيمة الخاصتين CurrentX و CurrentY آلياً من قبل البرنامج بعد استخدام شتى الطرق الرسومية. فمثلاً، يسند البرنامج بعد رسم خط بطريقة Line إحداثيات نقطة نهاية الخط إلى الخاصتين CurrentX و CurrentY. وهكذا فبعد تنفيذ العبارة التالية:

```
Line (2000, 1500) - (5000, 6000),QBColor(0)
```

تصبح قيمة CurrentX تساوي ٥٠٠٠ وقيمة CurrentY تساوي ٦٠٠٠.

**نص الإجراء Form\_MouseDown()**

ينفذ الإجراء Form\_MouseDown() آلياً، عند ضغط زر الفأرة ضمن منطقة النموذج:

```
Private Sub Form_MouseDown(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    frmDraw.CurrentX = X
    frmDraw.CurrentY = Y
End Sub
```

يسند هذا الإجراء الوسيطين X و Y إلى الخاصيتين CurrentX و CurrentY للنموذج frmDraw على التوالي. وهكذا فبعد ضغط زر الفأرة، يشير الإحداثيان CurrentX و CurrentY إلى المكان الذي ضُغَط فيه زر الفأرة.

**نص الإجراء Form\_MouseMove()**

ينفذ هذا الإجراء آلياً عند تحريك الفأرة ضمن منطقة النموذج. يحمل الوسيطان X و Y لهذا الإجراء نفس معنى X و Y في الإجراء Form\_MouseMove() (أي بمعنى أن X و Y يمثلان إحداثيي مؤشر الفأرة).

```
Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then
        Line (frmDraw.CurrentX, frmDraw.CurrentY)- _
            (X, Y), QBColor(0)
    End If
End Sub
```

يختبر الإجراء قيمة Button للتحقق من أنه يجري الآن الضغط على الزر الأيسر للفأرة. فإذا كانت قيمة Button تساوي "١" فهذا معناه أنه يتم الضغط على الزر الأيسر للفأرة، وبالتالي تنفذ العبارة التالية:

```
Line (frmDraw.CurrentX, frmDraw.CurrentY)-(X,Y), QBColor(0)
```

ترسم هذه العبارة خطاً بلون أسود بدءاً من الموقع المحدد بالخاصيتين CurrentX و CurrentY إلى الموقع الراهن للفأرة.

تذكر دائماً أن البرنامج يُحدَّث قيمة الخاصيتين CurrentX و CurrentY بعد تنفيذ طريقة Line آلياً بإحداثيات نقطة نهاية الخط. مما يعني أن التنفيذ التالي للإجراء Form\_MouseMove() سيتسبب برسم خط جديد بدءاً من نقطة نهاية الخط السابق. ينفذ البرنامج الإجراء Form\_MouseMove() عند تحريك الفأرة، ويرسم خطاً جديداً بدءاً من نهاية الخط السابق وحتى الموقع الحالي للفأرة في حال ضغط المستخدم على الزر الأيسر للفأرة.

لاحظ أن هذا الإجراء ينفذ مجدداً ومجدداً كلما حركت الفأرة. وإذا تم الضغط أثناء التحريك على زر الفأرة الأيسر فسيُرسَم خط مع كل تنفيذ للإجراء Form\_MouseMove(). والخط يرسم بدءاً من نقطة نهاية الخط السابق وحتى الموقع الحالي لمؤشر الفأرة.

#### نص الإجراء cmdExit\_Click()

```
Private Sub cmdExit_Click()  
    End  
End Sub
```

### الخاصية AutoRedraw

هناك خلل في برنامج الرسم. اتبع الخطوات التالية للتعرف على المشكلة:

- ❑ نفذ برنامج الرسم وارسم عدة خطوط بواسطته.
- ❑ صغّر إطار البرنامج. (انقر رمز إشارة الناقص الذي يظهر عند الزاوية العليا من الإطار لتصغير النافذة).
- يستجيب برنامج الرسم بإظهار نفسه كرمز (Icons) على شريط مهام ويندوز.
- ❑ استرجع إطار برنامج الرسم. (انقر رمز البرنامج الموجود حالياً على شريط مهام ويندوز Windows).

كما تشاهد، اختفت الخطوط التي رسمتها.

تظهر نفس المشكلة عند تغطية جزء من إطار برنامج الرسم بإطار آخر. تحل هذه المشكلة ببساطة، بإسناد القيمة True إلى الخاصية AutoRedraw للنموذج frmRedraw في طور التصميم.

□ أسند القيمة True إلى الخاصية AutoRedraw للنموذج frmRedraw.

□ احفظ المشروع باختيار **Save Project** من القائمة **File** لفيجول بيسك.

يعيد فيجول بيسك رسم الإطار عندما تدعو الحاجة، عندما تكون قيمة الخاصية AutoRedraw تساوي True (حسب ما يتوضح من اسم الخاصية نفسها). لنشاهد ذلك على أرض الواقع.

□ نفذ برنامج الرسم وارسم بضعة خطوط.

□ صغر إطار البرنامج. (بنقر رمز التصغير الموجود عند الزاوية العليا اليمنى من الإطار).

يستجيب البرنامج بإظهار نفسه كرمز على شريط مهام ويندوز.

□ استرجع إطار برنامج الرسم. (انقر رمز البرنامج الموجود الآن على شريط مهام ويندوز).

كما تشاهد، تظهر الخطوط التي كانت مرسومة.

إذاً تعلمنا أن إسناد قيمة True إلى الخاصية AutoRedraw يمكن البرنامج من إعادة رسم محتويات الإطار عندما تدعو الحاجة إلى ذلك.

حاول ألا تستخدم هذه الخاصية، رغم أنها قد تبدو ملائمة للاستخدام ضمن البرنامج. قد تستغرب!! فهذه الخاصية تتسبب بانخفاض أداء البرنامج. سنتعلم طرائق أخرى من هذا الكتاب أكثر ملائمة لإعادة إنعاش الإطار.

### برنامج الفأرة الراسمة

ينبغي أن لا يُشغل البرنامج المكتوب بلغة فيجول بيسك نفسه على الدوام بتنفيذ الإجراء Form\_MouseMove() عند تحريك الفأرة، لأن ذلك يمنع من تنفيذ أي برنامج آخر أثناء تحريك الفأرة. يتفحص برنامجنا هذا حالة الفأرة خلال فترات زمنية ثابتة

فقط مما يسمح له بتنفيذ المهام الأخرى عند تحريك الفأرة. فإذا اكتشف البرنامج أن الفأرة حركت منذ آخر اختبار، فسوف ينفذ الإجراء `Form_MouseMove()`.

### التمثيل المرئي لبرنامج الفأرة الراسمة

سنبدأ كعادتنا بطور التمثيل المرئي لنموذج البرنامج:

- أنشئ مشروعاً جديداً من نوع Standard EXE.
  - احفظ نموذج المشروع بالاسم `HowOften.Frm` في الدليل `C:\VB5Prg\Ch04`، واحفظ ملف المشروع بالاسم `HowOften.Vbp` في نفس الدليل.
  - أنشئ النموذج `frmHowOften` طبقاً للجدول ٤-٤.
- يفترض أن يبدو النموذج عند اكتماله كما في الشكل ٤-١٦.

الشكل ٤-١٦

النموذج `frmHowOften`  
(في طور التصميم).



## الجدول ٤-٤. جدول خصائص النموذج frmHowOften.

القيمة	الخاصية	الكائن
frmHowOften	Name	Form
برنامج الفأرة الراسمة	Caption	
True	RightToLeft	
cmdExit	Name	CommandButton
&خروج	Caption	
True	RightToLeft	

## إدخال نص برنامج الفأرة الراسمة

□ كالعادة، تحقق من وجود العبارة Option Explicit ضمن قسم التصاريح العامة General Declaration للنموذج frmHowOften. (أي اكتبها إذا لم تكن موجودة).

```
' يجب التصريح عن كل المتحولات
Option Explicit
```

□ أدخل النص التالي ضمن الإجراء :Form\_MouseMove()

```
Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    Circle (X, Y), 40
End Sub
```

□ أدخل النص التالي ضمن الإجراء :cmdExit\_Click()

```
Private Sub cmdExit_Click()
    End
End Sub
```

## تنفيذ برنامج الفأرة الراسمة

لنشاهد أثر ما كتبناه:

□ تنفيذ برنامج الفأرة الراسمة.

□ ضع مؤشر الفأرة في منطقة النموذج، ومن ثم حرك الفأرة.

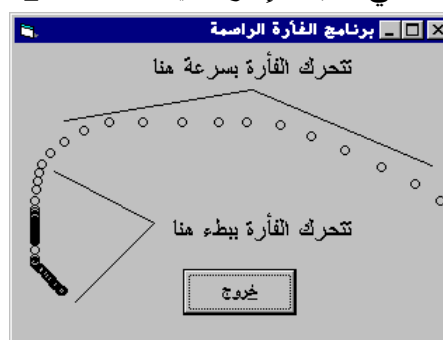
يرسم الإجراء Form\_MouseMove() مع تحريك الفأرة دوائر صغيرة عند الموقع الحالي للفأرة، وعموماً ستلاحظ أن الإجراء Form\_MouseMove() لن ينفذ مع كل حركة للفأرة. (انظر الشكل ١٧-٤).

يرسم الإجراء Form\_MouseMove() قطعاً من الدوائر لدى تحريك الفأرة، وتزداد كثافة الدوائر المرسومة عند تحريك الفأرة ببطء. ويقل عدد الدوائر المرسومة عند تحريك الفأرة بسرعة.

تذكر أن كل دائرة مرسومة هي دلالة على وقوع حادثة تحريك فأرة MouseMove. وبالتالي تنفيذ للإجراء Form\_MouseMove().

الشكل ١٧-٤

تحريك الفأرة في منطقة نموذج  
برنامج الفأرة الراسمة.



### كيف يعمل برنامج الفأرة الراسمة

يستخدم برنامج الفأرة الراسمة الإجراء Form\_MouseMove() لإنجاز العمل.

### نص الإجراء Form\_MouseMove()

ينفذ الإجراء Form\_MouseMove() عندما يكتشف البرنامج أنه تم تحريك الفأرة بعد آخر عملية تحقق:

```
Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    Circle (X, Y), 40
End Sub
```

يرسم هذا الإجراء عند الموقع الحالي لمؤشر الفأرة باستخدام العبارة:

```
Circle (x, y) , 40
```

إلى رسم دائرة بقطر يساوي ٤٠ وحدة قياس Unit (أي حسب وحدة القياس المستخدمة) وباعتبار أن الخاصية ScaleMode للنموذج تشير إلى Twip، فهذا يعني أن الطريقة Circle ستستخدم وحدة القياس Twip (القطر يقاس بالـ twips ويقاس مركز الإحداثيات بالـ twips).

### نص الإجراء cmdExit\_Click()

ينفذ هذا الإجراء عند نقر الزر خروج وينهي بدوره تنفيذ البرنامج:

```
Private Sub cmdExit_Click()  
    End  
End Sub
```

### برنامج أزرار الفأرة

سنكتب الآن برنامجاً يدعى برنامج أزرار الفأرة، يستخدم الوسيط Button لكل من الإجراءات Form\_MouseDown() و Form\_MoveUp() لتحديد زر الفأرة الذي تم ضغطه أو تحريره.

### التمثيل المرئي لبرنامج أزرار الفأرة

سنبنى الآن نموذج برنامج أزرار الفأرة:

❑ أنشئ مشروعاً جديداً من نوع Standard EXE.

❑ احفظ نموذج المشروع بالاسم Button.Frm في الدليل C:\VB5Prg\Ch04

واحفظ ملف المشروع بالاسم Button.Vbp في نفس الدليل.

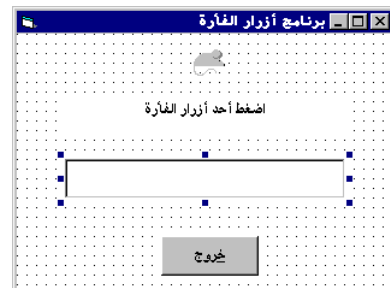
❑ أنشئ النموذج frmButton طبقاً للجدول ٤-٥.

يفترض أن يبدو النموذج عند اكتماله كما في الشكل ٤-١٨.

الشكل ٤-١٨

النموذج frmButton

(في طور التصميم).



الجدول ٤-٥. جدول خصائص النموذج frmButton.

القيمة	الخاصية	الكائن
<b>frmButton</b>	<b>Name</b>	<b>Form</b>
برنامج أزرار الفأرة	Caption	
White	BackColor	
True	RightToLeft	
<b>cmdExit</b>	<b>Name</b>	<b>CommandButton</b>
&خروج	Caption	
True	RightToLeft	
<b>txtResult</b>	<b>Name</b>	<b>Text Box</b>
(اجعله فارغاً)	Text	
2-Center	Alignment	
True	MultiLine	
False	Enabled	
True	RightToLeft	
<b>lblInstruction</b>	<b>Name</b>	<b>Label</b>
اضغط أحد أزرار الفأرة	Caption	
2-Center	Alignment	
White	BackColor	
True	RightToLeft	
<b>imgMouse</b>	<b>Name</b>	<b>Image</b>
Mouse04.Ico	Picture	
False	Stretch	

يطالبك الجدول ٤-٥ بإسناد اسم الملف Mouse04.Ico إلى الخاصية Picture لعنصر التحكم imgMouse. يقع هذا الملف في حال توفره ضمن الدليل الفرعي \Icons\Computer\ تحت الدليل الذي تم تنصيب لغة فيجول بيسك فيه. يمكنك طبعاً إسناد أي ملف آخر من نوع Ico إلى الخاصية Picture. ولا سيما إذا لم تجد الملف المذكور. فالغرض الوحيد من استخدام هذا الملف في برنامجنا أنه يتلاءم مع سياق العمل لأنه يمثل شكل الفأرة.

## إدخال نص برنامج أزرار الفأرة

□ كالعادة، تحقق (أو أدخل) من وجود العبارة التالية: Option Explicit ضمن قسم

التصاريح العامة General Declaration للنموذج :frmButton

```
يجب التصريح عن كل المتحولات'
Option Explicit
```

□ أدخل النص التالي ضمن الإجراء :Form\_MouseDown()

```
Private Sub Form_MouseDown(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then
        txtResult.Text = "زر الفأرة اليساري مضغوط الآن"
    End If
    If Button = 2 Then
        txtResult.Text = "زر الفأرة اليميني مضغوط الآن"
    End If
    If Button = 4 Then
        txtResult.Text = "زر الفأرة الوسطي مضغوط الآن"
    End If
End Sub
```

□ أدخل النص التالي ضمن الإجراء :Form\_MouseUp()

```
Private Sub Form_MouseUp(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    txtResult = ""
End Sub
```

□ أدخل النص التالي ضمن الإجراء :cmdExit\_Click()

```
Private Sub cmdExit_Click()
    End
End Sub
```

□ احفظ المشروع باختيار Save Project من القائمة File.

## تنفيذ برنامج أزرار الفأرة

□ نفذ برنامج أزرار الفأرة.

يُظهر الإطار المبين في الشكل ١٩-٤ عند بدء تشغيل برنامج أزرار الفأرة.

الشكل ١٩-٤

إطار برنامج أزرار الفأرة.



يُظهر مربع النص اسم الزر المضغوط عند ضغط أحد أزرار الفأرة في منطقة خالية ضمن النموذج. يوضح الشكل ٢٠-٤ محتوى مربع النص نتيجة ضغط الزر الأيمن للفأرة. ويبين الشكل ٢١-٤ محتوى مربع النص عند ضغط الزر الأيسر للفأرة. يُظهر برنامج أزرار الفأرة حالة زر الفأرة فقط عند الضغط على أحد أزرار الفأرة في منطقة خالية من النموذج.

الشكل ٢٠-٤

برنامج أزرار الفأرة عند

الضغط على الزر الأيمن للفأرة.



الشكل ٢١-٤

برنامج أزرار الفأرة عند

الضغط

على الزر الأيسر للفأرة.



### ملاحظة

تدعى المنطقة غير المغطاة بعنصر تحكم فعال Enabled ضمن النموذج، بمنطقة خالية. فمثلاً الخاصية Enabled للالفة IblInstruction تساوي True، ولهذا لا تعتبر

المنطقة التي تشغلها هذه اللافتة منطقة خالية.  
ومن جهة أخرى، تساوي الخاصية Enabled لمربع النص txtResult إلى False وبالتالي فالمنطقة التي يشغلها مربع النص هذا عبارة عن منطقة خالية ضمن النموذج.  
لا يؤدي النقر على اللافتة lblInstruction إلى تنفيذ الإجراء Form\_MouseDown()، لأن منطقة هذه اللافتة كما اتفقنا لا تمثل منطقة خالية.  
وعلى العكس تعتبر منطقة مربع النص txtResult منطقة خالية ويؤدي النقر عليها إلى تنفيذ الإجراء Form\_MouseDown().

لا يستجيب برنامج أزرار الفأرة عند النقر على الزر الأوسط للفأرة ما لم يتم تعريف الفأرة ضمن ويندوز بأنها ذات ثلاثة أزرار. (طبعاً في حال الفأرة ذات ثلاثة أزرار).

### كيف يعمل برنامج أزرار الفأرة

يستجيب برنامج أزرار الفأرة لحادثةMouseDown (الضغط على زر فأرة) بتنفيذ الإجراء Form\_MouseDown() ولحادثة تحرير الفأرة MouseUp بتنفيذ الإجراء Form\_MouseDown().

### نص الإجراء Form\_MouseDown()

ينفذ هذا الإجراء تلقائياً بضغط أحد أزرار الفأرة في منطقة خالية ضمن النموذج.  
مهمة الإجراء تحديد الزر الذي تم ضغطه بفحص قيمة الوسيط Button.  
عندما تكون قيمة الوسيط Button تساوي ١ فهذا معناه أنه تم الضغط على الزر الأيسر، وتعني القيمة ٢ الضغط على الزر الأيمن، والقيمة ٤ تشير إلى الضغط على الزر الأوسط:

```
Sub Form_MouseDown(Button As Integer, Shift As Integer, _
    X As Single, Y As Single)
    If Button = 1 Then
        txtResult.Text = "زر الفأرة اليساري مضغوط الآن"
    End If
    If Button = 2 Then
```

```

    txtResult.Text = "زر الفأرة اليميني مضغوط الآن"
End If
If Button = 4 Then
    txtResult.Text = "زر الفأرة الوسطي مضغوط الآن"
End If
End Sub

```

### نص الإجراء Form\_MouseUp()

ينفذ هذا الإجراء تلقائياً عند تحرير زر الفأرة. يمحو نص هذا الإجراء محتويات مربع النص:

```

Private Sub Form_MouseUp(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    txtResult = ""
End Sub

```

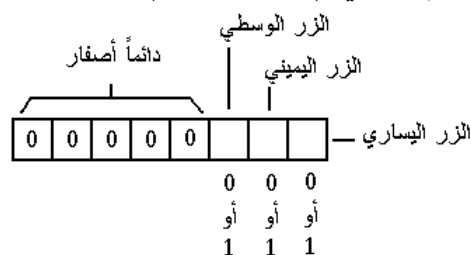
### الوسيط Button في الإجراء Form\_MouseMove()

يحدد الوسيط Button في إجراءات الفأرة الزر المضغوط لحظة وقوع الحادثة. وحسب ما ذكرنا فإن قيمة هذا الوسيط ضمن الإجراء Form\_MouseDown() يمكن أن تساوي ١ أو ٢ أو ٤ ولا يمكن أن تتخذ أي قيمة أخرى مما يعني أنك لا تستطيع استخدام الحادثة MouseDown للتحقق من الضغط على أكثر من زر واحد. أما بالنسبة للإجراء Form\_MouseMove() فالأمر يختلف، إذ يمكن للوسيط Button أن يأخذ أي قيمة بين ٠ و ٧، مما يشمل كل الإمكانيات المحتملة لأزرار الفأرة الثلاث. فمثلاً، يساوي Button إلى ٣ عند الضغط على الزرين الأيسر والأيمن والذي يكافئ الرقم الثنائي (٠٠٠٠٠٠١١)، انظر الشكل ٤-٢٢.

الشكل ٤-٢٢

الوسيط Button للإجراء

Form\_MouseMove()



انظر الجدول التالي إذا لم تكن ملماً بالنظام الثنائي:

التدوين الثنائي Binary Notation	التدوين العشري Decimal Notation
.....	٠
.....١	١
.....١٠	٢
.....١١	٣
.....١٠٠	٤
.....١٠١	٥
.....١١٠	٦
.....١١١	٧

فمثلاً عندما تكون قيمة Button تساوي صفر، فالقيمة الثنائية تساوي (.....) وحسب الشكل ٤-٢٢، الضغط على كل أزرار الفأرة سويًا تكون القيمة الثنائية المرافقة هي (.....١١١)، وعندما تكون القيمة مساوية إلى ٤ فهذا معناه الضغط على الزر الأوسط والقيمة الثنائية المرافقة (.....١٠٠).

## برنامج أزرار الفأرة ٢

يوضح برنامج أزرار الفأرة ٢ كيفية استخدام الوسيط Button، في الإجراء

.Form\_MouseMove()

## التمثيل المرئي لبرنامج أزرار الفأرة ٢

سنبدأ بطور التمثيل المرئي لبرنامج أزرار الفأرة ٢:

□ أنشئ مشروعاً جديداً من نوع Standard EXE.

□ احفظ نموذج المشروع بالاسم Button2.Frm في الدليل C:\VB5Prg\Ch04،

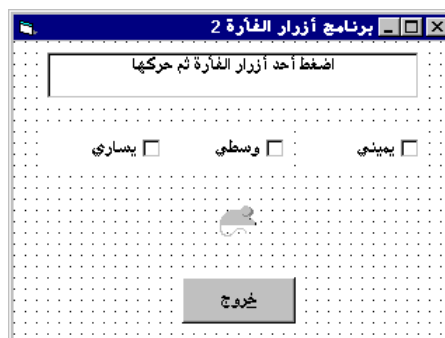
واحفظ ملف المشروع بالاسم Button.Vbp في نفس الدليل.

□ أنشئ نموذج البرنامج frmButton2 طبقاً للجدول ٤-٦.

الشكل ٤- ٢٣

النموذج frmButton2

في مرحلة التصميم.



الجدول ٤-٦. جدول خصائص النموذج frmButton2.

الكائن	الخاصية	القيمة
<b>Form</b>	<b>Name</b>	<b>frmButton2</b>
	Caption	برنامج أزرار الفأرة ٢
	BackColor	White
	RightToLeft	True
<b>CommandButton</b>	<b>Name</b>	<b>cmdExit</b>
	Caption	&خروج
	RightToLeft	True
<b>Check Box</b>	<b>Name</b>	<b>chkLeft</b>
	BackColor	White
	Caption	يساري
	Enabled	False
<b>Check Box</b>	<b>Name</b>	<b>chkMiddle</b>
	BackColor	White
	Caption	وسطي
	Enabled	False
<b>Check Box</b>	<b>Name</b>	<b>chkRight</b>

الكائن	الخاصية	القيمة
	BackColor	White
	Caption	يميني
	Enabled	False

Image	Name	imgMouse
	Picture	Mouse04.Ico
	Stretch	False
Label	Name	lblInstruction
	Caption	اضغط أحد أزرار الفأرة ثم حركها
	Alignment	2-Center
	BackColor	White
	BorderStyle	1-Fixed Single
	RightToLeft	True

■ يطالبك الجدول ٦-٤ بإسناد اسم الملف Mouse04.Ico إلى الخاصية Picture لعنصر التحكم Image. فإذا لم يكن هذا الملف موجوداً أو لم يكن الدليل \Icons\Computer\ موجوداً لسبب ما، ضمن دليل لغة فيجول بيسك، تستطيع استخدام أي ملف آخر من نوع Ico.

■ يطالبك الجدول بإسناد القيمة False إلى الخاصية Stretch لعنصر التحكم Image. هذا يعني أن عنصر التحكم Image لن يوسّع الصورة. فمثلاً إذا كانت الخاصية Picture لعنصر التحكم Image تشير إلى صورة بحجم 32x32 Pixel ، وكبرت بعدها حجم العنصر Image، فإن الصورة لن توسع ولن تغطي حجم عنصر التحكم Image. بل ستظهر الصورة بحجمها الأصلي فقط.

■ يسند الجدول القيمة False إلى الخاصية Enabled لكل مربع من مربعات الاختيار Check Boxes. مما يعني أن المستخدم لن يتمكن من اختيار هذه المربعات أو إلغاء اختيارها (أي حجبها) بالنقر بالفأرة عليها. وبدلاً من ذلك فإن نص البرنامج هو الذي سيختار أو يلغي اختيار مربعات التحكم بناءً على حالة أزرار الفأرة (وهو السبب وراء منع المستخدم من اختيار أو إلغاء هذه المربعات).

## إدخال نص برنامج أزرار الفأرة ٢

□ تحقق من وجود العبارة التالية ضمن قسم التصاريح العامة General

## :Declaration

يجب التصريح عن كل المتحولات  
Option Explicit

□ أدخل النص التالي ضمن الإجراء :Form\_MouseMove()

```
Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If (Button And 1) = 1 Then
        chkLeft.Value = 1
    Else
        chkLeft.Value = 0
    End If

    If (Button And 2) = 2 Then
        chkRight.Value = 1
    Else
        chkRight.Value = 0
    End If

    If (Button And 4) = 4 Then
        chkMiddle.Value = 1
    Else
        chkMiddle.Value = 0
    End If
End Sub
```

□ أدخل النص التالي ضمن الإجراء :Form\_MouseUp()

```
Private Sub Form_MouseUp(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then
        chkLeft.Value = 0
    End If

    If Button = 2 Then
        chkRight.Value = 0
    End If
```

```

If Button = 4 Then
    chkMiddle.Value = 0
End If
End Sub

```

□ أدخل النص التالي ضمن الإجراء cmdExit\_Click():

```

Private Sub cmdExit_Click()
    End
End Sub

```

□ احفظ المشروع باختيار **Save Project** من القائمة **File** لفيجول بيسك.

## تنفيذ برنامج أزرار الفأرة ٢

□ نفذ برنامج أزرار الفأرة ٢.

يبين الشكل ٢٤-٤ كيف يبدو إطار هذا البرنامج، لاحظ وجود ثلاثة مربعات اختيار:

يميني و وسطي و يساري.

الشكل ٢٤-٤

إطار برنامج أزرار الفأرة ٢.



يختار برنامج أزرار الفأرة ٢ ويلغي مربعات الاختيار تبعاً لحالة أزرار الفأرة " أثناء تحريك الفأرة ". فمثلاً يختار البرنامج مربعي الاختيار يميني و يساري إذا ضغط المستخدم على زري الفأرة اليساري واليميني وحرك الفأرة، (انظر الشكل ٢٥-٤).

## كيف يعمل برنامج أزرار الفأرة ٢

يستخدم برنامج أزرار الفأرة ٢ الوسيط Button2 للإجراء Form\_MouseMove() لتحديد الزر أو مجموعة أزرار الفأرة التي ضغط عليها.

الشكل ٤-٢٥

برنامج أزرار الفأرة ٢ عند

الضغط على زر الفأرة

اليساري واليميني (أثناء تحريكها).



### نص الإجراء Form\_MouseMove()

ينفذ الإجراء تلقائياً عند تحريك الفأرة ضمن منطقة نموذج البرنامج:

```
Private Sub Form_MouseMove(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If (Button And 1) = 1 Then
        chkLeft.Value = 1
    Else
        chkLeft.Value = 0
    End If

    If (Button And 2) = 2 Then
        chkRight.Value = 1
    Else
        chkRight.Value = 0
    End If

    If (Button And 4) = 4 Then
        chkMiddle.Value = 1
    Else
        chkMiddle.Value = 0
    End If
End Sub
```

يختار نص هذا الإجراء أو يلغي مربعات الاختيار تبعاً لقيمة الوسيط Button. تحدد أول عبارة If في الإجراء إذا كان زر الفأرة الأيسر مضغوطاً وذلك بإجراء ضرب منطقي AND للوسيط Button مع القيمة ١.

فإذا كان ناتج حاصل الضرب يساوي ١ فهذا يعني أنه تم الضغط على الزر الأيسر، وبشكل مشابه إذا كان ناتج حاصل الضرب Button مع ٢ منطقياً AND يساوي ٢، فهذا معناه أنه تم الضغط على الزر الأيمن، وإذا كان ناتج حاصل الضرب المنطقي AND لـ Button مع ٤ يساوي ٤ فهذا دلالة على أن الزر الأوسط مضغوط حالياً، لا تكثر لحاصل الضرب المنطقي AND إذا لم تكن على اطلاع به أو لست على علم بالنظام الثنائي، المهم أن تعرف ما يحدث عند الضغط على أزرار الفأرة أثناء تحريكها.

### نص الإجراء Form\_MouseUp()

ينفذ هذا الإجراء عند تحرير أي من أزرار الفأرة:

```
Private Sub Form_MouseUp(Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then
        chkLeft.Value = 0
    End If
    If Button = 2 Then
        chkRight.Value = 0
    End If
    If Button = 4 Then
        chkMiddle.Value = 0
    End If
End Sub
```

يتحقق هذا الإجراء من الزر الذي حرر، ويلغي مربع الاختيار المرافق. فمثلاً تتحقق عبارة الشرط If الأولى عند تحرير الزر الأيسر للفأرة ويلغى اختيار مربع الاختيار يساري.

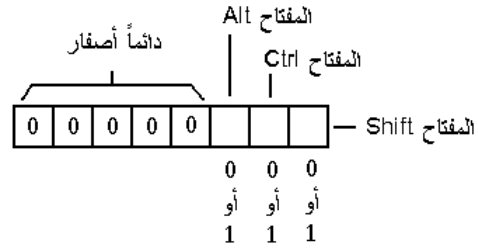
### ضغط المفاتيح Shift و Ctrl و Alt إلى جانب ضغط أزرار الفأرة

تمتلك الحوادث MouseDown و MouseUp و MouseMove القيمة الصحيحة Shift كثنائي وسيط ضمن الإجراءات المرافقة لها، ويشير هذا الوسيط إلى المفتاح Shift و Ctrl و Alt المضغوط إلى جانب أزرار الفأرة.

يبين الشكل ٢٦-٤ كيف يشير الوسيط Shift إلى المفتاح المضغوط.

الشكل ٢٦-٤

الوسيط Shift.



حسب ما يوضحه الشكل ٢٦-٤. تمثل البتات الأدنى من الوسيط Shift حالة المفاتيح Shift و Ctrl و Alt أثناء الضغط على زر (أو أزرار) الفأرة.

يوضح الجدول ٧-٤ القيم الثمان المحتملة التي يمكن أن يمتلكها الوسيط Shift.

الجدول ٧-٤ القيم المحتملة للوسيط Shift.

القيمة الثنائية	القيمة العشرية	المفتاح Shift	المفتاح Ctrl	المفتاح Alt
00000000	0	لا	لا	لا
00000001	1	نعم	لا	لا
00000010	2	لا	نعم	لا
00000011	3	نعم	نعم	لا
00000100	4	لا	لا	نعم
00000101	5	نعم	لا	نعم
00000110	6	لا	نعم	نعم
00000111	7	نعم	نعم	نعم

فمثلاً، إذا كانت قيمة الوسيط Shift تساوي ٦ في أحد الإجراءات Form\_MouseDown أو Form\_MouseUp و Form\_MouseMove فهذا يعني ضغط المفاتيح Ctrl و Alt إلى جانب ضغط أحد أزرار الفأرة.

لا تتس أن الوسيط Button يمكن أن يأخذ القيم ١ و ٢ و ٤ فقط في الإجراء Form\_MouseDown(). بعكس الوسيط Shift لنفس الإجراء والذي يمكن أن يأخذ أي قيمة ضمن الجدول ٤-٧.

### برنامج السحب

تعلمنا حتى الآن كيفية استخدام حوادث الفأرة التي تقع عند ضغط زر الفأرة MouseDown، أو عند تحرير زر فأرة MouseUp. أو عند تحريك الفأرة MouseMove. سنتعلم الآن كيف تستخدم حوادث الفأرة لبناء آلية تسمح للمستخدم بسحب وإسقاط (إفلات) عناصر التحكم ضمن إطار البرنامج. تعرف عملية السحب بأنها ضغط على الزر الأيسر للفأرة عندما يكون مؤشرها فوق عنصر تحكم، ثم تحريك الفأرة مع إبقاء الضغط على الزر. أما تحرير زر الفأرة بعد سحب العنصر فيدعى بالإفلات. سنكتب الآن برنامج السحب، يوضح هذا البرنامج مدى سهولة إنجاز آلية السحب في البرنامج.

### التمثيل المرئي لبرنامج السحب

سنشرع الآن بطور التمثيل المرئي لبرنامج السحب:

□ أنشئ مشروعاً جديداً من نوع Standard EXE.

□ احفظ نموذج المشروع بالاسم Drag.Frm في الدليل C:\VB5Prg\Ch04 واحفظ

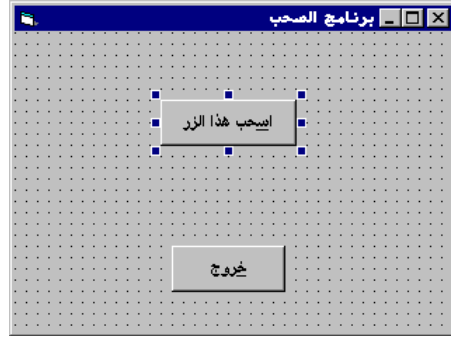
ملف المشروع بالاسم Drag.Vbp في نفس الدليل.

□ أنشئ نموذج البرنامج طبقاً للجدول ٤-٨.

## الشكل ٤-٢٧

النموذج frmDrag

(طور التصميم).



الجدول ٤-٨. جدول خصائص برنامج السحب.

القيمة	الخاصية	الكائن
frmDrag	Name	Form
برنامج السحب	Caption	
True	RightToLeft	
cmdExit	Name	CommandButton
&خروج	Caption	
True	RightToLeft	
cmdDragMe	Name	CommandButton
&اسحب هذا الزر	Caption	
1-Automatic	DragMode	
True	RightToLeft	

## نص برنامج السحب

□ تحقق من وجود العبارة Option Explicit ضمن قسم التصاريح العامة General

Declaration للنموذج frmDrag:

```
يجب التصريح عن كل المتحولات '
Option Explicit
```

□ أدخل النص التالي ضمن الإجراء cmdExit\_Click():

```
Private Sub cmdExit_Click()
    End
End Sub
```

□ احفظ المشروع باختيار Save Project من القائمة File لفيجول بيسك.

### تنفيذ برنامج السحب

تسند القيمة 1-Automatic إلى الخاصية DragMode للزر اسحب هذا الزر. يسمح هذا للمستخدم بسحب الزر أثناء زمن التنفيذ.

اتبع الخطوات التالية لرؤية كيفية سحب الزر اسحب هذا الزر:

□ نفذ برنامج السحب.

□ اضغط زر الفأرة الأيسر عندما يكون مؤشر الفأرة فوق الزر اسحب هذا الزر واستمر بالضغط على الزر مع تحريك الفأرة.

كما تشاهد، يظهر مستطيل بحجم الزر اسحب هذا الزر ويتبع حركة الفأرة.

□ جرب سحب الزر اسحب هذا الزر خارج النموذج.

يستجيب البرنامج بإظهار دائرة مع خط مائل داخلها (رمز عملية غير نظامية)، وهي دلالة بأنك تحاول سحب عنصر التحكم إلى منطقة ممنوعة.

يختفي المستطيل عند تحرير زر الفأرة (تدعى هذه العملية بالإفلات). لاحظ أن الزر اسحب هذا الزر يبقى في موقعه الأصلي.

□ انقر على الزر خروج لبرنامج السحب لإنهائه.

### تحسين برنامج السحب

كما لاحظنا، بسبب إسناد 1-Automatic إلى الخاصية DragMode يظهر مستطيل بنفس حجم الزر اسحب هذا الزر، ويتحرك استجابة لحركة الفأرة. يمكنك هذا المستطيل من مشاهدة المكان الذي يجري سحب الزر إليه.

لنولد شكلاً مختلفاً عند سحب عنصر التحكم، باتباع الخطوات التالية:

□ أسند اسم الملف Drag1Pg.Ico إلى الخاصية DragIcon للزر اسحب هذا الزر

(يفترض أن تجد الملف Drag1Pg.Ico في الدليل Icons\DragDrop تحت دليل

لغة فيجول بيسك، ويمكنك استخدام أي ملف آخر من نوع \*.Ico إذا لم تعثر على هذا الملف).

□ احفظ المشروع باختيار Save Project من القائمة File.

اتبع الخطوات التالية لرؤية تأثير تعديل الخاصية DragIcon:

□ نفذ برنامج السحب.

□ اسحب الزر اسحب هذا الزر.

يستجيب البرنامج بإظهار الرمز Drag.Ico عند سحب الزر اسحب هذا الزر (انظر الشكل ٤-٢٨).

والآن ستجد أن الرمز Drag1Pg.Ico سيتحرك تبعاً لحركة الفأرة بدلاً من المستطيل، قبل إجراء التعديل.

الشكل ٤-٢٨

الرمز Drag1Pg.Ico وهو يتحرك  
تبعاً لحركة الفأرة أثناء  
سحب الزر اسحب هذا الزر.



## برنامج الإفلات

يوضح برنامج الإفلات مفهوم الإفلات (ويقصد به تحرير زر الفأرة بعد السحب) وكيف يستخدم في البرنامج.

### التمثيل المرئي لبرنامج الإفلات

□ أنشئ مشروعاً جديداً من نوع Standard EXE.

□ احفظ نموذج المشروع بالاسم Drop.Frm في الدليل C:\VB5Prg\Ch04 واحفظ

ملف المشروع بالاسم Drop.Vbp في نفس الدليل.

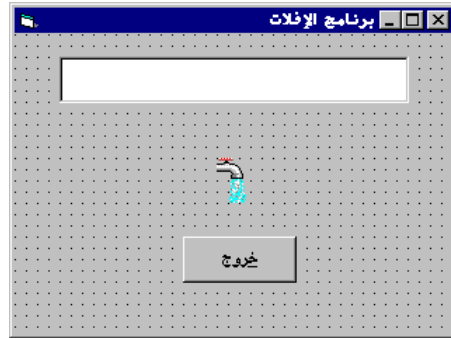
□ أنشئ النموذج طبقاً للجدول ٤-٩.

يفترض أن يبدو الشكل المكتمل كما في الشكل ٤-٢٩.

الشكل ٤ - ٢٩

النموذج frmDrop

في مرحلة التصميم.



الجدول ٤-٩. جدول خصائص برنامج الإفلات.

القيمة	الخاصية	الكائن
<b>frmDrop</b>	<b>Name</b>	<b>Form</b>
برنامج الإسقاط	Caption	
True	RightToLeft	
<b>cmdExit</b>	<b>Name</b>	<b>CommandButton</b>
&خروج	Caption	
True	RightToLeft	
<b>txtInfo</b>	<b>Name</b>	<b>Text Box</b>
(اجعله فارغاً)	Text	
2-Center	Alignment	
True	MultiLine	
False	Enabled	
True	RightToLeft	
<b>imgWater</b>	<b>Name</b>	<b>Image</b>
Water.Ico	Picture	
1-Automatic	DragMode	
1-True	Stretch	
Water image	Tag	

يطالبك الجدول ٤-٩ بإسناد اسم الملف Water.Ico إلى الخاصية Picture لعنصر التحكم imgWater. تستطيع العثور على هذا الملف ضمن الدليل Icons\Elements تحت دليل فيجول بيسك. استخدم أي ملف آخر من نوع Ico إذا لم تجد هذا الملف.

## إدخال نص برنامج الإفلات

□ تحقق من وجود العبارة Option Explicit ضمن قسم التصاريح العامة General

:Declaration

```
' يجب التصريح عن كل المتحولات'
Option Explicit
```

□ أدخل النص التالي ضمن الإجراء :Form\_DragOver()

```
Private Sub Form_DragOver(Source As Control, _
    X As Single, Y As Single, State As Integer)
    Dim sInfo As String
    sInfo = " الآن يتم سحب "
    sInfo = sInfo + Source.Tag
    sInfo = sInfo + " فوق النموذج "
    sInfo = sInfo + Str(State)
    txtInfo.Text = sInfo
End Sub
```

□ أدخل النص التالي ضمن الإجراء :cmdExit\_Click()

```
Private Sub cmdExit_Click()
    End
End Sub
```

□ أدخل النص التالي ضمن الإجراء :Form\_DragDrop()

```
Private Sub Form_DragDrop(Source As Control, _
    X As Single, Y As Single)
    'clear the text box.
    txtInfo.Text = ""
    'move the control
    Source.Move X, Y
End Sub
```

□ أدخل النص التالي ضمن الإجراء :cmdExit\_DragOver()

```
Private Sub cmdExit_DragOver(Source As Control, _
    X As Single, Y As Single, State As Integer)
    Dim sInfo
```

```
sInfo = " الآن يتم سحب "
sInfo = sInfo + Source.Tag
sInfo = sInfo + " فوق الزر خروج "
sInfo = sInfo + Str(State)
txtInfo.Text = sInfo
```

**End Sub**

□ احفظ المشروع باختيار **Save Project** من القائمة **File** لفيجول بيسك.

### تنفيذ برنامج الإفلات

□ نفذ برنامج الإفلات.

□ اسحب صورة صنوبر المياه.

تظهر رسالة ضمن مربع النص أثناء سحب الصورة، تشير إلى حالة عملية السحب. يتسبب تحرير الفأرة بنقل الصورة إلى النقطة التي تم تحرير الفأرة فيها (نقطة الإفلات).

□ أنه البرنامج بالنقر على الزر خروج.

### كيف يعمل برنامج الإفلات

يستخدم برنامج الإفلات الإجراءات `Form_DragOver()` و `cmdExit_DragOver()` و `Form_DragDrop()`.

### نص الإجراء `Form_DragOver()`

ينفذ الإجراء `Form_DragOver()` عند سحب عنصر التحكم `imgWater` فوق النموذج. يمتلك هذا الإجراء أربعة وسائط `Source` و `X` و `Y` و `Start`:

```
Private Sub Form_DragOver(Source As Control, _
    X As Single, Y As Single, State As Integer)
    Dim sInfo As String
    sInfo = " الآن يتم سحب "
    sInfo = sInfo + Source.Tag
    sInfo = sInfo + " فوق النموذج "
```

```
sInfo = sInfo + Str(State)
txtInfo.Text = sInfo
```

**End Sub**

يمثل الوسيط Source اسم عنصر التحكم الذي يجري سحبه. وبما أن عنصر التحكم الذي يتم سحبه يدعى imgWater، وبالتالي يسند imgWater آلياً إلى الوسيط Source. يمثل الوسيطان x و y الإحداثيات الراهنة x,y لمؤشر الفأرة (منسوبة إلى نظام إحداثيات النموذج). يمتلك الوسيط State إحدى القيم ٠ أو ١ أو ٢:

■ تسحب الصورة من نقطة حرة إلى نقطة حرة أخرى ضمن النموذج، عندما تكون قيمة State تساوي ٢.

■ تسحب الصورة من نقطة حرة ضمن النموذج إلى نقطة غير نظامية (مثلاً نقطة خارج النموذج)، عندما تكون قيمة State تساوي ١.

■ تسحب الصورة من نقطة غير نظامية إلى نقطة حرة ضمن النموذج، عندما تساوي قيمة State الصفر.

يجوز الإجراء Form\_DragOver() سلسلة تدعى sInfo ويظهر السلسلة في مربع النص :txtInfo

```
Dim sInfo As String
sInfo = " الآن يتم سحب "
sInfo = sInfo + Source.Tag
sInfo = sInfo + " فوق النموذج "
sInfo = sInfo + Str(State)
txtInfo.Text = sInfo
```

فمثلاً يُسند إلى السلسلة ما يلي عند سحب الصورة من منطقة حرة ضمن النموذج إلى منطقة أخرى:

الآن يتم سحب Water Image فوق النموذج 2

استخدمنا العبارة التالية أثناء تجهيز السلسلة sInfo:

```
sInfo = sInfo + Source.Tag
```

كما أسندنا أثناء طور التصميم، إلى الخاصية Tag لعنصر الصورة imgWater، القيمة Water Image، لذلك فإن قيمة Source.Tag ستساوي Water Image.

## الخاصية Tag

تستخدم الخاصية Tag غالباً كمنطقة تخزين للمعطيات. فمثلاً، تحوي الخاصية Tag للعنصر imgWater على سلسلة مهمتها التعريف بعنصر التحكم هذا. استخدمنا في الإجراء Form\_DragOver() هذه السلسلة لتعريف العنصر المسحوب، طبعاً يمكن إسناد أي شيء آخر ترغب به (مثل MyWater) إلى الخاصية Tag.

### نص الإجراء cmdExit\_DragOver()

ينفذ هذا الإجراء عند سحب عنصر التحكم فوق الزر خروج، ولهذا يتشابه هذا الإجراء مع الإجراء Form\_DragOver() باستثناء أن السلسلة sInfo تفيد بأن عنصر التحكم water يجري سحبه فوق الزر خروج.

```
Private Sub cmdExit_DragOver(Source As Control, _
    X As Single, Y As Single, State As Integer)
    Dim sInfo As String
    sInfo = " الآن يتم سحب "
    sInfo = sInfo + Source.Tag
    sInfo = sInfo + " فوق الزر خروج "
    sInfo = sInfo + Str(State)
    txtInfo.Text = sInfo
End Sub
```

فمثلاً ستحتوي السلسلة الرسالة التالية عند سحب عنصر التحكم Water فوق الزر خروج:

الآن يتم سحب Water Image فوق الزر خروج .

قيمة State = 0 هنا، وذلك بسبب سحب العنصر imgWater من خارج الزر خروج إلى داخله. ومن وجهة نظر الزر خروج فإن أي نقطة خارجه هي نقطة غير نظامية.

### نص الإجراء Form\_DragDrop()

ينفذ هذا الإجراء عند إفلات عنصر التحكم ضمن النموذج. يجب إنجاز شيئين عند حدوث هذا: محو مربع النص، ونقل عنصر التحكم Water إلى نقطة الإفلات:

```
Private Sub Form_DragDrop(Source As Control, _
                        X As Single, Y As Single)
    'clear the text box.
    txtInfo.Text = ""
    'move the control
    Source.Move X, Y
End Sub
```

لهذا استخدمنا العبارتين التاليتين ضمن الإجراء:

```
txtInfo.Text = ""
Source.Move X, Y
```

ينقل عنصر التحكم imgWater إلى نقطة الإفلات باستخدام الطريقة Move. تتحدد نقطة الإفلات بالوسيطين X و Y لهذا الإجراء.

### الخلاصة

تتناول هذا الفصل حوادث الفأرة. فتعلمنا الحوادث MouseDown و MouseMove و DragOver و DragDrop. تقدم وسائط إجراءات هذه الحوادث معلومات عن حالة الفأرة زمن وقوع الحادثة. فالوسيط Button مثلاً يشير إلى زر الفأرة الذي تم النقر عليه والوسيط Shift يخبر أي مفتاح من المفاتيح Shift أو Ctrl أو Alt تم الضغط عليه إلى جانب الفأرة. كما يحدد الوسيطان X و Y موقع مؤشر الفأرة زمن وقوع الحادثة، وهكذا.