

Tổng hợp & Biên soạn
KS. ĐÌNH XUÂN LAM

VN-GUIDE®



LẬP TRÌNH GAME

BẰNG

VISUAL BASIC

Với những trò chơi phổ biến
đầy tính hấp dẫn.

- Dò mìn
- Xếp gạch
- Caro
- Xạ thủ
- Không chiến
- ...vv.

NHÀ XUẤT BẢN THỐNG KÊ

LẬP TRÌNH GAME
bằng
Visual Basic

Tổng hợp và biên soạn:
KS. Đinh Xuân Lâm

NHÀ XUẤT BẢN THỐNG KÊ

LỜI MỞ ĐẦU

Những trò chơi điện tử nhẹ nhàng, hấp dẫn và thú vị không chỉ giúp người chơi giải trí sau những giờ học tập, làm việc căng thẳng, mà còn giúp phát huy khả năng tư duy, suy đoán của người chơi. Đó cũng là một trong các lý do khiến những trò chơi điện tử đã lôi cuốn khá nhiều người tham gia đủ mọi lứa tuổi trên phạm vi toàn thế giới. Cho nên, việc viết các phần mềm trò chơi trên máy tính hiện nay đã được nhiều hãng và trung tâm phần mềm lớn quan tâm không kém việc nghiên cứu và viết những phần mềm khác.

Về phương diện lập trình, viết phần mềm trò chơi trên máy tính là một công việc phức tạp, có khi là một thử thách thực sự. Bởi vì, lập trình Game đòi hỏi lập trình viên không những là người thành thạo một ngôn ngữ lập trình, mà còn phải nắm rõ nguyên tắc, cách thức trò chơi mình định viết. Để xây dựng một trò chơi, đôi khi bạn phải vận dụng nhiều kiến thức khác nhau chẳng hạn Toán học, Vật lý, ..., chứ không chỉ biết ngôn ngữ lập trình là đủ.

Lập trình game bằng Visual Basic được xây dựng với mục đích giúp các bạn yêu thích máy tính và trò chơi điện tử tiếp cận với thế giới lập trình game, mà không đòi hỏi bạn phải là một cao thủ chơi game và lập trình giỏi. Với một kiến thức Visual Basic tương đối, bạn có thể tự mình xây dựng được các trò chơi nhẹ nhàng, phổ biến và đầy hấp dẫn.

Có thể tóm tắt công việc xây dựng một chương trình trò chơi gồm 3 công đoạn chính như sau :

- ◆ Phân tích những yêu cầu trò chơi, cách thức chơi
- ◆ Xây dựng các thuật toán xử lý hợp lý
- ◆ Sử dụng ngôn ngữ lập trình Visual Basic để thiết kế chương trình trò chơi dựa trên những thuật toán đã xây dựng

Hy vọng, tập sách này sẽ mang lại cho bạn nhiều điều bổ ích, tạo tiền đề để bạn tìm hiểu, tiếp cận với thế giới lập trình game, tuy phức tạp, bí ẩn, nhưng cũng đầy kỳ thú.

BÀI 1.

TRÒ CHƠI SẮP SỐ

I. Mô tả trò chơi

Trò chơi sắp số còn gọi là trò chơi Puzzle, một trò chơi đơn giản nhưng khá thú vị, đòi hỏi khả năng tính toán và sự sắp xếp khéo léo của người chơi.

- ♦ Thẻ lệ trò chơi như sau:

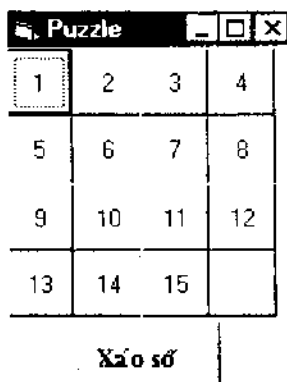
Không gian chơi gồm 16 ô số, được sắp xếp trên 4 hàng, mỗi hàng gồm 4 cột. Trong đó có 15 ô chứa số có giá trị từ 1 đến 15 được phân bố theo thứ tự ngẫu nhiên và 1 ô trống (Hình 1.1).

Puzzle			
6	1	9	7
13	2	10	8
11		4	5
14	15	12	3

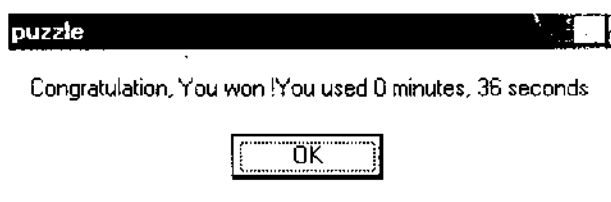
Xáo số

Hình 1.1 Một trong các màn hình trò chơi sắp số khi mới bắt đầu.

Người chơi phải tiến hành sắp xếp lại các ô số này theo thứ tự để được kết quả như Hình 1.2 thì trò chơi sẽ kết thúc. Chương trình hiển thị câu chúc mừng "Congratulation, You won !" và tổng thời gian chơi (Hình 1.3).



Hình 1.2 Trò chơi hoàn tất.



Hình 1.3 Thông báo chúc mừng và thời gian chơi.

Người chơi có thể thực hiện chơi lại bằng cách nhấp button "Xáo số". Chương trình sẽ tự động xáo lại các ô chứa số và ô trống theo thứ tự ngẫu nhiên.

Việc sắp xếp lại các ô số được thực hiện bằng cách sử dụng ô trống. Người dùng có thể chuyển một số từ các ô lân cận đến ô trống bằng cách nhấp mouse lên ô số đó. Chẳng hạn, trên Hình 1.1

người dùng có thể nhấp ô chứa số 2 để chuyển nó đến ô trống bên dưới và khi đó ô chứa số 2 cũ sẽ trở thành ô trống mới.

Chú ý rằng người dùng chỉ có thể di chuyển các số thuộc những ô lân cận ô trống. Chẳng hạn các ô lân cận ô trống trong Hình 1.1 là những ô chứa số 2, 4, 11 và 15.

Hình 1.4 là một ví dụ khác về các ô lân cận ô trống. Trong trường hợp này đó là các ô chứa số 9, 2 và 12. Để ý rằng hai ô chứa số 3 và 1 không được xem là các ô lân cận.

Puzzle			
9	3	5	8
	2	15	10
12	1	11	14
6	4	13	7

Xáo số

Hình 1.4 Trò chơi trong quá trình sắp xếp.

II. Các thuật toán xử lý

1. Xác định một ô có phải là ô lân cận của ô trống hiện hành hay không

- ♦ Đánh số thứ tự các ô từ 0 đến 15 theo thứ tự từ phải qua trái và từ trên xuống dưới.
- ♦ Xác định các ô lân cận cho từng ô.

Để ý rằng các ô có thứ tự 0 (hàng 1 cột 1), 3 (hàng 1 cột 4), 12 (hàng 4 cột 1) và 15 (hàng 4 cột 4) có số ô lân cận là 2.

8 ô có thứ tự 1 (hàng 1 cột 2), 2 (hàng 1 cột 3), 4 (hàng 2 cột 1), 7 (hàng 2 cột 4), 8 (hàng 3 cột 1), 11 (hàng 3 cột 4), 13 (hàng 4 cột 2) và 14 (hàng 4 cột 4) có số ô lân cận là 3.

4 ô còn lại có số thứ tự 5, 6, 9, 10 có số ô lân cận là 4.

Ví dụ: Trong Hình 1.4

- Ô có thứ tự 6 (chứa số 2), có số ô lân cận là 4, đó là các ô có thứ tự 2 (chứa số 3), 5 (không chứa số), 7 (chứa số 15), 10 (chứa số 1).
- Ô có thứ tự 16 (chứa số 7), có số ô lân cận là 2, đó là các ô có thứ tự 12 (chứa số 14), 15 (chứa số 13).
- ♦ Nếu một trong các ô lân cận của “ô được click” có giá trị trống (ô trống) thì hoán đổi nội dung “ô được click” với “ô trống”. Ngược lại thì không làm gì cả.

MÁCH NƯỚC

- ✓ Để ý rằng trong Visual Basic chỉ số mảng bắt đầu từ 0, nên để thuận tiện khi chuyển thuật toán qua ngôn ngữ lập trình bạn nên diễn giải thứ tự các ô từ 0 đến 15 thay vì từ 1 đến 16.
-

2. Thuật toán xáo số:

Có thể thực hiện như sau:

- ♦ Xem như không gian chơi của bạn có 16 ô đều trống.

- ♦ Chọn ngẫu nhiên 1 trong 16 số (từ 0 đến 15) để đặt vào ô trống đầu tiên.
- ♦ Sau đó tiếp tục chọn các số còn lại (15 con số chưa được chọn) để đặt vào ô thứ 2.
- ♦ Và cứ tiếp tục cho đến ô cuối cùng, sao cho đảm bảo nguyên tắc các số đã được chọn, sẽ không được chọn lại. Điều này tránh được tình trạng có 2 hay nhiều ô có cùng giá trị số.
- ♦ Xóa trống nội dung ô chứa số 0 để tạo "ô trống".

3. Di chuyển số đến ô trống:

Thực chất là hoán vị nội dung "ô được click" với "ô trống".

- ♦ Gán nội dung "ô được click" cho "ô trống".
- ♦ Xóa trống nội dung "ô được click".

III. Thiết kế chương trình

1. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start ➡ Programs ➡ Microsoft Visual Studio 6.0 ➡ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➡ New Project).

Bước 2. Visual Basic sẽ tạo một project mới form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➡ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name : frmPuzzle
- Caption : Puzzle

- WindowState : 0 – Normal
- ScaleMode : 1 - Twip
- Width : 2250
- Height : 3030

Chọn lệnh File ➡ Save Project, lưu form với tên *puzzle.frm* và lưu project với tên *puzzle.vbp* trong thư mục xác định, chẳng hạn *c:\itgames*.

Bước 3. Đặt các điều khiển vào form như Hình 1.5.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Xáo số

Hình 1.5 Màn hình thiết kế form Puzzle.

Đặt lên form 1 điều khiển Timer và định lại các thuộc tính sau:

Name = Timer1, Interval = 1000

Lần lượt đặt lên form 17 command button và định lại các thuộc tính theo bảng sau:

1. Command button 1	
Name : button	Index : 0

Caption : 1	Style : 1 – Graphical
Top : 0	Left : 0
2. Command button 2	
Name : button	Index : 1
Caption : 2	Style : 1 – Graphical
Top : 0	Left : 500
3. Command button 3	
Name : button	Index : 2
Caption : 3	Style : 1 – Graphical
Top : 0	Left : 1000
4. Command button 4	
Name : button	Index : 3
Caption : 4	Style : 1 – Graphical
Top : 0	Left : 1500
5. Command button 5	
Name : button	Index : 4
Caption : 5	Style : 1 – Graphical
Top : 500	Left : 0
6. Command button 6	
Name : button	Index : 5
Caption : 6	Style : 1 – Graphical
Top : 500	Left : 500
7. Command button 7	
Name : button	Index : 6

Caption : 7	Style : 1 – Graphical
Top : 500	Left : 1000
8. Command button 8	
Name : button	Index : 7
Caption : 8	Style : 1 – Graphical
Top : 500	Left : 1500
9. Command button 9	
Name : button	Index : 8
Caption : 9	Style : 1 – Graphical
Top : 1000	Left : 0
10. Command button 10	
Name : button	Index : 9
Caption : 10	Style : 1 – Graphical
Top : 1000	Left : 500
11. Command button 11	
Name : button	Index : 10
Caption : 11	Style : 1 – Graphical
Top : 1000	Left : 1000
12. Command button 12	
Name : button	Index : 11
Caption : 12	Style : 1 – Graphical
Top : 1000	Left : 1500
13. Command button 13	
Name : button	Index : 12

Caption : 13	Style : 1 – Graphical
Top : 1500	Left : 0
14. Command button 14	
Name : button	Index : 13
Caption : 14	Style : 1 – Graphical
Top : 1500	Left : 500
15. Command button 15	
Name : button	Index : 14
Caption : 15	Style : 1 – Graphical
Top : 1500	Left : 1000
16. Command button trống	
Name : button	Index : 15
Caption : (trống)	Style : 1 – Graphical
Top : 1500	Left : 1500
17. Command button Xáo số	
Name : cmdXao	Index : (rỗng)
Caption : Xáo số	Style : 0 – Standard
Top : 2040	Left : 360
BackColor : &H8000000F&	Font : VNI-Times, Bold, 12
Height : 495	Width : 1215

16 button đầu đều có thuộc tính:

Height = 500, Width = 500, BackColor : &H80000009&

MÁCH NƯỚC

- ✓ Để ý rằng có tới 16 button tạo thành một mảng cùng tên button, có kích cỡ, hình dạng giống nhau ngoại trừ thuộc tính Caption và vị trí là khác nhau mà thôi. Do đó để thiết kế nhanh và dễ dàng hơn, bạn hãy đặt lên form 1 command button, định xong thuộc tính cho điều khiển và lần lượt sao chép thành 15 command button khác (dùng lệnh Edit ➡ Copy và Edit ➡ Paste). Xong chỉnh sửa lại các thuộc tính cho phù hợp.
-

Bước 4. Viết mã lệnh

' Định nghĩa mảng lưu giá trị Caption của các button lân cận

Dim lancan(16, 4)

Dim sogiay *' Biến xác định số giây trôi qua kể từ khi bắt đầu chơi*

' Khi nhấp 1 trong 16 button

Private Sub button_Click(Index As Integer)

For i = 0 To 3 *' duyệt 4 phần tử mảng ứng với button được nhấp*

If lancan(Index, i) <> "" Then *' Nếu là button lân cận của button*

' được click

If button(lancan(Index, i)).Caption = "" Then *' button lân cận là button*

' trống gán trị số button được click cho button trống

button(lancan(Index, i)).Caption = button(Index).Caption

' đổi button được click thành button trống

button(Index).Caption = ""

End If

End If

Next

Kiemtra *' Gọi thủ tục kiểm tra*

End Sub

' Kiểm tra xem giá trị các button đã sắp theo thứ tự từ 1 đến 15 hay chưa

Sub kiểmtra()

dung = 1 *' Giả sử đã sắp đúng thứ tự*

For i = 0 To 14 *' Duyệt 15 button đầu tiên*

If button(i).Caption <> i + 1 Then *' Nếu có 1 button chưa sắp đúng thứ tự*

dung = 0 *' biến báo chưa sắp đúng thứ tự*

End If

Next

If button(15).Caption <> "" Then *' Nếu button cuối không phải là
' button trống*

dung = 0 *' biến báo chưa sắp đúng thứ tự*

End If

If dung = 1 Then *' Nếu đã sắp đúng thứ tự, hiển thị thông báo
' chúc mừng và thời gian người chơi đã
' thực hiện việc sắp xếp*

thoigian = "You used " & sogiay \ 60 & " minutes, " & sogiay Mod 60 & "
seconds"

MsgBox "Congratulation, You won ! " + thoigian

End If

End Sub

' Khi click button Xáo số

Private Sub cmdxao_Click()

For i = 0 To 15 *' Xóa trống Caption của 16 command
button*

button(i).Caption = ""

Next

For i = 0 To 15 *' Hiển thị số ngẫu nhiên từ 0 đến 15 trên các
' command button*

```

mso = Int(Rnd * 16)
j = 0
Do While j <= 1      ' Vòng lặp đảm bảo không chọn trùng số
    If mso = button(j).Caption Then
        mso = Int(Rnd * 16)
        j = 0
    Else
        j = j + 1
    End If
Loop
button(i).Caption = mso      ' Gán số được chọn cho thuộc tính Caption
                              ' của button

Next

For i = 0 To 15      ' Vòng lặp tìm button có thuộc tính Caption = 0
                    ' để xóa trống

    If button(i).Caption = 0 Then
        button(i).Caption = ""
    End If
Next

sogiyay = 0      ' Bắt đầu đếm thời gian chơi
End Sub

' Khi form được nạp
Private Sub Form_Load()
    Sogiyay = 0
    ' Xác định button lân cận cho từng button trong không gian làm việc
    lancan(0, 0) = 1      ' Button 1 có 2 button lân cận
    lancan(0, 1) = 4
    lancan(0, 2) = ""

```

$\text{lancan}(0, 3) = ""$

$\text{lancan}(1, 0) = 0$

' Button 2 có 3 button lân cận

$\text{lancan}(1, 1) = 2$

$\text{lancan}(1, 2) = 5$

$\text{lancan}(1, 3) = ""$

$\text{lancan}(2, 0) = 1$

' Button 3 có 3 button lân cận

$\text{lancan}(2, 1) = 3$

$\text{lancan}(2, 2) = 6$

$\text{lancan}(2, 3) = ""$

$\text{lancan}(3, 0) = 2$

' Button 4 có 2 button lân cận

$\text{lancan}(3, 1) = 7$

$\text{lancan}(3, 2) = ""$

$\text{lancan}(3, 3) = ""$

$\text{lancan}(4, 0) = 0$

' Button 5 có 3 button lân cận

$\text{lancan}(4, 1) = 5$

$\text{lancan}(4, 2) = 8$

$\text{lancan}(4, 3) = ""$

$\text{lancan}(5, 0) = 1$

' Button 6 có 4 button lân cận

$\text{lancan}(5, 1) = 4$

$\text{lancan}(5, 2) = 6$

$\text{lancan}(5, 3) = 9$

$\text{lancan}(6, 0) = 2$

' Button 7 có 4 button lân cận

$\text{lancan}(6, 1) = 5$

$\text{lancan}(6, 2) = 7$

$\text{lancan}(6, 3) = 10$

<code>lancan(7, 0) = 3</code>	<i>' Button 8 có 3 button lân cận</i>
<code>lancan(7, 1) = 6</code>	
<code>lancan(7, 2) = 11</code>	
<code>lancan(7, 3) = ""</code>	
<code>lancan(8, 0) = 4</code>	<i>' Button 9 có 3 button lân cận</i>
<code>lancan(8, 1) = 9</code>	
<code>lancan(8, 2) = 12</code>	
<code>lancan(8, 3) = ""</code>	
<code>lancan(9, 0) = 5</code>	<i>' Button 10 có 4 button lân cận</i>
<code>lancan(9, 1) = 8</code>	
<code>lancan(9, 2) = 10</code>	
<code>lancan(9, 3) = 13</code>	
<code>lancan(10, 0) = 6</code>	<i>' Button 11 có 4 button lân cận</i>
<code>lancan(10, 1) = 9</code>	
<code>lancan(10, 2) = 11</code>	
<code>lancan(10, 3) = 14</code>	
<code>lancan(11, 0) = 7</code>	<i>' Button 12 có 3 button lân cận</i>
<code>lancan(11, 1) = 10</code>	
<code>lancan(11, 2) = 15</code>	
<code>lancan(11, 3) = ""</code>	
<code>lancan(12, 0) = 8</code>	<i>' Button 13 có 2 button lân cận</i>
<code>lancan(12, 1) = 13</code>	
<code>lancan(12, 2) = ""</code>	
<code>lancan(12, 3) = ""</code>	
<code>lancan(13, 0) = 9</code>	<i>' Button 14 có 3 button lân cận</i>

```

lancan(13, 1) = 12
lancan(13, 2) = 14
lancan(13, 3) = ""
lancan(14, 0) = 10      ' Button 15 có 3 button lân cận
lancan(14, 1) = 13
lancan(14, 2) = 15
lancan(14, 3) = ""
lancan(15, 0) = 11      ' Button 16 có 2 button lân cận
lancan(15, 1) = 14
lancan(15, 2) = ""
lancan(15, 3) = ""
Randomize                ' Khởi động chế độ chọn số ngẫu nhiên
For i = 0 To 15          ' Xóa trống giá trị số hiển thị trên các button
    button(i).Caption = ""
Next
For i = 0 To 15          ' Hiển thị số ngẫu nhiên từ 0 đến 15 trên các button
    mso = Int(Rnd * 16)
    j = 0
    Do While j <= 1      ' Vòng lặp đảm bảo các số hiển thị là không trùng
        If mso = button(j).Caption Then
            mso = Int(Rnd * 16)
            j = 0
        Else
            j = j + 1
        End If
    Loop
    button(i).Caption = mso

```



```
Next
For i = 0 To 15      ' Tìm button có thuộc tính Caption = 0 để xóa trống
    If button(i).Caption = 0 Then
        button(i).Caption = ""
    End If
Next
End Sub

' Thủ tục được gọi cứ sau 1 giây một lần
Private Sub Timer1_Timer()
    sogiay = sogiay + 1      ' đếm số giây đã trôi qua
End Sub
```

Bước 5. Chọn Start ➡ Run hay nhấn F5 để chạy thử chương trình.

Nếu mọi thứ đều như ý, chọn lệnh File ➡ Make Puzzle.exe để compile chương trình thành tập tin Puzzle.exe có thể thi hành trên môi trường Windows 98, không cần khởi động Visual Basic.

IV. Phân tích chương trình

Phân thuật toán xử lý đã trình bày các kỹ thuật tiến hành trò chơi. Giờ đây, bạn chỉ cần chú ý thêm các điểm sau:

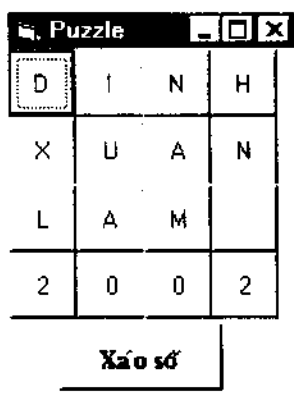
- ♦ Biến *sogiay* và mảng *lancan* được sử dụng trong nhiều thủ tục khác nhau. Để giá trị của biến duy trì khi chuyển từ thủ tục này sang thủ tục, bạn phải khai báo chúng như là các biến toàn cục của form Puzzle.
- ♦ Số button lân cận tối đa của một button bất kỳ là 4 và tối thiểu là 2. Với các button có ít hơn 4 button lân cận, gán giá trị rỗng "" cho thành phần mảng *lancan* tương ứng.

- ♦ Giá trị của tham số Index trong thủ tục biến cố button_Click giúp bạn xác định button tương ứng được click. Tham số này chỉ tồn tại khi các command button được tổ chức thành mảng có cùng tên button.

V. Bài tập thực hành

Bài 1. Chỉnh sửa chương trình Puzzle trên để mở rộng không gian chơi thành 8 hàng x 8 cột. Có thể sắp các số từ 1 đến 63 với 1 ô trống cuối cùng.

Bài 2. Cải tiến chương trình Puzzle để tiến hành sắp xếp chữ. Trò chơi sẽ kết thúc khi người chơi sắp được chữ như Hình 1.6.



Hình 1.6 Kết quả trò chơi xếp chữ.

Nhận xét: Thực hiện trò chơi xếp chữ như trên dễ chơi hơn nhiều so với xếp số, vì có các ký tự và số trùng nhau.

MÁCH NƯỚC

- ✓ Để khỏi phải chỉnh sửa chương trình nhiều, bạn có thể đặt các ký tự cần hiển thị vào 1 mảng 16 phần tử chẳng hạn Noidung(16). Sau đó tiến hành quy trình *xáo số* như bình thường nhưng thay lệnh :

button(i).Caption = mso

bằng lệnh : button(i).Caption = Noidung(mso)

- ✓ Tương tự, bạn có thể chỉnh sửa đoạn lệnh đầu thủ tục **Kiểmtra** thành:

For i = 0 To 15 ' Duyệt tất cả 16 button

If button(i).Caption <> Noidung(i) Then ' Nếu có 1 button chưa
'sắp đúng thứ tự

 dung = 0 ' biến báo chưa sắp đúng thứ tự

End If

Next

BÀI 2.

TRÒ CHƠI DÒ Mìn

I. Mô tả trò chơi

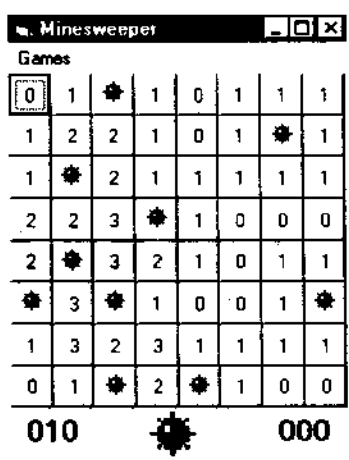
Trò chơi dò mìn còn gọi là trò chơi Minesweeper mà bạn có thể chơi khi sử dụng Windows 98, Windows Me,... Cách thức chơi cũng khá đơn giản, chỉ cần người chơi chịu khó tính toán và một chút may mắn là thành công.

♦ Thể lệ trò chơi như sau:

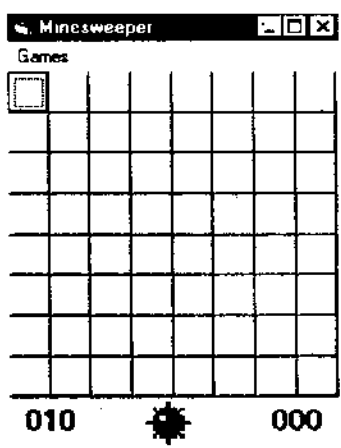
Không gian chơi gồm 64 ô, được sắp xếp trên 8 hàng, mỗi hàng gồm 8 cột. Trong đó có 10 ô ngẫu nhiên có chứa mìn. Các ô còn lại chứa giá trị số xác định số mìn của 8 ô bao quanh nó (Hình 2.1).

Khi trò chơi bắt đầu, bạn sẽ được màn hình như Hình 2.2. Trong đó chữ số ở góc trái dưới cho biết tổng số mìn có trong không gian chơi. Chữ số ở góc phải dưới là đồng hồ đếm tổng số thời gian chơi (tính bằng giây) kể từ khi người chơi nhấp ô đầu tiên.

Người chơi sẽ tiến hành nhấp một ô bất kỳ trong không gian chơi. Nếu ô được nhấp có chứa mìn, chương trình sẽ hiển thị vị trí tất cả 10 trái mìn, người chơi sẽ không thể truy xuất tới các ô còn lại, đồng hồ đếm thời gian chơi sẽ dừng lại tại thời điểm người chơi nhấp phải ô chứa mìn. Trò chơi sẽ kết thúc với kết quả người chơi thua (Hình 2.3).



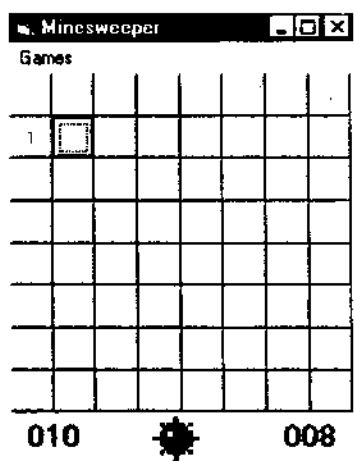
Hình 2.1 Một trong những kiểu phân bố min và cách đếm min của trò chơi.



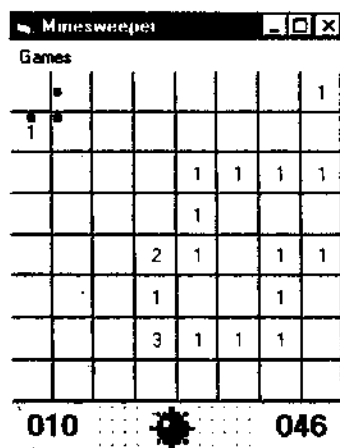
Hình 2.2 Màn hình bắt đầu trò chơi.

Để bắt đầu lần chơi mới, nhấn F2 hoặc chọn lệnh Games ➔ New trên menu chương trình. Bạn sẽ được một màn hình mới như Hình 2.2. Muốn kết thúc chương trình, nhấn tổ hợp phím Ctrl+Q hoặc chọn lệnh Games ➔ Exit.

ô thuộc hàng 4, cột 6 / 7 / 8, ô thuộc hàng 5, cột 6 hoặc ô thuộc hàng 6, cột 5 / 6 (để ý đây là các ô chứa số 0 liên thông với nhau).

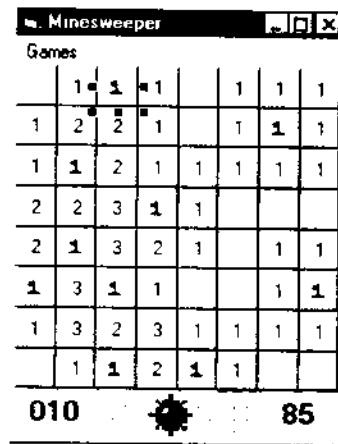


Hình 2.4 Nhấp ô không chứa mìn, nhưng các ô xung quanh có 1 ô chứa mìn.

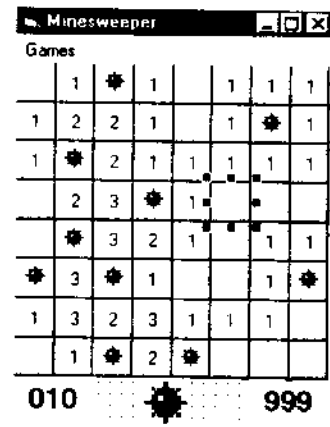


Hình 2.5 Nhấp nút phải mouse vào ô không chứa mìn và các ô xung quanh cũng không chứa mìn.

- Người chơi sẽ chiến thắng khi lật hết các ô không chứa mìn trong không gian chơi. Trò chơi sẽ kết thúc và các vị trí chứa mìn sẽ hiển thị lá cờ chiến thắng. Còn đồng hồ đếm thời gian chơi ở góc phải dưới sẽ dừng lại cho biết tổng thời gian chơi (Hình 2.6).



Hình 2.6 Trò chơi hoàn tất.



Hình 2.7 Người chơi lâu quá thời gian qui định.

- Nếu người chơi quá lâu (quá 999 giây) mà vẫn chưa hoàn tất lượt chơi, chương trình xem như người chơi thua cuộc và tự động hiển thị vị trí các quả mìn (Hình 2.7 trên).

II. Các thuật toán xử lý

1. Xác định ô chứa mìn

- ♦ Đánh số thứ tự các ô từ 0 đến 63 theo thứ tự từ phải qua trái và từ trên xuống dưới.

Bước 1. Chọn ngẫu nhiên 1 ô có vị trí bất kỳ trong không gian chơi (số thứ tự ô trong khoảng từ 0 đến 63).

Bước 2. Đặt quả mìn đầu tiên vào ô.

Bước 3. Chọn ngẫu nhiên 1 ô có vị trí bất kỳ trong không gian chơi.

Bước 4. Kiểm tra xem ô được chọn đã chứa mìn hay chưa? Nếu rồi, lặp lại bước 3. Nếu chưa, đặt quả mìn tiếp theo vào ô.

Bước 5. Kiểm tra xem số mìn đã đặt đã đủ 10 quả hay chưa? Nếu rồi, kết thúc thuật toán. Nếu chưa thì lặp lại bước 3.

Nhận xét: Rõ ràng thuật toán đảm bảo được 1 ô có tối đa 1 quả mìn và luôn có đủ 10 quả mìn trong không gian chơi.

2. Đếm số mìn có trong các ô xung quanh ô hiện hành

- ♦ Xác định các ô xung quanh ô hiện hành.

Để ý rằng 4 ô nằm 4 góc không gian chơi, có số thứ tự 0 (hàng 1 cột 1), 7 (hàng 1 cột 8), 56 (hàng 8 cột 1) và 63 (hàng 8 cột 8) có số ô xung quanh là 3.

Các ô thuộc hàng 1 (có số thứ tự 1, 2, 3, 4, 5, 6), hàng 8 (có số thứ tự 57, 58, 59, 60, 61, 62), cột 1 (có số thứ tự 8, 16, 24, 32, 40, 48), cột 8 (có số thứ tự 15, 23, 31, 39, 47, 55) có số ô xung quanh là 5.

Các ô còn lại trong không gian chơi có số ô xung quanh là 8.

- ♦ Thuật toán đếm số mìn xung quanh.

Bước 1. Duyệt lần lượt từng ô từ 0 đến 63.

Bước 2. Nếu ô đang duyệt là ô chứa mìn, trở lại bước 1, duyệt ô kế tiếp. Ngược lại, ô đang duyệt không chứa mìn, đặt "số mìn" ban đầu là 0, xong thực hiện tiếp bước 3.

Bước 3. Nếu ô đang duyệt thuộc cột 1 thì bỏ qua bước 3. Ngược lại, phải xác định có mìn ở ô bên trái ô đang duyệt hay không. Nếu có, tăng giá trị "số mìn" lên 1.

Bước 4. Nếu ô đang duyệt thuộc cột 8 thì bỏ qua bước 4. Ngược lại, xác định có mìn ở ô bên phải ô đang duyệt hay không. Nếu có, tăng giá trị "số mìn" lên 1.

Bước 5. Nếu ô đang duyệt thuộc hàng 1 thì bỏ qua bước 5. Ngược lại, xác định có mìn ở ô phía trên ô đang duyệt hay không. Nếu có, tăng giá trị "số mìn" lên 1.

Bước 6. Nếu ô đang duyệt thuộc hàng 8 thì bỏ qua bước 6. Ngược lại, xác định có mìn ở ô bên dưới ô đang duyệt hay không. Nếu có, tăng giá trị "số mìn" lên 1.

Bước 7. Nếu ô đang duyệt thuộc cột 1 hoặc hàng 1 thì bỏ qua bước 7. Ngược lại, xác định có mìn ở ô chéo trên bên trái ô đang duyệt hay không. Nếu có tăng giá trị "số mìn" lên 1.

Bước 8. Nếu ô đang duyệt thuộc **cột 8** hoặc **hàng 1** thì bỏ qua bước 8. Ngược lại, xác định có min ở ô chéo trên bên phải ô đang duyệt hay không. Nếu có, tăng giá trị “số min” lên 1.

Bước 9. Nếu ô đang duyệt thuộc **cột 1** hoặc **hàng 8** thì bỏ qua bước 9. Ngược lại, xác định có min ở ô chéo dưới bên trái ô đang duyệt hay không. Nếu có, tăng giá trị “số min” lên 1.

Bước 10. Nếu ô đang duyệt thuộc **cột 8** hoặc **hàng 8** thì bỏ qua bước 10. Ngược lại, xác định có min ở ô chéo dưới bên phải ô đang duyệt hay không. Nếu có, tăng giá trị “số min” lên 1.

Bước 11. Lưu giá trị số min xung quanh cho ô đang duyệt.

Bước 12. Nếu chưa duyệt đến ô cuối cùng thì lặp lại bước 1.

MÁCH NƯỚC

- ✓ Để ý rằng các ô thuộc cột 1 chỉ có tối đa 5 ô xung quanh, không có 3 ô bên trái, ô chéo bên trái phía trên và ô chéo bên trái phía dưới. Các ô thuộc cột 1 là các ô có số thứ tự 0, 8, 16, 32, 40, 48, 56 đều là những số chia hết cho 8.
- ✓ Các ô thuộc cột 8 cũng chỉ có tối đa 5 ô xung quanh, không có 3 ô bên phải, ô chéo bên phải phía trên và ô chéo bên phải phía dưới. Các ô thuộc cột 8 là những ô có số thứ tự 7, 15, 23, 31, 39, 47, 55, 63 đều là các số mà nếu đem trừ bớt 7 thì trở thành các số chia hết cho 8.
- ✓ Các ô thuộc hàng 1 có số thứ tự từ 0 đến 7 chỉ có tối đa 5 ô xung quanh, không có 3 ô phía trên, ô chéo phía trên bên trái và ô chéo phía trên bên phải.

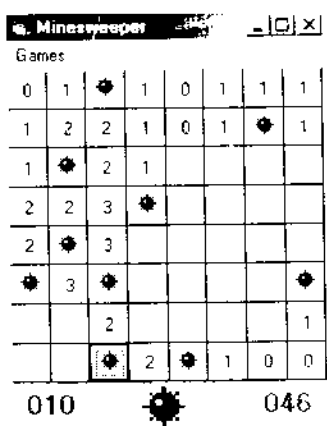
- ✓ Các ô thuộc hàng 8 có số thứ tự từ 56 đến 63 chỉ có tối đa 5 ô xung quanh, không có 3 ô phía dưới, ô chéo phía dưới bên trái và ô chéo phía dưới bên phải.
- ✓ Riêng 4 ô nằm 4 góc có 3 ô xung quanh cũng thỏa điều kiện trên. Chẳng hạn, hãy xét ô đầu tiên, do thuộc hàng 1 lại đồng thời thuộc cột 1 nên không có 5 ô phía trên, ô chéo phía trên bên trái, ô chéo phía trên bên phải, ô bên trái và ô chéo phía dưới bên trái.

3. Thuật toán “lan rộng” – tự động lật các ô không có mìn xung quanh: Có thể nói đây là thuật toán phức tạp nhất trong trò chơi này.

Nếu ô được nhấp là ô có giá trị 0 (các ô xung quanh nó không chứa mìn), chương trình sẽ lật ô vừa nhấp và tất cả các ô xung quanh nó. Sau đó, chương trình tiếp tục tìm các ô xung quanh đã lật có giá trị 0 để lật tiếp những ô lân cận quanh ô đó.

Ví dụ 1. Hình 2.8 là một diễn hình cho cách lật tự động các ô không có mìn xung quanh của thuật toán để có kết quả như Hình 2.5.

- ♦ Giả sử người chơi lật ô có giá trị 0 ở hàng 4, cột 7. Chương trình sẽ tự động lật 8 ô xung quanh nó. Trong đó có 3 ô (hàng 4, cột 8), (hàng 4 cột 6) và (hàng 5 cột 6) là các ô có giá trị 0.
- ♦ Sau đó chương trình sẽ tự động lật các ô xung quanh ô (hàng 4 cột 8). Do những ô này đều đã được lật, nên không có ô nào được lật thêm.
- ♦ Tiếp theo chương trình sẽ tự động lật các ô xung quanh ô (hàng 4 cột 6). Kết quả có thêm 3 ô (hàng 3, cột 5), (hàng 4, cột 5) và (hàng 5, cột 5) được lật.

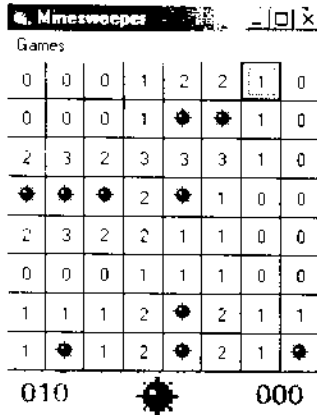


Hình 2.8 Tự động lật các ô không có mìn xung quanh.

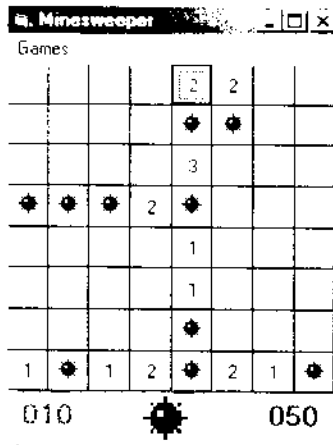
- ♦ Tiếp theo chương trình sẽ tự động lật các ô xung quanh ô (hàng 5 cột 6). Kết quả có thêm 3 ô nữa được lật. Trong đó có 2 ô (hàng 6, cột 5), (hàng 6, cột 6) có giá trị 0.
- ♦ Đến đây các ô xung quanh những ô trống khi lật ô (hàng 4, cột 7) đều đã lật hết. Nhưng lại có thêm 2 ô có giá trị 0 vừa được bổ sung khi lật ô (hàng 5, cột 6). Nên thuật toán chưa kết thúc.
- ♦ Tiếp theo chương trình sẽ tự động lật các ô xung quanh ô (hàng 6 cột 6). Kết quả có thêm 3 ô (hàng 7, cột 5), (hàng 7, cột 6) và (hàng 7, cột 7) được lật.
- ♦ Cuối cùng chương trình sẽ tự động lật các ô xung quanh ô (hàng 6 cột 5). Kết quả có thêm 3 ô (hàng 5, cột 4), (hàng 6, cột 4) và (hàng 7, cột 4) được lật.
- ♦ Đến lúc này các ô xung quanh những ô có giá trị 0 đã được lật đều có giá trị khác 0 nên thuật toán chấm dứt.

Kết quả chỉ cần nhấp mouse tại vị trí ô (hàng 4, cột 7), người chơi đã lật được một lúc tới 21 ô như Hình 2.5.

Ví dụ 2. Giả sử các quả mìn phân bố như Hình 2.9. Hình 2.10 là một minh họa khác cho 3 kiểu lật tự động các ô không có mìn xung quanh, khi người chơi nhấp các ô thuộc (hàng 1, cột 1), (hàng 1, cột 8) và (hàng 6, cột 2).



Hình 2.9 Một kết quả phân bố mìn trong không gian chơi.



Hình 2.10 Kết quả lật tự động các ô không có mìn chung quanh (các ô màu trắng).

Chú ý: Để ý rằng việc hiển thị số và vị trí các quả mìn trên những ô màu xám để bạn đọc tiện theo dõi thuật toán. Còn người chơi chỉ thấy được nội dung các ô đã được lật (những ô màu trắng).

Sau đây là nội dung thuật toán “Tự lật các ô không có mìn xung quanh” :

Bước 1. Nếu ô được nhấp là ô có giá trị 0, lật ô được nhấp.

Bước 2. Nếu ô bên phải có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô bên phải ô được nhấp.

Bước 3. Nếu ô bên trái có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô bên trái ô được nhấp.

Bước 4. Nếu ô bên trên có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô bên trên ô được nhấp.

Bước 5. Nếu ô bên dưới có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô bên dưới ô được nhấp.

Bước 6. Nếu ô bên trên có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô bên trên ô được nhấp.

Bước 7. Nếu ô ở góc chéo trên bên trái có giá trị 0 và chưa được lật ghi nhận lại vị trí này trong mảng *ncell*. Lật ô ở góc chéo trên bên trái ô được nhấp.

Bước 8. Nếu ô ở góc chéo trên bên phải có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô ở góc chéo trên bên phải ô được nhấp.

Bước 9. Nếu ô ở góc chéo dưới bên trái có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô ở góc chéo dưới bên trái ô được nhấp.

Bước 10. Nếu ô ở góc chéo dưới bên phải có giá trị 0 và chưa được lật, ghi nhận lại vị trí này trong mảng *ncell*. Lật ô ở góc chéo dưới bên phải ô được nhấp.

Bước 11. Duyệt lại mảng *ncell*. Nếu không còn vị trí nào được ghi nhận trong mảng thì kết thúc thuật toán. Ngược lại sẽ tiếp tục bước 12.

Bước 12. Lấy vị trí ô được ghi nhận trong mảng *ncell* ra như một ô vừa được nhấp và xóa vị trí ghi nhận vừa lấy khỏi mảng *ncell*. Lặp lại bước 1.

III. Thiết kế chương trình

1. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start ➡ Programs ➡ Microsoft Visual Studio 6.0 ➡ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➡ New Project).

Bước 2. Visual Basic sẽ tạo một project mới, form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➡ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name : frmMines
- Caption : Minesweeper
- WindowState : 0 – Normal
- ScaleMode : 1 - Twip
- Width : 3345
- Height : 4380

Chọn lệnh File ➡ Save Project, lưu form với tên *mines.frm* và lưu project với tên *mines.vbp* trong thư mục xác định, chẳng hạn *c:\ltgames*.

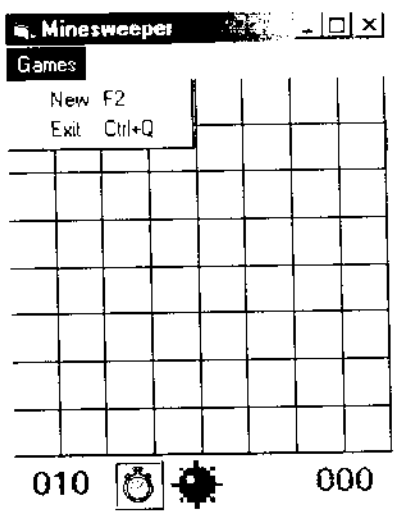
Bước 3. Chuẩn bị 2 tập tin hình *bom.jpg* (hình trái mìn) và *win.jpg* (hình lá cờ) (Hình 2.11) và đặt vào thư mục xây dựng ứng dụng *C:\ltgames*.



Hình 2.11 Tập tin *bom.jpg* và *win.jpg*.

Bạn có thể sưu tầm 2 hình này trong các đĩa CD icon hoặc dùng chương trình Snap32.exe để cắt chúng từ chương trình Minesweeper của Windows. Nếu kích thước hình quá lớn, bạn cần dùng chương trình Paint để định lại kích thước cho chúng khoảng 13x14 pixels.

Bước 4. Đặt các điều khiển vào form như Hình 2.12.



Hình 2.12 Màn hình thiết kế form *frmMines*.

Đặt lên form 1 điều khiển Timer và định lại các thuộc tính sau:

Name = Timer1, Interval = 1000

Đặt lên form 1 điều khiển Image và định lại các thuộc tính sau:

Name = Image1, Stretch = True, Picture = C:\Itgames\hom.jpg

Đặt lên form 2 điều khiển Label và định lại các thuộc tính như sau:

Name = Label1, Caption = 010, Alignment = 2 - Center

Name = Label2, Caption = 000, Alignment = 2 - Center

Các Label có cùng thuộc tính:

ForeColor = &H000000FF&

Fontname = Ms Sans Serif, FontStyle = Bold, FontSize = 14

Lần lượt đặt lên form 64 Command button và định lại các thuộc tính theo bảng sau:

1. Command button thứ 1 (hàng 1 cột 1)	
Name : button	Index : 0
Top : 0	Left : 0
2. Command button thứ 2 (hàng 1, cột 2)	
Name : button	Index : 1
Top : 0	Left : 400
3. Command button thứ 3 (hàng 1, cột 3)	
Name : button	Index : 2
Top : 0	Left : 800
4. Command button thứ 4 (hàng 1, cột 4)	
Name : button	Index : 3
Top : 0	Left : 1200

5. Command button thứ 5 (hàng 1, cột 5)	
Name : button	Index : 4
Top : 0	Left : 1600
6. Command button thứ 6 (hàng 1, cột 6)	
Name : button	Index : 5
Top : 0	Left : 2000
7. Command button thứ 7 (hàng 1, cột 7)	
Name : button	Index : 6
Top : 0	Left : 2400
8. Command button thứ 8 (hàng 1, cột 8)	
Name : button	Index : 7
Top : 0	Left : 2800
9. Command button thứ 9 (hàng 2, cột 1)	
Name : button	Index : 8
Top : 400	Left : 0
10. Command button thứ 10 (hàng 2, cột 2)	
Name : button	Index : 9
Top : 400	Left : 400
11. Command button thứ 11 (hàng 2, cột 3)	
Name : button	Index : 10
Top : 400	Left : 800
12. Command button thứ 12 (hàng 2, cột 4)	
Name : button	Index : 11
Top : 400	Left : 1200

13. Command button thứ 13 (hàng 2, cột 5)	
Name : button	Index : 12
Top : 400	Left : 1600
14. Command button thứ 14 (hàng 2, cột 6)	
Name : button	Index : 13
Top : 400	Left : 2000
15. Command button thứ 15 (hàng 2, cột 7)	
Name : button	Index : 14
Top : 400	Left : 2400
16. Command button thứ 16 (hàng 2, cột 8)	
Name : button	Index : 15
Top : 400	Left : 2800
17. Command button thứ 17 (hàng 3, cột 1)	
Name : button	Index : 16
Top : 800	Left : 0
18. Command button thứ 18 (hàng 3, cột 2)	
Name : button	Index : 17
Top : 800	Left : 400
19. Command button thứ 19 (hàng 3, cột 3)	
Name : button	Index : 18
Top : 800	Left : 800
20. Command button thứ 20 (hàng 3, cột 4)	
Name : button	Index : 19
Top : 800	Left : 1200

21. Command button thứ 21 (hàng 3, cột 5)	
Name : button	Index : 20
Top : 800	Left : 1600
22. Command button thứ 22 (hàng 3, cột 6)	
Name : button	Index : 21
Top : 800	Left : 2000
23. Command button thứ 23 (hàng 3, cột 7)	
Name : button	Index : 22
Top : 800	Left : 2400
24. Command button thứ 24 (hàng 3, cột 8)	
Name : button	Index : 23
Top : 800	Left : 2800
25. Command button thứ 25 (hàng 4 cột 1)	
Name : button	Index : 24
Top : 1200	Left : 0
26. Command button thứ 26 (hàng 4, cột 2)	
Name : button	Index : 25
Top : 1200	Left : 400
27. Command button thứ 27 (hàng 4, cột 3)	
Name : button	Index : 26
Top : 1200	Left : 800
28. Command button thứ 28 (hàng 4, cột 4)	
Name : button	Index : 27
Top : 1200	Left : 1200

29. Command button thứ 29 (hàng 4, cột 5)

Name : button	Index : 28
Top : 1200	Left : 1600

30. Command button thứ 30 (hàng 4, cột 6)

Name : button	Index : 29
Top : 1200	Left : 2000

31. Command button thứ 31 (hàng 4, cột 7)

Name : button	Index : 30
Top : 1200	Left : 2400

32. Command button thứ 32 (hàng 4, cột 8)

Name : button	Index : 31
Top : 1200	Left : 2800

33. Command button thứ 33 (hàng 5 cột 1)

Name : button	Index : 32
Top : 1600	Left : 0

34. Command button thứ 34 (hàng 5, cột 2)

Name : button	Index : 33
Top : 1600	Left : 400

35. Command button thứ 35 (hàng 5, cột 3)

Name : button	Index : 34
Top : 1600	Left : 800

36. Command button thứ 36 (hàng 5, cột 4)

Name : button	Index : 35
Top : 1600	Left : 1200

37. Command button thứ 37 (hàng 5, cột 5)

Name : button

Index : 36

Top : 1600

Left : 1600

38. Command button thứ 38 (hàng 5, cột 6)

Name : button

Index : 37

Top : 1600

Left : 2000

39. Command button thứ 39 (hàng 5, cột 7)

Name : button

Index : 38

Top : 1600

Left : 2400

40. Command button thứ 40 (hàng 5, cột 8)

Name : button

Index : 39

Top : 1600

Left : 2800

41. Command button thứ 41 (hàng 6, cột 1)

Name : button

Index : 40

Top : 2000

Left : 0

42. Command button thứ 42 (hàng 6, cột 2)

Name : button

Index : 41

Top : 2000

Left : 400

43. Command button thứ 43 (hàng 6, cột 3)

Name : button

Index : 42

Top : 2000

Left : 800

44. Command button thứ 44 (hàng 6, cột 4)

Name : button

Index : 43

Top : 2000

Left : 1200

45. Command button thứ 45 (hàng 6, cột 5)	
Name : button	Index : 44
Top : 2000	Left : 1600
46. Command button thứ 46 (hàng 6, cột 6)	
Name : button	Index : 45
Top : 2000	Left : 2000
47. Command button thứ 47 (hàng 6, cột 7)	
Name : button	Index : 46
Top : 2000	Left : 2400
48. Command button thứ 48 (hàng 6, cột 8)	
Name : button	Index : 47
Top : 2000	Left : 2800
49. Command button thứ 49 (hàng 7, cột 1)	
Name : button	Index : 48
Top : 2400	Left : 0
50. Command button thứ 50 (hàng 7, cột 2)	
Name : button	Index : 49
Top : 2400	Left : 400
51. Command button thứ 51 (hàng 7, cột 3)	
Name : button	Index : 50
Top : 2400	Left : 800
52. Command button thứ 52 (hàng 7, cột 4)	
Name : button	Index : 51
Top : 2400	Left : 1200

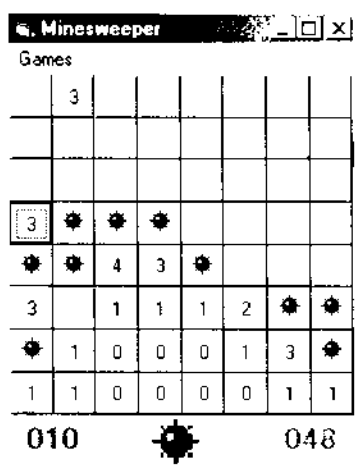
53. Command button thứ 53 (hàng 7, cột 5)	
Name : button	Index : 52
Top : 2400	Left : 1600
54. Command button thứ 54 (hàng 7, cột 6)	
Name : button	Index : 53
Top : 2400	Left : 2000
55. Command button thứ 55 (hàng 7, cột 7)	
Name : button	Index : 54
Top : 2400	Left : 2400
56. Command button thứ 56 (hàng 7, cột 8)	
Name : button	Index : 55
Top : 2400	Left : 2800
57. Command button thứ 57 (hàng 8, cột 1)	
Name : button	Index : 56
Top : 2800	Left : 0
58. Command button thứ 58 (hàng 8, cột 2)	
Name : button	Index : 57
Top : 2800	Left : 400
59. Command button thứ 59 (hàng 8, cột 3)	
Name : button	Index : 58
Top : 2800	Left : 800
60. Command button thứ 60 (hàng 8, cột 4)	
Name : button	Index : 59
Top : 2800	Left : 1200

61. Command button thứ 61 (hàng 8, cột 5)	
Name : button	Index : 60
Top : 2800	Left : 1600
62. Command button thứ 62 (hàng 8, cột 6)	
Name : button	Index : 61
Top : 2800	Left : 2000
63. Command button thứ 63 (hàng 8, cột 7)	
Name : button	Index : 62
Top : 2800	Left : 2400
64. Command button thứ 64 (hàng 8, cột 8)	
Name : button	Index : 63
Top : 2800	Left : 2800

64 button đều có cùng thuộc tính:

Height = 400, Width = 400, Caption = Rỗng, Style = 1 – Graphical,
BackColor : &H80000000F&

Chú ý quan trọng: Hết sức cẩn thận với vị trí và thứ tự các Command button. Chỉ cần xáo lộn vị trí 2 command button thứ 2 (hàng 1 cột 2) và thứ 42 (hàng 6 cột 2), bạn có thể nhận được những kết quả khó hiểu như Hình 2.13 khi nhập ô thuộc hàng 6 cột 2 (mà thoát nhìn bạn dễ lầm lẫn nguyên nhân là do thuật toán hay do câu lệnh lập trình của bạn sai, mà không hề nghĩ là do bạn đã đặt sai vị trí các Command button). Nếu bạn đặt vị trí nhiều Command button hơn, kết quả sẽ bất nháo hơn rất nhiều.



Hình 2.13 Sai sót do lập trình viên đặt lộn vị trí các Command button.

MÁCH NƯỚC

Để thiết kế nhanh các Command button và giảm thiểu việc nhầm lẫn vị trí, bạn nên làm như sau:

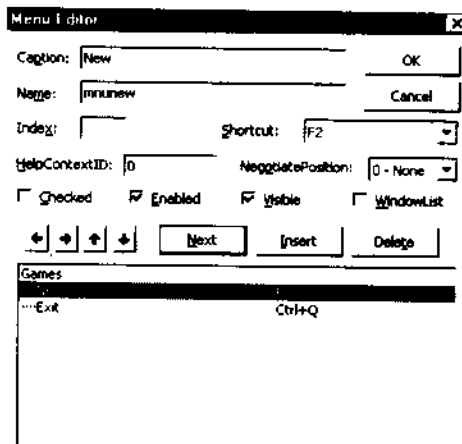
- ✓ Đầu tiên, đặt lên form Command button thứ 1, định xong các thuộc tính cho nó.
- ✓ Kế tiếp dùng lệnh Edit ➡ Copy, Edit ➡ Paste để sao chép thành Command button thứ 2 (cùng tên), xong tăng giá trị left thành 400.
- ✓ Tương tự sao chép thành các Command button 3,4,5,6,7,8. Mỗi lần tăng left thêm 400 twip thành 800,1200,1600,2000,2400,2800.
- ✓ Sao chép cùng lúc cả 8 Command button hàng 1 thành hàng 2. Sửa lại Top thành 400. Visual Basic sẽ tự động điều chỉnh giá trị thuộc tính Index trong quá trình sao chép.
- ✓ Tiếp tục sao chép thành các hàng 3, 4, 5, 6, 7, 8.

Bước 5. Thiết kế menu

Sử dụng Menu Editor thiết kế menu hệ thống cho chương trình theo bảng sau:

Caption	Name	Shortcut
Games	Mnugames	
New	Mnunew	F2
Exit	MnuExit	Ctrl+Q

- Mở màn hình thiết kế form `frmmines`, chọn lệnh **Tools** ➡ **Menu Editor** để mở cửa sổ Menu Editor và nhập vào từng lệnh Menu, xong nhấp **Next**. Để tạo các lệnh trong menu xuống, nhấp nút mũi tên phải. Muốn chuyển lệnh trở về menu cấp cao hơn, nhấp mũi tên trái. Kết quả thiết kế menu như Hình 2.14.
- Để đặt phím nóng gọi lệnh, chọn lệnh trong Menu Editor, mở popup mục **Shortcut** và chọn tổ hợp phím thích hợp.



Hình 2.14 Kết quả thiết kế menu chương trình trong cửa sổ Menu Editor.

Bước 6. Viết mã lệnh

' Khai báo các biến và mảng toàn cục của form

Dim nsweep(64), ncell(64)

Dim ngay, nbatdau

' Khi nhấp một ô bất kỳ trong không gian chơi

Private Sub button_Click(Index As Integer)

nbatdau = True *' bắt đầu đếm giờ*

If nsweep(Index) = "X" Then *' Nếu ô được nhấp có chứa mìn*

For i = 0 To 63 *' duyệt lần lượt 64 ô trong không gian chơi*

If nsweep(i) = "X" Then *' nếu là ô chứa mìn*

button(i).Caption = "" *' thì xóa trống nội dung ô*

button(i).Picture = LoadPicture(App.Path & "\bom.jpg")

' và hiển thị quả mìn

End If

button(i).Enabled = False *' khóa không cho truy xuất tới*

' ô (command button) nữa

Next

nbatdau = False *' kết thúc việc đếm giờ*

Else *' nếu ô được nhấp không chứa mìn*

If nsweep(Index) = 0 Then *' nếu ô được nhấp có các ô bao quanh*

' không chứa mìn

For i = 0 To 63

ncell(i) = 0 *' mặc định các ô không nằm trong danh sách ô tự động lật*

Next

button(Index).Caption = "" *' lật ô được nhấp (không cho truy xuất*

button(Index).Enabled = False *' button và đổi màu button thành trắng)*

button(Index).BackColor = &H80000005

lanrong (Index) *' gọi thủ tục lanrong, lật tự động*

' các ô bao quanh ô được nhấp

For i = 0 To 63 *' duyệt tìm các ô nằm trong lật tự động*

If ncell(i) <> 0 Then *' nếu tìm thấy*

ncell(i) = 0 *' loại ô khỏi danh sách lật ô tự động*

lanrong (i) *' gọi thủ tục lanrong, lật tự động các ô*

End If *' bao quanh ô tìm thấy*

Next

Else *' nếu ô được nhấp có các ô bao quanh có chứa mìn*

button(Index).Enabled = False *' lật ô được nhấp*

button(Index).BackColor = &H80000005

button(Index).Caption = nsweep(Index) *' hiển thị số mìn thuộc các ô bao quanh*

End If

Kiemtra *' gọi thủ tục kiểm tra, kiểm tra xem đã hoàn tất trò chơi chưa*

End If

End Sub

' Thủ tục được gọi sau mỗi lần nhấp một ô (command button)

' Kiểm tra xem người chơi đã lật hết các ô (trừ các ô chứa mìn) hay chưa

Sub kiểmtra()

xong = True

For i = 0 To 63 *' duyệt lần lượt 64 command button tương ứng cho 64 ô*

If button(i).Enabled = True And nsweep(i) <> "X" Then

xong = False *' nếu còn ô không chứa mìn chưa bị lật*

Exit For *' thì thoát lặp, không cần duyệt tiếp*

End If

Next

' nếu đã lật tất cả các ô không chứa mìn

If xong = True Then

For i = 0 To 63 ' vòng lặp tìm lật và bật các lá cờ trên các ô chứa mìn

If nsweep(i) = "X" Then

button(i).Caption = ""

button(i).Picture = LoadPicture(App.Path & "\win.jpg")

End If

button(i).Enabled = False ' lật ô (không cho truy xuất tới command
' button nữa)

Next

End If

End Sub

' Thủ tục lật tự động các ô "không có mìn bao quanh" ở xung quanh
' "ô quy định"

Sub lanrong(ncur As Byte)

hienhanh = ncur ' xác định vị trí "ô quy định" trong không gian chơi

' Lật tất cả các ô "không có mìn bao quanh" nằm ở bên phải "ô quy định"
' cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của cột cuối cùng
' trong không gian chơi mới dừng

Do While nsweep(hienhanh) = 0 ' lặp đến khi gặp "ô có mìn
' xung quanh"

If button(hienhanh).Enabled = True And hienhanh <> ncur Then

ncell(hienhanh) = 1 ' đánh dấu vị trí ô cho quá trình lật tự động sau đó

End If

button(hienhanh).Caption = ""

button(hienhanh).Enabled = False

button(hienhanh).BackColor = &H800000005

If (hienhanh - 7) Mod 8 = 0 Then ' nếu là ô thuộc cột cuối cùng thì
' thoát lặp

Exit Do

```

End If
hienhanh = hienhanh + 1
Loop
If nsweep(hienhanh) <> 0 Then ' nếu là ô có mìn xung quanh thì lật ô
    button(hienhanh).Caption = nsweep(hienhanh) ' hiển thị số mìn bao
        ' quanh trên ô
    button(hienhanh).Enabled = False
    button(hienhanh).BackColor = &H80000005
End If
' Lật tất cả các ô "không có mìn bao quanh" nằm ở bên trái "ô quy định"
' cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của cột đầu tiên
' trong không gian chơi mới dừng
hienhanh = ncur
Do While nsweep(hienhanh) = 0
    If button(hienhanh).Enabled = True And hienhanh <> ncur Then
        ncell(hienhanh) = 1 ' đánh dấu vị trí ô cho quá trình lật tự động sau đó
    End If
    button(hienhanh).Caption = ""
    button(hienhanh).Enabled = False
    button(hienhanh).BackColor = &H80000005
    If hienhanh Mod 8 = 0 Then ' nếu là ô thuộc cột đầu tiên thì thoát lặp
        Exit Do
    End If
    hienhanh = hienhanh - 1
Loop
If nsweep(hienhanh) <> 0 Then ' nếu là ô có mìn xung quanh thì lật ô
    button(hienhanh).Caption = nsweep(hienhanh)
    button(hienhanh).Enabled = False

```



```
button(hienhanh).BackColor = &H80000005
```

```
End If
```

```
' Lật tất cả các ô "không có mìn bao quanh" nằm ở phía dưới "ô quy định"  
' cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của hàng cuối cùng  
' trong không gian chơi mới dừng
```

```
hienhanh = ncuc
```

```
Do While nsweep(hienhanh) = 0
```

```
    If button(hienhanh).Enabled = True And hienhanh <> ncuc Then
```

```
        ncell(hienhanh) = 1
```

```
    End If
```

```
    button(hienhanh).Caption = ""
```

```
    button(hienhanh).Enabled = False
```

```
    button(hienhanh).BackColor = &H80000005
```

```
    If hienhanh >= 56 Then ' nếu là ô thuộc hàng cuối cùng
```

```
        Exit Do
```

```
    End If
```

```
    hienhanh = hienhanh + 8
```

```
Loop
```

```
If nsweep(hienhanh) <> 0 Then
```

```
    button(hienhanh).Caption = nsweep(hienhanh)
```

```
    button(hienhanh).Enabled = False
```

```
    button(hienhanh).BackColor = &H80000005
```

```
End If
```

```
' Lật tất cả các ô "không có mìn bao quanh" nằm phía trên "ô quy định"  
' cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của hàng đầu tiên  
' trong không gian chơi mới dừng
```

```
hienhanh = ncuc
```

```
Do While nsweep(hienhanh) = 0
```

```
    If button(hienhanh).Enabled = True And hienhanh <> ncuc Then
```

```
ncell(hienhanh) = 1
End If
button(hienhanh).Caption = ""
button(hienhanh).Enabled = False
button(hienhanh).BackColor = &H80000005
If hienhanh <= 7 Then
    Exit Do
End If
hienhanh = hienhanh - 8
Loop
If nsweep(hienhanh) <> 0 Then
    button(hienhanh).Caption = nsweep(hienhanh)
    button(hienhanh).Enabled = False
    button(hienhanh).BackColor = &H80000005
End If
' Lật tất cả các ô "không có mìn bao quanh " nằm ở phía trên bên trái "ô
' quy định" cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của hàng
' hoặc cột đầu tiên trong không gian chơi mới dừng
hienhanh = ncuc
Do While nsweep(hienhanh) = 0
    If button(hienhanh).Enabled = True And hienhanh <> ncuc Then
        ncell(hienhanh) = 1
    End If
    button(hienhanh).Caption = ""
    button(hienhanh).Enabled = False
    button(hienhanh).BackColor = &H80000005
    If hienhanh Mod 8 = 0 Or hienhanh <= 7 Then
        Exit Do
    End If
    hienhanh = hienhanh + 8
End Do
```

End If

hienhanh = hienhanh - 9

Loop

If nsweep(hienhanh) <> 0 Then

button(hienhanh).Caption = nsweep(hienhanh)

button(hienhanh).Enabled = False

button(hienhanh).BackColor = &H80000005

End If

' Lật tất cả các ô "không có mìn bao quanh" nằm ở phía trên bên phải "ô quy định" cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của hàng đầu tiên hoặc cột cuối cùng trong không gian chơi mới dừng

hienhanh = ncuc

Do While nsweep(hienhanh) = 0

If button(hienhanh).Enabled = True And hienhanh <> ncuc Then

ncell(hienhanh) = 1

End If

button(hienhanh).Caption = ""

button(hienhanh).Enabled = False

button(hienhanh).BackColor = &H80000005

If (hienhanh - 7) Mod 8 = 0 Or hienhanh <= 7 Then

Exit Do

End If

hienhanh = hienhanh - 7

Loop

If nsweep(hienhanh) <> 0 Then

button(hienhanh).Caption = nsweep(hienhanh)

button(hienhanh).Enabled = False

button(hienhanh).BackColor = &H80000005

End If

' Lật tất cả các ô "không có mìn bao quanh " nằm ở góc trái bên dưới "ô quy định" cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của cột đầu tiên hay hàng cuối cùng trong không gian chơi mới dừng

hienhanh = ncur

Do While nsweep(hienhanh) = 0

If button(hienhanh).Enabled = True And hienhanh <> ncur Then

ncell(hienhanh) = 1

End If

button(hienhanh).Caption = ""

button(hienhanh).Enabled = False

button(hienhanh).BackColor = &H80000005

If hienhanh Mod 8 = 0 Or hienhanh >= 56 Then

Exit Do

End If

hienhanh = hienhanh + 7

Loop

If nsweep(hienhanh) <> 0 Then

button(hienhanh).Caption = nsweep(hienhanh)

button(hienhanh).Enabled = False

button(hienhanh).BackColor = &H80000005

End If

' Lật tất cả các ô "không có mìn bao quanh " nằm ở góc phải phía dưới "ô quy định" cho đến khi gặp phải ô "có mìn bao quanh" hoặc là ô của hàng hay cột cuối cùng trong không gian chơi mới dừng

hienhanh = ncur

Do While nsweep(hienhanh) = 0

If button(hienhanh).Enabled = True And hienhanh <> ncur Then

ncell(hienhanh) = 1

```
End If
button(hienhanh).Caption = ""
button(hienhanh).Enabled = False
button(hienhanh).BackColor = &H80000005
If (hienhanh - 7) Mod 8 = 0 Or hienhanh >= 56 Then
    Exit Do
End If
hienhanh = hienhanh + 9
Loop
If nsweep(hienhanh) <> 0 Then
    button(hienhanh).Caption = nsweep(hienhanh)
    button(hienhanh).Enabled = False
    button(hienhanh).BackColor = &H80000005
End If
```

End Sub

' Khi form được nạp

Private Sub Form_Load()

```
nbatdau = False
sobom = 0
Randomize
' Phát sinh 10 vị trí có bom một cách ngẫu nhiên
For i = 1 To 10
    nvitri = Int(Rnd * 64)
    Do While nsweep(nvitri) = "X"
        nvitri = Int(Rnd * 64)
    Loop
    nsweep(nvitri) = "X" ' ghi nhận lại vị trí có bom
Next
```

' Giải thuật đếm bom chung quanh

For i = 0 To 63 ' duyệt tất cả 64 command button (đại diện cho 64 ô)

sobom = 0

' Đếm bom ở ô bên trái, trừ khi ô thuộc cột đầu tiên

If i Mod 8 <> 0 Then

 If nsweep(i - 1) = "X" Then

 sobom = sobom + 1

 End If

End If

' Đếm bom ở ô bên phải, trừ khi ô thuộc cột cuối cùng

If (i - 7) Mod 8 <> 0 Then

 If nsweep(i + 1) = "X" Then

 sobom = sobom + 1

 End If

End If

' Đếm bom ở ô phía trên, trừ khi ô thuộc hàng trên cùng

If i > 7 Then

 If nsweep(i - 8) = "X" Then

 sobom = sobom + 1

 End If

End If

' Đếm bom ở ô bên dưới, trừ khi ô thuộc hàng cuối cùng

If i < 56 Then

 If nsweep(i + 8) = "X" Then

 sobom = sobom + 1

 End If

End If

' Đếm bom ở ô chéo phía trên bên trái, trừ khi ô thuộc hàng hoặc cột

' đầu tiên

If i Mod 8 <> 0 And i > 7 Then

 If nsweep(i - 9) = "X" Then

 sobom = sobom + 1

 End If

End If

' Đếm bom ở ô chéo phía trên bên phải, trừ khi ô thuộc hàng đầu tiên

' hoặc cột cuối cùng

If (i - 7) Mod 8 <> 0 And i > 7 Then

 If nsweep(i - 7) = "X" Then

 sobom = sobom + 1

 End If

End If

' Đếm bom ở ô chéo phía dưới bên trái, trừ khi ô thuộc hàng cuối cùng

' hoặc cột đầu tiên

If i Mod 8 <> 0 And i < 56 Then

 If nsweep(i + 7) = "X" Then

 sobom = sobom + 1

 End If

End If

' Đếm bom ở ô chéo phía dưới bên phải, trừ khi ô thuộc hàng cuối cùng

' hoặc cột cuối cùng

If (i - 7) Mod 8 <> 0 And i < 56 Then

 If nsweep(i + 9) = "X" Then

 sobom = sobom + 1

 End If

End If

If nsweep(i) <> "X" Then

 nsweep(i) = sobom

End If

Next

' Xóa trống và cho phép truy xuất tới tất cả các ô trong không gian chơi

For i = 0 To 63

 button(i).Caption = ""

 button(i).Enabled = True

 button(i).BackColor = &H8000000F

 button(i).Picture = LoadPicture("")

Next

End Sub

' Khi chọn lệnh Games ➡ Exit hay nhấn CTRL + Q

Private Sub mnuExit_Click()

 End

End Sub

' Khi chọn lệnh Games ➡ New hay nhấn F2

Private Sub mnunew_Click()

For k = 0 To 63 *' Xóa mảng chứa giá trị tổng số bom xung quanh
 ' mỗi ô*

 nsweep(k) = 0

Next

ngiay = 0

lblclock.Caption = "000" *' Đặt đồng hồ trở về 0 giây*

nbatdau = False *' Dừng việc đếm giờ*

Form_Load *' Gọi thủ tục Form_load để phát sinh bom
 ' và đếm số bom xung quanh mỗi ô*

End Sub

' Thủ tục được gọi mỗi giây một lần

Private Sub Timer1_Timer()

If nbatdau Then *' nếu đang ở trạng thái đếm thời gian trôi qua*


```

ngiay = ngiay + 1
If ngiay >= 999 Then    ' Nếu ván chơi quá 999 giây
    For i = 0 To 63      ' hiển thị các vị trí có bom
        If nsweep(i) = "X" Then
            button(i).Caption = ""
            button(i).Picture = LoadPicture(App.Path & "\bom.jpg")
        End If
        button(i).Enabled = False    ' không cho truy xuất tới các ô nữa
        nbatdau = False              ' dùng để đếm thời gian, đợi người chơi thoát
    Next                             ' chương trình hay chơi lại ván khác
End If
lblclock.Caption = Right("000" & ngiay, 3)    ' Hiển thị thời gian chơi
                                              ' trôi qua

End If
End Sub

```

Bước 7. Chọn Start ➡ Run hay nhấn F5 để thi hành chạy thử chương trình.

Nếu mọi thứ đều như ý, chọn lệnh File ➡ Make Mines.exe để compile chương trình thành tập tin Mines.exe có thể thi hành trên môi trường Windows 98, không cần khởi động Visual Basic.

IV. Phân tích chương trình

- ♦ Mảng *nsweep* dùng để lưu thông tin số mìn của các ô xung quanh. Mỗi phần tử mảng *nsweep*, đại diện cho một ô có chỉ số index tương ứng trong không gian chơi. Giá trị lưu trong phần tử mảng nếu là "X" cho biết ô đó có chứa mìn. Nếu là giá trị số, cho biết ô không chứa mìn. Nếu giá trị số > 0, cho biết số mìn chứa

trong các ô bao quanh ô đó. Nếu giá trị số = 0, cho biết các ô bao quanh ô đó cũng không có mìn.

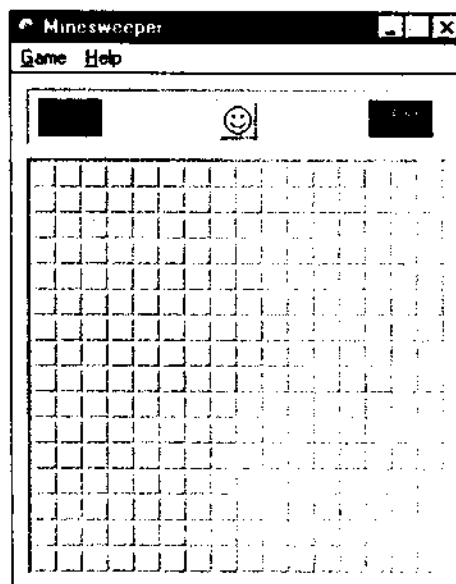
- ♦ Mảng *ncell* dùng lưu thông tin tổng số ô không có mìn bao quanh chưa được lật trong lần nhấp mới nhất. Tương tự mảng *nsweep*, mỗi phần tử mảng *ncell*, đại diện cho một ô có chỉ số index tương ứng trong không gian chơi. Nếu giá trị lưu trong phần tử mảng là 1, cho biết ô đang nằm trong danh sách các “ô không có mìn bao quanh” cần được tự động lật. Ngược lại, phần tử mảng sẽ có giá trị 0.
- ♦ Biến *ngiay* xác định số giây trôi qua của một lượt chơi.
- ♦ Biến *nbatdau* xác định trạng thái bắt đầu hay kết thúc việc đếm giờ.
- ♦ Giá trị của tham số Index trong thủ tục biến cố button_Click giúp bạn xác định button tương ứng được nhấp. Tham số này chỉ tồn tại khi các command button được tổ chức thành mảng có cùng tên button.

V. Bài tập thực hành

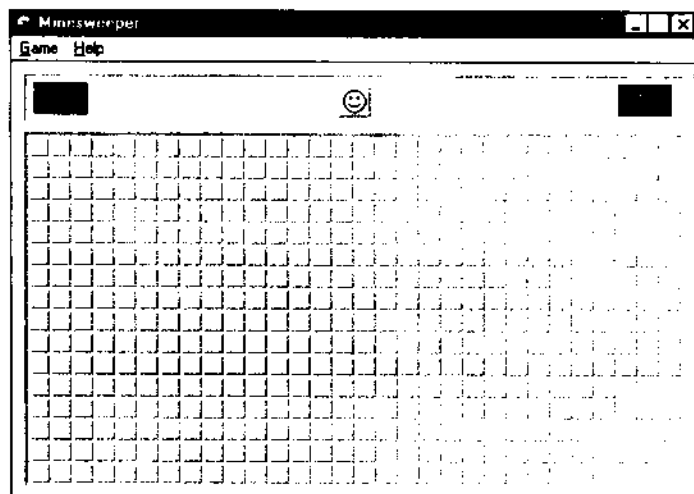
Bài 1. Năng cấp chương trình *Mines* đã xây dựng như sau:

Bổ sung vào menu chương trình 3 tùy chọn quy định mức độ chơi:

- ♦ Beginner: Mức độ sơ cấp, có 10 quả mìn trong không gian chơi 8 hàng x 8 cột (như đã trình bày ở trên).
- ♦ Intermediate: Mức độ trung cấp, có 40 quả mìn trong không gian chơi 16 hàng x 16 cột (Hình 2.15).
- ♦ Expert: Mức độ chuyên gia, có 99 quả mìn trong không gian chơi 16 hàng x 30 cột (Hình 2.16).



Hình 2.15 40 quả mìn trong không gian 16 hàng x 16 cột.



Hình 2.16 99 quả mìn trong không gian 16 hàng x 30 cột.

BÀI 3.

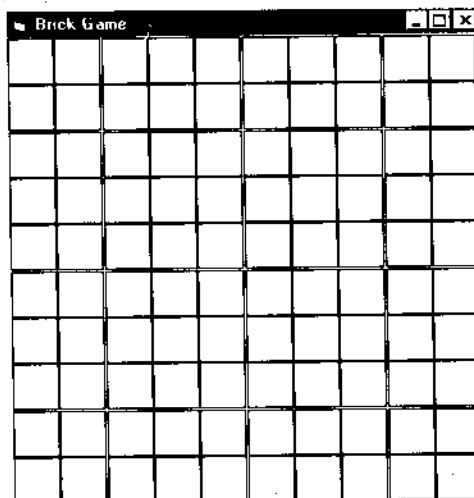
TRÒ CHƠI XẾP GẠCH

I. Mô tả trò chơi

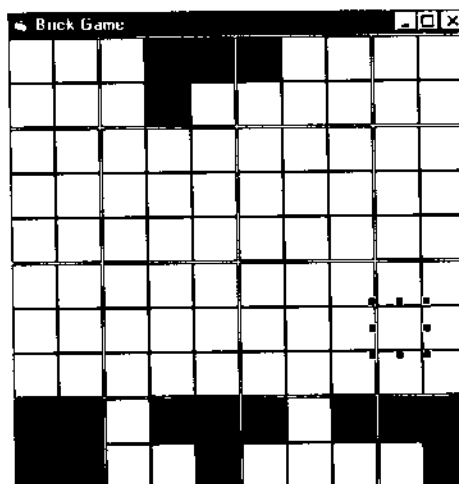
Có thể nói trò chơi xếp gạch là một trong những trò chơi quen thuộc đối với nhiều bạn trẻ. Có lẽ bạn cũng đã không ít lần chơi trò chơi này bằng máy chơi games, trên điện thoại di động hoặc bằng máy tính cá nhân. Còn bây giờ, điều bạn muốn biết là các lập trình viên đã xây dựng trò chơi này như thế nào.

Trước tiên chúng ta hãy thử điểm lại trò chơi định xây dựng:

- ♦ Người chơi thực hiện việc xếp gạch trong không gian chơi là 10×10 như Hình 3.1, mỗi ô có thể chứa một viên gạch.
- ♦ Có năm khối gạch khác nhau như Hình 3.2 và Hình 3.3, mỗi khối gạch đều có bốn viên.
- ♦ Mỗi lần một khối gạch ngẫu nhiên thuộc một trong bốn khối đã nêu sẽ xuất hiện trung tâm của hàng trên cùng. Khối gạch sẽ dần dần dịch chuyển xuống dưới sau mỗi khoảng thời gian qui định là 0.5 giây (mỗi lần một hàng). Quá trình dịch chuyển sẽ hoàn tất khi có ít nhất một trong bốn viên gạch của khối di chuyển đến một ô thuộc hàng cuối cùng (Hình 3.2) hay đụng phải các ô đang chứa gạch ở bên dưới (Hình 3.3).

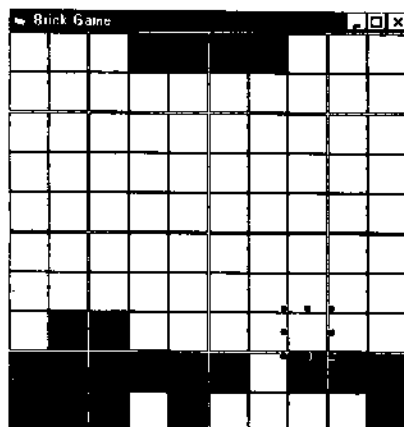


Hình 3.1 Không gian xếp gạch 10x10.



Hình 3.2 Hình dạng 4 trong 5 khối gạch mà chương trình sẽ phát sinh.

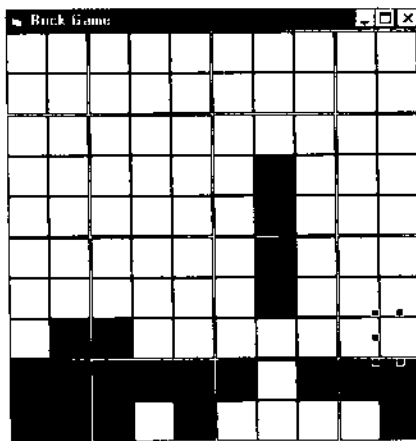
Trong Hình 3.2, ba khối gạch đầu di chuyển xuống dưới cho đến khi viên gạch đầu tiên trong khối đụng hàng cuối cùng mới dừng lại, do các ô bên dưới chưa chứa gạch.



Hình 3.3 Khối gạch thứ 5 vừa mới phát sinh.

Trong Hình 3.3, nếu bạn cứ cho khối gạch thứ 5 vừa mới phát sinh, dịch chuyển xuống dưới, khối gạch này sẽ bị dừng lại ở hàng thứ 3 (kể từ dưới lên), do các ô hàng 2 đã chứa gạch.

- ♦ Có thể dùng phím **↓** để tăng tốc độ dịch chuyển của khối gạch.
- ♦ Khi khối gạch hoàn tất việc di chuyển, chương trình sẽ tự động phát sinh ra khối gạch mới và bắt đầu lại quá trình di chuyển như cũ. Khối gạch mới được chọn ngẫu nhiên nên có thể có dạng trùng với khối gạch cũ hoặc không.
- ♦ Người chơi có thể dùng phím **←** hay phím **→** để dịch chuyển khối gạch sang trái hoặc phải trong quá trình di chuyển. Và có thể dùng phím **↑** để quay khối gạch, để chọn lựa kiểu dáng thích hợp cho việc lắp ghép với các viên gạch đang có ở bên dưới. Mỗi lần khối gạch sẽ quay ngược chiều kim đồng hồ 90 độ. Chẳng hạn như Hình 3.4, cho thấy người chơi đã quay khối gạch trong Hình 3.3 đi 90 độ và dịch chuyển sang phải, trong quá trình khối gạch dịch chuyển xuống dưới.

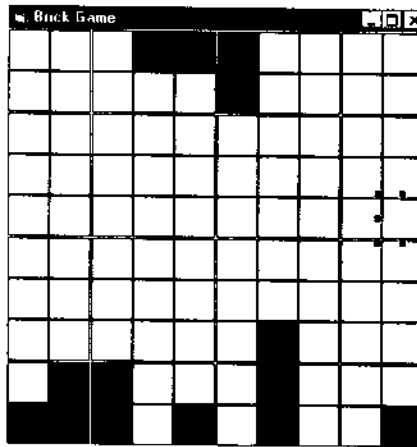


Hình 3.4 Quay khối gạch 90 độ và dịch chuyển sang phải.

- Chỉ có các hàng xếp đầy gạch mới được xóa khỏi không gian chơi, khi đó những hàng gạch phía trên sẽ được dịch xuống thay chỗ cho hàng vừa bị xóa, đồng thời chương trình sẽ tiến hành cộng điểm cho người chơi. Số điểm cộng sẽ thay đổi tùy vào số hàng bị xóa trong một lần.

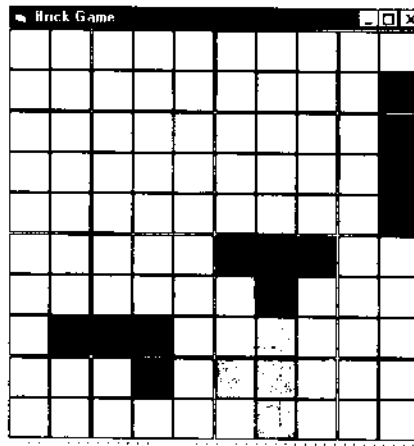
1 hàng bị xóa / 1 lần	Cộng 10 điểm
2 hàng bị xóa / 1 lần	Cộng 30 điểm
3 hàng bị xóa / 1 lần	Cộng 50 điểm
4 hàng bị xóa / 1 lần	Cộng 80 điểm

Chẳng hạn ở Hình 3.4, khi khối gạch đang dịch chuyển dừng, hàng thứ 2 (kể từ dưới lên) sẽ đầy gạch, nên sẽ bị xóa khỏi màn hình. Các viên gạch ở hàng 3 sẽ tự chuyển xuống hàng 2, hàng 4 chuyển xuống hàng 3. Đồng thời một khối gạch mới sẽ tự động xuất hiện ở phía trên như Hình 3.5.



Hình 3.5 Hàng 2 đầy gạch đã bị xóa khỏi không gian chơi.

Một ví dụ khác trong Hình 3.6 dễ thấy khi khối gạch đang di chuyển tiến đến hàng cuối cùng thì không chỉ một mà có đến bốn hàng đều đầy gạch, đây là một minh họa của việc xóa bốn hàng gạch một lần khỏi không gian chơi.



Hình 3.6 Bốn hàng xếp đầy gạch cùng một lúc.

- Tốc độ dịch chuyển của khối gạch sẽ thay đổi ngày càng nhanh tùy vào điểm số đạt được của người chơi.

Ban đầu: 1 giây / lần

Trên 1000 điểm: 0.5 giây / lần

Trên 2000 điểm: 0.4 giây / lần

Trên 3000 điểm: 0.3 giây / lần

Trên 5000 điểm: 0.2 giây / lần

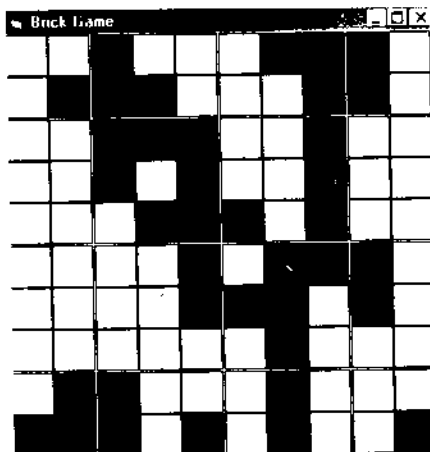
- ◆ Ngoài ra, chương trình còn có điểm thưởng dành cho người chơi nhanh theo qui định sau:

Nếu đạt 200 điểm sau 3 phút chơi: thưởng 100 điểm

Nếu đạt 1000 điểm sau 10 phút chơi: thưởng 200 điểm

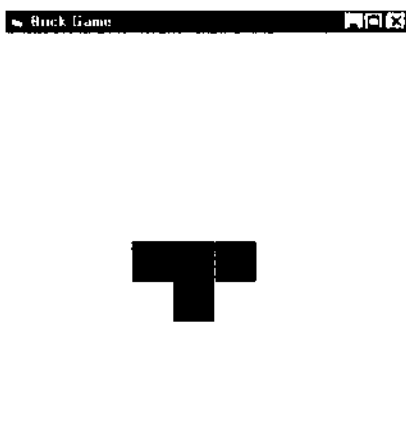
Nếu đạt 2000 điểm sau 15 phút chơi: thưởng 300 điểm

- ◆ Trò chơi sẽ kết thúc khi khối gạch bốn viên hình vuông không thể dịch đi đâu được nữa – Game Over.



Hình 3.7 Khối gạch bốn viên hình vuông không thể dịch đi đâu được nữa – Game Over.

Chú ý: Việc hiển thị các ô chưa chứa gạch chẳng qua để bạn dễ theo dõi khi xem sách. Còn khi lập trình, chỉ có những ô đang chứa gạch mới được hiển thị (Hình 3.8). Điều này khiến người chơi phải linh hoạt hơn trong việc xoay khối gạch và giống hàng.



Hình 3.8 Màn hình kết quả trò chơi xếp gạch.

II. Các thuật toán xử lý

1. Xác định vị trí các viên gạch trong khối gạch sẽ phát sinh

- ♦ Đánh số các ô trong không gian chơi theo thứ tự từ trái sang phải, hết hàng trên tiếp đến là hàng dưới. Trong đó ô đầu tiên bắt đầu là 0 như Hình 3.9.

Nhận xét: Cách đánh số này giúp bạn liên tưởng một ô trong không gian chơi tương ứng với một thành phần trong một mảng 100 phần tử. Và thứ tự ô lúc này chính là chỉ số mảng tương ứng.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Hình 3.9 Đánh số các ô trong không gian chơi.

- ♦ Tất cả các khối gạch đều có bốn viên gạch. Để lưu vị trí các viên gạch trong khối, ta có thể dùng biến chuỗi.

`m_dang = "vị trí viên 1, vị trí viên 2, vị trí viên 3, vị trí viên 4"`

Trong đó vị trí các viên gạch là chỉ số mảng của ô tương ứng trong không gian chơi.

Sau đây là vị trí xuất hiện đầu tiên của từng khối gạch trong không gian chơi và giá trị biến `m_dang` tương ứng:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19

Hình 3.10 Giá trị `m_dang = "03,04,05,06"`.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.11 Giá trị $m_dang = "03,04,05,13"$.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.12 Giá trị $m_dang = "03,04,05,14"$.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.13 Giá trị $m_dang = "03,04,05,15"$.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.14 Giá trị $m_dang = "04,05,14,15"$.

2. Xác định vị trí kế tiếp của các viên gạch trong khối gạch khi di chuyển xuống

Dễ thấy, mỗi lần dịch chuyển xuống một hàng, dù là khối gạch nào, vị trí của mỗi viên gạch trong khối đều tăng thêm 10 đơn vị.

Ví dụ: Sau khi dịch chuyển khối gạch trong Hình 3.13 xuống một hàng, ta có vị trí mới của từng viên gạch là $m_dang = "13,14,15,25"$.

3. Xác định vị trí kế tiếp của các viên gạch trong khối gạch khi di chuyển sang trái

Mỗi lần dịch chuyển sang trái một cột, dù là khối gạch nào, vị trí của mỗi viên gạch trong khối đều giảm một đơn vị.

Ví dụ: Sau khi dịch chuyển khối gạch trong Hình 3.12 sang trái một cột, ta có vị trí mới của từng viên gạch là $m_dang = "2,3,4,13"$.

4. Xác định vị trí kế tiếp của các viên gạch trong khối gạch khi di chuyển sang phải

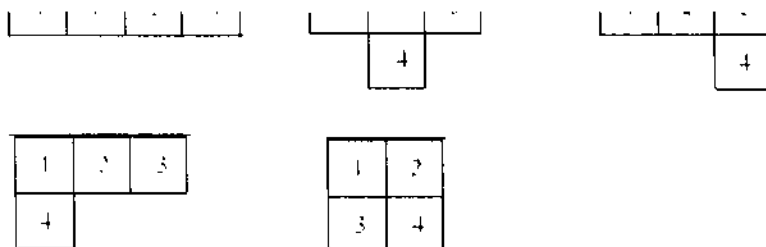
Mỗi lần dịch chuyển sang phải một cột, dù là khối gạch nào, vị trí của mỗi viên gạch trong khối đều tăng một đơn vị.

Ví dụ: Sau khi dịch chuyển khối gạch trong Hình 3.11 sang phải một cột, ta có vị trí mới của từng viên gạch là $m_dang = "4,5,6,14"$.

5. Phép biến hình khối gạch

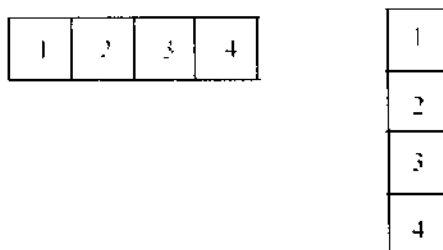
Trò chơi có tất cả năm khối gạch khác nhau, nhưng mỗi khối gạch trong quá trình di chuyển có thể biến đổi thành một số dạng khối mới bằng cách quay khối gạch 90 độ. Để tiện cho việc lập trình, kiểm tra các viên gạch trong khối, bạn cần đảm bảo cách xác định tâm quay mỗi khối gạch và thứ tự các viên gạch trong khối trước và sau khi biến hình là không thay đổi.

- ♦ Trong tập sách này, chúng tôi đánh số thứ tự các viên gạch trong khối như Hình 3.15.



Hình 3.15 Cách đánh số thứ tự các viên gạch trong từng khối gạch.

- Với khối gạch đầu tiên có thể biến đổi qua lại giữa hai dạng như Hình 3.16. Bạn có thể có dạng gạch 2 bằng cách quay dạng gạch 1 theo chiều kim đồng hồ 90 độ. Ngược lại, quay ngược chiều kim đồng hồ 90 độ, để đổi dạng gạch 2 trở lại dạng gạch 1. Quá trình quay luôn lấy viên gạch 1 làm tâm quay.



Hình 3.16 Các dạng biến đổi qua lại của khối gạch đầu tiên.

Sau đây là công thức biến đổi vị trí của từng viên gạch trong khối gạch thứ nhất:

- ♦ Dạng 1 - > Dạng 2:

$$VT1 \text{ mới} = VT1 \text{ cũ}$$

$$VT2 \text{ mới} = VT1 \text{ cũ} + 10$$

$$VT3 \text{ mới} = VT1 \text{ cũ} + 20$$

$$VT4 \text{ mới} = VT1 \text{ cũ} + 30$$

Trong đó: VT1 cũ là vị trí viên gạch 1 trước khi biến đổi

♦ Dạng 2 -> Dạng 1:

$$VT1 \text{ mới} = VT1 \text{ cũ}$$

$$VT2 \text{ mới} = VT1 \text{ cũ} + 1$$

$$VT3 \text{ mới} = VT1 \text{ cũ} + 2$$

$$VT4 \text{ mới} = VT1 \text{ cũ} + 3$$

Hình 3.17 là một minh họa cho phép biến đổi khối gạch đầu tiên. Rõ ràng công thức chuyển đổi đã trình bày là hoàn toàn đúng với khối gạch này.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59

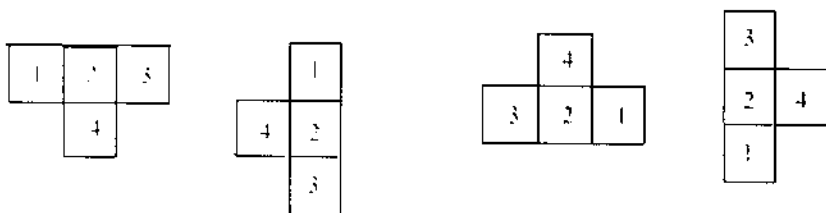
Hình 3.17 Ví dụ về phép biến đổi khối gạch đầu tiên trong không gian chơi.

- b. Với khối gạch thứ 2 có thể biến đổi qua lại giữa 4 dạng như Hình 3.18. Bạn có thể có dạng gạch 2 bằng cách quay dạng gạch 1 theo chiều kim đồng hồ 90 độ với tâm quay là viên gạch 2. Tương tự, bạn có thể có dạng gạch 3 bằng cách quay dạng gạch 2 90 độ theo chiều kim đồng hồ, dạng gạch 4 chính là dạng gạch 3 quay 90 độ theo chiều kim đồng hồ. Bạn cũng có thể quay dạng gạch 4 90 độ theo chiều kim đồng hồ để trở lại dạng gạch 1. Một điểm lưu ý là quá trình quay luôn lấy viên gạch 2 làm tâm quay và quá trình biến đổi dạng gạch luôn theo chu trình:

Dạng 1 -> Dạng 2 -> Dạng 3 -> Dạng 4



Nói cách khác, khi khối gạch đang ở dạng 3, bạn không thể biến đổi ngay thành dạng 2, mà phải biến đổi lần lượt theo chu trình.



Hình 3.18 Các dạng biến đổi qua lại của khối gạch thứ 2.

Sau đây là công thức biến đổi vị trí của từng viên gạch trong khối gạch thứ hai:

- ♦ Dạng 1 -> Dạng 2:

$$VT1 \text{ mới} = VT2 \text{ cũ} - 10$$

$$VT2 \text{ mới} = VT2 \text{ cũ}$$

$$VT3 \text{ mới} = VT2 \text{ cũ} + 10$$

$$VT4 \text{ mới} = VT2 \text{ cũ} - 1$$

♦ Dạng 2 - > Dạng 3:

$$VT1 \text{ mới} = VT2 \text{ cũ} + 1$$

$$VT2 \text{ mới} = VT2 \text{ cũ}$$

$$VT3 \text{ mới} = VT2 \text{ cũ} - 1$$

$$VT4 \text{ mới} = VT2 \text{ cũ} - 10$$

♦ Dạng 3 - > Dạng 4:

$$VT1 \text{ mới} = VT2 \text{ cũ} + 10$$

$$VT2 \text{ mới} = VT2 \text{ cũ}$$

$$VT3 \text{ mới} = VT2 \text{ cũ} - 10$$

$$VT4 \text{ mới} = VT2 \text{ cũ} + 1$$

♦ Dạng 4 - > Dạng 1:

$$VT1 \text{ mới} = VT2 \text{ cũ} - 1$$

$$VT2 \text{ mới} = VT2 \text{ cũ}$$

$$VT3 \text{ mới} = VT2 \text{ cũ} + 1$$

$$VT4 \text{ mới} = VT2 \text{ cũ} + 10$$

Hình 3.19 là một minh họa cho phép biến đổi khối gạch thứ hai.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

Hình 3.19a Ví dụ về phép biến đổi khối gạch thứ hai trong không gian chơi.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.19b Ví dụ về phép biến đổi khối gạch thứ hai trong không gian chơi.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.19c Ví dụ về phép biến đổi khối gạch thứ hai trong không gian chơi.

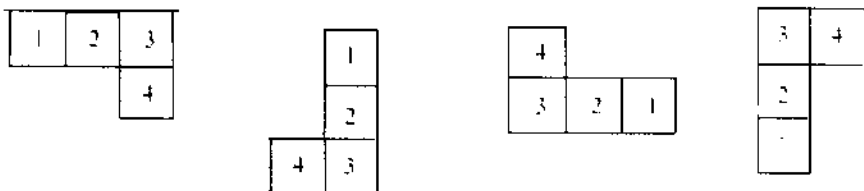
0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.19d Ví dụ về phép biến đổi khối gạch thứ hai trong không gian chơi.

- c. Với khối gạch thứ 3 có thể biến đổi qua lại giữa bốn dạng như Hình 3.20. Bạn có thể có dạng gạch 2 bằng cách quay dạng gạch 1 theo chiều kim đồng hồ 90 độ với tâm quay là viên gạch 3. Tương tự, bằng cách quay dạng gạch đang có 90 độ theo chiều kim đồng hồ, với tâm quay là viên gạch 3, bạn sẽ có dạng gạch kế tiếp theo đúng chu trình sau:

Dạng 1 -> Dạng 2 -> Dạng 3 -> Dạng 4





Hình 3.20 Các dạng biến đổi qua lại của khối gạch thứ 3.

Sau đây, là công thức biến đổi vị trí của từng viên gạch trong khối gạch thứ ba:

♦ Dạng 1 - > Dạng 2:

$$VT1 \text{ mới} = VT3 \text{ cũ} - 20$$

$$VT2 \text{ mới} = VT3 \text{ cũ} - 10$$

$$VT3 \text{ mới} = VT3 \text{ cũ}$$

$$VT4 \text{ mới} = VT3 \text{ cũ} - 1$$

♦ Dạng 2 - > Dạng 3:

$$VT1 \text{ mới} = VT3 \text{ cũ} + 2$$

$$VT2 \text{ mới} = VT3 \text{ cũ} + 1$$

$$VT3 \text{ mới} = VT3 \text{ cũ}$$

$$VT4 \text{ mới} = VT3 \text{ cũ} - 10$$

♦ Dạng 3 - > Dạng 4:

$$VT1 \text{ mới} = VT3 \text{ cũ} + 20$$

$$VT2 \text{ mới} = VT3 \text{ cũ} + 10$$

$$VT3 \text{ mới} = VT3 \text{ cũ}$$

$$VT4 \text{ mới} = VT3 \text{ cũ} + 1$$

- ♦ Dạng 4 - > Dạng 1:

$$VT1 \text{ mới} = VT3 \text{ cũ} - 2$$

$$VT2 \text{ mới} = VT3 \text{ cũ} - 1$$

$$VT3 \text{ mới} = VT3 \text{ cũ}$$

$$VT4 \text{ mới} = VT3 \text{ cũ} + 10$$

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

Hình 3.21a Ví dụ về phép biến đổi khối gạch thứ ba trong không gian chơi.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

Hình 3.21b Ví dụ về phép biến đổi khối gạch thứ ba trong không gian chơi.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29

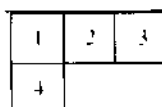
Hình 3.21c Ví dụ về phép biến đổi khối gạch thứ ba trong không gian chơi.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49

Hình 3.21d Ví dụ về phép biến đổi khối gạch thứ ba trong không gian chơi.

- d. Với khối gạch thứ 4 cũng có thể biến đổi qua lại giữa bốn dạng như Hình 3.22. Bạn có thể có dạng gạch 2 bằng cách quay dạng gạch 1 theo chiều kim đồng hồ 90 độ với tâm quay là viên gạch 1. Tương tự, bằng cách quay dạng gạch đang có 90 độ theo chiều kim đồng hồ, với tâm quay là viên gạch 1, bạn sẽ có dạng gạch kế tiếp theo đúng chu trình sau:

Dạng 1 -> Dạng 2 -> Dạng 3 -> Dạng 4



Hình 3.22 Các dạng biến đổi qua lại của khối gạch thứ 4.

Sau đây là công thức biến đổi vị trí của từng viên gạch trong khối gạch thứ tư:

- ♦ Dạng 1 -> Dạng 2:

$$VT1 \text{ mới} = VT1 \text{ cũ}$$

$$VT2 \text{ mới} = VT1 \text{ cũ} + 10$$

$$VT3 \text{ mới} = VT1 \text{ cũ} + 20$$

$$VT4 \text{ mới} = VT1 \text{ cũ} - 1$$

- ♦ Dạng 2 - > Dạng 3:

$$VT1 \text{ mới} = VT1 \text{ cũ}$$

$$VT2 \text{ mới} = VT1 \text{ cũ} - 1$$

$$VT3 \text{ mới} = VT1 \text{ cũ} - 2$$

$$VT4 \text{ mới} = VT1 \text{ cũ} - 10$$

- ♦ Dạng 3 - > Dạng 4:

$$VT1 \text{ mới} = VT1 \text{ cũ}$$

$$VT2 \text{ mới} = VT1 \text{ cũ} - 10$$

$$VT3 \text{ mới} = VT1 \text{ cũ} - 20$$

$$VT4 \text{ mới} = VT1 \text{ cũ} + 1$$

- ♦ Dạng 4 - > Dạng 1:

$$VT1 \text{ mới} = VT1 \text{ cũ}$$

$$VT2 \text{ mới} = VT1 \text{ cũ} + 1$$

$$VT3 \text{ mới} = VT1 \text{ cũ} + 2$$

$$VT4 \text{ mới} = VT1 \text{ cũ} + 10$$

Hình 3.23 là một minh họa cho phép biến đổi khối gạch thứ tư.

- e. Với khối gạch thứ 5 (khối gạch hình vuông), là một khối gạch đối xứng, nên sẽ không phát sinh dạng gạch mới khi quay nó 90 độ. Do vậy, không áp dụng lệnh biến đổi với khối gạch này.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

Hình 3.23 Ví dụ về phép biến đổi khối gạch thứ tư trong không gian chơi.

MẠCH NƯỚC

Khi biến đổi một khối gạch, trước tiên bạn cần đảm bảo khối gạch mới phải nằm hoàn toàn trong không gian chơi. Nếu không, thì không tiến hành phép biến đổi. Sau đây là các điều kiện cần phải thỏa trước khi biến đổi một khối gạch:

- ✓ Khối gạch 1 (Dạng 1 -> Dạng 2 - Hình 3.16):
 $VT1 < 70$ và các vị trí $VT1+10$, $VT1+20$, $VT1+30$ phải chưa có gạch
- ✓ Khối gạch 1 (Dạng 2 -> Dạng 1 - Hình 3.16):
 $VT1 \bmod 10 \leq 6$ và các vị trí $VT1+1$, $VT1+2$, $VT1+3$ phải chưa có gạch
- ✓ Khối gạch 2 (Dạng 1 -> Dạng 2 - Hình 3.18):
 $VT2 > 10$ và vị trí $VT2 - 10$ phải chưa có gạch
- ✓ Khối gạch 2 (Dạng 2 -> Dạng 3 - Hình 3.18):
 $VT2 \bmod 10 < 9$ và vị trí $VT2 + 1$ phải chưa có gạch
- ✓ Khối gạch 2 (Dạng 3 -> Dạng 4 - Hình 3.18):
 $VT2 < 90$ và vị trí $VT2+10$ phải chưa có gạch
- ✓ Khối gạch 2 (Dạng 4 -> Dạng 1 - Hình 3.18):
 $VT2 \bmod 10 > 0$ và vị trí $VT2-1$ phải chưa có gạch
- ✓ Khối gạch 3 (Dạng 1 -> Dạng 2 - Hình 3.20):
 $VT3 > 19$ và các vị trí $VT3-10$, $VT3-20$ phải chưa có gạch
- ✓ Khối gạch 3 (Dạng 2 -> Dạng 3 - Hình 3.20):
 $VT3 \bmod 10 \leq 7$ và các vị trí $VT3+1$, $VT3+2$ phải chưa có gạch
- ✓ Khối gạch 3 (Dạng 3 -> Dạng 4 - Hình 3.20):
 $VT3 < 79$ và các vị trí $VT3+10$, $VT3+20$ phải chưa có gạch
- ✓ Khối gạch 3 (Dạng 4 -> Dạng 3 - Hình 3.20):
 $VT3 \bmod 10 \geq 2$ và các vị trí $VT3-1$, $VT3-2$ phải chưa có gạch

- ✓ Khối gạch 4 (Dạng 1 -> Dạng 2 - Hình 3.22):
 $VT1 \bmod 10 \geq 1$, $VT1 < 79$ và các vị trí $VT1-1$, $VT1+20$ phải chưa có gạch
 - ✓ Khối gạch 4 (Dạng 2 -> Dạng 3 - Hình 3.22):
 $VT1 \bmod 10 \geq 2$, $VT1 > 9$ và các vị trí $VT1-2$, $VT1-10$ phải chưa có gạch
 - ✓ Khối gạch 4 (Dạng 3 -> Dạng 4 - Hình 3.22):
 $VT1 \bmod 10 \leq 8$, $VT1 > 19$ và các vị trí $VT1-20$, $VT1+1$ phải chưa có gạch
 - ✓ Khối gạch 4 (Dạng 4 -> Dạng 1 - Hình 3.22):
 $VT1 \bmod 10 \leq 7$, $VT1 < 89$ và các vị trí $VT1+2$, $VT1+10$ phải chưa có gạch
 - ✓ Khối gạch 5 - khối gạch hình vuông: không biến đổi
-

6. Xác định vị trí dừng của khối gạch

a. Khi dịch chuyển sang trái:

Nếu có ít nhất một trong bốn viên gạch của khối nằm trong cột đầu tiên trong không gian chơi hay đụng phải ô ở bên trái có chứa gạch thì sẽ dừng lại.

Ví dụ: Trong Hình 3.24 khối gạch "12,22,23,24" không thể dịch chuyển sang trái được nữa, vì bên trái hai viên gạch vị trí 12 và 22 là các ô đã chứa gạch (vị trí 11 và 21). Trong khi đó khối gạch "43,44,45,46" vẫn có thể tiếp tục di chuyển sang trái, vì ô bên trái khối gạch (vị trí 42) chưa có gạch. Còn khối gạch "10,11,20,21" không thể di chuyển sang trái được nữa vì có hai viên gạch vị trí 10, 20 thuộc cột đầu tiên trong không gian chơi.

Chú ý: Khi kiểm tra bên trái các viên gạch trong khối có gạch hay không, phải chú ý xác định đúng viên gạch cần kiểm tra tùy theo khối gạch và hình dạng hiện hành của nó. Chẳng hạn với khối gạch “43,44,45,46”, nếu bạn kiểm tra các viên gạch ở vị trí 44,45,46, thì chắc chắn bên trái nó luôn có gạch, nhưng đó là các viên gạch trong cùng khối, do đó không thể đảm bảo điều kiện dừng được, mà phải kiểm tra viên gạch ở vị trí 43.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69

Hình 3.24 Khối gạch “12,22,23,24” không thể dịch chuyển sang trái nữa.

Sau đây là danh sách các viên gạch cần kiểm tra bên trái theo từng khối gạch cụ thể để đảm bảo tính chính xác của điều kiện dừng bên trái:

- Khối gạch 1 – Dạng 1 (Hình 3.16): kiểm tra viên gạch 1
- Khối gạch 1 – Dạng 2 (Hình 3.16): kiểm tra cả bốn viên gạch
- Khối gạch 2 – Dạng 1 (Hình 3.18): kiểm tra viên gạch 1 và 4
- Khối gạch 2 – Dạng 2 (Hình 3.18): kiểm tra viên gạch 1, 3 và 4

- Khối gạch 2 – Dạng 3 (Hình 3.18): kiểm tra viên gạch 3 và 4
- Khối gạch 2 – Dạng 4 (Hình 3.18): kiểm tra viên gạch 1, 2 và 3
- Khối gạch 3 – Dạng 1 (Hình 3.20): kiểm tra viên gạch 1 và 4
- Khối gạch 3 – Dạng 2 (Hình 3.20): kiểm tra viên gạch 1, 2, và 4
- Khối gạch 3 – Dạng 3 (Hình 3.20): kiểm tra viên gạch 3 và 4
- Khối gạch 3 – Dạng 4 (Hình 3.20): kiểm tra viên gạch 1, 2 và 3
- Khối gạch 4 – Dạng 1 (Hình 3.22): kiểm tra viên gạch 1 và 4
- Khối gạch 4 – Dạng 2 (Hình 3.22): kiểm tra viên gạch 2, 3 và 4
- Khối gạch 4 – Dạng 3 (Hình 3.22): kiểm tra viên gạch 3 và 4
- Khối gạch 4 – Dạng 4 (Hình 3.22): kiểm tra viên gạch 1, 2 và 3
- Khối gạch 5 - khối gạch hình vuông (Hình 3.15): kiểm tra viên gạch 1 và 3

MÁCH NƯỚC

- ✓ Số thứ tự các ô trong cột đầu tiên luôn là một số chia hết cho 10.
 - ✓ Số thứ tự các ô trong cột cuối cùng là một số khi chia cho 10 sẽ dư 9.
-

b. Khi dịch chuyển sang phải:

Ngược lại với việc dịch chuyển sang trái, giờ đây điều bạn cần kiểm tra là:

Nếu có ít nhất một trong bốn viên gạch của khối nằm trong cột cuối cùng trong không gian chơi hay đưng phải ở ở bên phải có chứa gạch thì sẽ dừng lại.

Sau đây là danh sách các viên gạch cần kiểm tra bên trái theo từng khối gạch cụ thể để đảm bảo tính chính xác của điều kiện dừng bên trái:

- Khối gạch 1 – Dạng 1 (Hình 3.16): kiểm tra viên gạch 4
- Khối gạch 1 – Dạng 2 (Hình 3.16): kiểm tra cả 4 viên gạch
- Khối gạch 2 – Dạng 1 (Hình 3.18): kiểm tra viên gạch 3 và 4
- Khối gạch 2 – Dạng 2 (Hình 3.18): kiểm tra viên gạch 1, 2 và 3
- Khối gạch 2 – Dạng 3 (Hình 3.18): kiểm tra viên gạch 1 và 4
- Khối gạch 2 – Dạng 4 (Hình 3.18) : kiểm tra viên gạch 1, 3 và 4
- Khối gạch 3 – Dạng 1 (Hình 3.20) : kiểm tra viên gạch 3 và 4
- Khối gạch 3 – Dạng 2 (Hình 3.20) : kiểm tra viên gạch 1, 2 và 3
- Khối gạch 3 – Dạng 3 (Hình 3.20) : kiểm tra viên gạch 1 và 4
- Khối gạch 3 – Dạng 4 (Hình 3.20) : kiểm tra viên gạch 1, 2 và 4
- Khối gạch 4 – Dạng 1 (Hình 3.22) : kiểm tra viên gạch 3 và 4

- Khối gạch 4 – Dạng 2 (Hình 3.22) : kiểm tra viên gạch 1, 2 và 3
- Khối gạch 4 – Dạng 3 (Hình 3.22) : kiểm tra viên gạch 1 và 4
- Khối gạch 4 – Dạng 4 (Hình 3.22) : kiểm tra viên gạch 2, 3 và 4
- Khối gạch 5 - khối gạch hình vuông (Hình 3.15) : kiểm tra viên gạch 2 và 4

c. Khi dịch chuyển xuống dưới :

Nếu có ít nhất một trong bốn viên gạch của khối chuyển đến hàng cuối cùng trong không gian chơi hoặc đụng phải ô ở bên dưới có chứa gạch thì khối gạch sẽ dừng lại.

Sau đây là danh sách các viên gạch cần kiểm tra bên dưới theo từng khối gạch cụ thể để đảm bảo tính chính xác của điều kiện dừng lại:

- Khối gạch 1 – Dạng 1 (Hình 3.16): kiểm tra cả bốn viên gạch
- Khối gạch 1 – Dạng 2 (Hình 3.16): kiểm tra viên gạch 4
- Khối gạch 2 – Dạng 1 (Hình 3.18): kiểm tra viên gạch 1, 3 và 4
- Khối gạch 2 – Dạng 2 (Hình 3.18): kiểm tra viên gạch 3 và 4
- Khối gạch 2 – Dạng 3 (Hình 3.18): kiểm tra viên gạch 1, 2 và 3
- Khối gạch 2 – Dạng 4 (Hình 3.18): kiểm tra viên gạch 1 và 4
- Khối gạch 3 – Dạng 1 (Hình 3.20): kiểm tra viên gạch 1, 2 và 4

- Khối gạch 3 – Dạng 2 (Hình 3.20): kiểm tra viên gạch 3 và 4
- Khối gạch 3 – Dạng 3 (Hình 3.20): kiểm tra viên gạch 1, 2 và 3
- Khối gạch 3 – Dạng 4 (Hình 3.20): kiểm tra viên gạch 1 và 4
- Khối gạch 4 – Dạng 1 (Hình 3.22): kiểm tra viên gạch 2, 3 và 4
- Khối gạch 4 – Dạng 2 (Hình 3.22): kiểm tra viên gạch 3 và 4
- Khối gạch 4 – Dạng 3 (Hình 3.22): kiểm tra viên gạch 1, 2 và 3
- Khối gạch 4 – Dạng 4 (Hình 3.22): kiểm tra viên gạch 1 và 4
- Khối gạch 5 - khối gạch hình vuông (Hình 3.15): kiểm tra viên gạch 3 và 4

7. Xác định hàng gạch đầy

Một hàng gạch đầy là hàng gạch có tất cả các ô đều chứa gạch. Ô chứa gạch là ô được hiển thị trên màn hình (*thuộc tính Visible của điều khiển Shape1 tương ứng bằng True*).

8. Xóa các hàng gạch đầy

Sau khi xóa một hàng gạch đầy, nếu các ô hàng gạch phía trên hàng bị xóa có chứa gạch, bạn cần phải dịch chuyển chúng xuống một hàng để lấp vào hàng vừa bị xóa. Nói cách khác, những viên gạch phía trên hàng bị xóa sẽ dịch chuyển 10 vị trí.

$$VT \text{ mới} = VT \text{ cũ} + 10$$

Sau đây là các bước thực hiện giải thuật xóa hàng đầy gạch:

Bước 1. Duyệt tìm các hàng chứa đầy gạch từ trên xuống dưới để luôn đảm bảo rằng phía trên hàng chứa đầy gạch đầu tiên được tìm thấy sẽ không có hàng nào chứa đầy gạch nữa.

Bước 2. Nếu tìm thấy một hàng chứa đầy gạch (tất cả các ô trên hàng đều hiển thị), thì tiến hành xóa hàng như sau:

- Xóa hàng chứa đầy gạch (loại bỏ gạch khỏi hàng đầy).
- Dời các hàng phía trên hàng vừa xóa xuống một hàng. Bắt đầu từ hàng kế trên hàng bị xóa trở ngược lên đến hàng trên cùng.
- Xóa trống hàng gạch trên cùng.

Bước 3. Trở lại bước 1, tiếp tục duyệt tìm hàng đầy gạch, bắt đầu từ hàng vừa bị xóa trở xuống cho đến hàng cuối cùng thì dừng.

III. Thiết kế chương trình ➤

1. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start ➤ Programs ➤ Microsoft Visual Studio 6.0 ➤ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➤ New Project).

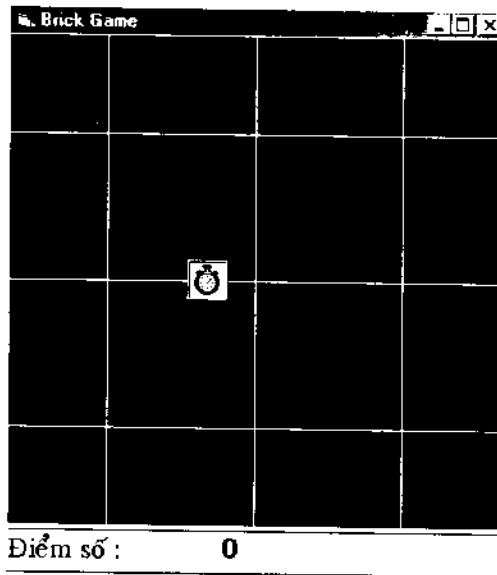
Bước 2. Visual Basic sẽ tạo một project mới có form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➤ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name: frmBrick
- Caption: Brick Game

- WindowState: 0 – Normal
- ScaleMode: 1 - Twip
- Width: 5145
- Height: 5835

Chọn lệnh File ➔ Save Project, lưu form với tên **Brick.frm** và lưu project với tên **Brick.vbp** trong thư mục xác định, chẳng hạn c:\ltgames.

Bước 3. Đặt các điều khiển vào form như Hình 3.25.



Hình 3.25 Màn hình thiết kế form frmBrick.

Đặt lên form 1 điều khiển Timer và định lại các thuộc tính sau:

Name = Timer1, Interval = 1000

Đặt lên form 1 điều khiển Shape và định lại các thuộc tính sau:

Name = Shape1, Shape = 0 – Rectangle, Index = 0, Visible = True

BackColor = &H80000006&, BackStyle = 1 – Opaque

BorderColor = &H80000006&, BorderStyle = 1 – Solid, BorderWidth = 1

FillColor = &H00FF0000&, FillStyle = 0 – Solid

Height = 500, Width = 500, Left = 0, Top = 0

Sao chép điều khiển Shape1 thành 9 điều khiển và định lại các thuộc tính sau:

1. Điều khiển Shape thứ 2 (hàng 1 cột 2)	
Name: Shape1	Index: 1
Top: 0	Left: 500
2. Điều khiển Shape thứ 3 (hàng 1 cột 3)	
Name: Shape1	Index: 2
Top: 0	Left: 1000
3. Điều khiển Shape thứ 4 (hàng 1 cột 4)	
Name: Shape1	Index: 3
Top: 0	Left: 1500
4. Điều khiển Shape thứ 5 (hàng 1 cột 5)	
Name: Shape1	Index: 4
Top: 0	Left: 2000
5. Điều khiển Shape thứ 6 (hàng 1 cột 6)	
Name: Shape1	Index: 5
Top: 0	Left: 2500
6. Điều khiển Shape thứ 7 (hàng 1 cột 7)	
Name: Shape1	Index: 6
Top: 0	Left: 3000

7. Điều khiển Shape thứ 8 (hàng 1 cột 8)	
Name: Shape1	Index: 7
Top: 0	Left: 3500
8. Điều khiển Shape thứ 9 (hàng 1 cột 9)	
Name: Shape1	Index: 8
Top: 0	Left: 4000
9. Điều khiển Shape thứ 10 (hàng 1 cột 10)	
Name: Shape1	Index: 9
Top: 0	Left: 4500

Sau đó, bao vùng chọn tất cả 10 điều khiển *Shape1* trên hàng đầu tiên. Thực hiện lệnh Edit ➡ Copy, rồi Edit ➡ Paste để sao thành hàng thứ 2. Định lại thuộc tính Top = 500 cho cả 10 điều khiển *Shape1* trên hàng thứ 2. Kiểm tra lại thuộc tính Index của các điều khiển *Shape1* trên hàng 2 sẽ tăng dần từ 10 đến 19 (tính từ trái sang phải).

Tương tự, bạn tiến hành sao thêm 8 hàng nữa. Như vậy, bạn đã có một mảng điều khiển *Shape1* gồm 100 phần tử có chỉ số Index chạy từ 0 đến 99.

Đặt tiếp lên form 1 điều khiển Line và 2 điều khiển Label với các thuộc tính sau:

1. Label 1	
Name: Label1	Caption: Điểm số
Width: 2175	Left: 0
Height: 495	Top: 5040
BackColor: &H8000000F&	BackStyle: 1 – Opaque

BorderStyle: 0 – None	ForeColor: &H80000012&
Font: VNI-Times, Normal, 14	
2. Label 2	
Name: Label2	Caption: 0
Width: 2895	Left: 2160
Height: 495	Top: 5040
BackColor: &H8000000F&	BackStyle: 1 – Opaque
BorderStyle: 0 – None	ForeColor: &H80000012&
Font: MS Sans Serif, Bold, 14	
3. Line 1	
Name: Line1	BorderWidth: 1
X1: 0	Y1: 5040
X2: 5040	Y2: 5040

Chú ý quan trọng: Hết sức cẩn thận để định vị trí và thứ tự các điều khiển Shape1 giống như Hình 3.9. Vì chỉ cần xáo lộn vị trí của 2 điều khiển Shape1, dạng khối hiển thị sẽ sai ngay.

2. Viết mã lệnh

‘ Khai báo các biến toàn cục có tác dụng trong toàn form

Dim dang(5), curshape, mstop, bienthe, diemso, thoigian, diembd

‘ Thủ tục được gọi khi người chơi nhấn một phím

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)

Select Case KeyCode

Case 39: moveright *‘ Nhấn phím mũi tên phải - right arrow*

Case 37: moveleft *‘ Nhấn phím mũi tên trái - left arrow*

Case 38: bienhinh *‘ Nhấn phím mũi tên hướng lên - Up arrow*

Case 40: movedown ' Nhấn phím mũi tên hướng xuống - Down arrow

End Select

End Sub

' Thủ tục được gọi khi form bắt đầu được nạp

Private Sub Form_Load()

' Khởi tạo giá trị ban đầu cho các biến

diemso = 0

diembd = 0

thoigian = 0

dang(0) = "03,04,05,06" ' Vị trí bắt đầu khối gạch 1

dang(1) = "03,04,05,14" ' Vị trí bắt đầu khối gạch 2

dang(2) = "03,04,05,15" ' Vị trí bắt đầu khối gạch 3

dang(3) = "03,04,05,13" ' Vị trí bắt đầu khối gạch 4

dang(4) = "04,05,14,15" ' Vị trí bắt đầu khối gạch 5

Randomize ' Khởi động chế độ chọn số ngẫu nhiên

curshape = Int(Rnd * 5) ' Chọn ngẫu nhiên một khối gạch trong năm khối

bienthe = 1 ' Mặc định khối gạch được chọn có dạng 1

For i = 0 To 99 ' đảm bảo không có ô nào trong không gian chơi

Shape1(i).Visible = False ' có chứa gạch

Next

show_brick ' Hiển thị vị trí bắt đầu khối gạch trong không gian chơi

End Sub

' Thủ tục được gọi khi người chơi phím nhấn mũi tên hướng lên

' Dùng để thực hiện việc chuyển đổi dạng của khối gạch đang di chuyển

Sub bienhinh()

m_dang = dang(curshape) ' Xác định khối gạch hiện hành

r_bien = True

Select Case curshape

Case 0: *' Biến đổi dạng khối gạch 1*

Select Case bienthe

Case 1: *' Biến đổi khối gạch 1 – dạng 1 thành dạng 2*

vt = Val(Mid(m_dang, 1, 2)) *' Vị trí viên gạch đầu tiên trong khối*

' Nếu thuộc một trong ba hàng cuối cùng trong không gian chơi

If vt >= 70 Then

 r_bien = False *' Không cho biến hình*

Else

' Nếu một trong ba ô bên phải ô chứa viên gạch đầu tiên đã có gạch

 If Shape1(Val(vt + 10)).Visible = True Or _

 Shape1(Val(vt + 20)).Visible = True Or _

 Shape1(Val(vt + 30)).Visible = True Then

 r_bien = False *' Không cho biến dạng khối gạch*

 End If

End If

' Nếu khối gạch có thể biến dạng

If r_bien = True Then

 hide_brick *' ẩn khối gạch 1 dạng 1 hiện hành*

' Định lại vị trí mới cho khối gạch 1 dạng 2

 vt = Val(Mid(m_dang, 1, 2))

 dang(curshape) = Right("0" + CStr(vt), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 10), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 20), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 30), 2) + ","

 show_brick *' hiển thị khối gạch 1, dạng 2*

 bienthe = 2

End If

Select Case bignthe

```

Case 1: ' Biến đổi khối gạch 2, dạng 1 thành khối gạch 2 dạng 2
vt = Val(Mid(m_dang, 4, 2)) ' Xác định vị trí viên gạch thứ 2 trong khối
If vt <= 9 Then ' nếu thuộc hàng đầu tiên
    r_bien = False ' không cho biến dạng
Else
    ' nếu ô phía trên có gạch
    If Shape1(Val(vt - 10)).Visible = True Then
        r_bien = False ' Không cho biến dạng
    End If
End If
If r_bien = True Then
    hide_brick
    vt = Val(Mid(m_dang, 4, 2))
    dang(curshape) = Right("0" + CStr(vt - 10), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 10), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 1), 2) + ","
    show_brick
    bienthe = 2
End If

Case 2: ' Biến đổi khối gạch 2, dạng 2 thành khối gạch 2 dạng 3
vt = Val(Mid(m_dang, 4, 2)) ' xác định vị trí viên gạch thứ 2
                                ' trong khối
If (vt Mod 10 = 9) Then ' nếu thuộc cột cuối cùng
    r_bien = False ' không cho biến hình
Else
    ' nếu ô bên phải có chứa gạch
    If Shape1(Val(vt + 1)).Visible = True Then

```

```
    r_bien = False          ' không cho biến hình
End If
End If
If r_bien = True Then
    hide_brick
    vt = Val(Mid(m_dang, 4, 2))
    dang(curshape) = Right("0" + CStr(vt + 1), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 1), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 10), 2) + ","
    show_brick
    bienthe = 3
End If
```

Case 3: ' *Biến đổi khối gạch 2, dạng 3 thành khối gạch 2 dạng 4*

```
vt = Val(Mid(m_dang, 4, 2)) ' xác định vị trí viên gạch thứ 2 trong khối
If vt >= 90 Then          ' nếu thuộc hàng cuối cùng trong khối
    r_bien = False        ' Không cho biến hình
Else
    ' nếu ô bên dưới có gạch
    If Shape1(Val(vt + 10)).Visible = True Then
        r_bien = False    ' Không cho biến hình
    End If
End If
If r_bien = True Then
    hide_brick
    vt = Val(Mid(m_dang, 4, 2))
    dang(curshape) = Right("0" + CStr(vt + 10), 2) + ","
```



```

dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 10), 2) + ","
dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 1), 2) + ","
show_brick
bienthe = 4

```

```
End If
```

Case 4: ' *Biến đổi khối gạch 2, dạng 4 thành khối gạch 2 dạng 1*

```
vt = Val(Mid(m_dang, 4, 2)) ' xác định vị trí viên gạch thứ 2 trong khối
```

```
If (vt Mod 10 = 0) Then ' nếu thuộc cột đầu tiên
```

```
    r_bien = False ' không cho biến hình
```

```
Else
```

' *Nếu ô bên trái có chứa gạch*

```
    If Shape1(Val(vt - 1)).Visible = True Then
```

```
        r_bien = False
```

```
    End If
```

```
End If
```

```
If r_bien = True Then
```

```
    hide_brick
```

```
    vt = Val(Mid(m_dang, 4, 2))
```

```
    dang(curshape) = Right("0" + CStr(vt - 1), 2) + ","
```

```
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
```

```
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 1), 2) + ","
```

```
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 10), 2) + ","
```

```
    show_brick
```

```
    bienthe = 1
```

```
End If
```

```
End Select
```

Case 2: ' *Biến đổi dạng khối gạch 3*

Select Case bienthe

Case 1: ' *Biến đổi khối gạch 3, dạng 1 thành khối gạch 3 dạng 2*

vt = Val(Mid(m_dang, 7, 2))

If vt <= 19 Then

 r_bien = False

Else

 If Shape1(Val(vt - 10)).Visible = True Or _

 Shape1(Val(vt - 20)).Visible = True Then

 r_bien = False

 End If

End If

If r_bien = True Then

 hide_brick

 vt = Val(Mid(m_dang, 7, 2))

 dang(curshape) = Right("0" + CStr(vt - 20), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 10), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 1), 2) + ","

 show_brick

 bienthe = 2

End If

Case 2: ' *Biến đổi khối gạch 3, dạng 2 thành khối gạch 3 dạng 3*

vt = Val(Mid(m_dang, 7, 2))

If vt Mod 10 > 7 Then

 r_bien = False

Else

 If Shape1(Val(vt + 1)).Visible = True Or _

 Shape1(Val(vt + 2)).Visible = True Then

 r_bien = False

 End If

End If

If r_bien = True Then

 hide_brick

 vt = Val(Mid(m_dang, 7, 2))

 dang(curshape) = Right("0" + CStr(vt + 2), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 1), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","

 dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 10), 2) + ","

 show_brick

 bienthe = 3

End If

Case 3: ' *Biến đổi khối gạch 3, dạng 3 thành khối gạch 3 dạng 4*

vt = Val(Mid(m_dang, 7, 2))

If vt > 79 Then

 r_bien = False

Else

 If Shape1(Val(vt + 10)).Visible = True Or _

 Shape1(Val(vt + 20)).Visible = True Then

 r_bien = False

 End If

End If

If r_bien = True Then

```
hide_brick
vt = Val(Mid(m_dang, 7, 2))
dang(curshape) = Right("0" + CStr(vt + 20), 2) + ","
dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 10), 2) + ","
dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 1), 2) + ","
show_brick
bienthe = 4
End If

Case 4: ' Biến đổi khối gạch 3, dạng 4 thành khối gạch 3 dạng 1
vt = Val(Mid(m_dang, 7, 2))
If vt Mod 10 < 2 Then
    r_bien = False
Else
    If Shape1(Val(vt - 1)).Visible = True Or _
        Shape1(Val(vt - 2)).Visible = True Then
        r_bien = False
    End If
End If
If r_bien = True Then
    hide_brick
    vt = Val(Mid(m_dang, 7, 2))
    dang(curshape) = Right("0" + CStr(vt - 2), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 1), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 10), 2) + ","
    show_brick
```

```

    bienthe = 1
End If
End Select

```

Case 3: ' *Biến đổi dạng khối gạch 4*

```
Select Case bienthe
```

Case 1: ' *Biến đổi khối gạch 4, dạng 1 thành khối gạch 4 dạng 2*

```
vt = Val(Mid(m_dang, 1, 2))
```

```
If vt Mod 10 = 0 Or vt > 79 Then
```

```
    r_bien = False
```

```
Else
```

```
    If Shape1(Val(vt - 1)).Visible = True Or _
```

```
        Shape1(Val(vt + 20)).Visible = True Then
```

```
        r_bien = False
```

```
    End If
```

```
End If
```

```
If r_bien = True Then
```

```
    hide_brick
```

```
    vt = Val(Mid(m_dang, 1, 2))
```

```
    dang(curshape) = Right("0" + CStr(vt), 2) + ","
```

```
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 10), 2) + ","
```

```
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 20), 2) + ","
```

```
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 1), 2) + ","
```

```
    show_brick
```

```
    bienthe = 2
```

```
End If
```

Case 2: ' *Biến đổi khối gạch 4, dạng 2 thành khối gạch 4 dạng 3*

```
vt = Val(Mid(m_dang, 1, 2))
```

```
If vt Mod 10 < 2 Or vt <= 9 Then
    r_bien = False
Else
    If Shape1(Val(vt - 10)).Visible = True Or _
        Shape1(Val(vt - 2)).Visible = True Then
        r_bien = False
    End If
End If

If r_bien = True Then
    hide_brick
    vt = Val(Mid(m_dang, 1, 2))
    dang(curshape) = Right("0" + CStr(vt), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 1), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 2), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 10), 2) + ","
    show_brick
    bienthe = 3
End If

Case 3: 1 Biến đổi khối gạch 4, dạng 3 thành khối gạch 4 dạng 4
vt = Val(Mid(m_dang, 1, 2))
If vt Mod 10 = 9 Or vt <= 19 Then
    r_bien = False
Else
    If Shape1(Val(vt - 20)).Visible = True Or _
        Shape1(Val(vt + 1)).Visible = True Then
        r_bien = False
    End If
```

```

End If
If r_bien = True Then
    hide_brick
    vt = Val(Mid(m_dang, 1, 2))
    dang(curshape) = Right("0" + CStr(vt), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 10), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt - 20), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 1), 2) + ","
    show_brick
    bienthe = 4
End If

```

Case 4: ' *Biến đổi khối gạch 4, dạng 4 thành khối gạch 4 dạng 1*

```

vt = Val(Mid(m_dang, 1, 2))
If vt Mod 10 >= 8 Or vt >= 90 Then
    r_bien = False
Else
    If Shape1(Val(vt + 2)).Visible = True Or _
        Shape1(Val(vt + 10)).Visible = True Then
        r_bien = False
    End If
End If
If r_bien = True Then
    hide_brick
    vt = Val(Mid(m_dang, 1, 2))
    dang(curshape) = Right("0" + CStr(vt), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 1), 2) + ","
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 2), 2) + ","

```

```

    dang(curshape) = dang(curshape) + Right("0" + CStr(vt + 10), 2) + ","
    show_brick
    bienthe = 1
End If
End Select

Case 4: ' Với khối gạch 5 thì không làm gì cả
End Select

End Sub

' Thủ tục được gọi khi người chơi nhấn phím mũi tên phải
' Kiểm tra và thực hiện việc dịch chuyển khối gạch sang phải
Sub moveright()
' Biến m_stopr xác định liệu khối gạch còn có thể dịch sang phải hay không
mstop_r = False ' mặc định xem như khối gạch có thể tiếp tục dịch sang phải
m_dang = dang(curshape) ' xác định khối gạch hiện hành
' Kiểm tra điều kiện dừng (không cho dịch sang phải)
Select Case curshape
    Case 0: ' Đối với khối gạch 1
        Select Case bienthe
            Case 1: ' Nếu khối gạch 1 đang ở dạng 1
                vt = Val(Mid(m_dang, 10, 2)) ' lấy vị trí viên gạch 4 trong khối
                If vt Mod 10 = 9 Then ' Nếu đang ở cột cuối cùng
                    mstop_r = True ' không cho dịch sang phải nữa
                Else
                    ' nếu ở bên phải viên gạch 4 có chứa gạch
                    If Shape1(Val(vt + 1)).Visible = True Then
                        mstop_r = True ' cũng không cho dịch khối sang phải
                    End If
                End If
            End Case
        End Select
    End Case
End Select

```


Case 2: ' Nếu khối gạch 1 đang ở dạng 2

For ii = 1 To 4 ' kiểm tra cả bốn viên gạch trong khối

vt = Val(Mid(m_dang, ii * 3 - 2, 2))

If vt Mod 10 = 9 Then ' Nếu là cột cuối cùng

mstop_r = True

Exit For

Else

If Shape1(Val(vt + 1)).Visible = True Then

mstop_r = True

Exit For

End If

End If

Next

End Select

Case 1: ' Đối với khối gạch 2

Select Case bienthe

Case 1: ' Nếu khối gạch 2 đang ở dạng 1

For ii = 3 To 4 ' kiểm tra viên gạch thứ 3 và thứ 4 trong khối

vt = Val(Mid(m_dang, ii * 3 - 2, 2))

If vt Mod 10 = 9 Then ' Nếu là cột cuối cùng

mstop_r = True

Exit For

Else

If Shape1(Val(vt + 1)).Visible = True Then

mstop_r = True

Exit For

End If

End If

Next

Case 2: ' Nếu khối gạch 2 đang ở dạng 2

For ii = 1 To 3 ' Kiểm tra ô bên phải ba viên gạch 1, 2 và 3 trong khối

 vt = Val(Mid(m_dang, ii * 3 - 2, 2))

 If vt Mod 10 = 9 Then ' Nếu viên gạch kiểm tra thuộc ô cuối cùng

 mstop_r = True

 Exit For

Else

 If Shape1(Val(vt + 1)).Visible = True Then

 mstop_r = True

 Exit For

End If

End If

Next

Case 3: ' Nếu khối gạch 2 đang ở dạng 3

vt = Val(Mid(m_dang, 1, 2)) ' Kiểm tra ô bên phải viên gạch đầu tiên

 If vt Mod 10 = 9 Then

 mstop_r = True

Else

 If Shape1(Val(vt + 1)).Visible = True Then

 mstop_r = True

 End If

End If

If mstop_r = False Then ' Kiểm tra ô bên phải viên gạch thứ 4

 vt = Val(Mid(m_dang, 10, 2))

 If vt Mod 10 = 9 Then ' Nếu là hàng cuối cùng

 mstop_r = True

Else

```

    If Shape1(Val(vt + 1)).Visible = True Then
        mstop_r = True
    End If
End If
End If
Case 4:    ' Nếu khối gạch thứ 2 đang ở dạng 4
For ii = 1 To 4        ' Kiểm tra ô bên phải viên gạch 1, 3 và 4
    If ii = 1 Or ii = 3 Or ii = 4 Then
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
        If vt Mod 10 = 9 Then ' Nếu là cột cuối cùng
            mstop_r = True
        Exit For
    Else
        If Shape1(Val(vt + 1)).Visible = True Then
            mstop_r = True
        Exit For
    End If
End If
End If
Next
End Select
Case 2:    ' Đối với khối gạch thứ 3
Select Case bienthe
Case 1:    ' Nếu khối gạch 3 đang ở dạng 1
For ii = 3 To 4        ' Kiểm tra hai viên gạch thứ 3 và thứ 4
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt Mod 10 = 9 Then ' Nếu ô chứa gạch là cột cuối cùng
        mstop_r = True

```

```

Exit For
Else
    If Shape1(Val(vt + 1)).Visible = True Then
        mstop_r = True
    Exit For
End If
End If
Next

Case 2:    ' Nếu khối gạch thứ 3 đang ở dạng 2
For ii = 1 To 3        ' Kiểm tra ba viên gạch 1, 2, 3
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt Mod 10 = 9 Then    ' Nếu ô chứa gạch thuộc cột cuối cùng
        mstop_r = True
    Exit For
Else
    If Shape1(Val(vt + 1)).Visible = True Then
        mstop_r = True
    Exit For
End If
End If
Next

Case 3:    ' Nếu khối gạch thứ 3 đang ở dạng 3
For ii = 1 To 4        ' kiểm tra ô bên phải viên gạch 1 và 4 của khối
    If ii = 1 Or ii = 4 Then
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
        If vt Mod 10 = 9 Then    ' Nếu ô chứa gạch thuộc cột cuối cùng
            mstop_r = True    ' không cho dịch sang phải nữa
        Exit For
    
```

```

Else
    If Shape1(Val(vt + 1)).Visible = True Then
        mstop_r = True
    Exit For
End If
End If
End If
Next

Case 4:      ' Nếu khối gạch thứ 3 đang ở dạng 4
For ii = 1 To 4      ' Kiểm tra ô bên phải viên gạch 1, 2 và 4 của khối
    If ii = 1 Or ii = 2 Or ii = 4 Then
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
        If vt Mod 10 = 9 Then ' Nếu ô thuộc cột cuối cùng
            mstop_r = True
        Exit For
    Else
        If Shape1(Val(vt + 1)).Visible = True Then
            mstop_r = True
        Exit For
    End If
End If
End If
Next
End Select

Case 3:      ' Đối với khối gạch thứ 4
Select Case bienthe
Case 1:      ' Nếu là khối gạch 4, dạng 1
    For ii = 3 To 4      ' Kiểm tra ô bên phải viên gạch 3 và 4 trong khối

```

```
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt Mod 10 = 9 Then      ' Nếu ô thuộc cột cuối cùng
    mstop_r = True
    Exit For
Else
    If Shape1(Val(vt + 1)).Visible = True Then
        mstop_r = True
        Exit For
    End If
End If
Next

Case 2:      ' Nếu là khối gạch 4, dạng 2
For ii = 1 To 3      ' Kiểm tra ô bên phải viên gạch 1, 2 và 3 trong khối
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt Mod 10 = 9 Then      ' Nếu ô thuộc cột cuối cùng
        mstop_r = True
        Exit For
    Else
        If Shape1(Val(vt + 1)).Visible = True Then
            mstop_r = True
            Exit For
        End If
    End If
Next

Case 3:      ' Nếu là khối gạch 4, dạng 3
For ii = 1 To 4
    If ii = 1 Or ii = 4 Then      ' Kiểm tra ô bên phải viên gạch 1 và 4
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
```

```
If vt Mod 10 = 9 Then      ' Nếu ô kiểm tra thuộc cột cuối cùng
    mstop_r = True
    Exit For
Else
    If Shape1(Val(vt + 1)).Visible = True Then
        mstop_r = True
        Exit For
    End If
End If
End If
Next

Case 4:      ' Nếu là khối gạch 4, dạng 4
For ii = 2 To 4      ' Kiểm tra ô bên phải viên gạch 2, 3 và 4 trong khối
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt Mod 10 = 9 Then      ' Nếu ô thuộc cột cuối cùng
        mstop_r = True
        Exit For
    Else
        If Shape1(Val(vt + 1)).Visible = True Then
            mstop_r = True
            Exit For
        End If
    End If
Next
End Select

Case 4:      ' Nếu là khối gạch 5 - Dạng 1
For ii = 2 To 4      ' Kiểm tra ô bên phải viên gạch 2 và 4 trong khối
    If ii = 2 Or ii = 4 Then
```

```
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt Mod 10 = 9 Then      ' Nếu ô thuộc cột cuối cùng
    mstop_r = True
    Exit For
Else
    If Shape1(Val(vt + 1)).Visible = True Then
        mstop_r = True
        Exit For
    End If
End If
End If
Next
End Select

' Dịch chuyển tất cả bốn viên gạch trong khối sang phải một cột
If mstop_r = False Then
    hide_brick
    vt = Val(Mid(m_dang, 1, 2)) + 1
    dang(curshape) = Right("0" + CStr(vt), 2) + ","
    vt = Val(Mid(m_dang, 4, 2)) + 1
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
    vt = Val(Mid(m_dang, 7, 2)) + 1
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
    vt = Val(Mid(m_dang, 10, 2)) + 1
    dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2)
    show_brick
End If
End Sub

' Thủ tục được gọi khi người chơi nhấn phím mũi tên trái
```


' Kiểm tra và thực hiện việc dịch chuyển khối gạch sang trái

' Biến **m_stopl** xác định liệu khối gạch còn có thể dịch sang trái hay không

' mstop_l = False ' mặc định xem như khối gạch có thể tiếp tục dịch sang trái

Sub moveleft()

mstop_l = False ' không cho dịch sang trái

m_dang = dang(curshape)

' Kiểm tra điều kiện đúng

vt = Val(Mid(m_dang, 1, 2))

Select Case curshape

Case 0: ' Đối với khối gạch 1

Select Case bienthe

Case 1: ' Nếu là khối gạch 1, dạng 1

' Kiểm tra ô chứa viên gạch đầu tiên trong khối

vt = Val(Mid(m_dang, 1, 2))

If vt Mod 10 = 0 Then ' Nếu ô kiểm tra thuộc cột đầu tiên

mstop_l = True ' Không cho dịch chuyển khối sang trái nữa

Else

' Nếu ở bên trái viên gạch 1 có chứa gạch

If Shape1(Val(vt - 1)).Visible = True Then

mstop_l = True ' cũng không cho dịch chuyển khối gạch sang trái

End If

End If

Case 2: ' Nếu là khối gạch 1, dạng 2

For ii = 1 To 4 ' Lần lượt kiểm tra ô bên trái của cả bốn viên gạch trong khối

vt = Val(Mid(m_dang, ii * 3 - 2, 2)) ' Vị trí viên gạch

If vt Mod 10 = 0 Then ' Nếu vị trí viên gạch thuộc cột đầu tiên

mstop_l = True ' Không cho dịch chuyển khối sang trái

Exit For

```

Else
    If Shape1(Val(vt - 1)).Visible = True Then ' Nếu ô bên trái có gạch
        mstop_1 = True          ' không cho dịch chuyển khối sang trái
    Exit For
End If
End If
Next
End Select

Case 1: ' Đối với khối gạch 2
    Select Case bienthe
    Case 1: ' Nếu là khối gạch 2, dạng 1
        ' Kiểm tra ô bên trái viên gạch 1 và 4 trong khối
        vt = Val(Mid(m_dang, 1, 2))          ' Lấy vị trí viên gạch 1
        If vt Mod 10 = 0 Then                ' Nếu thuộc cột đầu tiên
            mstop_1 = True                    ' không cho dịch khối sang trái
        Else
            ' Kiểm tra ô bên trái viên gạch 1 liệu có chứa gạch
            If Shape1(Val(vt - 1)).Visible = True Then
                mstop_1 = True
            End If
        End If
        If mstop_1 = False Then
            vt = Val(Mid(m_dang, 10, 2)) ' Lấy vị trí viên gạch thứ 1 trong khối
            If vt Mod 10 = 0 Then          ' Nếu thuộc cột đầu tiên
                mstop_1 = True
            Else
                ' Kiểm tra ô bên trái viên gạch 4 liệu có chứa gạch
                If Shape1(Val(vt - 1)).Visible = True Then

```

```
mstop_l = True
End If
End If
End If
Case 2: ' Nếu là khối gạch 2, dạng 2
' Kiểm tra các viên gạch 1, 3 và 4 trong khối
For ii = 1 To 4
If ii = 1 Or ii = 3 Or ii = 4 Then
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt Mod 10 = 0 Then
mstop_l = True
Exit For
Else
If Shape1(Val(vt - 1)).Visible = True Then
mstop_l = True
Exit For
End If
End If
End If
Next
Case 3: ' Nếu là khối gạch 2, dạng 3
' Kiểm tra các viên gạch 3 và 4 trong khối
For ii = 3 To 4
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt Mod 10 = 0 Then
mstop_l = True
Exit For
Else
```

```
    If Shape1(Val(vt - 1)).Visible = True Then
        mstop_l = True
    Exit For
End If
End If
Next

Case 4:      ' Nếu là khối gạch 2, dạng 4
' Kiểm tra các viên gạch 1, 2 và 3 trong khối
For ii = 1 To 3
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt Mod 10 = 0 Then
        mstop_l = True
    Exit For
Else
    If Shape1(Val(vt - 1)).Visible = True Then
        mstop_l = True
    Exit For
End If
End If
Next

End Select

Case 2:      ' Đối với khối gạch 3
Select Case bienthe
Case 1:      ' Nếu là khối gạch 3, dạng 1
' Kiểm tra viên gạch 1 và 4 trong khối
For ii = 1 To 4
    If ii = 1 Or ii = 4 Then
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
```

```
If vt Mod 10 = 0 Then
    mstop_l = True
    Exit For
Else
    If Shape1(Val(vt - 1)).Visible = True Then
        mstop_l = True
        Exit For
    End If
End If
End If
Next
```

Case 2: ' Nếu là khối gạch 3, dạng 2

' Kiểm tra các viên gạch 1, 2 và 4 trong khối

```
For ii = 1 To 4
    If ii = 1 Or ii = 2 Or ii = 4 Then
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
        If vt Mod 10 = 0 Then
            mstop_l = True
            Exit For
        Else
            If Shape1(Val(vt - 1)).Visible = True Then
                mstop_l = True
            Exit For
        End If
    End If
End If
Next
```

Case 3: ' Nếu là khối gạch 3, dạng 3

```
' Kiểm tra các viên gạch 3 và 4 trong khối
For ii = 3 To 4
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt Mod 10 = 0 Then
        mstop_l = True
        Exit For
    Else
        If Shape1(Val(vt - 1)).Visible = True Then
            mstop_l = True
            Exit For
        End If
    End If
Next ii
Next

Case 4:      ' Nếu là khối gạch 3, dạng 4
' Kiểm tra các viên gạch 1, 2 và 3 trong khối
For ii = 1 To 3
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt Mod 10 = 0 Then
        mstop_l = True
        Exit For
    Else
        If Shape1(Val(vt - 1)).Visible = True Then
            mstop_l = True
            Exit For
        End If
    End If
Next ii
Next
End Select
```

Case 3: ' Đối với khối gạch 4

Select Case bienthe

Case 1: ' Nếu là khối gạch 4, dạng 1

 ' Kiểm tra viên gạch 1 và 4 trong khối

For ii = 1 To 4

 If ii = 1 Or ii = 4 Then

 vt = Val(Mid(m_dang, ii * 3 - 2, 2))

 If vt Mod 10 = 0 Then

 mstop_l = True

 Exit For

Else

 If Shape1(Val(vt - 1)).Visible = True Then

 mstop_l = True

 Exit For

 End If

End If

End If

Next

Case 2: ' Nếu là khối gạch 4, dạng 2

 ' Kiểm tra các viên gạch 2, 3 và 4 trong khối

For ii = 2 To 4

 vt = Val(Mid(m_dang, ii * 3 - 2, 2))

 If vt Mod 10 = 0 Then

 mstop_l = True

 Exit For

Else

 If Shape1(Val(vt - 1)).Visible = True Then

 mstop_l = True

```

Exit For
End If
End If
Next
Case 3:    ' Nếu là khối gạch 4, dạng 4
' Kiểm tra viên gạch 3 và 4 trong khối
For ii = 3 To 4
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt Mod 10 = 0 Then
mstop_l = True
Exit For
Else
If Shape1(Val(vt - 1)).Visible = True Then
mstop_l = True
Exit For
End If
End If
Next
Case 4:    ' Nếu là khối gạch 4, dạng 4
' Kiểm tra các viên gạch 1, 2 và 3 trong khối
For ii = 1 To 3
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt Mod 10 = 0 Then
mstop_l = True
Exit For
Else
If Shape1(Val(vt - 1)).Visible = True Then
mstop_l = True

```



```

Exit For
End If
End If
Next
End Select
Case 4:  ' đối với khối gạch 5 (có duy nhất 1 dạng)
' Kiểm tra viên gạch 1 và 3 trong khối
For ii = 1 To 3
If ii = 1 Or ii = 3 Then
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt Mod 10 = 0 Then
mstop_l = True
Exit For
Else
If Shape1(Val(vt - 1)).Visible = True Then
mstop_l = True
Exit For
End If
End If
End If
Next
End Select
' Nếu khối gạch còn được phép dịch sang trái sau khi kiểm tra,
' tiến hành dịch khối gạch sang trái một cột
If mstop_l = False Then
hide_brick
vt = Val(Mid(m_dang, 1, 2)) - 1
dang(curshape) = Right("0" + CStr(vt), 2) + ","

```

```
vt = Val(Mid(m_dang, 4, 2)) - 1
dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
vt = Val(Mid(m_dang, 7, 2)) - 1
dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
vt = Val(Mid(m_dang, 10, 2)) - 1
dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2)
show_brick
```

End If

End Sub

' Thủ tục được gọi khi người chơi nhấn phím mũi tên xuống

' Kiểm tra và dịch chuyển khối gạch xuống một dòng ngay lập tức

' mà không đợi đến thời gian qui định mới thực hiện

Sub movedown()

mstop = False *' mặc định xem như khối gạch vẫn có thể dịch chuyển*
' xuống một dòng

m_dang = dang(curshape) *' Xác định khối gạch hiện hành*

' Kiểm tra điều kiện dừng

Select Case curshape

Case 0: *' Đối với khối gạch 1*

Select Case bienthe

Case 1: *' Nếu là khối gạch 1, dạng 1*

' Kiểm tra cả bốn viên gạch trong khối

For ii = 1 To 4

vt = Val(Mid(m_dang, ii * 3 - 2, 2))

If vt > 89 Then *' Nếu vị trí viên gạch thuộc hàng cuối cùng*

mstop = True *' thì dừng không cho di chuyển xuống nữa*

Exit For

Else

' Kiểm tra ô bên dưới viên gạch liệu có chứa gạch không

```

    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
        Exit For
    End If
End If
Next

Case 2:    ' Nếu là khối gạch 1, dạng 2
' Kiểm tra viên gạch thứ 4 trong khối
vt = Val(Mid(m_dang, 10, 2))
If vt > 89 Then    ' Nếu thuộc hàng cuối cùng
    mstop = True    ' không cho dịch chuyển khối xuống dưới nữa
Else
' Kiểm tra ô dưới viên gạch thứ 4, liệu có chứa gạch không
    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
    End If
End If
End Select

Case 1:    ' Đối với khối gạch 2
Select Case bienthe
Case 1:    ' Nếu khối gạch 2 có dạng 1
' Kiểm tra các viên gạch 1, 3 và 4
For ii = 1 To 4
    If ii = 1 Or ii = 3 Or ii = 4 Then
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
        If vt > 89 Then
            mstop = True
        Exit For
    End If
Next ii

```

```
Else
    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
    Exit For
End If
End If
End If
Next

Case 2:      ' Nếu khối gạch 2 có dạng 2
' Kiểm tra viên gạch 3 và 4
For ii = 3 To 4
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt > 89 Then
        mstop = True
    Exit For
Else
    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
    Exit For
End If
End If
Next

Case 3:      ' Nếu khối gạch 2 có dạng 3
' Kiểm tra viên gạch 1, 2 và 3 trong khối
For ii = 1 To 3
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt > 89 Then
        mstop = True
```

```
Exit For
Else
    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
    Exit For
End If
End If
Next

Case 4:      ' Nếu khối gạch 2 có dạng 4
' Kiểm tra viên gạch 1 và 4 trong khối
vt = Val(Mid(m_dang, 1, 2))
If vt > 89 Then
    mstop = True
Else
    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
    End If
End If
If mstop = False Then
    vt = Val(Mid(m_dang, 10, 2))
    If vt > 89 Then
        mstop = True
    Else
        If Shape1(Val(vt + 10)).Visible = True Then
            mstop = True
        End If
    End If
End If
```

End Select

Case 2: ' Đối với khối gạch 3

Select Case bienthe

Case 1: ' Nếu khối gạch 3 có dạng 1

 ' Kiểm tra các viên gạch 1, 2 và 4 trong khối

For ii = 1 To 4

 If ii = 1 Or ii = 2 Or ii = 4 Then

 vt = Val(Mid(m_dang, ii * 3 - 2, 2))

 If vt > 89 Then

 mstop = True

 Exit For

 Else

 If Shape1(Val(vt + 10)).Visible = True Then

 mstop = True

 Exit For

 End If

End If

End If

Next

Case 2: ' Nếu khối gạch 3 có dạng 2

 ' Kiểm tra viên gạch 3 và 4 trong khối

For ii = 3 To 4

 vt = Val(Mid(m_dang, ii * 3 - 2, 2))

 If vt > 89 Then

 mstop = True

 Exit For

Else

 If Shape1(Val(vt + 10)).Visible = True Then

```
mstop = True
Exit For
End If
End If
Next

Case 3:    ' Nếu khối gạch 3 có dạng 3
' Kiểm tra 3 viên gạch 1, 2 và 3 trong khối
For ii = 1 To 3
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt > 89 Then
        mstop = True
        Exit For
    Else
        If Shape1(Val(vt + 10)).Visible = True Then
            mstop = True
            Exit For
        End If
    End If
Next

Case 4:    ' Nếu khối gạch 3 có dạng 4
' Kiểm tra viên gạch 1 và 4 trong khối
For ii = 1 To 4
    If ii = 1 Or ii = 4 Then
        vt = Val(Mid(m_dang, ii * 3 - 2, 2))
        If vt > 89 Then    ' Nếu là hàng cuối cùng
            mstop = True
            Exit For
        Else
```

```
If Shape1(Val(vt + 10)).Visible = True Then
    mstop = True
    Exit For
End If
End If
End If
Next
End Select
Case 3:      ' Đối với khối gạch 4
Select Case bienthe
Case 1:      ' Nếu khối gạch 4 đang có dạng 1
' Kiểm tra các viên gạch 2, 3 và 4
For ii = 2 To 4
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt > 89 Then
        mstop = True
        Exit For
    Else
        If Shape1(Val(vt + 10)).Visible = True Then
            mstop = True
            Exit For
        End If
    End If
Next
Case 2:      ' Nếu khối gạch 4 có dạng 2
' Kiểm tra viên gạch 3 và 4 trong khối
For ii = 3 To 4
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
```



```

If vt > 89 Then
    mstop = True
    Exit For
Else
    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
        Exit For
    End If
End If
Next

```

Case 3: ' Nếu khối gạch 4 có dạng 3

' Kiểm tra các viên gạch 1, 2 và 3 trong khối

```

For ii = 1 To 3
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt > 89 Then
        mstop = True
        Exit For
    Else
        If Shape1(Val(vt + 10)).Visible = True Then
            mstop = True
            Exit For
        End If
    End If
Next

```

Case 4: ' Nếu khối gạch 4 có dạng 4

' Kiểm tra viên gạch 1 và 4 trong khối

```

For ii = 1 To 4
    If ii = 1 Or ii = 4 Then

```

```
vt = Val(Mid(m_dang, ii * 3 - 2, 2))
If vt > 89 Then
    mstop = True
    Exit For
Else
    If Shape1(Val(vt + 10)).Visible = True Then
        mstop = True
        Exit For
    End If
End If
End If
Next
End Select

Case 4: ' Đối với khối gạch 5 (chỉ có 1 dạng)
' Kiểm tra viên gạch 3 và 4 trong khối
For ii = 3 To 4
    vt = Val(Mid(m_dang, ii * 3 - 2, 2))
    If vt > 89 Then
        mstop = True
        Exit For
    Else
        If Shape1(Val(vt + 10)).Visible = True Then
            mstop = True
            Exit For
        End If
    End If
Next
End Select
```

' Nếu khối gạch vẫn còn được phép dịch chuyển xuống dưới sau khi kiểm tra

' Dịch chuyển khối gạch xuống một hàng

If mstop = False Then

hide_brick

vt = Val(Mid(m_dang, 1, 2)) + 10

dang(curshape) = CStr(vt) + ","

vt = Val(Mid(m_dang, 4, 2)) + 10

dang(curshape) = dang(curshape) + CStr(vt) + ","

vt = Val(Mid(m_dang, 7, 2)) + 10

dang(curshape) = dang(curshape) + CStr(vt) + ","

vt = Val(Mid(m_dang, 10, 2)) + 10

dang(curshape) = dang(curshape) + CStr(vt)

show_brick

Else

' Nếu khối gạch bị dừng lại, không được dịch chuyển xuống nữa

' Phát sinh một khối gạch mới ở hàng đầu tiên trong không gian chơi

Me.Enabled = False

dang(0) = "03,04,05,06"

dang(1) = "03,04,05,14"

dang(2) = "03,04,05,15"

dang(3) = "03,04,05,13"

dang(4) = "04,05,14,15"

' Gọi thủ tục testhangday, kiểm tra các hàng đầy để loại bỏ khối

' không gian chơi và cộng điểm cho người chơi

testhangday

curshape = Int(Rnd * 5)

' Mặc định khối gạch mới phát sinh luôn có dạng 1

bienthe = 1

show_brick

mstop = False

Me.Enabled = True

End If

End Sub

' Được gọi từ trong thủ tục movedown, dùng kiểm tra hàng có đầy gạch chưa

Sub testhangday()

sohangxoa = 0 *' biến đếm số hàng được xóa cùng một lúc*

For i = 0 To 9 *' Duyệt từng hàng trong không gian chơi (10 hàng)*

mday = True *' mặc định xem hàng đã đầy gạch*

For j = 0 To 9 *' Duyệt từng ô trên một hàng (một ô)*

' Chỉ cần ít nhất một ô trong hàng chưa có gạch

If Shape1(i * 10 + j).Visible = False Then

mday = False *' Xem như hàng chưa đầy*

Exit For *' mà không cần kiểm tra các ô còn lại trong hàng*

End If

Next

If mday = True Then *' nếu hàng vừa kiểm tra đầy gạch*

sohangxoa = sohangxoa + 1 *' tăng biến lưu số hàng được xóa lên 1*
' tiến hành xóa hàng đầy

For j = 0 To 9 *' bằng cách ẩn tất cả các ô trên hàng*

Shape1(i * 10 + j).Visible = False

Next

' Dịch chuyển các hàng phía trên hàng vừa bị xóa xuống một hàng

For j = i * 10 - 1 To 0 Step -1

Shape1(j + 10).Visible = Shape1(j).Visible

Next

' Xóa trống hàng đầu tiên trong không gian chơi

For j = 0 To 9

Shape1(j).Visible = False

Next

End If

' Tiếp tục duyệt qua hàng kể cho đến khi đủ 10 hàng

Next

' Tính điểm dựa trên giá trị biến lưu số hàng được xóa cùng lúc

Select Case sohangxoa

Case 1: *' Nếu chỉ xóa được 1 hàng, cộng 10*

diemso = diemso + 10

Case 2: *' Nếu xóa được 2 hàng, cộng 30*

diemso = diemso + 30

Case 3: *' Nếu xóa được 3 hàng, cộng 50*

diemso = diemso + 50

Case 4: *' nếu xóa được 4 hàng, cộng 80*

diemso = diemso + 80

End Select

' Hiển thị điểm số của người chơi đến thời điểm hiện tại

Label2 = diemso

' Tính điểm thưởng cho người chơi

' Nếu người chơi kiếm được trên 2000 điểm trong khoảng thời gian

' ít hơn 15 phút thì được thưởng 300 điểm

If thoigian <= 900 And diemso - diembd >= 2000 Then

diemso = diemso + 300

thoigian = 0 *' đặt lại mốc thời gian để tính thưởng cho lần kế tiếp*

diembd = diemso

Else

' Nếu người chơi kiếm được trên 1000 điểm trong khoảng thời gian

' ít hơn 10 phút thì được thưởng 200

If thoigian <= 600 And diemso - diembd >= 1000 Then

```
diemso = diemso + 200
thoigian = 0
diembd = diemso
Else
    ' Nếu người chơi kiếm được trên 200 điểm trong khoảng thời gian
    ' ít hơn 3 phút thì được thưởng 100
    If thoigian <= 180 And diemso - diembd >= 200 Then
        diemso = diemso + 100
        thoigian = 0
        diembd = diemso
    End If
End If
End If
End If
' Định lại tốc độ tự dịch chuyển khối gạch theo điểm số hiện có của người chơi
If diemso > 5000 Then      ' điểm càng cao tốc độ dịch chuyển càng nhanh
    Timer1.Interval = 200
ElseIf diemso > 3000 Then
    Timer1.Interval = 300
ElseIf diemso > 2000 Then
    Timer1.Interval = 400
ElseIf diemso > 1000 Then
    Timer1.Interval = 500
End If
End Sub
' Thủ tục hiển thị vị trí mới của bốn viên gạch trong khối gạch sau khi dịch chuyển
Sub show_brick()
    vt = Mid(dang(curshape), 1, 2)
    Shape1(Val(vt)).Visible = True
```

```
vt = Mid(dang(curshape), 4, 2)
Shape1(Val(vt)).Visible = True
vt = Mid(dang(curshape), 7, 2)
Shape1(Val(vt)).Visible = True
vt = Mid(dang(curshape), 10, 2)
Shape1(Val(vt)).Visible = True
```

End Sub

’ Thủ tục ẩn vị trí của bốn viên gạch trong khối gạch trước khi dịch chuyển

Sub hide_brick()

```
vt = Mid(dang(curshape), 1, 2)
Shape1(Val(vt)).Visible = False
vt = Mid(dang(curshape), 4, 2)
Shape1(Val(vt)).Visible = False
vt = Mid(dang(curshape), 7, 2)
Shape1(Val(vt)).Visible = False
vt = Mid(dang(curshape), 10, 2)
Shape1(Val(vt)).Visible = False
```

End Sub

’ Thủ tục tự động được gọi sau một khoảng thời gian xác định

Private Sub Timer1_Timer()

’ Định lại thời gian gọi thủ tục cũng có nghĩa định lại thời gian

’ tự dịch chuyển của khối gạch

```
If diemso > 5000 Then
    thoigian = thoigian + 0.2
Elseif diemso > 3000 Then
    thoigian = thoigian + 0.3
Elseif diemso > 2000 Then
    thoigian = thoigian + 0.4
```

```

Elseif diemso > 1000 Then
    thoigian = thoigian + 0.5
Else
    thoigian = thoigian + 1
End If
' Gọi thủ tục movedown để kiểm tra và dịch chuyển khối gạch
movedown
End Sub

```

IV. Phân tích chương trình

1. Ý nghĩa các biến toàn cục đã khai báo

- Mảng **dang**: lưu năm khối gạch dạng 1 và vị trí bắt đầu xuất hiện
- Biến **curshape**: xác định khối gạch đang di chuyển
- Biến **bienthe**: xác định dạng hiện hành của khối gạch đang di chuyển
- Biến **mstop**: xác định trạng thái khối gạch là đang tiếp tục di chuyển hay dừng
- Biến **diemso**: lưu kết quả điểm hiện hành của lượt chơi
- Biến **thoigian**: đếm số giây trôi qua kể từ lần cuối cùng người chơi được thưởng điểm
- Biến **diembd**: lưu điểm số của người chơi ở lần thưởng điểm cuối cùng

2. Cách viết một câu lệnh dài:

Thay vì viết liên tục trên một dòng, bạn có thể viết thành nhiều dòng, nhưng cần thêm dấu _ ở cuối mỗi dòng trước khi xuống, trừ dòng cuối cùng.

Ví dụ: Thay vì viết liên tục trên một dòng câu lệnh IF sau

```

If Shape1(Val(vt + 10)).Visible = True Or Shape1(Val(vt + 20)).
Visible = True Or Shape1(Val(vt + 30)).Visible = True Then

```



```

r_bien = False           ' Không cho biến hình
End If

```

Bạn có thể viết:

```

If Shape1(Val(vt + 10)).Visible = True Or _
   Shape1(Val(vt + 20)).Visible = True Or _
   Shape1(Val(vt + 30)).Visible = True Then
    r_bien = False       ' Không cho biến hình
End If

```

Rõ ràng là cách viết 2 sẽ rõ ràng và sáng sủa hơn nhiều.

3. Cách lấy vị trí một viên gạch trong khối

Chúng ta đã biết, biến `m_dang` lưu vị trí bốn viên gạch trong một khối. Chẳng hạn:

```
m_dang = "03,04,05,13"
```

Trong đó, vị trí viên gạch 1 là hai ký tự đầu tiên trong chuỗi

Vị trí viên gạch 2 là ký tự 4 và 5 trong chuỗi

Vị trí viên gạch 3 là ký tự 7 và 8 trong chuỗi

Vị trí viên gạch 4 là ký tự 10 và 11 trong chuỗi

Do đó, có thể rút ra quy luật xác định vị trí viên gạch chung như sau:

Vị trí viên gạch i : là ký tự $i*3-2$ và ký tự $i*3-1$ trong chuỗi

Nói cách khác, chúng ta sẽ có công thức lấy vị trí viên gạch i :

```
vt = Val(Mid(m_dang, i * 3 - 2, 2))
```

Với i có giá trị từ 1 đến 4. Vì trong các khối gạch đã định nghĩa, không có khối gạch nào có nhiều hơn bốn viên gạch.

4. Giải thích và minh họa kỹ thuật xóa hàng đầy gạch

Giả sử sau khi khối gạch ngừng, không gian chơi có hai hàng đầy gạch là hàng dưới cùng (hàng thứ 10) và hàng thứ 8. Giờ đây, chúng ta sẽ xem cách chương trình tiến hành xóa các hàng đầy gạch.

Bước 1. Đầu tiên, chương trình tìm hàng đầy gạch bằng cách duyệt từ hàng trên cùng (hàng thứ nhất) trở xuống. Do đó sẽ gặp hàng đầy gạch đầu tiên là hàng thứ 8 (7 hàng trên hàng 8 chưa đầy gạch).

Bước 2. Xóa gạch trong hàng thứ 8. Kết quả hàng thứ 8 là hàng trống.

Bước 3. Chuyển gạch trong hàng 7 xuống hàng 8.

Bước 4. Chuyển gạch trong hàng 6 xuống hàng 7. Tiếp tục tương tự, cho đến khi chuyển gạch của hàng trên cùng xuống hàng thứ 2.

Bước 5. Rõ ràng, đến lúc này trên nguyên tắc hàng đầu tiên phải trống, vì không có gạch nơi khác chuyển tới, còn bản thân gạch trong hàng đã chuyển xuống hàng 2 nên cần xóa trống hàng đầu tiên.

Bước 6. Tiếp tục duyệt hàng thứ 9.

Bước 7. Khi duyệt tới hàng cuối cùng, lại là một hàng đầy gạch. Vì thế nên tiến hành xóa gạch trong hàng 10.

Bước 8. Chuyển gạch từ hàng 9 xuống 10, hàng 8 xuống 9, ..., hàng 1 xuống hàng 2.

Bước 9. Xóa trống hàng 1.

Chú ý: Thuật toán sẽ cho kết quả sai nếu bạn tiến hành duyệt các hàng theo thứ tự từ dưới lên. Hoặc xóa các hàng theo thứ tự từ trên xuống.

Nói cách khác, **không được** viết câu lệnh:

For i = 9 To 0 step -1 thay cho For i = 0 To 9 khi duyệt các hàng và

For j=0 to i*10-1 thay cho For j = i * 10 - 1 To 0 Step -1
khi dịch chuyển hàng

V. Bài tập thực hành

Bài 1. Trò chơi bị giới hạn trong không gian chơi 10×10 là hơi chật hẹp để bạn xoay xở. Hãy phát triển chương trình để không gian chơi mở rộng thành $16 \text{ hàng} \times 10 \text{ cột}$.

Bài 2. Xem kỹ lại các câu lệnh tính điểm thưởng trong thủ tục *Testhangday*. Xong hãy chỉ ra điểm bất hợp lý của cấu trúc IF đã sử dụng. Đề nghị một phương án khác để thực hiện việc tính điểm thưởng công bằng hơn.

MÁCH NƯỚC

- ✓ Hãy suy nghĩ theo hướng khi nào người chơi mới có khả năng nhận được 300 điểm thưởng. Liệu người chơi có thể nhận được 300 điểm thưởng khi dành được 2000 điểm trong thời gian 14 phút hay không?
-

Bài 3. Hãy suy nghĩ một cách viết khác để có thể rút ngắn được đoạn mã lệnh trong các thủ tục *moveleft*, *moveright*, *movedown* và *bienhinh*.

MÁCH NƯỚC

Thay vì xét từng khối và dạng gạch trước, sau đó mới xác định những viên gạch cần kiểm tra theo từng trường hợp riêng lẻ, bạn có thể tập hợp các trường hợp giống nhau lại để kiểm tra cùng một lúc. Cụ thể là:

- ✓ Kiểm tra viên gạch 1: Nếu khối gạch đang di chuyển thuộc khối 1 dạng 1, khối 1 dạng 2, khối 2 dạng 1, khối 2 dạng 2, khối 2 dạng 4, khối 3 dạng 1, khối 3 dạng 2, khối 3 dạng 4, khối 4 dạng 1 hoặc khối 4 dạng 4, khối 5.
- ✓ Kiểm tra viên gạch 2: Nếu khối gạch đang di chuyển thuộc khối 1 dạng 2, khối 2 dạng 4, khối 3 dạng 2, khối 3 dạng 4, khối 4 dạng 2, khối 4 dạng 4.
- ✓ Kiểm tra viên gạch 3: Nếu khối gạch đang di chuyển thuộc khối 1 dạng 2, khối 2 dạng 2, khối 2 dạng 3, khối 2 dạng 4, khối 3 dạng 3, khối 3 dạng 4, khối 4 dạng 2, khối 4 dạng 3, khối 4 dạng 4 hoặc khối 5.
- ✓ Kiểm tra viên gạch 4: Nếu khối gạch đang di chuyển thuộc khối 1 dạng 2, khối 2 dạng 1, khối 2 dạng 2, khối 2 dạng 3, khối 3 dạng 1, khối 3 dạng 2, khối 3 dạng 3, khối 4 dạng 1, khối 4 dạng 2, khối 4 dạng 3.

Nội dung thủ tục **Moveleft** sau là kết quả minh họa cho cách viết này:

Sub moveleft()

```
mstop_1 = False      ' Biến xác định dừng dịch khối sang trái hay không
m_dang = dang(curshape)
' Kiểm tra điều kiện dừng
```

‘Kiểm tra viên gạch 1 trong khối gạch đang di chuyển

```
If (bienthe = 1) Or (curshape = 5) Or _  
    (bienthe = 2 And curshape <> 3) Or _  
    (bienthe = 4 And curshape <> 1) Then  
    vt = Val(Mid(m_dang, 1, 2))  
    If vt Mod 10 = 0 Then  
        mstop_l = True  
    Else  
        If Shape1(Val(vt - 1)).Visible = True Then  
            mstop_l = True  
        End If  
    End If  
End If
```

‘Kiểm tra viên gạch 2 trong khối gạch đang di chuyển

```
If ((bienthe = 2 And (curshape = 0 Or curshape = 2 Or curshape = 3)) Or _  
    (bienthe = 4 And (curshape = 1 Or curshape = 2 Or curshape = 3))) And _  
    m_stop_l = False Then  
    vt = Val(Mid(m_dang, 4, 2))  
    If vt Mod 10 = 0 Then  
        mstop_l = True  
    Else  
        If Shape1(Val(vt - 1)).Visible = True Then  
            mstop_l = True  
        End If  
    End If  
End If
```

‘Kiểm tra viên gạch 3 trong khối gạch đang di chuyển

```
If ((bienthe = 2 And curshape <> 2) Or _
```

```
(bienthe = 3 And curshape <> 0) Or _  
(bienthe = 4 And curshape = 1)) And m_stop_l = False Then  
vt = Val(Mid(m_dang, 7, 2))  
If vt Mod 10 = 0 Then  
    mstop_l = True  
Else  
    If Shape1(Val(vt - 1)).Visible = True Then  
        mstop_l = True  
    End If  
End If  
End If  
'Kiểm tra viên gạch 4 trong khối gạch đang di chuyển  
If (bienthe = 2 Or _  
    (bienthe = 3 And curshape <> 0) Or _  
    (curshape = 0 And bienthe = 2)) And _  
    m_stop_l = False Then  
vt = Val(Mid(m_dang, 10, 2))  
If vt Mod 10 = 0 Then ' Neu la cot dau tien  
    mstop_l = True  
Else  
    If Shape1(Val(vt - 1)).Visible = True Then  
        mstop_l = True  
    End If  
End If  
End If  
' Dịch chuyển khối gạch sang trái một cột  
If mstop_l = False Then  
    hide_brick
```

```
vt = Val(Mid(m_dang, 1, 2)) - 1
dang(curshape) = Right("0" + CStr(vt), 2) + ","
vt = Val(Mid(m_dang, 4, 2)) - 1
dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
vt = Val(Mid(m_dang, 7, 2)) - 1
dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2) + ","
vt = Val(Mid(m_dang, 10, 2)) - 1
dang(curshape) = dang(curshape) + Right("0" + CStr(vt), 2)
show_brick
End If
End Sub
```

BÀI 4.

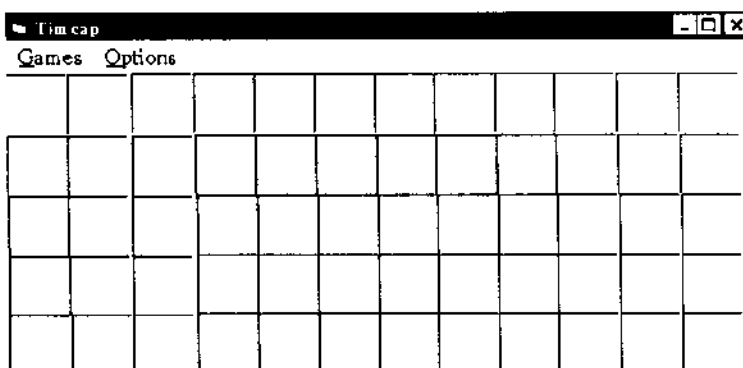
TRÒ CHƠI TÌM CẶP

I. Mô tả trò chơi

Trò chơi tìm cặp là trò chơi lật ảnh để tìm cặp với số lần lật càng ít càng tốt. Dùng các phím mũi tên **←,→,↑,↓** để di chuyển con trỏ qua lại các ô. Nhấn phím **Page Down** để lật ảnh.

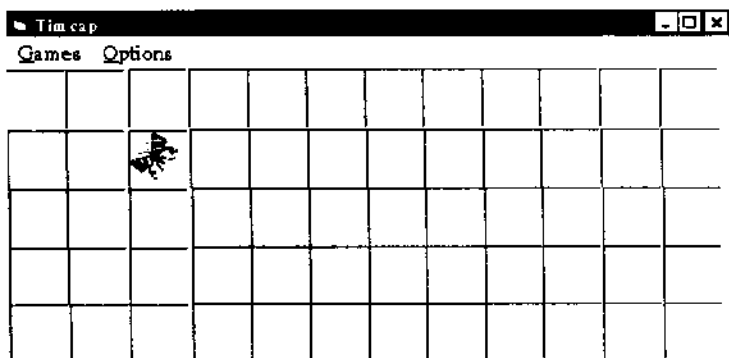
Trò chơi chia làm 7 mức từ thấp đến cao. Số ô trong không gian chơi thay đổi theo từng mức. Cụ thể là:

- Mức 1 : 4 ô, phân thành 2 hàng 2 cột
 - Mức 2 : 6 ô, phân thành 3 hàng 2 cột
 - Mức 3 : 12 ô, phân thành 3 hàng 4 cột
 - Mức 4 : 24 ô, phân thành 4 hàng 6 cột
 - Mức 5 : 32 ô, phân thành 4 hàng 8 cột
 - Mức 6 : 50 ô, phân thành 5 hàng 10 cột
 - Mức 7 : 60 ô, phân thành 5 hàng 12 cột
- ♦ Khi trò chơi bắt đầu, mỗi ô sẽ được gán cho hình xác định, nhưng không hiển thị ra cho người chơi thấy.
 - ♦ Dù ở mức độ nào, người chơi đều có thể tìm thấy trong không gian chơi 2 ô chứa cùng một hình giống nhau.



Hình 4.1 Không gian chơi của trò chơi ứng với mức 7.

- ◆ Khi người chơi nhấp trái hay nhấn phím Page Down trên một ô, hình trong ô đó sẽ hiển thị, còn gọi là lật hình (Hình 4.2). Lúc đó, người chơi cần xác định ô có chứa hình giống hình vừa lật để lật nó.



Hình 4.2 Chọn một ô khác có chứa hình giống như ô đã lật.

- ◆ Nếu hai ô được lật chứa hình giống nhau thì chúng sẽ được loại khỏi không gian chơi.

- ♦ Ngược lại, xem như người chơi lật không thành công, 2 hình vừa lật sẽ được úp trở lại (không hiển thị) và chờ đợi lần lật kế tiếp của người chơi.
- ♦ Người chơi sẽ hoàn tất mức chơi khi loại bỏ thành công tất cả các ô chứa hình khối không gian chơi. Khi đó, người chơi có thể chuyển đến mức chơi cao hơn để thử sức.

Để làm phong phú cách chơi, người chơi có thể chọn lựa một trong hai hình thức chơi:

1. *Chơi tính điểm:* Người chơi có thể lựa chọn bất kỳ 1 trong 7 mức độ chơi. Mỗi mức độ chơi khi hoàn tất sẽ được một thang điểm tối đa xác định. Cụ thể là:

- Mức 1 : 10 điểm
- Mức 2 : 20 điểm
- Mức 3 : 30 điểm
- Mức 4 : 40 điểm
- Mức 5 : 60 điểm
- Mức 6 : 80 điểm
- Mức 7 : 100 điểm

Tuy nhiên, người chơi chỉ có thể đạt được điểm tối đa chỉ khi hoàn tất mức độ chơi mà không có lần lật nào thất bại. Ngược lại, mỗi lần lật không thành công, người chơi sẽ bị trừ đi 1 điểm. Hình thức chơi này không khống chế thời gian chơi.

2. *Chơi theo thời gian:* Người chơi phải hoàn tất mỗi mức độ chơi trong khoảng thời gian quy định. Khi đó, người chơi mới có thể lên được mức cao hơn. Nói cách khác, người chơi phải lần lượt chơi từ mức thấp đến cao. Để lên được mức cao hơn, người chơi phải ghép được tất cả các hình trước khi bom phát nổ.

Thời gian chơi được quy định cụ thể cho mỗi mức như sau:

- Mức 1 : 20 giây
- Mức 2 : 30 giây
- Mức 3 : 1 phút
- Mức 4 : 2 phút
- Mức 5 : 3 phút
- Mức 6 : 4 phút
- Mức 7 : 5 phút

Ngoài ra, để tăng độ khó cho trò chơi, những hình sử dụng trong trò chơi được chọn ngẫu nhiên trong tập hình khá phong phú và đa dạng. Tập hình được pha trộn giữa hai nhóm hình khác nhau:

- ♦ *Nhóm 1:* Các hình hoàn toàn khác nhau, được xem như nhóm hình dễ chơi, vì người chơi có thể dễ nhớ và phân biệt các hình với nhau trong quá trình tìm cặp.

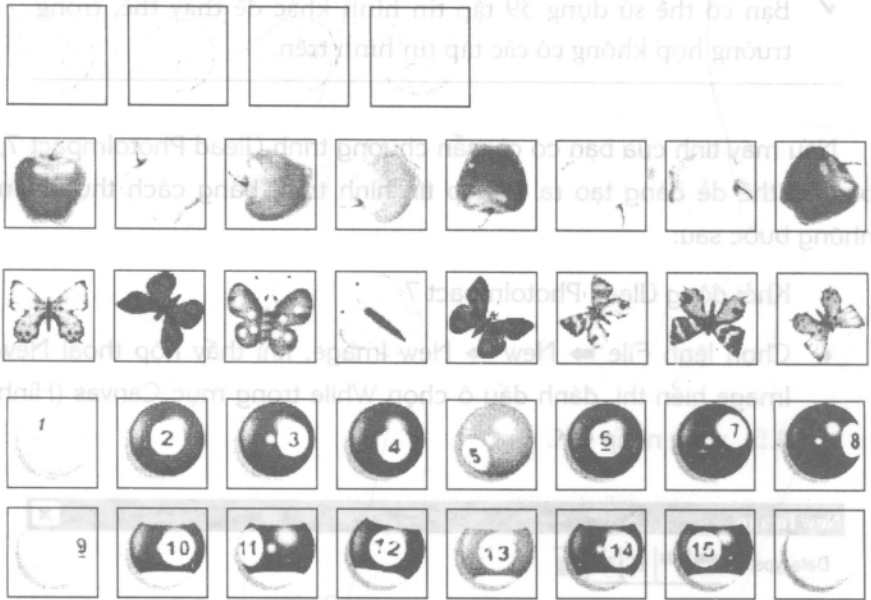
Ví dụ: Gà con, cá, bướm, quả táo, đồng tiền, nút áo, quả bóng (Hình 4.3).



Hình 4.3 Nhóm hình dễ phân biệt.

- ♦ *Nhóm 2:* Các hình cùng chủng loại, nhưng khác nhau về góc xoay, màu sắc, hình dạng. Đây là nhóm hình khó chơi hơn, đòi hỏi người chơi không chỉ cần một trí nhớ tốt mà còn có khả năng nhận xét, phán đoán nhanh nhạy.

Ví dụ: Nhóm hình các quả bóng khác nhau, con bướm khác nhau, quả táo khác nhau và đồng tiền khác nhau (Hình 4.4).



Hình 4.4 Những nhóm hình khó phân biệt.

Nhận xét: Chương trình không tách biệt rõ ràng hai nhóm Hình 1 và 2 mà trộn lẫn chúng với nhau trong quá trình chọn hình.

II. Chuẩn bị các tập tin hình *.bmp

Chương trình sử dụng một tập hợp hình của bạn gồm tất cả 39 hình (1 gà con, 1 cá, 1 nút áo, 4 đồng tiền, 8 con bướm, 8 quả táo và 16 trái bóng khác nhau).

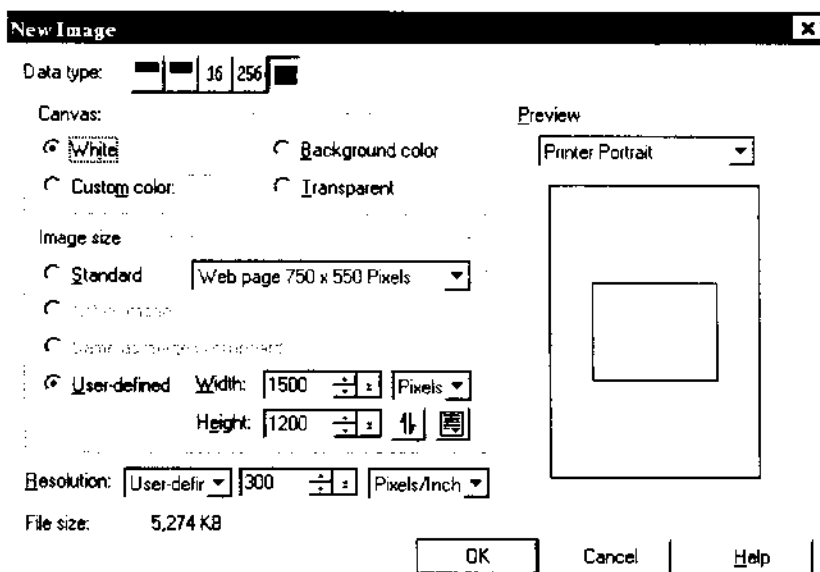
Mỗi hình ảnh cần được lưu trong một tập tin dạng bitmap *.bmp riêng biệt (gacon.bmp, ca.bmp, nut.bmp, tien1.bmp, tien2.bmp, tien3.bmp, tien4.bmp, buom1.bmp, ..., buom8.bmp, tao1.bmp, ..., tao8.bmp, bong1.bmp, ..., bong16.bmp).

GHI CHÚ

- ✓ Bạn có thể sử dụng 39 tập tin hình khác để thay thế, trong trường hợp không có các tập tin hình trên.

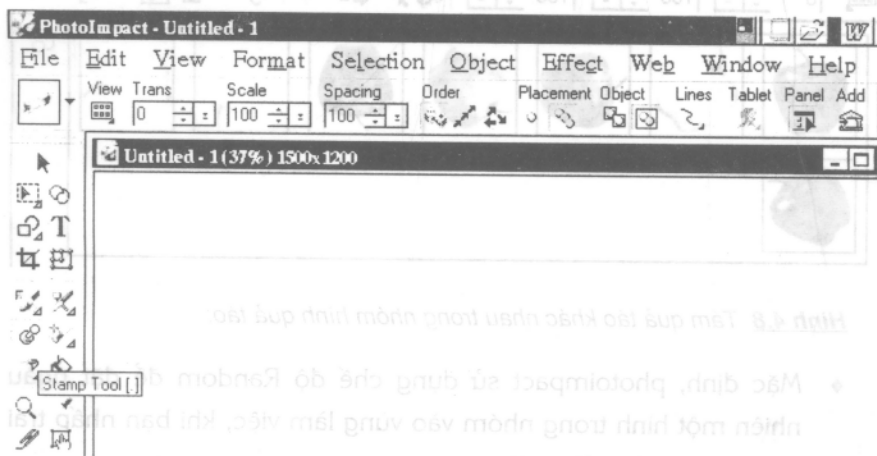
Nếu máy tính của bạn có cài sẵn chương trình Ulead PhotoImpact 7, bạn có thể dễ dàng tạo ra 39 tập tin hình trên, bằng cách thực hiện những bước sau:

- ♦ Khởi động Ulead PhotoImpact 7
- ♦ Chọn lệnh File ➡ New ➡ New Image, khi thấy hộp thoại New Image hiển thị, đánh dấu ô chọn White trong mục Canvas (Hình 4.5), xong nhấp OK.



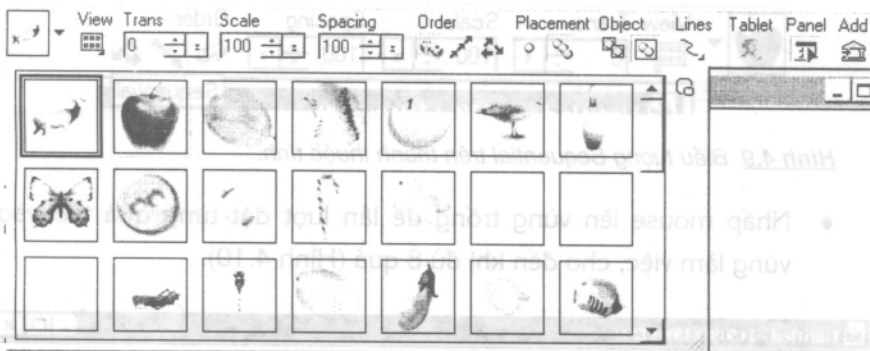
Hình 4.5 Hộp thoại New Image.

- ◆ Nhấp công cụ Stamp trên Toolbox (Hình 4.6).



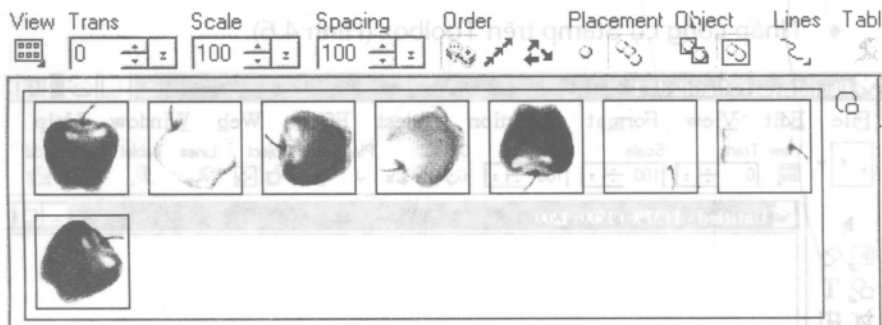
Hình 4.6 Công cụ Stamp trên Toolbox.

- ◆ Nhấp vào hình con nai trên thanh thuộc tính, bạn sẽ được danh sách các nhóm như Hình 4.7.



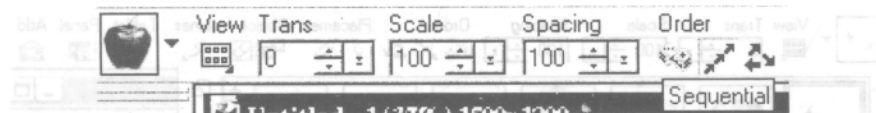
Hình 4.7 Thanh thuộc tính và các nhóm hình trong công cụ Stamp.

- ◆ Chọn nhóm hình quả táo. Sau đó, nhấp vào mục View trên thanh thuộc tính, bạn sẽ thấy nhóm hình này có tất cả 8 quả táo khác nhau (Hình 4.8).



Hình 4.8 Tám quả táo khác nhau trong nhóm hình quả táo.

- ◆ Mặc định, photoimpact sử dụng chế độ Random để đặt ngẫu nhiên một hình trong nhóm vào vùng làm việc, khi bạn nhấp trái mouse trên vùng làm việc.
- ◆ Để đặt đủ 8 quả táo lên vùng làm việc theo đúng thứ tự như bạn nhìn thấy trong Hình 4.8, nhấp biểu tượng **Sequential** trên thanh thuộc tính (Hình 4.9).



Hình 4.9 Biểu tượng Sequential trên thanh thuộc tính.

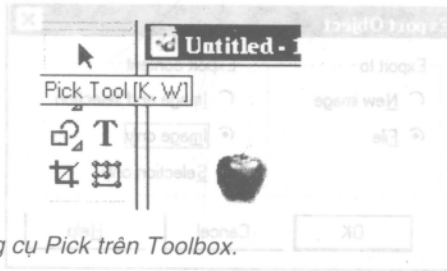
- ◆ Nhấp mouse lên vùng trống để lần lượt đặt từng quả táo vào vùng làm việc, cho đến khi đủ 8 quả (Hình 4.10).



Hình 4.10 Các quả táo đã được đặt trên vùng làm việc.

- ♦ Xuất quả táo đầu tiên ra tập tin hình *Tao1.bmp*.

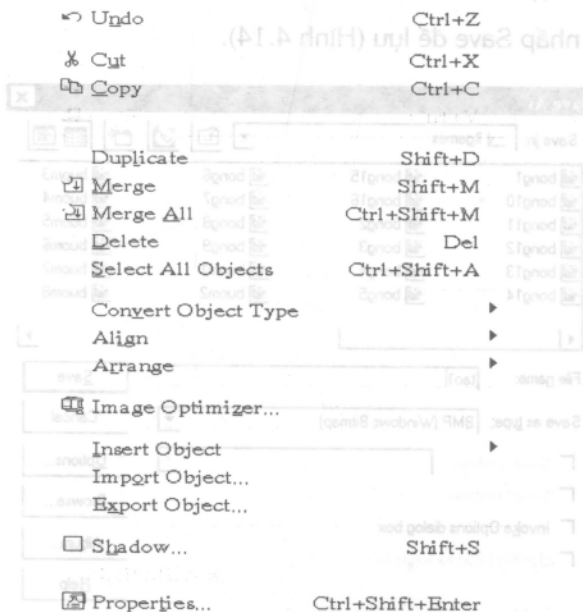
Chọn công cụ Pick trên Toolbox (Hình 4.11).



Hình 4.11 Công cụ Pick trên Toolbox.

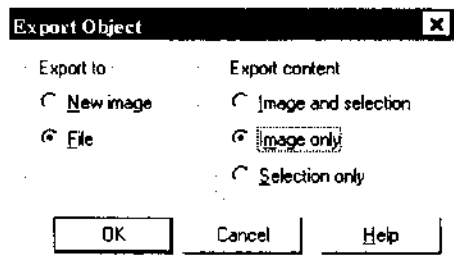
Nhấp nút trái mouse lên quả táo thứ nhất để chọn nó. Khi quả táo được chọn, bạn sẽ thấy đường viền đứt quãng bao quanh nó.

Sau đó, nhấp nút phải mouse lên quả táo đã chọn và chọn lệnh Export Object trên menu nhanh (Hình 4.12) để gọi hộp thoại Export Object.



Hình 4.12 Menu nhanh khi nhấp nút phải mouse lên quả táo đã chọn.

Trong mục Export to, chọn File. Trong mục Export content, chọn Image only (Hình 4.13). Xong nhấn OK.



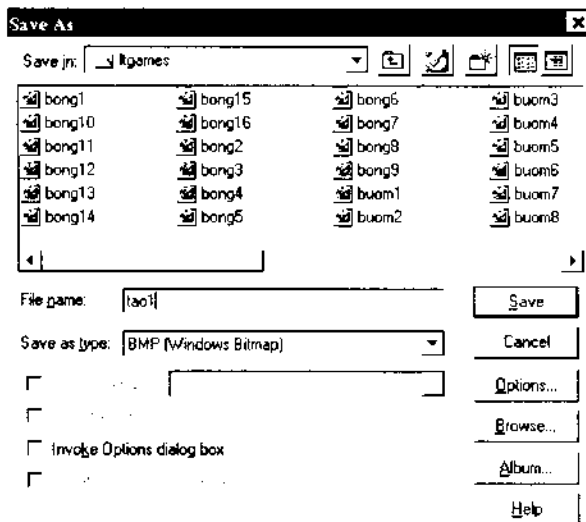
Hình 4.13 Hộp thoại Export Object.

Để lưu riêng hình quả táo được chọn thành tập tin Tao1.bmp, trong hộp thoại Save As, đặt:

Filename = tao1

Save as Type = BMP (Windows Bitmap).

Xong nhấn Save để lưu (Hình 4.14).



Hình 4.14 Hộp thoại Save As.

- ♦ Thực hiện tương tự với các quả táo còn lại để tạo những tập tin tao1.bmp đến tao8.bmp.
- ♦ Xong lại chọn nhóm hình khác và lặp lại các bước trên để tạo những tập tin hình còn lại.

III. Các thuật toán xử lý

1. Chọn hình sử dụng trong tập hợp hình có sẵn

- ♦ Như đã đề cập ở trên, tập hợp hình có sẵn gồm tất cả 39 hình. Do số lượng hình ít và để tiện cho việc biên dịch và sao chép chương trình về sau, bạn có thể đặt tất cả các hình trong một mảng điều khiển Image có tên là *HINH*, có chỉ số từ 0 đến 38.
- ♦ Sau khi đặt các hình vào những điều khiển Image, bạn có thể chạy chương trình mà không cần đến sự hiện diện của các tập tin hình trong thư mục chương trình nữa. Điều này sẽ tiện lợi hơn việc tham chiếu đến các tập tin hình đặt ở ngoài bằng lệnh LoadPicture. Vì nếu chỉ cần xóa mất một tập tin hình, chương trình của bạn sẽ báo lỗi và có thể không chạy được.
- ♦ Tuy nhiên, không phải lúc nào bạn cũng sử dụng hết 39 hình trong một lần chơi. Cụ thể ở mức độ 1, không gian chơi chỉ có 4 ô, nên chỉ cần sử dụng hai hình. Tương tự, số hình sử dụng ở mức độ 2 là 3 hình, mức độ 3 là 6 hình, mức độ 4 là 12 hình, mức độ 5 là 16 hình, mức độ 6 là 25 hình và mức độ 7 là 30 hình. Để thấy, ngay cả mức độ cao nhất, bạn cũng không sử dụng 1 lượt hết 39 hình.
- ♦ Nói cách khác tùy theo mức độ chơi, mà bạn phải thực hiện chọn:

- 2 hình trong số 39 hình
- 3 hình trong số 39 hình
- 6 hình trong số 39 hình
- 12 hình trong số 39 hình
- 16 hình trong số 39 hình
- 25 hình trong số 39 hình
- 30 hình trong số 39 hình

Việc chọn hình phải luôn đảm bảo:

- ♦ Các hình được chọn là không giống nhau, nói cách khác mỗi hình được chọn chỉ hiển thị trong 2 ô và chỉ 2 ô mà thôi.
- ♦ Hình nào trong tập hợp cũng sẽ được chọn dù ở mức độ chơi nào, sau một số lần chọn.
- ♦ Do vậy, bạn cần một giải thuật chọn hình chung đúng cho cả 7 trường hợp trên

Bước 1. Xác định tổng số hình cần chọn trong tập hợp.

Bước 2. Kiểm tra xem đã chọn đủ số hình yêu cầu chưa. Nếu rồi kết thúc quá trình chọn. Nếu chưa thực hiện bước 3.

Bước 3. Chọn ngẫu nhiên một trong số 39 hình thuộc tập hợp.

Bước 4. Kiểm tra xem hình vừa chọn đã được chọn trong các lần chọn trước hay chưa. Nếu rồi lặp lại bước 3. Nếu chưa thực hiện bước 5.

Bước 5. Tăng số hình được chọn lên 1. Lặp lại bước 2.

2. Phân bổ những hình đã chọn vào các ô trong không gian chơi

- ♦ Lưu ý rằng số ô trong không gian chơi luôn nhiều gấp đôi số hình được chọn. Nói cách khác mỗi hình được chọn luôn nằm trong hai ô nào đó thuộc không gian chơi.
- ♦ Do vậy, bạn cần một giải thuật chung để thực hiện việc phân bổ:
 - 2 hình được chọn vào 4 ô
 - 3 hình được chọn vào 6 ô
 - 6 hình được chọn vào 12 ô
 - 12 hình được chọn vào 24 ô
 - 16 hình được chọn vào 32 ô
 - 25 hình được chọn vào 50 ô
 - 30 hình được chọn vào 60 ô

sao cho đảm bảo mỗi hình chỉ chứa trong 2 ô và chỉ 2 ô mà thôi.

- ♦ Sau đây là giải thuật phân bổ.

Bước 1. Xác định tổng số hình được chọn cần phân bổ (số ô phân bổ hình phải luôn nhiều gấp đôi).

Bước 2. Chọn từng hình cần phân bổ. Nếu đã phân bổ đủ số hình được chọn thì kết thúc giải thuật. Ngược lại thực hiện bước 3.

Bước 3. Chọn ngẫu nhiên một ô trong không gian chơi.

Bước 4. Nếu ô đã được phân bổ hình, lặp lại bước 3. Nếu ô chưa được phân bổ hình, ô sẽ được phân bổ hình đang duyệt. Xong tiếp tục bước 5.

Bước 5. Chọn ngẫu nhiên một ô nữa trong không gian chơi.

Bước 6. Nếu ô đã được phân bổ hình, lặp lại bước 5. Ngược lại, ô sẽ được phân bổ hình đang duyệt.

Bước 7. Trở lại bước 2 để chọn hình cần phân bổ kế tiếp.

3. Thuật toán xác định các ô có hiệu lực trong không gian chơi:

Để ý rằng số ô sử dụng trong không gian chơi thay đổi tùy theo mức độ chơi. Số ô sử dụng nhiều nhất là 60 ô ứng với mức 7 và ít nhất là 4 ô ứng với mức 1.

Nếu bạn thiết kế không gian chơi riêng cho từng mức độ chơi thì không cần quan tâm đến thuật toán này. Nhưng nếu bạn muốn sử dụng một không gian chơi duy nhất gồm 60 ô cho cả 7 mức độ chơi, bạn cần phải xác định các ô có hiệu lực với mức độ chơi hiện tại. Vì chỉ có mức 7 mới sử dụng hết tất cả 60 ô trong không gian chơi mà thôi.

Giả sử bạn đánh số thứ tự các ô trong không gian chơi từ trái sang phải, lần lượt từ trên xuống, với ô đầu tiên có thứ tự là 0 (Hình 4.15).

Tim cap

Games Options											
0	1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59

Hình 4.15 Đánh số thứ tự các ô trên không gian chơi.

MÁCH NƯỚC

- ✓ Số thứ tự này cũng chính là chỉ số mảng của điều khiển Image chứa hình sẽ hiển thị trên mỗi ô.

- ♦ Ở mức độ chơi 1, chỉ có 4 ô có hiệu lực trong không gian chơi là 0 (hàng 1, cột 1), 1 (hàng 1, cột 2), 12 (hàng 2, cột 1) và 13 (hàng 2, cột 2). Những ô còn lại trong không gian chơi sẽ được ẩn đối với mức độ chơi 1 (Hình 4.16).

4. Tìm cặp				
Games	Options			
0	1	2	3	4
12	13	14	15	16
24	25	26	27	28

Hình 4.16 Ở mức độ chơi 1, chỉ có 4 ô có hiệu lực.

- ♦ Tương tự những ô có hiệu lực ở mức độ chơi 2: 0, 1, 12, 13, 24 và 25.
- ♦ Mức độ chơi 3: 0, 1, 2, 3, 12, 13, 14, 15, 24, 25, 26 và 27.
- ♦ Mức độ chơi 4: 0, 1, 2, 3, 4, 5, 12, 13, 14, 15, 16, 17, 24, 25, 26, 27, 28, 29, 36, 37, 38, 39, 40, 41.
- ♦ Mức độ chơi 5: 0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19, 24, 25, 26, 27, 28, 29, 30, 31, 36, 37, 38, 39, 40, 41, 42, 43.
- ♦ Mức độ chơi 6: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57.

- ♦ Mức độ chơi 7: tất cả 60 ô trong không gian chơi đều có hiệu lực.

IV. Thiết kế chương trình

1. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start Programs ➡ Microsoft Visual Studio 6.0 ➡ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➡ New Project).

Bước 2. Visual Basic sẽ tạo một project mới, form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➡ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name : frmTimcap
- Caption : Tim cap
- WindowState : 0 – Normal
- ScaleMode : 1 - Twip
- Width : 5010
- Height : 8430

Chọn lệnh File ➡ Save Project, lưu form với tên *timcap.frm* và lưu project với tên *mines.vbp* trong thư mục xác định, chẳng hạn *c:\ltgames*.

Bước 3. Định nghĩa menu chương trình.

Mở form timcap. Xong chọn lệnh Tools ➡ Menu Editor.

Lần lượt định nghĩa các mục trên menu ngang và menu xổ xuống theo bảng sau:

Caption	Name	Checked	Shortcut
&Games	MnuGame		
..... New games	MnuNew		Ctrl+N
..... Quit games			Ctrl+Q
&Options	MnuOption	Yes	
..... Chơi tính điểm	MnuDiem		
..... Chơi tính giờ	mnuThoigian		
..... -	MnuSep1		
..... Level 1	MnuL1	Yes	
..... Level 2	MnuL2		
..... Level 3	MnuL3		
..... Level 4	MnuL4		
..... Level 5	MnuL5		
..... Level 6	MnuL6		
..... Level 7	MnuL7		

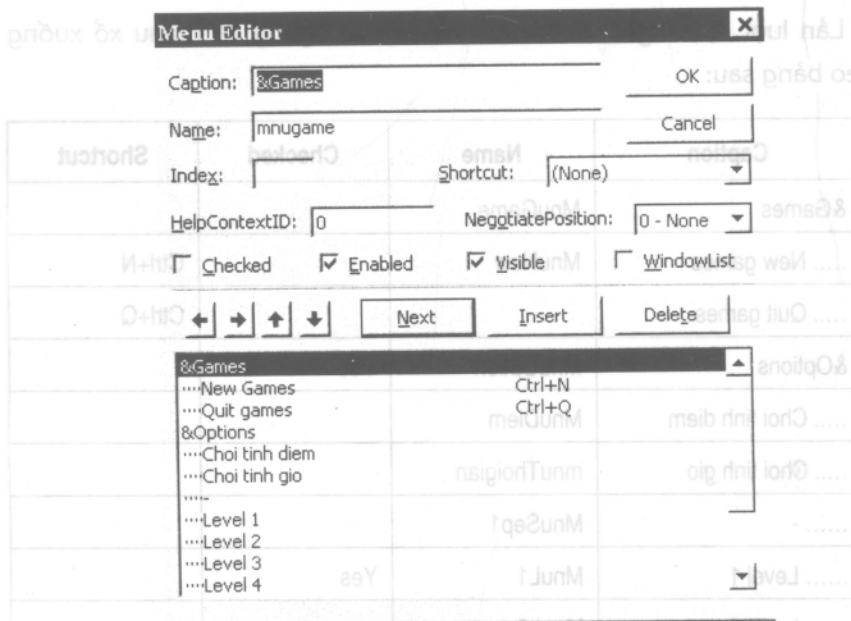
Kết quả của sổ Menu Editor sẽ như Hình 4.17.

Bước 4. Đặt các điều khiển vào form như Hình 4.18.

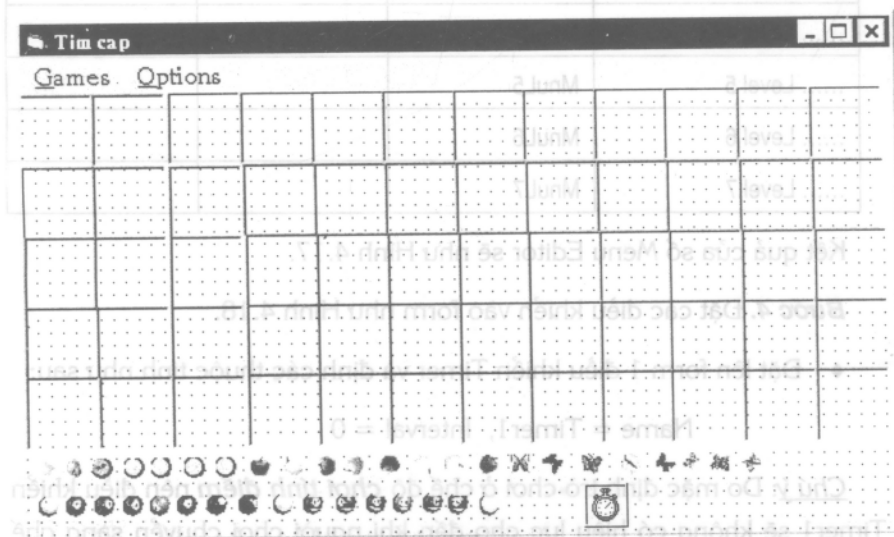
- ♦ Đặt lên form 1 điều khiển Timer và định các thuộc tính như sau:

Name = Timer1, Interval = 0

Chú ý: Do mặc định trò chơi ở chế độ **chơi tính điểm** nên điều khiển Timer1 sẽ không có hiệu lực cho đến khi người chơi chuyển sang chế độ **chơi tính giờ**.



Hình 4.17 Cửa sổ Menu Editor.



Hình 4.18 Màn hình thiết kế form frmTimcap.

Tạo mảng 39 điều khiển Image chứa tập hình có sẵn.

- ◆ Đặt lên form 1 điều khiển Image và định lại các thuộc tính sau:

Name = Hình, Index = 0, Stretch = True, Picture =

C:\tgames\gacon.bmp,

Width = 147, Height = 118, Visible = false

- ◆ Sau chép điều khiển Hình thành 38 điều khiển nữa và định lại tập tin hình cho từng điều khiển như sau:

Name	Index	Picture
Hình	1	C:\tgames\ca.bmp
Hình	2	C:\tgames\nut.bmp
Hình	3	C:\tgames\lien1.bmp
Hình	4	C:\tgames\lien2.bmp
Hình	5	C:\tgames\lien3.bmp
Hình	6	C:\tgames\lien4.bmp
Hình	7	C:\tgames\tao1.bmp
Hình	8	C:\tgames\tao2.bmp
Hình	9	C:\tgames\tao3.bmp
Hình	10	C:\tgames\tao4.bmp
Hình	11	C:\tgames\tao5.bmp
Hình	12	C:\tgames\tao6.bmp
Hình	13	C:\tgames\tao7.bmp
Hình	14	C:\tgames\tao8.bmp
Hình	15	C:\tgames\buom1.bmp
Hình	16	C:\tgames\buom2.bmp

Hình	17	C:\tgames\buom3.bmp
Hình	18	C:\tgames\buom4.bmp
Hình	19	C:\tgames\buom5.bmp
Hình	20	C:\tgames\buom6.bmp
Hình	21	C:\tgames\buom7.bmp
Hình	22	C:\tgames\buom8.bmp
Hình	23	C:\tgames\banh1.bmp
Hình	24	C:\tgames\banh2.bmp
Hình	25	C:\tgames\banh1.bmp
Hình	26	C:\tgames\banh2.bmp
Hình	27	C:\tgames\banh3.bmp
.....		
Hình	37	C:\tgames\banh14.bmp
Hình	38	C:\tgames\banh15.bmp
Hình	39	C:\tgames\banh16.bmp

MÁCH NƯỚC

- ✓ Do các điều khiển Hình đóng vai trò như những vật chứa hình, chứ không hiển thị trên form khi chương trình thi hành, nên bạn cần định thuộc tính Visible = False cho các điều khiển này. Mặt khác, việc định kích thước điều khiển nhỏ sẽ tiết kiệm không gian form và giảm kích thước tập tin nguồn chương trình.

- ♦ Bước tiếp theo là định nghĩa không gian chơi. Đặt lên form 1 điều khiển Image mới và định lại các thuộc tính như sau:

Name = oHinh, Index = 0, Stretch = True, Picture = None,
Width = 530, Height = 530, Visible = True, BorderStyle = 1- Fixed
Single, Top = 0, Left = 0

- ♦ Sao chép điều khiển oHinh thành 59 điều khiển nữa để tạo mảng oHinh 60 phần tử. Xong định lại các thuộc tính theo bảng sau:

1. Image thứ 1 (hàng 1 cột 1)	
Name : oHinh	Index : 0
Top : 0	Left : 0
2. Image thứ 2 (hàng 1, cột 2)	
Name : oHinh	Index : 1
Top : 0	Left : 530
3. Image thứ 3 (hàng 1, cột 3)	
Name : oHinh	Index : 2
Top : 0	Left : 1060
4. Image thứ 4 (hàng 1, cột 4)	
Name : oHinh	Index : 3
Top : 0	Left : 1590
5. Image thứ 5 (hàng 1, cột 5)	
Name : oHinh	Index : 4
Top : 0	Left : 2120
6. Image thứ 6 (hàng 1, cột 6)	
Name : oHinh	Index : 5
Top : 0	Left : 2650
7. Image thứ 7 (hàng 1, cột 7)	

Name : oHinh	Index : 6
Top : 0	Left : 3180
8. Image thứ 8 (hàng 1, cột 8)	
Name : oHinh	Index : 7
Top : 0	Left : 3710
9. Image thứ 9 (hàng 1, cột 9)	
Name : oHinh	Index : 8
Top : 0	Left : 4240
10. Image thứ 10 (hàng 1, cột 10)	
Name : oHinh	Index : 9
Top : 0	Left : 4770
11. Image thứ 11 (hàng 1, cột 11)	
Name : oHinh	Index : 10
Top : 0	Left : 5300
12. Image thứ 12 (hàng 1, cột 12)	
Name : oHinh	Index : 11
Top : 0	Left : 5830
13. Image thứ 13 (hàng 2, cột 1)	
Name : oHinh	Index : 12
Top : 530	Left : 0
14. Image thứ 14 (hàng 2, cột 2)	
Name : oHinh	Index : 13
Top : 530	Left : 530
15. Image thứ 15 (hàng 2, cột 3)	

Name : oHinh	Index : 14
Top : 530	Left : 1060
16. Image thứ 16 (hàng 2, cột 4)	
Name : oHinh	Index : 15
Top : 530	Left : 1590
17. Image thứ 17 (hàng 2, cột 5)	
Name : oHinh	Index : 16
Top : 530	Left : 2120
..... Và cứ tiếp tục như vậy cho đến Image thứ 60 (Index = 59)	

60 Image đều có cùng thuộc tính:

Height = 530, Width = 530, Picture = None, Stretch = True,
Visible = True

MÁCH NƯỚC

Để thiết kế nhanh các Command button và giảm thiểu việc nhầm lẫn vị trí, bạn nên làm như sau:

- ✓ Thiết kế và sắp xếp đúng vị trí 12 Image thuộc hàng đầu tiên
- ✓ Xong copy 12 Image để tạo hàng 2 (chỉ sửa lại thuộc tính Top)
- ✓ Làm tương tự để tạo các hàng 3, 4 và 5.

Bước 5. Viết mã lệnh

· Khai báo các biến và mảng toàn cục của form

Dim soHinh(7), quydoi(7), diem(7), thoigian(7)

Dim hinhchon(30), ochua(60), khonggian(60)

Dim nlevel, nhinhthuc, totals, totals2, nlat, nhinhcu, nvtcu, ndiem, nsogiai

' Thực hiện khi form vừa được nạp

Private Sub Form_Load()

' Khởi tạo vị trí các ô có hiệu lực trong không gian chơi ứng với từng mức độ chơi

quydoi(1) = " 00,01,12,13"

quydoi(2) = " 00,01,12,13,24,25"

quydoi(3) = " 00,01,02,03,12,13,14,15,24,25,26,27"

quydoi(4) = " 00,01,02,03,04,05,12,13,14,15,16,17,24,25,26,27,28,29,36,
37,38,39,40,41"

quydoi(5) = " 00,01,02,03,04,05,06,07,12,13,14,15,16,17,18,19,24,25,26,
27,28,29, 30,31,36,37,38,39,40,41,42,43"

quydoi(6) = " 00,01,02,03,04,05,06,07,08,09,12,13,14,15,16,17,18,19,20,
21,24,25,26,27,28,29,30,31,32,33,36,37,38,39,40,41,42,43,
44,45,48,49,50,51,52,53,54,55,56,57"

quydoi(7) = " 00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,
19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,
38,39,40,41,42,43,44,45, 46,47,48,49,50,51,52,53,54,55,56,
57,58,59"

' Khởi tạo giá trị tổng số hình sử dụng ứng với mức độ chơi

sohinh(1) = 2

sohinh(2) = 3

sohinh(3) = 6

sohinh(4) = 12

sohinh(5) = 16

sohinh(6) = 25

sohinh(7) = 30

' Khởi tạo số điểm tối đa cho từng mức độ chơi

diem(1) = 10

diem(2) = 20

diem(3) = 30

diem(4) = 40

diem(5) = 60

diem(6) = 80

diem(7) = 100

*' Khởi tạo thời gian tối đa mà người chơi phải lật hết các ô ứng với từng
' mức độ chơi*

thoigian(1) = 20

thoigian(2) = 30

thoigian(3) = 60

thoigian(4) = 120

thoigian(5) = 180

thoigian(6) = 240

thoigian(7) = 300

*' Mặc định, trò chơi bắt đầu với hình thức chơi tính điểm, mức độ 1
nhinhthuc = 1*

nlevel = 1

*' Gọi thủ tục khoitao để định nghĩa và khởi tạo không gian chơi
khoitao*

End Sub

' Định nghĩa không gian chơi, chọn hình và gán hình vào các ô

Sub khoitao()

nsogiyay = 0 *' giá trị ban đầu trước khi thực hiện việc đếm giây*

nlat = 0 *' chưa có hình nào được lật*

If nhinhthuc = 2 Then *' Nếu hình thức chơi là "Tính thời gian"*

Timer1.Interval = 1000 *' Khởi động đồng hồ đếm giây*

Else *' ngược lại nếu "Tính điểm"*

Timer1.Interval = 0 *' ngừng đồng hồ đếm giây*

End If

ndiem = diem(nlevel) ' điểm tối đa của người chơi ứng với mức chơi hiện hành

totals = sohinhh(nlevel) ' tổng số hình sử dụng ứng với mức độ chơi hiện hành

totals2 = sohinhh(nlevel) * 2 ' tổng số ô có hiệu lực trong không gian chơi

' Lưu vị trí các ô có hiệu lực trong không gian chơi

For i = 0 To totals2 - 1

 khonggian(i) = Val(Mid(quydoi(nlevel), 3 * i + 1, 2))

Next

' Gán giá trị âm cho những thành phần thuộc mảng **khonggian**

' không lưu vị trí ô có hiệu lực

For i = totals2 To 59

 khonggian(i) = -1

Next

' ẩn tất cả các điều khiển Image oHinh trên không gian chơi

For i = 0 To 59

 ohinh(i).Visible = False

Next

' Bật lại các điều khiển Image có hiệu lực ứng với mức độ chơi hiện hành,

' có vị trí được lưu trong mảng khonggian

For i = 0 To 59

 If khonggian(i) >= 0 Then

 ohinh(khonggian(i)).Visible = True

 End If

Next

' Chọn ra cho đủ số hình cần sử dụng trong tổng số 39 hình

' của tập hợp hình (sao cho các hình được chọn không trùng lặp nhau)

Randomize

i = 1

Do While i <= totals

```

hinhchon(i) = 0
nhinh = Int(Rnd * 39)
nduplicate = 0
For j = 1 To i - 1
    If hinhchon(j) = nhinh Then ' Nếu hình vừa chọn đã được chọn rồi
        nduplicate = 1
    End If
Next j
If nduplicate <> 1 Then ' Nếu hình vừa chọn chưa được chọn trước đó
    hinhchon(i) = nhinh ' chọn hình đó
    i = i + 1
End If
Loop
' Phân bố số hình đã chọn vào các ô có hiệu lực trong không gian chơi
' sao cho mỗi hình được phân bố vào 2 ô có hiệu lực và chỉ đúng 2 mà thôi
For i = 0 To 59
    ochua(i) = 999 ' giá trị 999 xem như ô chưa có hình nào phân bố vào
Next
For i = 1 To totals ' duyệt từng hình đã chọn
    ' Xác định ô có hiệu lực thứ nhất cần phân bố hình
    nduplicate = 1
    Do While nduplicate = 1
        ncell = Int(Rnd * totals2)
        If ochua(khonggian(ncell)) = 999 Then ' phải là ô chưa có hình nào
            ochua(khonggian(ncell)) = hinhchon(i)
            nduplicate = 0 ' hoàn tất việc phân bố hình vào ô đầu tiên
        End If
    Loop

```

' Xác định ô có hiệu lực thứ 2 cần phân bố hình

nduplicate = 1

Do While nduplicate = 1

ncell = Int(Rnd * totals2)

If ochua(khonggian(ncell)) = 999 Then *' phải là ô chưa có hình nào*

ochua(khonggian(ncell)) = hinhchon(i)

nduplicate = 0

End If

Loop

Next

End Sub

' Khi chọn chức năng Level 1 trong menu Options

Private Sub mnul1_Click()

nlevel = 1 *' đặt mức độ chơi hiện hành thành 1*

mnul1.Checked = True *' bật dấu chọn trước mục Level 1*

mnul2.Checked = False *' xóa dấu chọn trước các mục Level còn lại*

mnul3.Checked = False

mnul4.Checked = False

mnul5.Checked = False

mnul6.Checked = False

mnul7.Checked = False

End Sub

' Khi chọn chức năng Level 2 trong menu Options

Private Sub mnul2_Click()

nlevel = 2 *' đặt mức độ chơi hiện hành thành 1*

mnul1.Checked = False

mnul2.Checked = True *' Bật dấu chọn trước mục Level 2*

mnul3.Checked = False *' xóa dấu chọn trước các mục Level còn lại*

mnul4.Checked = False

mnul5.Checked = False

mnul6.Checked = False

mnul7.Checked = False

End Sub

' Khi chọn chức năng Level 3 trong menu Options

Private Sub mnul3_Click()

nlevel = 3 *' đặt mức độ chơi hiện hành thành 3*

mnul1.Checked = False

mnul2.Checked = False

mnul3.Checked = True *' Bật dấu chọn trước mục Level 3*

mnul4.Checked = False *' Xóa dấu chọn trước các mục Level còn lại*

mnul5.Checked = False

mnul6.Checked = False

mnul7.Checked = False

End Sub

' Khi chọn chức năng Level 4 trong menu Options

Private Sub mnul4_Click()

nlevel = 4 *' đặt mức độ chơi hiện hành thành 4*

mnul1.Checked = False

mnul2.Checked = False

mnul3.Checked = False

mnul4.Checked = True *' Bật dấu chọn trước mục Level 4*

mnul5.Checked = False *' Xóa dấu chọn trước các mục Level còn lại*

mnul6.Checked = False

mnul7.Checked = False

End Sub

' Khi chọn chức năng Level 5 trong menu Options

Private Sub mnul5_Click()

nlevel = 5

mnul1.Checked = False

mnul2.Checked = False

mnul3.Checked = False

mnul4.Checked = False

mnul5.Checked = True

mnul6.Checked = False

mnul7.Checked = False

End Sub

' Khi chọn chức năng Level 6 trong menu Options

Private Sub mnul6_Click()

nlevel = 6

mnul1.Checked = False

mnul2.Checked = False

mnul3.Checked = False

mnul4.Checked = False

mnul5.Checked = False

mnul6.Checked = True

mnul7.Checked = False

End Sub

' Khi chọn chức năng Level 7 trong menu Options

Private Sub mnul7_Click()

nlevel = 7

mnul1.Checked = False

mnul2.Checked = False

mnul3.Checked = False

mnul4.Checked = False

mnul5.Checked = False

mnul6.Checked = False

mnul7.Checked = True

End Sub

' Khi chọn lệnh Games ➡ New Games hay nhấn tổ hợp phím Ctrl+N.

Private Sub mnunew_Click()

For i = 0 To 59

 ohinh(i).Visible = True *' hiển thị lại toàn bộ không gian chơi*

 ohinh(i).Picture = LoadPicture("") *' bỏ các hình đã lật nếu có*

Next

' Gọi thủ tục khoitao để tiến hành khởi tạo lại không gian chơi mới

' chọn và phân bố hình cho các ô có hiệu lực ứng với mức độ chơi đã chọn

khoitao

End Sub

' Khi chọn lệnh Games ➡ Quit Games hay nhấn tổ hợp phím CTRL + Q.

Private Sub mnuquit_Click()

End

End Sub

' Khi chọn chức năng Chơi tính giờ trong menu Options

Private Sub mnuthoigian_Click()

mnudiem.Checked = False *' bỏ dấu chọn trước mục Chơi tính điểm*

mnuthoigian.Checked = True *' đánh dấu chọn trước mục Chơi tính giờ*

nhinhthuc = 2 *' chuyển hình thức chơi thành chơi tính giờ*

nlevel = 1 *' đặt mức độ chơi khởi đầu là 1*

mnul1.Visible = False *' ẩn các mục Level trong menu Options*

mnul2.Visible = False *' người chơi bắt buộc phải chơi từ thấp đến cao*

mnul3.Visible = False

mnul4.Visible = False

mnul5.Visible = False

mnul6.Visible = False

mnul7.Visible = False

End Sub

' Khi chọn chức năng Chơi tính điểm trong menu Options

Private Sub mnudiem_Click()

mnudiem.Checked = True *' Đặt dấu chọn trước mục Chơi tính điểm*

mnuthoigian.Checked = False *' bỏ dấu chọn trước mục Chơi tính giờ*

nhinhthuc = 1 *' chuyển hình thức chơi thành chơi tính điểm*

nlevel = 1 *' đặt mức độ chơi mặc định là 1*

mnul1.Visible = True *' kích hoạt các mục Level trong menu Options*

mnul2.Visible = True *' người chơi có thể chọn một mức độ chơi tùy thích*

mnul3.Visible = True

mnul4.Visible = True

mnul5.Visible = True

mnul6.Visible = True

mnul7.Visible = True

End Sub

' Khi người chơi nhấp lên một ô có hiệu lực trên không gian chơi

Private Sub ohinh_Click(Index As Integer)

ohinh(Index).Picture = hinh(ochua(Index)).Picture *' Hình sẽ được lật*

nlat = nlat + 1 *' tăng số hình đang được lật lên 1*

' nếu số hình đang được lật hiện hành bằng 2, thì đặt trở về 0

' điều này đảm bảo trong không gian chơi không bao giờ có quá hơn 2

' hình được lật

If nlat = 2 Then

 nlat = 0

' nếu hình được lật đầu tiên trùng với hình vừa lật

If nhinhcu = ochua(Index) Then

```
ohinh(Index).Visible = False      ' loại bỏ cả 2 ô khỏi không gian chơi
ohinh(nvtcu).Visible = False
totals = totals - 1                ' giảm số hình chưa lật đi 1
Else                               ' ngược lại
MsgBox "Sai rồi"                  ' hiển thị thông báo "Sai rồi"
ohinh(Index).Picture = LoadPicture("") ' úp hình trở lại và chờ lần
ohinh(nvtcu).Picture = LoadPicture("") ' lật tiếp theo của người chơi
ndiem = ndiem - 1                 ' trừ bớt của người chơi 1 điểm
End If

' Kiểm tra xem đã lật hết tất cả các hình chưa
If totals = 0 Then                ' nếu rồi
If nhinhthuc = 1 Then             ' và hình thức chơi là "Chơi tính điểm"
MsgBox "Điểm của bạn là : " & Str(ndiem) ' hiển thị kết quả điểm
Else                               ' Ngược lại nếu còn hình chưa lật hết
If nlevel = 7 Then               ' nếu đã chơi tới mức cuối cùng
MsgBox "Congratulation !"        ' hiển thị thông báo chúc mừng
Else                               ' ngược lại
nlevel = nlevel + 1              ' chuyển lên mức chơi cao hơn
mnunew_Click                      ' bắt đầu mức chơi mới
End If
nsogiyay = 0                      ' bắt đầu tính lại thời gian chơi ở mức mới
End If
End If

Else                               ' nếu không gian chơi mới có một hình được lật
nhinhcu = ochua(Index)           ' lưu lại vị trí lấy hình trong tập hợp hình có sẵn
nvtcu = Index                     ' và vị trí ô chứa hình để chuẩn bị cho lần lật tiếp theo
End If
End Sub
```


' Khi người chơi đang chọn hình thức Chơi tính giờ

' thủ tục sẽ được gọi sau mỗi giây trôi qua

Private Sub Timer1_Timer()

nsogiai = nsogiai + 1

' nếu đã hết thời gian chơi quy định mà người chơi chưa hoàn tất việc lật

' tất cả các ô thủ tục sẽ hiện thông báo hết giờ và bắt đầu chơi lại mức độ

' chơi hiện hành

If nsogiai >= thoigian(nlevel) Then

MsgBox "Đã hết thời gian chơi. Bạn thua rồi !"

nsogiai = 0

mnunew_Click

End If

End Sub

Bước 6. Chọn Start ➡ Run hay nhấn F5 để chạy thử chương trình.

Nếu mọi thứ đều như ý chọn lệnh File ➡ Make Timcap.exe để compile chương trình thành tập tin **Timcap.exe** có thể thi hành trên môi trường Windows 98, không cần khởi động Visual Basic.

V. Phân tích chương trình

1. Ý nghĩa của các mảng và biến toàn cục

- ♦ Mảng **quydoi** xác định các ô có hiệu lực tương ứng với mức độ chơi hiện hành.
- ♦ Mảng **sohinh** xác định tổng số hình được sử dụng trong không gian chơi.

Lưu ý rằng số hình sử dụng = số ô có hiệu lực / 2.

- ♦ Mảng **dlem** xác định điểm số tối đa người chơi có thể đạt ứng với từng mức độ chơi.

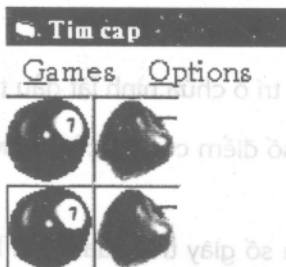
- ♦ Mảng **thoigian** xác định số giây tối đa người chơi phải hoàn tất việc lật ô ứng với mức độ chơi hiện hành.
- ♦ Mảng **Hinhchon** chứa thứ tự các hình được chọn trong tập hợp hình có sẵn (mảng **Hinh**) ứng với mức độ chơi hiện hành.
- ♦ Mảng **ochua** lưu vị trí hình được chọn trong tập hợp hình sẽ hiển thị lên mảng điều khiển **oHinh**, tại vị trí tương ứng.
- ♦ Mảng **khonggian** lưu vị trí các ô có hiệu lực trong không gian chơi.
- ♦ Mảng điều khiển Image **oHinh** dùng làm không gian chơi.
- ♦ Mảng điều khiển Image **Hinh** dùng chứa tập hợp các hình có sẵn.
- ♦ Biến **nlevel** cho biết mức độ chơi hiện hành.
- ♦ Biến **nhinhthuc** cho biết hình thức chơi hiện hành là tính điểm hay tính giờ.
- ♦ Biến **totals** xác định tổng số hình được chọn sử dụng cho lượt chơi.
- ♦ Biến **totals2** xác định tổng số ô có hiệu lực trong không gian chơi.
- ♦ Biến **nlat** xác định hình vừa lật là hình đã lật đầu tiên trong không gian chơi hay hình thứ 2.
- ♦ Biến **nhinhcu** lưu vị trí hình lật đầu tiên trong tập hợp hình đã chọn.
- ♦ Biến **nvtcu** lưu vị trí ô chứa hình lật đầu tiên.
- ♦ Biến **ndiem** lưu số điểm còn lại của người chơi tới thời điểm hiện tại.
- ♦ Biến **nsoglay** lưu số giây trôi qua kể từ khi bắt đầu mức độ chơi mới khi người chơi chọn hình thức Chơi tính giờ.

Hãy xét một ví dụ sau để hiểu thêm về vai trò của các mảng toàn cục trong quá trình vận hành chương trình:

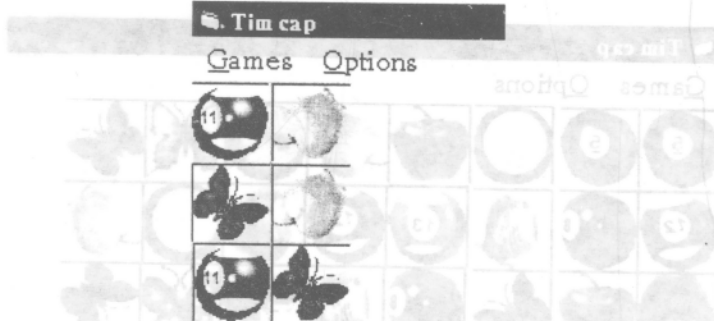
Giả sử người chơi đang ở mức độ chơi 1. Nghĩa là chỉ có 4 ô trong không gian chơi 60 ô đã định nghĩa bởi mảng điều khiển image **oHinh** có hiệu lực là 0, 1, 12, và 13.

- ♦ Khi đó mảng 4 thành phần đầu tiên của mảng **khonggian** lưu vị trí các ô có hiệu lực: $khonggian(0) = 0$, $khonggian(1) = 1$, $khonggian(2) = 12$, $khonggian(3) = 13$, các thành phần còn lại từ $khonggian(4)$ đến $khonggian(59)$ đều có giá trị -1.
- ♦ Sau khi hoàn tất việc chọn hình, 2 thành phần đầu tiên của mảng **Hinhchon** sẽ lưu vị trí 2 hình được chọn trong tập hợp hình có sẵn (mảng điều khiển image **Hinh**). Các thành phần còn lại không cần xét đến.
- ♦ Mảng **ochua** có 4 thành phần $ochua(0)$, $ochua(1)$, $ochua(12)$ và $ochua(13)$ lưu vị trí của một trong hai hình đã được chọn (mảng **Hinhchon**). Những thành phần còn lại ứng với các ô mất hiệu lực trong không gian chơi, sẽ lưu giá trị 999.

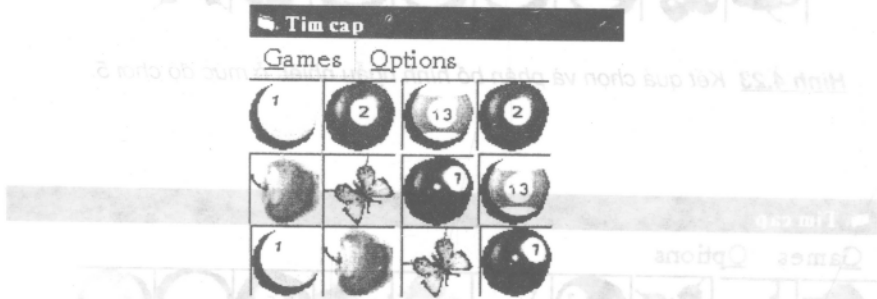
2. Những hình sau minh họa cho một số kết quả chọn và phân bố hình vào các ô trong không gian chơi tùy theo mức độ chơi khác nhau.



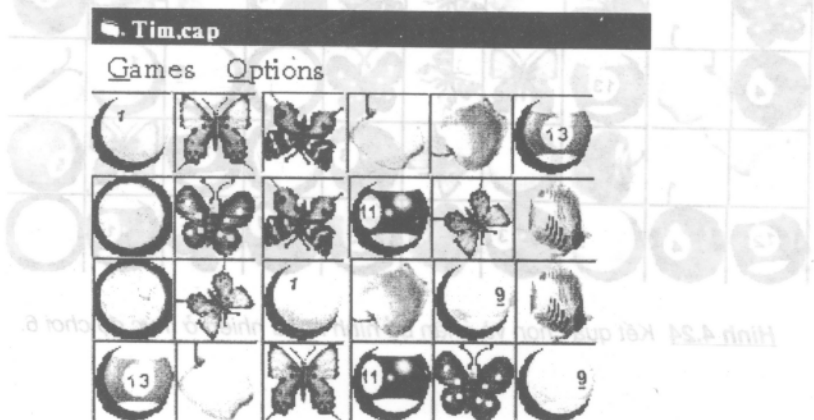
Hình 4.19 Kết quả chọn và phân bố hình ngẫu nhiên ở mức độ chơi 1.



Hình 4.20 Kết quả chọn và phân bố hình ngẫu nhiên ở mức độ chơi 2.



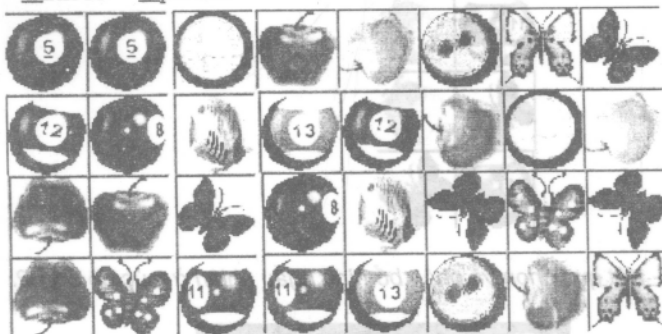
Hình 4.21 Kết quả chọn và phân bố hình ngẫu nhiên ở mức độ chơi 3.



Hình 4.22 Kết quả chọn và phân bố hình ngẫu nhiên ở mức độ chơi 4.

Tim cap

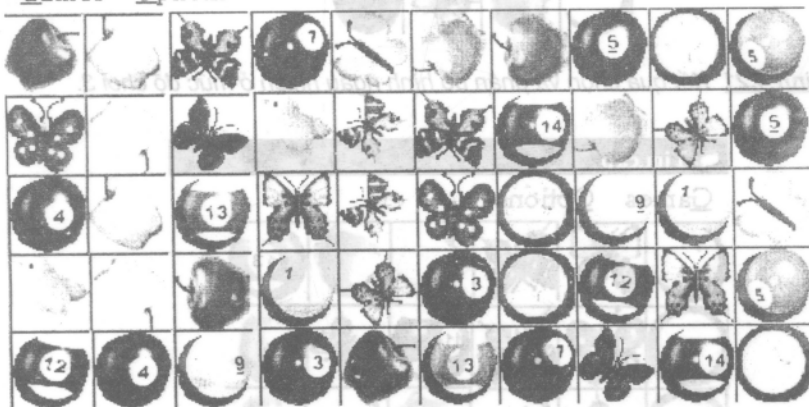
Games Options



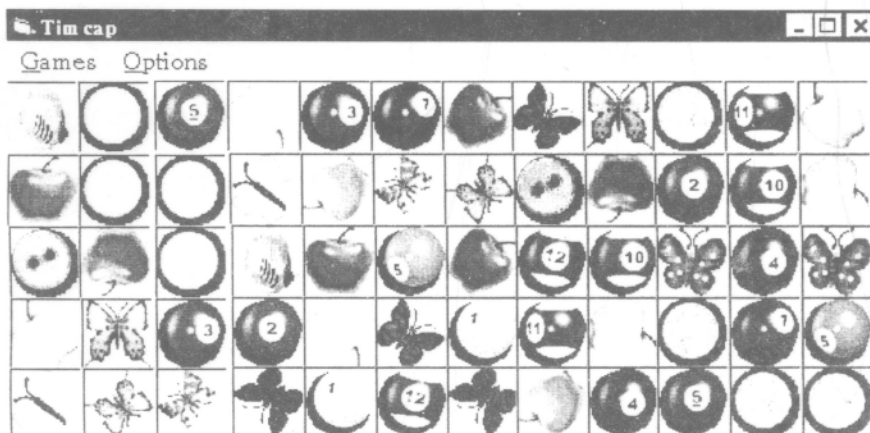
Hình 4.23 Kết quả chọn và phân bố hình ngẫu nhiên ở mức độ chơi 5.

Tim cap

Games Options



Hình 4.24 Kết quả chọn và phân bố hình ngẫu nhiên ở mức độ chơi 6.



Hình 4.25 Kết quả chọn và phân bố hình ngẫu nhiên ở mức độ chơi 7.

Nhận xét: Điểm chung duy nhất là mỗi hình được phân bố vào hai ô trong không gian chơi và chỉ 2 mà thôi.

VI. Bài tập thực hành

Bài 1. Nâng cấp chương trình *Timcap* đã xây dựng như sau:

Người dùng có thể thay đổi hình thức chơi bất kỳ lúc nào khi đang chơi. Nhưng việc thay đổi này thực sự chỉ có tác dụng cho lần chơi mới (Games ➔ New Games). Điều này sẽ dẫn đến trường hợp, màn hình đang hiển thị không gian chơi ứng với mức độ 1, nhưng trong menu tùy chọn lại là Level 3. Hãy suy nghĩ một cách để ngăn không cho người chơi thay đổi các tùy chọn mức độ hay hình thức chơi, khi trò chơi của mức độ hiện hành chưa kết thúc.

Bài 2. Nếu muốn tập hợp hình có sẵn giờ đây là 50 chứ không phải 39 hình nữa, theo bạn phải điều chỉnh những gì trong chương trình đã xây dựng. Ngoài ra, trong trò chơi đã xây dựng, bạn phải lật hình bằng cách nhấp nút trái mouse. Hãy điều chỉnh chương trình để bạn có thể sử dụng những phím mũi tên để di chuyển qua lại giữa các ô và lật hình bằng phím PageDown.

BÀI 5.

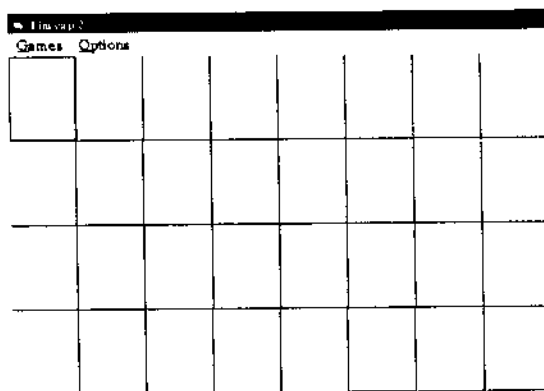
TRÒ CHƠI TÌM CẶP HAI

I. Mô tả trò chơi

Trò chơi tìm cặp hai thực chất là trò chơi tìm từ hoặc cụm từ tiếng Anh đồng nghĩa. Đây không chỉ là một trò chơi, mà còn là phương thức giúp bạn luyện tập tiếng Anh. Trò chơi có thể thức khá giống với tìm cặp 1, điểm khác biệt ở đây là thay vì lật hình, trò chơi lại lật các từ tiếng Anh để tìm cặp từ đồng nghĩa, với số lần lật càng ít càng tốt. Người chơi cũng có thể dùng các phím mũi tên **←,→,↑,↓** để di chuyển con trỏ qua lại các ô. Nhấn phím **Enter** để lật ảnh.

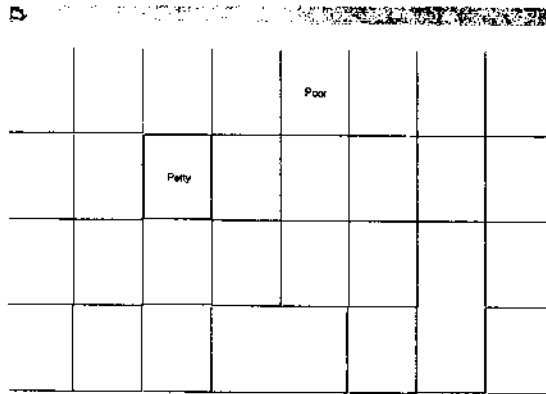
Trò chơi cũng chia làm 7 mức từ thấp đến cao. Số ô trong không gian chơi thay đổi theo từng mức. Cụ thể là:

- Mức 1 : 4 ô, phân thành 2 hàng 2 cột
- Mức 2 : 6 ô, phân thành 3 hàng 2 cột
- Mức 3 : 12 ô, phân thành 3 hàng 4 cột
- Mức 4 : 24 ô, phân thành 4 hàng 6 cột
- Mức 5 : 32 ô , phân thành 4 hàng 8 cột
- Mức 6 : 50 ô, phân thành 5 hàng 10 cột
- Mức 7 : 60 ô, phân thành 6 hàng 10 cột

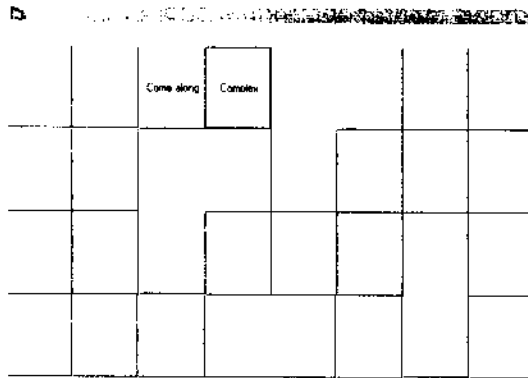


Hình 5.1 Không gian chơi của trò chơi ứng với mức 5.

- ◆ Khi trò chơi bắt đầu, mỗi ô sẽ được gán cho từ / cụm từ xác định, nhưng không hiển thị ra cho người chơi thấy.
- ◆ Dù ở mức độ nào, người chơi đều có thể tìm thấy trong không gian chơi hai ô chứa các từ đồng nghĩa nhau.
- ◆ Khi người chơi nhấp trái hay nhấn phím Enter trên một ô, từ trong ô đó sẽ hiển thị. Lúc đó, người chơi cần xác định ô có chứa từ vừa lật có nghĩa giống từ đã lật trước đó hay không (từ đang hiển thị).
- ◆ Nếu hai ô được lật chứa các từ / cụm từ đồng nghĩa nhau thì chúng sẽ được loại khỏi không gian chơi (Hình 5.2).
- ◆ Ngược lại, xem như người chơi chọn từ lật không thành công, hai từ vừa hiển thị sẽ được úp trở lại và chờ đợi lần lật kế tiếp của người chơi (Hình 5.3).
- ◆ Người chơi sẽ hoàn tất mức chơi khi loại bỏ thành công tất cả các ô chứa từ khỏi không gian chơi. Khi đó, người chơi có thể chuyển đến mức chơi cao hơn để thử sức.



Hình 5.2 Hai ô vừa lật là một cặp từ đồng nghĩa, nên sẽ được loại khỏi không gian chơi.



Hình 5.3 Hai ô vừa lật không phải là cặp từ đồng nghĩa, nên sẽ tự động úp trở lại.

Để làm phong phú cách chơi, người chơi cũng có thể chọn một trong hai hình thức chơi:

1. **Chơi tính điểm:** Người chơi có thể lựa chọn bất kỳ 1 trong 7 mức độ chơi. Mỗi mức độ chơi khi hoàn tất sẽ được một thang điểm tối đa xác định. Cụ thể là:

- Mức 1 : 10 điểm
- Mức 2 : 20 điểm
- Mức 3 : 30 điểm
- Mức 4 : 40 điểm
- Mức 5 : 60 điểm
- Mức 6 : 80 điểm
- Mức 7 : 100 điểm

Tuy nhiên, người chơi chỉ có thể đạt được điểm tối đa chỉ khi hoàn tất mức độ chơi mà không có lần lặt nào thất bại. Ngược lại, mỗi lần lặt không thành công, người chơi sẽ bị trừ đi 1 điểm. Hình thức chơi này không khống chế thời gian chơi.

2. *Chơi theo thời gian*: Người chơi phải hoàn tất mỗi mức độ chơi trong khoảng thời gian quy định. Khi đó, người chơi mới có thể lên được mức cao hơn. Nói cách khác, người chơi phải lần lượt chơi từ mức thấp đến cao. Để lên mức cao hơn, người chơi phải ghép được tất cả các từ / cụm từ trước khi hết giờ.

Thời gian chơi được quy định cụ thể cho mỗi mức như sau:

- Mức 1 : 20 giây
- Mức 2 : 30 giây
- Mức 3 : 1 phút
- Mức 4 : 2 phút
- Mức 5 : 3 phút
- Mức 6 : 4 phút

- Mức 7 : 5 phút

Rõ ràng, tuy thể thức chơi hoàn toàn giống như tìm cặp 1, nhưng đòi hỏi khó hơn, vì để chơi thành công, người chơi còn phải có chút đỉnh vốn từ vựng tiếng Anh.

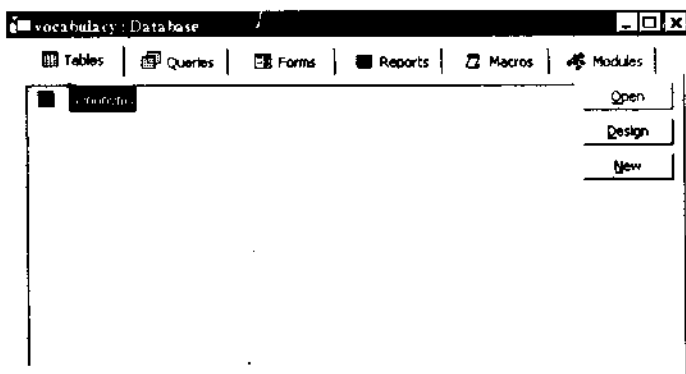
Trong bài tập này, chúng tôi cung cấp cho bạn 50 cặp từ / nhóm từ đồng nghĩa. Tuy nhiên, để người chơi không bị gò bó với số cặp từ đồng nghĩa ít ỏi đó, nên chúng tôi sẽ xây dựng chương trình theo hướng cho phép người chơi có thể cập nhật bổ sung số cặp từ đồng nghĩa với số lượng không hạn chế, mà không cần chỉnh sửa chương trình nguồn. Để làm được điều này, chúng ta cần phải có nơi lưu trữ các từ đồng nghĩa độc lập với chương trình nguồn. Trong trường hợp này, đối với nhiều bạn không có gì thuận tiện hơn là sử dụng một cơ sở dữ liệu Access để lưu trữ các cặp từ đồng nghĩa. Bởi vì đó là cơ sở dữ liệu khá phổ biến hiện nay và dễ cài đặt trên máy tính của bạn.

II. Chuẩn bị dữ liệu với hệ quản trị cơ sở dữ liệu Microsoft Access

Khởi động Microsoft Access, trong hộp thoại Microsoft Access, chọn mục Blank Database, xong nhấp OK. Hộp thoại File New Database sẽ hiển thị, chọn thư mục cần lưu cơ sở dữ liệu, chẳng hạn C:\LTGames, và nhập tên cơ sở dữ liệu sẽ tạo:

File name = *Vocabulary*, xong nhấp nút Create

Cửa sổ cơ sở dữ liệu sẽ hiển thị như Hình 5.4.



Hình 5.4 Cửa sổ cơ sở dữ liệu Vocabulary.mdb sau khi định nghĩa xong table Synonyms.

1. Định nghĩa cấu trúc table Synonyms:

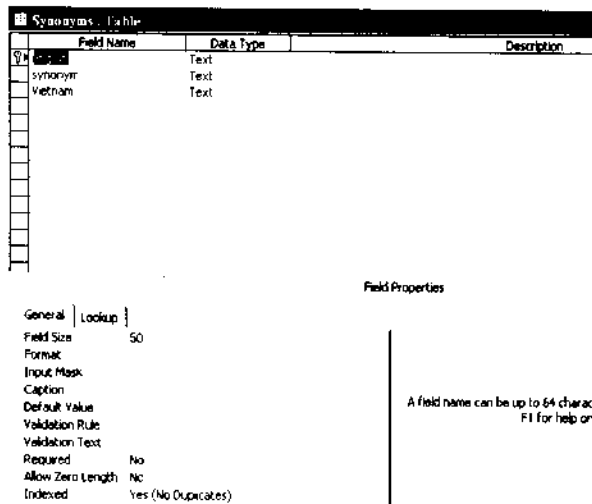
Chọn tab Tables, nhấp nút New, xong chọn Design View trong hộp thoại New table và nhấp OK.

Khai báo cấu trúc table Synonyms theo bảng sau:

Field name	Data Type	Field Size
English	Text	50
Synonyms	Text	50
Vietnamese	Text	50

Đặt field English làm khóa chính trong table Synonyms:

Nhấp tên field English, chọn lệnh Edit ➡ Primary Key. Kết quả màn hình thiết kế cấu trúc table sẽ như Hình 5.5.



Hình 5.5 Màn hình thiết kế cấu trúc table Synonyms.

Đóng màn hình và lưu table với tên Synonyms.

2. Cập nhật dữ liệu vào table Synonyms

Cuối bài, bạn sẽ được hướng dẫn cách thiết kế form nhập liệu cho table Synonyms. Còn bây giờ để có dữ liệu sử dụng trong chương trình tìm cặp 2, tạm thời bạn có thể nhập dữ liệu trực tiếp vào table Synonyms với Access.

Chọn table Synonyms, nhấp Open và tiến hành cập nhật dữ liệu theo bảng sau:

English	Synonyms	Vietnamese
At last	Finally	Sau cùng
Awkward	Whimsical	Oái ăm
Band	Gang	Bộ sậu
Charming	Lovely	Có duyên

Clearly	Distinctly	Rạch ròi
Complex	Complicated	Nhiều khê
Deceitful	Dishonest	Ba que (xỏ lá)
Demoded	Outdated	Lỗi thời
Doubtful	Uncertain	Nghi vấn
Fairlady	Maiden	Giai nhân
Fine	Silky	Mịn màng
Flock	Stream	Nuộm nược
Fordone	Exhausted	Mệt lử
Forward	Haft-back	Tiến về
Gear	Shift	Sang số xe
Genuine	Unusual talent	Kỳ tài
Glutinous	Sticky rice	Gạo nếp
Haft-asleep	Haft-awake	Lơ mơ
Harmonious	Lilting	Du dương
Headstrong	Self-illed	Khó bảo
Hero	He-man	Người hùng
High-born	Aristocratic	Quý phái
Hooligan	Rowdy	Du côn
Human life	Human livelihood	Nhân mạng
Hurry up	Come along	Nhanh lên
Initial	Approach	Khởi đầu
Lonely	Solitary	Lẻ loi
Loveable	Likeable	Khả ái

Loyalty	Constancy	Lòng son
Master-key	Skeleton-key	Chìa khóa vạn năng
Noble	High-minded	Cao thượng
Outsider	Unrelated person	Người ngoài
Overflow	Boil over	Trào ra
Overturn	Upset	Lật nhào
Peach	Sneak	Mách lẻo
Poor	Petty	Quèn
Project	Scheme	Quy hoạch
Responsibility	Task	Nhiệm vụ
Shy	Bashful	E lệ
Sketch out	Draw out	Phác thảo
Sky-scraper	Sky-high	Ngất trời
Snobbish	Affected	Đai các
Specialist	Expert	Chuyên gia
Spruce up	Smarten	Chải chuốt
Sub-department	Section	Phân khoa
Surely	Certainly	Ắt hẳn
Sweetheart	Lover	Người yêu
Universalize	Generalize	Phổ cập
Unruly	Rowdy	Ba trợn
Unstable	Unsteady	Lắt léo

MÁCH NƯỚC

- ✓ Nội dung field Vietnamese chỉ để bạn tham khảo. Thực chất chương trình trò chơi “dò tìm 2” chỉ cần dữ liệu trong field English và Synonyms mà thôi. Tuy nhiên, bạn có thể sử dụng nội dung field này khi xây dựng thêm chức năng học từ đồng nghĩa, giúp “người chơi” ôn tập và học từ đồng nghĩa trước khi chơi trò chơi.
-

III. Các thuật toán xử lý

1. Chọn các cặp từ đồng nghĩa trong những cặp từ thuộc table Synonyms

- ◆ Như đã đề cập ở trên, số cặp từ mẫu là 50. Tuy nhiên, bạn có thể bổ sung số cặp từ đồng nghĩa với số lượng không hạn chế vào table Synonyms. Có thể là 100 hoặc vài ngàn cặp từ đồng nghĩa khác nhau, nếu bạn chịu khó sưu tầm và cập nhật.
- ◆ Để bạn không bị gò bó trong một cách chọn lựa, chúng tôi sẽ trình bày ba phương thức khác nhau chọn ra một số cặp từ đồng nghĩa trong table Synonyms
- ◆ Hãy xét ví dụ minh họa sau: Giả sử bạn đang chọn mức độ chơi 5, điều đó có nghĩa bạn cần chọn ra 16 cặp từ đồng nghĩa trong table Synonyms.

Cách 1. Đọc tất cả cặp từ đồng nghĩa từ table Synonyms vào một mảng 2 chiều có tên **dWord**, mỗi phần tử mảng lưu thông tin một cặp từ đồng nghĩa. Sau đó, tiến hành giải thuật chọn 16 cặp từ trong mảng dWord như đã trình bày trong bài tập 4.

Ưu điểm: Tốc độ chọn lựa nhanh, có thể sử dụng lại giải thuật đã trình bày trong bài 4 mà không cần phải điều chỉnh gì nhiều.

Nhược điểm: Số lượng record trong Synonyms thay đổi tùy người nhập. Do đó, có thể dẫn đến trường hợp tràn mảng dWord. Mặt khác, mảng dWord có quá nhiều thành phần sẽ chiếm dung lượng bộ nhớ. Cách này chỉ nên thực hiện khi biết trước giới hạn số record trong table Synonyms và số lượng này là nhỏ cỡ 100 record trở xuống.

Cách 2. Tiến hành trực tiếp việc chọn lựa 16 cặp từ đồng nghĩa cần sử dụng ngay trên table Synonyms.

Ưu điểm: Không cần sử dụng mảng trung gian, đỡ tốn bộ nhớ và không hạn chế số lượng record nhập trong table Synonyms.

Nhược điểm: Quá trình chọn lựa ngẫu nhiên 16 cặp từ đồng nghĩa ngay trong quá trình đọc trực tiếp record từ table Synonyms sẽ diễn ra chậm hơn khi xử lý trên mảng. Đặc biệt là khi table có số lượng record lớn. Cách này thường thích hợp khi bạn muốn chọn ra 16 record liên tiếp nhau bắt đầu từ vị trí xác định nào đó.

Cách 3. Thực chất là cách trung hòa hai phương pháp trên. Và đây cũng là cách được sử dụng trong bài tập này. Đầu tiên, bạn sẽ đọc 40 cặp từ đồng nghĩa liên tiếp nhau trong table Synonyms bắt đầu từ một vị trí ngẫu nhiên nào đó vào mảng dWord. Sau đó tiến hành giải thuật chọn 16 cặp từ đồng nghĩa trong tổng số 40 cặp từ đồng nghĩa chứa trong mảng aWord.

Ưu điểm: Không hạn chế số lượng record nhập trong table Synonyms. Kích thước mảng hai chiều dWord luôn cố định là 40 (> 30 là số lượng cặp từ tối đa được chọn sử dụng), nên tốn ít bộ nhớ và tránh được tình trạng tràn mảng. Giải thuật chọn n cặp từ

($n=2, 3, 6, 12, 16, 25, 30$) trong tổng số 40 cặp từ vẫn đảm bảo được tính linh động ngẫu nhiên. Cho phép tận dụng lại chương trình trong bài tập 4, mà không phải chỉnh sửa gì nhiều.

MÁCH NƯỚC

- ✓ Số cặp từ tối đa được sử dụng trong một lượt chơi là 30 cặp từ ứng với mức độ chơi 7. Và số cặp từ đọc ra mảng dWord trước khi thực hiện việc chọn ngẫu nhiên là 40. Do vậy, để đảm bảo chương trình chạy tốt, bạn cần lưu ý số cặp từ nhập trong table Synonyms phải lớn hơn 40.
-

2. Phân bố những cặp từ đồng nghĩa đã chọn vào các ô trong không gian chơi

Tương tự, như trong bài 4. Tuy nhiên, bạn cần lưu ý vài điểm sau:

- ◆ Cách bố trí các ô trong không gian chơi ở mức 7: 60 ô, phân thành 6 hàng 10 cột thay vì phân thành 5 hàng, 12 cột như bài 4.
- ◆ Các cặp từ đồng nghĩa, tuy có cùng nghĩa tiếng Việt, nhưng từ tiếng Anh là khác nhau. Do đó, sau khi phân bố, trong không gian chơi sẽ không có hai ô nào chứa từ tiếng Anh giống nhau.
- ◆ Sau đây là giải thuật phân bố.

Bước 1. Xác định tổng số cặp từ được chọn cần phân bố (số ô phân bố phải luôn nhiều gấp đôi).

Bước 2. Chọn từng cặp từ cần phân bố. Nếu đã phân bố đủ số cặp từ được chọn thì kết thúc giải thuật. Ngược lại thực hiện bước 3.

Bước 3. Chọn ngẫu nhiên một ô trong không gian chơi.

Bước 4. Nếu ô đã được phân bổ từ, lặp lại bước 3. Nếu ô chưa được phân bổ từ, ô sẽ được phân bổ từ đầu tiên trong cặp từ đồng nghĩa đang chọn. Xong tiếp tục bước 5.

Bước 5. Chọn ngẫu nhiên một ô nữa trong không gian chơi.

Bước 6. Nếu ô đã được phân bổ từ, lặp lại bước 5. Ngược lại, ô sẽ được phân bổ từ thứ 2 trong cặp từ đồng nghĩa đang chọn.

Bước 7. Trở lại bước 2 để chọn cặp từ đồng nghĩa cần phân bổ kế tiếp.

3. Thuật toán xác định các ô có hiệu lực trong không gian chơi:

Tương tự như bài 4. Thiết tường không cần nhắc lại.

IV. Thiết kế chương trình

1. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start ➡ Programs ➡ Microsoft Visual Studio 6.0 ➡ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➡ New Project).

Bước 2. Visual Basic sẽ tạo một project mới, form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➡ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name : frmTimcap2
- Caption : Tim cap 2
- WindowState : 2 – Maximized

- ScaleMode : 1 - Twip
- Width : 12000
- Height : 9000

Chọn lệnh File ➡ Save Project, lưu form với tên *timcap2.frm* và lưu project với tên *timcap2.vbp* trong thư mục xác định, chẳng hạn *c:\ltgames*.

Bước 3. Định nghĩa menu chương trình.

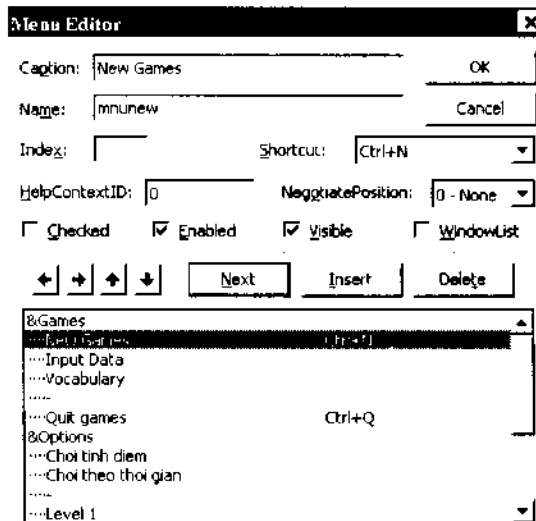
Mở form *frmTimcap2*. Xong chọn lệnh Tools ➡ Menu Editor.

Lần lượt định nghĩa các mục trên menu ngang và menu xổ xuống theo bảng sau:

Caption	Name	Checked	Shortcut
&Games	MnuGame		
..... New games	MnuNew		Ctrl+N
..... Input Data	MnuInput		
..... Vocabulary	MnuView		
..... *	MnuSep1		
..... Quit games			Ctrl+Q
&Options	MnuOption	Yes	
..... Chơi tính điểm	MnuDiem		
..... Chơi tính giờ	MnuThoigian		
..... -	MnuSep2		
..... Level 1	MnuL1	Yes	
..... Level 2	MnuL2		

..... Level 3	MnuL3		
..... Level 4	MnuL4		
..... Level 5	MnuL5		
..... Level 6	MnuL6		
..... Level 7	MnuL7		

Kết quả của sổ Menu Editor sẽ như Hình 5.6.



Hình 5.6 Kết quả của sổ Menu Editor.

Bước 4. Đặt các điều khiển vào form như Hình 5.7.

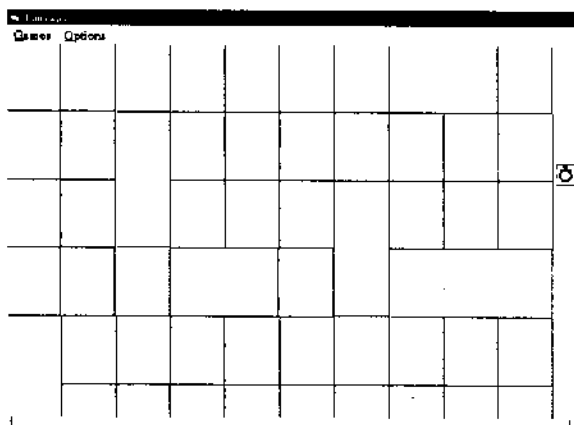
- ◆ Đặt lên form 1 điều khiển Timer và định các thuộc tính như sau:

Name = Timer1, Interval = 0

Chú ý: Do mặc định trò chơi ở chế độ **chơi tính điểm** nên điều khiển Timer1 sẽ không có hiệu lực cho đến khi người chơi chuyển sang chế độ **chơi tính giờ**.

- ◆ Bước tiếp theo là định nghĩa không gian chơi. Đặt lên form 1 điều khiển Command button mới và định lại các thuộc tính như sau:

Name = sWord, Index = 0, Style = 1 – Graphical, Caption = (trống),
Width = 1000, Height = 1200, Visible = True, BackColor =
&H00C0C0C0&, Top = 0, Left = 0



Hình 5.7 Màn hình thiết kế form frmTimcap2.

- ◆ Sao chép điều khiển sWord thành 59 điều khiển nữa để tạo mảng sWord 60 phần tử. Xong định lại các thuộc tính theo bảng sau:

1. Command button thứ 1 (hàng 1 cột 1)	
Name : sWord	Index : 0
Top : 0	Left : 0
2. Command button thứ 2 (hàng 1, cột 2)	
Name : sWord	Index : 1
Top : 0	Left : 1000
3. Command button thứ 3 (hàng 1, cột 3)	
Name : sWord	Index : 2

Top : 0	Left : 2000
4. Command button thứ 4 (hàng 1, cột 4)	
Name : sWord	Index : 3
Top : 0	Left : 3000
5. Command button thứ 5 (hàng 1, cột 5)	
Name : sWord	Index : 4
Top : 0	Left : 4000
6. Command button thứ 6 (hàng 1, cột 6)	
Name : sWord	Index : 5
Top : 0	Left : 5000
7. Command button thứ 7 (hàng 1, cột 7)	
Name : sWord	Index : 6
Top : 0	Left : 6000
8. Command button thứ 8 (hàng 1, cột 8)	
Name : sWord	Index : 7
Top : 0	Left : 7000
9. Command button thứ 9 (hàng 1, cột 9)	
Name : sWord	Index : 8
Top : 0	Left : 8000
10. Command button thứ 10 (hàng 1, cột 10)	
Name : sWord	Index : 9
Top : 0	Left : 9000
11. Command button thứ 11 (hàng 1, cột 11)	
Name : sWord	index : 10

Top : 1200	Left : 0
12. Command button thứ 12 (hàng 1, cột 12)	
Name : sWord	Index : 11
Top : 1200	Left : 1000
13. Command button thứ 13 (hàng 2, cột 1)	
Name : sWord	Index : 12
Top : 1200	Left : 2000
14. Command button thứ 14 (hàng 2, cột 2)	
Name : sWord	Index : 13
Top : 1200	Left : 3000
.....	
Và cứ tiếp tục như vậy cho đến Command button thứ 60 (Index = 59)	

60 Command button đều có cùng thuộc tính:

Height = 1200, Width = 1000, Caption = (trống), Visible = True

MÁCH NƯỚC

Để thiết kế nhanh các Command button và giảm thiểu việc nhầm lẫn vị trí, bạn nên làm như sau:

- ✓ Thiết kế và sắp xếp đúng vị trí 10 Command button thuộc hàng đầu tiên.
- ✓ Xong copy 10 Command button để tạo hàng 2 (chỉ sửa lại thuộc tính Top).
- ✓ Làm tương tự để tạo các hàng 3, 4, 5 và 6.

Bước 5. Viết mã lệnh

' Khai báo các biến và mảng toàn cục của form

Dim dword(40, 2), tuchon(30), ochua(60), khonggian(60)

Dim socaptu(7), diem(7), thoigian(7), quydoi(7)

Dim nlevel, nhinhthuc, totals, totals2, nlat, ntucu, nvtcu, studn, ndiem, nsogiy

' Thủ tục được gọi khi nạp form

Private Sub Form_Load()

' Đọc ra 40 cặp từ đồng nghĩa để sử dụng trong vô số cặp từ đã nhập

' Sao cho không có cặp từ nào được chọn quá một lần trong số 40 cặp từ sử dụng

Dim nrec

' Khởi tạo vị trí các ô có hiệu lực trong không gian chơi ứng với từng mức độ chơi

quydoi(1) = " 00,01,10,11"

quydoi(2) = " 00,01,10,11,20,21"

quydoi(3) = " 00,01,02,03,10,11,12,13,20,21,22,23"

quydoi(4) = " 00,01,02,03,04,05,10,11,12,13,14,15,20,21,22,23,24,25,30,31,32,33,34, 35"

quydoi(5) = " 00,01,02,03,04,05,06,07,10,11,12,13,14,15,16,17,20,21,22,23,24, 25,26,27,30,31,32,33,34,35,36,37"

quydoi(6) = " 00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19, 20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39, 40,41,42,43,44,45,46,47,48,49"

quydoi(7) = " 00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20, 21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36, 37,38,39, 40,41,42,43,44,45,46,47,48,49,50,51,52,53,54, 55,56,57,58,59"

' Khởi tạo giá trị tổng số cặp từ sử dụng ứng với mức độ chơi

socaptu(1) = 2

socaptu(2) = 3

socaptu(3) = 6

socaptu(4) = 12

socaptu(5) = 16

socaptu(6) = 25

socaptu(7) = 30

' Khởi tạo số điểm tối đa cho từng mức độ chơi

diem(1) = 10

diem(2) = 20

diem(3) = 30

diem(4) = 40

diem(5) = 60

diem(6) = 80

diem(7) = 100

' Khởi tạo thời gian tối đa mà người chơi phải lật hết các ô ứng với từng

' mức độ chơi

thoigian(1) = 20

thoigian(2) = 30

thoigian(3) = 60

thoigian(4) = 120

thoigian(5) = 180

thoigian(6) = 240

thoigian(7) = 300

' Mặc định, trò chơi bắt đầu với hình thức chơi tính điểm, mức độ 1

ninhthuc = 1

nlevel = 1

' Gọi thủ tục khoitao để định nghĩa và khởi tạo không gian chơi

khoitao

End Sub

' Khởi tạo không gian chơi tùy theo mức độ chơi

Sub khoitao()

Randomize *' khởi động chế độ chọn số ngẫu nhiên*

' Truy xuất dữ liệu trong table Synonyms thuộc cơ sở dữ liệu Access

Vocabulary.mdb

Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")

Set rstudien = dbs.OpenRecordset("Synonyms")

If rstudien.EOF = False Then *' Nếu có dữ liệu trong table Synonyms*

nrec = rstudien.RecordCount *' đếm số record có trong table Synonyms*

If nrec > 40 Then *' Nếu có nhiều hơn 40 record*

ncap = Int(Rnd * nrec - 40) *' chọn ngẫu nhiên vị trí record bắt đầu đọc
' dữ liệu*

Else *' ngược lại*

ncap = 1 *' bắt đầu đọc từ record đầu tiên*

End If

rstudien.MoveFirst *' Chuyển về record đầu tiên*

For i = 2 To ncap *' vòng lặp cho phép chuyển đến vị trí
' record bắt đầu*

rstudien.MoveNext *' đọc dữ liệu*

Next

k = 0

Do While Not rstudien.EOF *' Nếu chưa đọc đến record cuối cùng*

dword(k, 0) = rstudien!english *' đọc từ và từ đồng nghĩa từ*

dword(k, 1) = rstudien!Synonyms *' record hiện hành vào mảng dword*

rstudien.MoveNext

k = k + 1

If k >= 40 Then *' Nếu đã đọc đủ 40 cặp từ thì thoát*

Exit Do *' không cần đọc nữa*

```

End If
Loop
End If
rstudien.Close
' Xác định các cặp từ được sử dụng trong số 40 cặp từ đã chọn
nsoglay = 0          ' giá trị ban đầu trước khi thực hiện việc đếm giây
nlat = 0             ' chưa có từ nào được lật
If nhinhthuc = 2 Then    ' Nếu hình thức chơi là "Tính thời gian"
    Timer1.Interval = 1000    ' Khởi động đồng hồ đếm giây
Else                    ' ngược lại nếu "Tính điểm"
    Timer1.Interval = 0      ' ngừng đồng hồ đếm giây
End If
ndiem = diem(nlevel) ' điểm tối đa của người chơi ứng với mức chơi hiện hành
totals = socaptu(nlevel) ' tổng số cặp từ sử dụng ứng với mức độ chơi hiện hành
totals2 = socaptu(nlevel) * 2 ' tổng số ô có hiệu lực trong không gian chơi
' Chọn ra ngẫu nhiên n cặp từ trong số 40 cặp và phân bố vào m ô
' (với n = 2,3,6,12,16,25,30 và m = 4,6,12,24,32,50,60)
i = 1
Do While i <= totals
    tuchon(i) = 0
    nchon = Int(Rnd * 40)
    nduplicate = 0
    For j = 1 To i - 1
        If tuchon(j) = nchon Then
            nduplicate = 1
        End If
    Next j
    If nduplicate <> 1 Then

```

```

tuchon(i) = nchon
i = i + 1
End If
Loop
' Lưu vị trí các ô có hiệu lực trong không gian chơi
For i = 0 To totals2 - 1
    khonggian(i) = Val(Mid(quydoi(nlevel), 3 * i + 1, 2))
Next
' Gán giá trị âm cho những thành phần thuộc mảng khonggian
' không lưu vị trí ô có hiệu lực
For i = totals2 To 59
    khonggian(i) = -1
Next
' Phân bố n cặp từ vào m ô trong không gian chơi
For i = 0 To 59
    ochua(i) = ""
Next
For i = 1 To totals
    ' Chọn ô để phân bố từ, mỗi từ 1 ô
    ' Xác định ô thứ nhất
    nduplicate = 1
    Do While nduplicate = 1
        ncell = Int(Rnd * totals2)
        If ochua(khonggian(ncell)) = "" Then ' phải là ô chưa chứa từ nào
            ' phân bố từ đầu tiên trong cặp từ được chọn vào ô
            ochua(khonggian(ncell)) = dword(tuchon(i), 0)
            nduplicate = 0
        End If
    End If

```

Loop

' Xác định ô chứa từ thứ 2 trong cặp từ đồng nghĩa đang chọn

nduplicate = 1

Do While nduplicate = 1

ncell = Int(Rnd * totals2)

If ochua(khonggian(ncell)) = "" Then *' phải là ô chưa chứa từ nào*

ochua(khonggian(ncell)) = dword(tuchon(i), 1)

nduplicate = 0

End If

Loop

Next

' ẩn tất cả điều khiển Command button đang mang các từ đồng nghĩa

For i = 0 To 59

sword(i).Visible = False

Next

' Bật lại các ô có hiệu lực ứng với mức độ chơi hiện hành,

' có vị trí được lưu trong mảng khonggian

For i = 0 To 59

If khonggian(i) >= 0 Then

sword(khonggian(i)).Visible = True

sword(khonggian(i)).Enabled = True

End If

Next

End Sub

' Khi chọn chức năng Level 1 trong menu Options

Private Sub mnul1_Click()

nlevel1 = 1 *' đặt mức độ chơi hiện hành thành 1*

mnul1.Checked = True *' bật dấu chọn trước mục Level 1*

```
mnul2.Checked = False      ' xóa dấu chọn trước các mục Level còn lại
mnul3.Checked = False
mnul4.Checked = False
mnul5.Checked = False
mnul6.Checked = False
mnul7.Checked = False
End Sub
```

' Khi chọn chức năng Level 2 trong menu Options

Private Sub mnul2_Click()

```
nlevel = 2                  ' đặt mức độ chơi hiện hành thành 1
mnul1.Checked = False
mnul2.Checked = True        ' Bật dấu chọn trước mục Level 2
mnul3.Checked = False      ' Xóa dấu chọn trước các mục Level còn lại
mnul4.Checked = False
mnul5.Checked = False
mnul6.Checked = False
mnul7.Checked = False
```

End Sub

' Khi chọn chức năng Level 3 trong menu Options

Private Sub mnul3_Click()

```
nlevel = 3                  ' Đặt mức độ chơi hiện hành thành 3
mnul1.Checked = False
mnul2.Checked = False
mnul3.Checked = True        ' Bật dấu chọn trước mục Level 3
mnul4.Checked = False      ' Xóa dấu chọn trước các mục Level còn lại
mnul5.Checked = False
mnul6.Checked = False
mnul7.Checked = False
```

End Sub*' Khi chọn chức năng Level 4 trong menu Options***Private Sub mnul4_Click()**nlevel = 4 *' Đặt mức độ chơi hiện hành thành 4*

mnul1.Checked = False

mnul2.Checked = False

mnul3.Checked = False

mnul4.Checked = True *' Bật dấu chọn trước mục Level 4*mnul5.Checked = False *' Xóa dấu chọn trước các mục Level còn lại*

mnul6.Checked = False

mnul7.Checked = False

End Sub*' Khi chọn chức năng Level 5 trong menu Options***Private Sub mnul5_Click()**

nlevel = 5

mnul1.Checked = False

mnul2.Checked = False

mnul3.Checked = False

mnul4.Checked = False

mnul5.Checked = True

mnul6.Checked = False

mnul7.Checked = False

End Sub*' Khi chọn chức năng Level 6 trong menu Options***Private Sub mnul6_Click()**

nlevel = 6

mnul1.Checked = False

mnul2.Checked = False


```
mnul3.Checked = False  
mnul4.Checked = False  
mnul5.Checked = False  
mnul6.Checked = True  
mnul7.Checked = False
```

End Sub

' Khi chọn chức năng Level 7 trong menu Options

Private Sub mnul7_Click()

```
nlevel = 7  
mnul1.Checked = False  
mnul2.Checked = False  
mnul3.Checked = False  
mnul4.Checked = False  
mnul5.Checked = False  
mnul6.Checked = False  
mnul7.Checked = True
```

End Sub

' Khi chọn chức năng Chơi tính giờ trong menu Options

Private Sub mnuthoigian_Click()

```
mnudiem.Checked = False ' bỏ dấu chọn trước mục Chơi tính điểm  
mnuthoigian.Checked = True ' đánh dấu chọn trước mục Chơi tính giờ  
nhinhthuc = 2 ' chuyển hình thức chơi thành chơi tính giờ  
nlevel = 1 ' đặt mức độ chơi khởi đầu là 1  
mnul1.Visible = False ' ẩn các mục Level trong menu Options  
mnul2.Visible = False ' để người chơi bắt buộc phải chơi từ thấp đến cao  
mnul3.Visible = False  
mnul4.Visible = False  
mnul5.Visible = False
```

```
mnul6.Visible = False
```

```
mnul7.Visible = False
```

End Sub

' Khi chọn chức năng Chơi tính điểm trong menu Options

Private Sub mnudiem_Click()

```
mnudiem.Checked = True      ' Đặt dấu chọn trước mục Chơi tính điểm
```

```
mnuthoigian.Checked = False ' Bỏ dấu chọn trước mục Chơi tính giờ
```

```
nhinhthuc = 1              ' chuyển hình thức chơi thành chơi tính điểm
```

```
nlevel = 1                 ' đặt mức độ chơi mặc định là 1
```

```
mnul1.Visible = True      ' bật các mục Level trong menu Options
```

```
mnul2.Visible = True      ' người chơi có thể chọn một mức độ chơi tùy thích
```

```
mnul3.Visible = True
```

```
mnul4.Visible = True
```

```
mnul5.Visible = True
```

```
mnul6.Visible = True
```

```
mnul7.Visible = True
```

End Sub

' Khi chọn lệnh Games ➡ Quit Games hay nhấn tổ hợp phím CTRL + Q.

Private Sub mnuquit_Click()

```
End
```

End Sub

' Khi người chơi nhấp mouse lên một ô có hiệu lực trên không gian chơi.

Private Sub sword_Click(Index As Integer)

```
sword(Index).Caption = ochua(Index) ' Từ sẽ được lật
```

```
nlat = nlat + 1                    ' tăng số từ đang được lật lên 1
```

```
' nếu số từ đang được lật hiện hành bằng 2, thì đặt trở về 0
```

```
' điều này đảm bảo trong không gian chơi không bao giờ có quá hơn
```

```
' 2 từ được lật
```

```
If nlat = 2 Then
```

```
nlat = 0
' nếu từ được lật đầu tiên là từ đồng nghĩa với từ vừa lật
If studn = ochua(Index) Then ' Kiểm tra xem có cùng cặp từ đồng nghĩa không
sword(Index).Visible = False ' nếu phải, loại bỏ cả 2 ô khỏi không gian chơi
sword(nvtcu).Visible = False
totals = totals - 1 ' giảm số cặp từ chưa lật đi 1
Else ' ngược lại
MsgBox "Sai rồi" ' hiển thị thông báo "Sai rồi"
sword(Index).Enabled = True
sword(nvtcu).Enabled = True
sword(Index).Caption = "" ' úp từ trở lại và chờ lần
sword(nvtcu).Caption = "" ' lật tiếp theo của người chơi
ndiem = ndiem - 1 ' trừ bớt của người chơi 1 điểm
End If
' Kiểm tra xem đã lật hết tất cả các cặp từ đồng nghĩa hay chưa
If totals = 0 Then ' nếu rồi
If nhinhthuc = 1 Then ' và hình thức chơi là "Chơi tính điểm"
MsgBox "Điểm của bạn là : " & Str(ndiem) ' hiển thị kết quả điểm
Else ' Nếu hình thức chơi là "Chơi tính giờ"
If nlevel = 7 Then ' nếu đã chơi tới mức cuối cùng
MsgBox "Congratulation !" ' hiển thị thông báo chúc mừng
Else ' nếu chưa chơi tới mức cuối cùng
nlevel = nlevel + 1 ' chuyển lên mức chơi cao hơn
mnunew_Click ' bắt đầu mức chơi mới
End If
nsogiyay = 0 ' bắt đầu tính lại thời gian chơi ở mức mới
End If
End If
```

```

Else          ' nếu không gian chơi mới có một từ được lật
ntucu = ochua(Index) ' lưu lại nội dung và vị trí từ đã lật
nvtcu = Index
' Tìm từ đồng nghĩa với từ đang lật và lưu vào biến studn
For i = 0 To totals2 / 2 - 1
If dword(tuchon(i + 1), 0) = ntucu Then
studn = dword(tuchon(i + 1), 1)
Exit For
Else
If dword(tuchon(i + 1), 1) = ntucu Then
studn = dword(tuchon(i + 1), 0)
Exit For
End If
End If
Next
End If
End Sub

' Khi chọn lệnh Games ➡ New Games hay nhấn tổ hợp phím Ctrl+N
Private Sub mnunew_Click()
For i = 0 To 59
sword(i).Visible = True          ' hiển thị lại toàn bộ không gian chơi
sword(i).Caption = ""            ' bỏ các từ đã lật nếu có
Next
' Gọi thủ tục khoitao để tiến hành khởi tạo lại không gian chơi mới
' chọn và phân bố từ cho các ô có hiệu lực ứng với mức độ chơi đã chọn
khoitao
End Sub

' Khi người chơi đang chọn hình thức Chơi tính giờ
' thủ tục sẽ được gọi sau mỗi giây trôi qua

```

Private Sub Timer1_Timer()

 nsogiay = nsogiay + 1

 If nsogiay >= thoigian(nlevel) Then

 MsgBox "Đã hết thời gian chơi. Bạn thua rồi !"

 nsogiay = 0

 mnunew.Click

 End If

End Sub

Bước 6. Chọn Start ➡ Run hay nhấn F5 để chạy thử chương trình.

Nếu mọi thứ đều như ý, chọn lệnh File ➡ Make Timcap2.exe để compile chương trình thành tập tin **Timcap2.exe** có thể thi hành trên môi trường Windows 98, không cần khởi động Visual Basic.

V. Phân tích chương trình

1. Ý nghĩa của các mảng và biến toàn cục

- ♦ Mảng 2 chiều **dword**, lưu thông tin 40 cặp từ đồng nghĩa được đọc ra từ cơ sở dữ liệu.
- ♦ Mảng **quydoi** xác định các ô có hiệu lực tương ứng với mức độ chơi hiện hành.
- ♦ Mảng **khonggian** lưu vị trí các ô có hiệu lực trong không gian chơi.
- ♦ Mảng **diem** xác định điểm số tối đa người chơi có thể đạt ứng với từng mức độ chơi.
- ♦ Mảng **thoigian** xác định số giây tối đa người chơi phải hoàn tất việc lật ô ứng với mức độ chơi hiện hành.

- ♦ Mảng **socaptu** xác định tổng số cặp từ được sử dụng trong không gian chơi.
Lưu ý rằng số cặp từ sử dụng = số ô có hiệu lực / 2.
- ♦ Mảng **Tuchon** chứa thứ tự các từ được chọn trong tập hợp những cặp từ có sẵn (mảng **dWord**) ứng với mức độ chơi hiện hành.
- ♦ Mảng **ochua** lưu vị trí từ được chọn trong tập hợp các cặp từ sẽ hiển thị lên mảng điều khiển **sWord**, tại vị trí tương ứng.
- ♦ Mảng điều khiển Command button **sWord** dùng làm không gian chơi.
- ♦ Biến **nlevel** cho biết mức độ chơi hiện hành.
- ♦ Biến **nhinhthuc** cho biết hình thức chơi hiện hành là tính điểm hay tính giờ.
- ♦ Biến **totals** xác định tổng số cặp từ được chọn sử dụng cho lượt chơi.
- ♦ Biến **totals2** xác định tổng số ô có hiệu lực trong không gian chơi.
- ♦ Biến **nlat** xác định từ vừa lật là từ đã lật đầu tiên trong không gian chơi hay từ thứ 2.
- ♦ Biến **studn** lưu từ đồng nghĩa với từ đã được lật trước đó.
- ♦ Biến **ntucu** lưu vị trí từ lật đầu tiên trong tập hợp từ đã chọn.
- ♦ Biến **nvtcu** lưu vị trí ô chứa từ đầu tiên đã lật.
- ♦ Biến **ndiem** lưu số điểm còn lại của người chơi tới thời điểm hiện tại.
- ♦ Biến **nsoglay** lưu số giây trôi qua kể từ khi bắt đầu mức độ chơi mới khi người chơi chọn hình thức Chơi tính giờ.

2. Các hình sau minh họa cho một số kết quả chọn và phân bố những cặp từ đồng nghĩa vào các ô trong không gian chơi tùy theo mức độ chơi khác nhau.

Tìm cặp 2	
Games	Options
Outdated	Sweetheart
Demoded	Lover

Hình 5.8 Kết quả chọn và phân bố từ đồng nghĩa một cách ngẫu nhiên ở mức chơi 1.

Tìm cặp 2	
Games	Options
Fine	Silly
Demoded	Outdated
Lonely	Solitary

Hình 5.9 Kết quả chọn và phân bố từ đồng nghĩa một cách ngẫu nhiên ở mức chơi 2.

Tim cap 2			
Games	Options		
Upset	Come along	Hurry up	Awkwar
Haft-asleep	Lonely	Fordone	Overturn
Haft-awake	Exhausted	Whimsical	Solitary

Hình 5.10 Kết quả chọn và phân bố từ đồng nghĩa một cách ngẫu nhiên ở mức chơi 3.

Tim cap 2					
Games	Options				
Uncertain	Flock	Hurry up	Finally	Maiden	Fine
Dishonest	Clearly	At last	Distinctly	Outsider	Haft-asleep
Awkwar	Unsteady	Haft-awake	Unstable	Silky	Onceitful
Whimsical	Unrelied person	Fainlady	Doubtful	Come along	Stream

Hình 5.11 Kết quả chọn và phân bố từ đồng nghĩa một cách ngẫu nhiên ở mức chơi 4.

Tìm cặp 2							
Games				Options			
Fine	Petty	Complex	Human livelihood	Loveable	Project	Human life	Stream
Hero	Overflow	Boil over	High-born	Flock	Complicate	Likeable	He-man
Scheme	Poor	Solitary	Doubtful	Uncertain	Half-back	Forward	Aristocratic
Silky	Fairlady	Genuine	Sweetheart	Unusual talent	Lonely	Lover	Maiden

Hình 5.12 Kết quả chọn và phân bố từ đồng nghĩa một cách ngẫu nhiên ở mức chơi 5.

Tìm cặp 2									
Games					Options				
Sky-high	Shy	Project	Outdated	Flock	Hard	Petty	Silky	Stream	Likeable
Unusual talent	Sky-scraper	Generalize	Scheme	Human livelihood	Sketch out	Clearly	Invernalize	Maiden	Band
He-man	Demoted	Initial	Deceitful	Distinctly	Gang	Dishonest	Fordone	Exhausted	Half-back
Outsider	Fine	Uncertain	Poor	Half-asleep	Unrelated person	Human life	Approach	Fairlady	Bashful
Expert	Specialist	Loveable	Boil over	Forward	Genuine	Overflow	Doubtful	Half-awake	Draw out

Hình 5.13 Kết quả chọn và phân bố từ đồng nghĩa một cách ngẫu nhiên ở mức chơi 6.

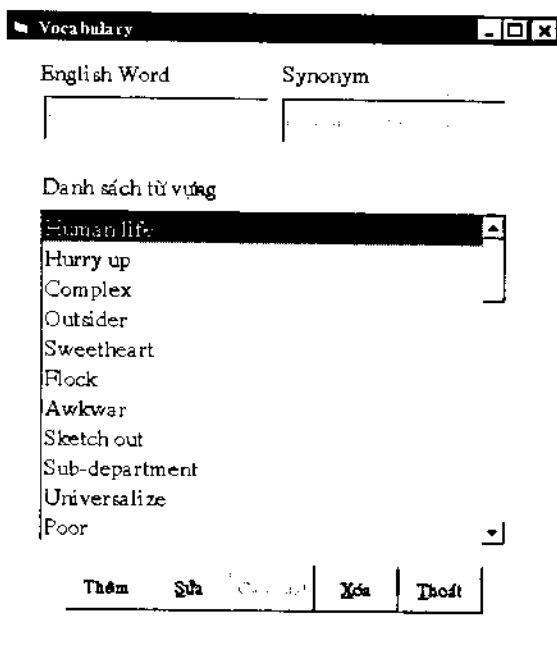
Tim cap 2									
Games					Options				
Haft-awake	Gang	Boil over	Unstable	Come along	Sketch out	Lover	Petty	Unsteady	Awkwar
Uncertain	Outdated	Overflow	Shift	Unusual talent	Shy	Bashful	Human life	Stream	Flock
Exhausted	Project	He-man	high-mindex	Haft-back	Fordone	Hurry up	Finally	Overhan	Fine
Task	At last	Complex	Sky-high	Sky-scrapes	Hero	Forward	Unrelated person	Human livelihood	Whimsical
Silky	Noble	Gear	Clearly	Fairledy	esponsibilit	Poor	Draw out	Maiden	Upset
Sweetheat	Distinctly	Haft-asleep	Band	Scheme	Complicates	Doubtful	Demoded	Outsider	Genuine

Hình 5.14 Kết quả chọn và phân bố từ đồng nghĩa một cách ngẫu nhiên ở mức chơi 7.

VI. Thiết kế form nhập liệu *frmInput*

Form *frmInput* dùng để cập nhật, xóa, điều chỉnh và lưu trữ danh sách các cặp từ đồng nghĩa vào table *Synonyms* trong cơ sở dữ liệu Access *Vocabulary.mdb* (Hình 5.15).

- ◆ Khi form được nạp (load lên), danh sách từ vựng trong cơ sở dữ liệu sẽ hiển thị trong Listbox “Danh sách từ vựng”, button “Cập nhật” không có tác dụng.

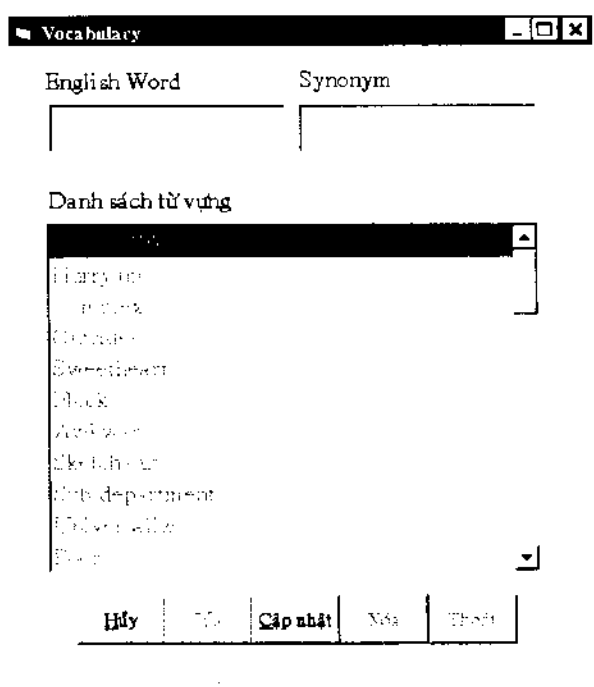


Hình 5.15 Màn hình form frmInput khi mới được nạp.

- ◆ Khi người dùng nhấp mouse lên một mục trong Listbox, nội dung từ vựng và từ đồng nghĩa tương ứng sẽ được hiển thị trong Text box "English Word" và "Synonyms".
- ◆ Khi nhấp button "Thêm", caption button sẽ được đổi thành "Hủy", tất cả button khác đều không có tác dụng trừ button "Cập nhật" (Hình 5.16). Nhấp lên button "Hủy" để hủy bỏ thao tác thêm. Khi nhấp lên button "Cập nhật", mẫu tin mới nhập trong text box sẽ được thêm vào table "Synonyms" và giao diện trở lại trạng thái bình thường như khi mới load form.
- ◆ Khi nhấp button "Sửa", caption cũng được đổi thành "Hủy", tất cả button khác đều không có tác dụng trừ button "Cập nhật". Nhấp lên button "Hủy" để hủy bỏ thao tác sửa. Khi nhấp lên

button “Cập nhật”, nội dung mẫu tin đang được chọn từ Listbox sẽ được cập nhật mới và giao diện trở lại trạng thái bình thường như khi mới load form. Nếu chưa nhập từ vựng hay từ đồng nghĩa với nó thì form hiển thị hộp thông báo “Các field không thể trống” và trở lại form để đợi người dùng nhập giá trị mới hay tiến hành thao tác “Hủy”. Bất kỳ sự thay đổi nào trên mẫu tin được chọn, sau khi “Cập nhật” sẽ phản ánh tức thì lên Listbox.

- ◆ Nhấp button “Xóa” để xóa “từ vựng” đang chọn khỏi Listbox và cơ sở dữ liệu. Từ vựng bị xóa trong table Synonyms, cũng lập tức bị loại bỏ khỏi Listbox.
- ◆ Nhấp button “Thoát” để kết thúc chương trình.



Hình 5.16 Giao diện form *frmInput* sau khi nhấp button “Thêm”.

Bước 1. Tạo form frmInput

Khởi động Visual Basic. Chọn lệnh File ➡ Open Project để mở lại project **Timcap2**. Chọn lệnh Project ➡ Add form, nhấp đúp biểu tượng form trong cửa sổ New để chèn một form mới vào project. Định lại các thuộc tính cho form mới như sau:

Name : frmInput

Caption : Vocabulary

Lưu form mới với tên **Input.frm**.

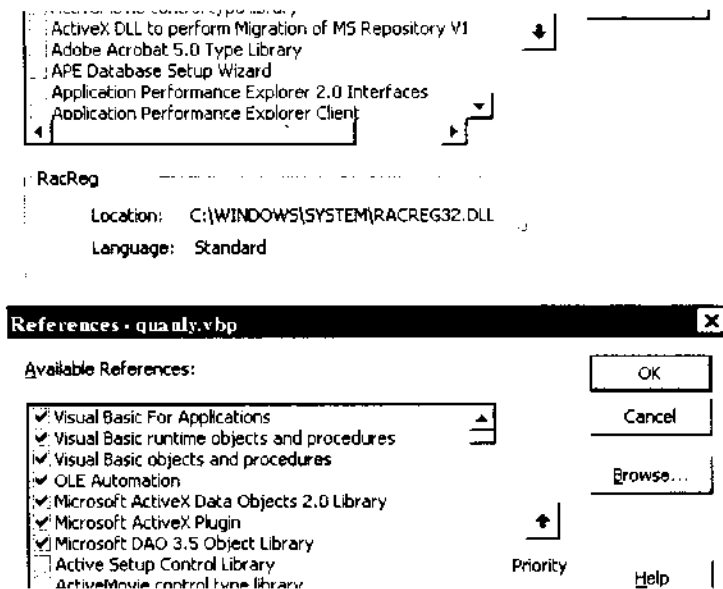
Bước 2. Thiết kế giao diện

- ♦ Đặt lên form 3 Label với Caption là “English Word”, “Synonyms” và “Danh sách từ vựng”.
- ♦ Đặt lên form 2 Text box và định lại các thuộc tính sau:
 Name = txteng, Text = (rỗng), Locked = False, Font = VNI-Times
 Name = txtsyn, Text = (rỗng), Locked = False, Font = VNI-Times
- ♦ Đặt lên form 1 Listbox và định lại các thuộc tính sau:
 Name = lstword, List = (rỗng), Font = VNI-Times
- ♦ Đặt lên form 5 Command button và thiết đặt các thuộc tính sau:

Name	Caption
CmdThem	Thêm
CmdSua	&Sửa
Cmdcn	&Cập nhật
CmdXoa	&Xóa
CmdThoat	&Thoát

Bước 3. Bổ sung thư viện đối tượng Microsoft DAO vào project.

Chọn lệnh Project ➡ References, đánh dấu ô chọn Microsoft DAO 3.5 Object Library trong cửa sổ References (Hình 5.17). Xong nhấp OK.



Hình 5.17 Cửa sổ References.

MÁCH NƯỚC

- ✓ Chỉ khi bạn bổ sung thư viện đối tượng Microsoft DAO 3.5 vào project, Visual Basic mới có thể hiểu được lệnh `OpenDatabase`.

Bước 4. Viết mã lệnh

‘ Khi nhấp button Cập nhật

`Private Sub Cmdcn_Click()`

```
If txteng.Text = "" Or txtsyn = "" Then
    MsgBox "Cac field khong the trong"
Exit Sub
End If

' Nếu nhấp nút Cập nhật khi form ở chế độ Thêm mới
If Cmdthem.Caption = "&Hủy" Then
    Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")
    Set rsSynonyms = dbs.OpenRecordset("Synonyms")
    rsSynonyms.AddNew
    rsSynonyms!english = txteng.Text
    rsSynonyms!Synonyms = txtsyn.Text
    rsSynonyms.Update
    Istword.AddItem txteng.Text
    rsSynonyms.Close
    Cmdthem.Enabled = True
    Cmdsua.Enabled = True
    Cmdxoa.Enabled = True
    Cmdcn.Enabled = False
    Cmdthoat.Enabled = True
    Cmdthem.Caption = "Thêm"
    Istword.Enabled = True
End If

' Nếu nhấp nút Cập nhật khi form ở chế độ Sửa
If Cmdsua.Caption = "&Hủy" Then
    Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")
    Set rsSynonyms = dbs.OpenRecordset("Synonyms")
    oldcv = Istword.List(Istword.ListIndex)
    SQL = "Select * from Synonyms where english = '" & oldcv & "'"
```

```
Set rsSynonyms = dbs.OpenRecordset(SQL, dbOpenDynaset)
rsSynonyms.Edit
rsSynonyms!english = txteng.Text
rsSynonyms!Synonyms = txtsyn.Text
rsSynonyms.Update
rsSynonyms.Close
Cmdthem.Enabled = True
Cmdsua.Enabled = True
Cmdxoax.Enabled = True
Cmdcn.Enabled = False
Cmdthoat.Enabled = True
Cmdsua.Caption = "&Sửa"
Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")
Set rsSynonyms = dbs.OpenRecordset("Synonyms")
lstword.Clear
If rsSynonyms.EOF = False Then
    While Not rsSynonyms.EOF
        lstword.AddItem rsSynonyms!english
        rsSynonyms.MoveNext
    Wend
End If
rsSynonyms.Close
End If
txteng.Enabled = False
txtsyn.Enabled = False
End Sub
' Khi nhấp nút Sửa
Private Sub Cmdsua_Click()
```



```
If Cmdsua.Caption = "&Sửa" Then
    Cmdsua.Caption = "&Hủy"
    Cmdthem.Enabled = False
    Cmdxoa.Enabled = False
    Cmdcn.Enabled = True
    Cmdthoat.Enabled = False
    txteng.Enabled = True
    txtsyn.Enabled = True
    txteng.SetFocus
```

Else

```
    Cmdsua.Caption = "&Sửa"
    Cmdthem.Enabled = True
    Cmdxoa.Enabled = True
    Cmdcn.Enabled = False
    Cmdthoat.Enabled = True
    txteng.Enabled = False
    txtsyn.Enabled = False
```

End If

End Sub

' Nếu nhấp nút Thêm

Private Sub Cmdthem_Click()

```
If Cmdthem.Caption = "Thêm" Then
    Cmdthem.Caption = "&Hủy"
    Cmdsua.Enabled = False
    Cmdxoa.Enabled = False
    Cmdcn.Enabled = True
    Cmdthoat.Enabled = False
    txteng.Enabled = True
```

```
txteng.Text = ""  
txtsyn.Enabled = True  
txtsyn.Text = ""  
lstword.Enabled = False  
txteng.SetFocus  
Else  
    Cmdthem.Caption = "Thêm"  
    Cmdsua.Enabled = True  
    Cmdxoá.Enabled = True  
    Cmdcn.Enabled = False  
    Cmdthoat.Enabled = True  
    txteng.Enabled = False  
    txtsyn.Enabled = False  
    lstword.Enabled = True  
End If
```

End Sub

' Nếu nhấp nút Thoát

Private Sub Cmdthoat_Click()

Unload Me ' đóng form, trở lại menu chính

End Sub

' Nếu nhấp nút Xóa

Private Sub Cmdxoá_Click()

```
Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")  
SQL = "Select * from Synonyms where english = '" & txteng.Text & "'" & ""  
Set rsSynonyms = dbs.OpenRecordset(SQL, dbOpenDynaset)  
rsSynonyms.Delete  
rsSynonyms.Close  
lstword.SetFocus
```

```

If lstword.ListIndex < 1 Then
    Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")
    Set rsSynonyms = dbs.OpenRecordset("Synonyms")
    lstword.Clear
    If rsSynonyms.EOF = False Then
        While Not rsSynonyms.EOF
            lstword.AddItem rsSynonyms!english
            rsSynonyms.MoveNext
        Wend
    End If
    rsSynonyms.Close
Else
    lstword.RemoveItem (lstword.ListIndex)
End If
txteng.Text = ""
txteng.Enabled = False
txtsyn.Text = ""
txtsyn.Enabled = False

```

End Sub

· Khi form được nạp

Private Sub Form_Load()

```

Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")
Set rsSynonyms = dbs.OpenRecordset("Synonyms")
lstword.Clear
If rsSynonyms.EOF = False Then
    While Not rsSynonyms.EOF
        lstword.AddItem rsSynonyms!english
        rsSynonyms.MoveNext
    
```

```

Wend
End If
rsSynonyms.Close
Cmdcn.Enabled = False

```

End Sub

' Khi chọn một từ vựng trong Listbox

Private Sub lstword_Click()

```

txteng = lstword.Text
Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")
SQL = "Select * from Synonyms where english = '" & txteng & "'"
Set rsSynonyms = dbs.OpenRecordset(SQL, dbOpenDynaset)
If rsSynonyms.EOF = False Then
    txtsyn = rsSynonyms!Synonyms
Else
    txtsyn = ""
End If
rsSynonyms.Close

```

End Sub**Bước 5.** Kết nối form vào chương trình chính

Mở lại form *frmTimcap2*. Bổ sung đoạn lệnh sau vào cửa sổ Code của form:

Private Sub mnulInput_Click()

```
FrmInput.Show 1
```

End Sub*** Giải thích chương trình**

- ♦ Thuộc tính Enabled của Command button cho phép lập trình viên dễ dàng vô hiệu hóa tạm thời tác dụng của một nút lệnh.

- ♦ Có thể tạo một đối tượng cơ sở dữ liệu tham chiếu tới một cơ sở dữ liệu Access và mở nó bằng lệnh OpenDatabase.

Ví dụ: *Set dbs = OpenDatabase(App.Path & "\Vocabulary.mdb")*

- ♦ Có thể tạo một đối tượng recordset tham chiếu tới một table bằng phương thức OpenRecordset của đối tượng cơ sở dữ liệu Access đang mở.

Ví dụ: *Set rsSynonyms = dbs.OpenRecordset("Synonyms")*

- ♦ Có thể tạo một đối tượng recordset tham chiếu tới một tập hợp dữ liệu của một table bằng cách kết hợp câu lệnh SQL trong phương thức OpenRecordset của đối tượng cơ sở dữ liệu Access đang mở.

Ví dụ: *SQL = "Select * from Synonyms where english = 'Glutinous'"*

Set rsSynonyms = dbs.OpenRecordset(SQL, dbOpenDynaset)

- ♦ Thêm một record vào table:

Dùng phương thức Addnew của đối tượng Recordset để thêm một record vào table.

Ví dụ: Thêm vào table record mới có nội dung là English = 'Charming', Synonyms = "Lovely"

.....

Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")

Set rsSynonyms = dbs.OpenRecordset("Synonyms")

rsSynonyms.AddNew

rsSynonyms!english = 'Charming'

rsSynonyms!Synonyms = 'Lovely'

rsSynonyms.Update

rsSynonyms.Close

- ♦ Sửa nội dung một record trong table:

Dùng phương thức Edit của đối tượng Recordset để sửa đổi nội dung một record.

Ví dụ: Điều chỉnh nội dung record có từ 'Harmoniou' trong table Synonyms thành 'Harmonious'

.....

```
Set dbs = OpenDatabase(App.Path & "\Vocabulary.mdb")
oldword = "Harmoniou"
SQL = "Select * from Synonyms where english = '" & oldword & "'"
Set rsSynonyms = dbs.OpenRecordset(SQL, dbOpenDynaset)
rsSynonyms.Edit
rsSynonyms!chucvu = 'Harmonious'
rsSynonyms.Update
rsSynonyms.Close
```

- ♦ Xóa một record khỏi table:

Dùng phương thức Delete của đối tượng recordset để xóa một record khỏi table.

Ví dụ: Xóa record chứa từ 'Harmonious' khỏi table Synonyms.

.....

```
Set dbs = OpenDatabase(App.Path & "\Vocabulary.mdb")
SQL = "Select * from Synonyms where english = 'Harmonious'"
Set rsSynonyms = dbs.OpenRecordset(SQL, dbOpenDynaset)
rsSynonyms.Delete
rsSynonyms.Close
```

VII. Thiết kế form tra cứu từ đồng nghĩa *frmTC*

Tra cứu từ đồng nghĩa trong cơ sở dữ liệu Access.

- Form *FRMTC* có giao diện như Hình 5.18. Người dùng có thể sử dụng form để nhập điều kiện lọc dữ liệu, tìm và tra cứu thông tin nhân viên trong table *Synonyms* của cơ sở dữ liệu Access *vocabulary.mdb*.

Hình 5.18 Màn hình form *FRMTC*.

- Form có thể được gắn vào chức năng "Vocabulary" trong menu chính của chương trình *Timcap2*. Khi nhấp mục "Games" > "Vocabulary", form *frmTC* sẽ được nạp.
- Trạng thái của form sẽ chuyển đổi qua lại giữa 2 chế độ "hiển thị thông tin thỏa điều kiện lọc" hoặc chế độ "nhập điều kiện lọc" khi nhấp button "Lọc". Khi đó thuộc tính Caption của button cũng chuyển đổi qua lại giữa "Lọc" và "Tìm".
- Khi ở chế độ "hiển thị thông tin thỏa điều kiện lọc", form sẽ hiển thị thông tin những từ vựng và từ đồng nghĩa thỏa điều kiện lọc. Khi ở chế độ "nhập điều kiện lọc", form dùng để nhập từ vựng hay từ đồng nghĩa cần tra cứu.

- ◆ Nếu có nhiều hơn một từ vựng thỏa điều kiện lọc, thì từ đầu tiên thỏa điều kiện lọc sẽ được hiển thị. Người dùng có thể nhấp button ">>" để chuyển tới từ tiếp theo. Và nhấp button "<<" để chuyển trở lại từ thỏa điều kiện trước đó.
- ◆ Nếu không có từ vựng nào thỏa điều kiện lọc, thì form sẽ không hiển thị thông tin nào và 2 button ">>", "<<" bị vô hiệu hóa.
- ◆ Mặc định khi form vừa được nạp hoặc khi không nhập từ vựng cần tìm nào thì tất cả các từ vựng sẽ được hiển thị.
- ◆ Khi nhấp button "Thoát", form sẽ trở lại màn hình chính.
- ◆ Khi nhấp button "Lọc", form sẽ chuyển sang chế độ nhập điều kiện lọc như Hình 5.19.

Hình 5.19 Tìm từ đồng nghĩa với từ "Project" với chế độ "nhập điều kiện lọc" của form FRMTC.

- ◆ Sau khi nhập điều kiện lọc, nhấp button "Tìm" để chương trình tìm kiếm các từ đồng nghĩa thỏa điều kiện lọc và hiển thị thông tin từ đồng nghĩa đầu tiên thỏa điều kiện lọc. Form trở về chế độ "Hiển thị thông tin thỏa điều kiện lọc".

- ♦ Lưu ý rằng có thể nhập cả từ vựng và từ đồng nghĩa. Khi đó, chỉ khi bạn nhập đúng cặp từ đồng nghĩa thông tin mới được giữ lại lúc bạn nhấp nút "Tìm". Nếu xóa trống, không nhập bất kỳ điều kiện nào có nghĩa là bỏ lọc. Có thể truy xuất trở lại tới mọi record trong table *Synonyms*.

Bước 1. Mở lại project *TIMCAP2*.

Chọn lệnh Project ➡ Add Form để bổ sung một form mới vào project.

Chọn lệnh File ➡ Save form1 As..., lưu form với tên *FRMTC.frm*.

Bước 2. Điều chỉnh thủ tục biến cố trong form chính *mnuView_click*.

```
Private Sub mnuView_Click()  
    frmTC.Show 1  
End Sub
```

Bước 3. Thiết kế giao diện cho form *frmTC*.

- ♦ Chọn form và định lại các thuộc tính form như sau:
Name = *frmTC*, **Caption** = *Tra cứu từ vựng*, **WindowState** = 0 – *Normal*.
- ♦ Đặt lên form 1 Frame. Định lại các thuộc tính cho Frame như sau:
Frame 1: **Name** = *Frame1*, **BorderStyle** = 1 – *Fixed Single*,
Backcolor = &H00E0E0E0&, **Caption** = *Thông tin từ đồng nghĩa*
- ♦ Lần lượt đặt 2 Label vào Frame1 với những thuộc tính sau:

Name	Caption	Backstyle
Label1	English	0 – Transparent
Label2	Synonyms	0 – Transparent

- ◆ Đặt 2 Text box vào Frame1. Định lại các thuộc tính cho Text box như sau:

Name	Text	Font
TxtEng	(rỗng)	VNI-Times
TxtSyn	(rỗng)	VNI-Times

- ◆ Đặt lên form 4 Command button và định lại các thuộc tính sau:

Name	Caption	Font
Cmdloc	Lọc	VNI-Times
CmdPrev	<<	VNI-Times
CmdNext	>>	VNI-Times
Cmdthoat	Thoát	VNI-Times

Bước 4. Viết mã lệnh xử lý biến cố cho form *FrmTC*:

Dim nrecc, ncurr, feng, fsyn

Dim rssyn As Recordset

Dim dbs As Database

' Khi nhấp nút "Lọc" hoặc "Tìm"

Private Sub cmdloc_Click()

If cmdloc.Caption = "Lọc" Then

cmdloc.Caption = "Tìm"

txteng = feng

Txtsyn = fsyn

txteng.Enabled = True

Txtsyn.Enabled = True

cmdprev.Enabled = False

cmdnext.Enabled = False

```
Else
    cmdloc.Caption = "Lọc"
    feng = txteng
    fsyn = Txtsyn
    rssyn.Close
    dieukien = ""
    If feng <> "" Then
        dieukien = dieukien & "english = " & Trim(feng) & ""
    If fsyn <> "" Then
        dieukien = dieukien & " and Synonyms = " & Trim(fsyn) & ""
    End If
Else
    If fsyn <> "" Then
        dieukien = dieukien & "Synonyms = " & Trim(fsyn) & ""
    End If
End If
If dieukien <> "" Then
    dieukien = "where " & dieukien
End If
SQL = "Select * from Synonyms " & dieukien
Set rssyn = dbs.OpenRecordset(SQL, dbOpenDynaset)
If rssyn.RecordCount = 0 Then
    txteng = ""
    Txtsyn = ""
    cmdprev.Enabled = False
    cmdnext.Enabled = False
Else
    txteng = rssyn!english
```

```
Txtsyn = rssyn!Synonyms
```

```
cmdprev.Enabled = True
```

```
cmdnext.Enabled = True
```

```
End If
```

```
txteng.Enabled = False
```

```
Txtsyn.Enabled = False
```

```
End If
```

```
End Sub
```

```
' Khi nhấp nút ">>"
```

```
Private Sub cmdnext_Click()
```

```
rssyn.MoveNext
```

```
cmdprev.Enabled = True
```

```
If rssyn.EOF Then
```

```
    rssyn.MoveLast
```

```
    cmdnext.Enabled = False
```

```
Exit Sub
```

```
End If
```

```
txteng = rssyn!english
```

```
Txtsyn = rssyn!Synonyms
```

```
End Sub
```

```
' Khi nhấp nút "<<"
```

```
Private Sub cmdprev_Click()
```

```
rssyn.MovePrevious
```

```
cmdnext.Enabled = True
```

```
If rssyn.BOF Then
```

```
    rssyn.MoveFirst
```

```
    cmdprev.Enabled = False
```

```
Exit Sub
```

```

End If
txteng = rssyn!english
Ttxtsyn = rssyn!Synonyms
End Sub
' Khi nhấp nút "Thoát"
Private Sub Cmdthoat_Click()
Unload Me
End Sub
' Khi form được nạp
Private Sub Form_Load()
Set dbs = OpenDatabase(App.Path & "\vocabulary.mdb")
Set rssyn = dbs.OpenRecordset("Synonyms")
If rssyn.RecordCount = 0 Then
    cmdprev.Enabled = False
    cmdnext.Enabled = False
    cmdloc.Enabled = False
Else
    txteng = rssyn!english
    Ttxtsyn = rssyn!Synonyms
End If
txteng.Enabled = False
Ttxtsyn.Enabled = False
End Sub

```

Bước 5. Giải thích chương trình

Có thể sử dụng câu lệnh SQL trong tham số đầu của phương thức OpenRecordset của đối tượng Databases để truy xuất dữ liệu có chọn lọc trong các table của cơ sở dữ liệu Access.

- ♦ Khối lệnh SELECT là câu lệnh cốt lõi của mọi truy vấn lấy về dữ liệu, giúp xác định những field nào sẽ được lấy về.

Khối lệnh gồm 3 mệnh đề chính: SELECT – FROM – WHERE. Ngoài ra khối lệnh Select còn được bổ sung thêm các mệnh đề GROUP BY, HAVING, ORDER BY, ... mà bạn sẽ gặp trong các bài thực hành sau.

- SELECT: Xác định nội dung các field, biểu thức, ... cần lấy dữ liệu.
- FROM: Xác định nguồn cần lấy dữ liệu, có thể là những table hay một câu lệnh truy vấn khác.
- WHERE: Chỉ ra các bộ lọc dữ liệu kết quả thỏa điều kiện, có thể dùng những toán tử logic để liên kết nhiều điều kiện lọc khác nhau trong mệnh đề where.

Ví dụ: Lấy thông tin từ vựng hoặc từ đồng nghĩa “Lovely” trong table Synonyms.

```
Select * from Synonyms  where english ='Lovely' or  
Synonyms='Lovely'
```

VIII. Bài tập thực hành

Dựa vào những kỹ thuật đã trình bày, hãy xây dựng một chương trình chơi ô chữ mà bạn đã từng chơi vào một dịp nào đó.

BÀI 6.

TRÒ CHƠI CARÔ

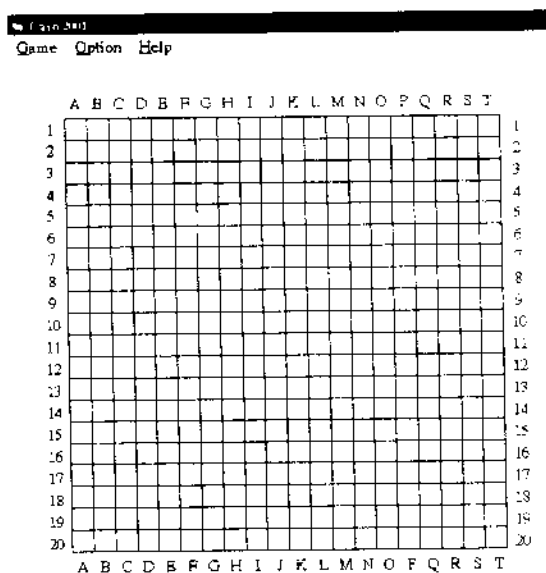
I. Mô tả trò chơi

Có thể nói trò chơi carô là một trong những trò chơi phổ biến mà hầu hết các bạn đều đã biết quy luật chơi. Chương trình có hai chế độ chơi:

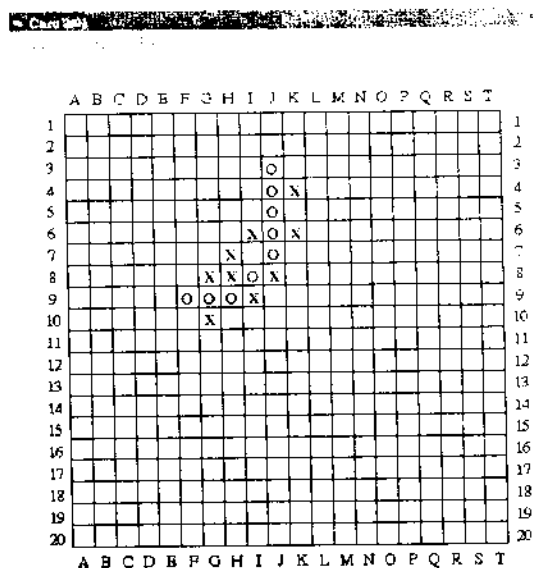
- ♦ Giữa hai người chơi với nhau, chương trình sẽ hiển thị màn hình carô để chơi thay vì phải chơi trên giấy và chương trình sẽ tự động kiểm tra và thông báo người thắng cuộc.
- ♦ Giữa người và máy, trong trường hợp này, người chơi có thể lựa chọn đi trước hoặc đi sau máy một nước.

Trước tiên chúng ta hãy thử điểm lại trò chơi định xây dựng :

- ♦ Không gian chơi có 400 ô phân thành 20 hàng $\times$ 20 cột (Hình 6.1)
- ♦ Tham dự trò chơi thường có hai người hoặc giữa người và máy. Trong trường hợp hai người chơi, mỗi bên sẽ lần lượt sử dụng mouse để thay phiên nhau đánh dấu vào các ô trống bằng ký tự quy định là "O" (người chơi sau) hoặc "X" (người chơi trước). Cuộc chơi sẽ kết thúc khi một trong hai bên đánh dấu được vào 5 ô liên nhau (theo chiều ngang, dọc hoặc trên hai đường chéo) mà không quan tâm chúng có bị chặn cả hai đầu hay không.

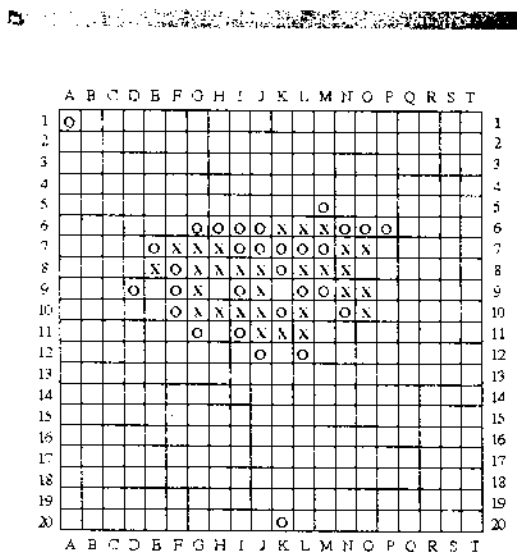


Hình 6.1 Không gian chơi carô 20×20.



Hình 6.2 Trò chơi kết thúc với phần thắng thuộc bên sử dụng ký tự "O".

- Trong trường hợp người chơi với máy. Máy tính sẽ tự động “suy nghĩ” và thực hiện việc đánh dấu khi đến lượt mình. Xong nhanh chóng trả quyền đánh dấu lại cho người chơi. Nếu chọn chế độ máy đi trước người chơi một nước, thì sau khi form được nạp, máy tính lập tức đi trước một nước, xong mới trả quyền đánh dấu cho người chơi. Ký tự đánh dấu của máy tính là “O” và của người chơi là “X”, dù máy tính đi trước hay sau người chơi một nước.



Hình 6.3 Phần thắng vẫn thuộc bên sử dụng ký tự “O” dù 5 ô đã đánh dấu liên tiếp (I7 đến M7) bị chặn hai đầu.

II. Các thuật toán xử lý

1. Đánh dấu ô trong không gian chơi

- Đánh số các ô trong không gian chơi theo thứ tự từ trái sang phải, hết hàng trên tiếp đến là hàng dưới. Mỗi ô lúc này sẽ tương ứng với một thành phần của mảng điều khiển Label *lbIO*.

- ♦ Sau khi chọn được ô cần đi bằng cách nhấp mouse (đối với người chơi) hoặc tự thi hành thuật toán *Timnuocdi* (đối với máy), tùy theo nước đi thuộc bên nào mà chương trình sẽ tự đặt vào ô ký tự thích hợp là "X" hoặc "O", đồng thời đổi màu ô từ vàng thành tím để người chơi dễ dàng nhận biết ô vừa được đi. Sau đó chuyển quyền chọn ô tiếp theo cho đối phương.

2. Đếm số ô liên tiếp cùng dấu bên trái ô hiện hành

Để ý rằng vị trí ô bên trái nhỏ hơn ô hiện hành một đơn vị.

Bước 1. Xác định dấu sử dụng của người chơi hiện hành.

Bước 2. Kiểm tra xem ô hiện hành có thuộc cột 0 (cột đầu tiên).

Bước 3. Nếu ô thuộc cột đầu tiên, kết thúc thuật toán và trả lại số dấu đếm được.

Ngược lại thực hiện tiếp bước 4.

Bước 4. Chuyển ô hiện hành sang trái một ô bằng cách giảm vị trí ô đi một đơn vị.

Bước 5. Nếu ô hiện hành cùng dấu với người chơi, tăng số dấu đếm được lên 1.

Xong lặp lại bước 2. Ngược lại nếu ô hiện hành chứa dấu khác với dấu người chơi sử dụng (dấu của đối phương) thì thực hiện tiếp bước 6.

Bước 6. Nếu ô hiện hành là trống (chưa chứa dấu nào) thì ghi nhận lại vị trí ô trống này (để làm cơ sở cho việc thực hiện nước đi kế tiếp). Xong kết thúc thuật toán và trả về số dấu đếm được.

3. Đếm số ô liên tiếp cùng dấu bên phải ô hiện hành

Để ý rằng vị trí ô bên phải lớn hơn ô hiện hành một đơn vị.

Bước 1. Xác định dấu sử dụng của người chơi hiện hành.

Bước 2. Kiểm tra xem ô hiện hành có thuộc cột 19 (cột cuối cùng) hay không.

Bước 3. Nếu ô thuộc cột 19, kết thúc thuật toán và trả lại số dấu đếm được.

Ngược lại thực hiện tiếp bước 4.

Bước 4. Chuyển ô hiện hành sang trái một ô bằng cách tăng vị trí ô lên 1 đơn vị.

Bước 5. Nếu ô hiện hành cùng dấu với người chơi, tăng số dấu đếm được lên 1.

Xong lặp lại bước 2. Ngược lại nếu ô hiện hành chứa dấu khác với dấu người chơi sử dụng (dấu của đối phương) thì thực hiện tiếp bước 6.

Bước 6. Nếu ô hiện hành là trống (chưa chứa dấu nào) thì ghi nhận lại vị trí ô trống này (để làm cơ sở cho việc thực hiện nước đi kế tiếp). Xong kết thúc thuật toán và trả về số dấu đếm được.

4. Đếm số ô liên tiếp cùng dấu ở trên ô hiện hành

Để ý rằng vị trí ô ở trên nhỏ hơn ô hiện hành 20 đơn vị.

Bước 1. Xác định dấu sử dụng của người chơi hiện hành.

Bước 2. Kiểm tra xem ô hiện hành có thuộc hàng 0 (hàng đầu tiên) hay không.

Bước 3. Nếu ô thuộc hàng đầu tiên, kết thúc thuật toán và trả lại số dấu đếm được.

Ngược lại thực hiện tiếp bước 4.

Bước 4. Chuyển ô hiện hành lên phía trên một hàng bằng cách giảm vị trí ô đi 20 đơn vị.

Bước 5. Nếu ô hiện hành cùng dấu với người chơi. Tăng số dấu đếm được lên 1.

Xong lặp lại bước 2. Ngược lại nếu ô hiện hành chứa dấu khác với dấu người chơi sử dụng (dấu của đối phương) thì thực hiện tiếp bước 6.

Bước 6. Nếu ô hiện hành là trống (chưa chứa dấu nào) thì ghi nhận lại vị trí ô trống này (để làm cơ sở cho việc thực hiện nước đi kế tiếp). Xong kết thúc thuật toán và trả về số dấu đếm được.

5. Đếm số ô liên tiếp cùng dấu ở dưới ô hiện hành

Để ý rằng vị trí ô ở dưới lớn hơn ô hiện hành 20 đơn vị.

Bước 1. Xác định dấu sử dụng của người chơi hiện hành.

Bước 2. Kiểm tra xem ô hiện hành có thuộc hàng 19 (hàng cuối cùng) hay không.

Bước 3. Nếu ô thuộc hàng cuối cùng, kết thúc thuật toán và trả lại số dấu đếm được. Ngược lại thực hiện tiếp bước 4.

Bước 4. Chuyển ô hiện hành xuống dưới một hàng bằng cách tăng vị trí ô đi 20 đơn vị.

Bước 5. Nếu ô hiện hành cùng dấu với người chơi, tăng số dấu đếm được lên 1.

Xong lặp lại bước 2. Ngược lại nếu ô hiện hành chứa dấu khác với dấu người chơi sử dụng (dấu của đối phương) thì thực hiện tiếp bước 6.

Bước 6. Nếu ô hiện hành là trống (chưa chứa dấu nào) thì ghi nhận lại vị trí ô trống này (để làm cơ sở cho việc thực hiện nước đi kế tiếp). Xong kết thúc thuật toán và trả về số dấu đếm được.

6. Đếm số ô liên tiếp cùng dấu ở phía trên bên trái ô hiện hành

Để ý rằng vị trí ô ở phía trên bên trái nhỏ hơn ô hiện hành 21 đơn vị. Và điều kiện kết thúc thuật toán là khi ô hiện hành nằm trong cột 0 (cột đầu tiên) hoặc hàng 0 (hàng đầu tiên) hoặc ô trống hay ô chứa ký tự khác với ký tự người chơi hiện hành.

Các bước thực hiện giải thuật tương tự như những thuật toán trên, thiết tưởng không cần phải nhắc lại chi cho rườm rà.

7. Đếm số ô liên tiếp cùng dấu ở phía dưới bên phải ô hiện hành

Để ý rằng vị trí ô ở phía dưới bên phải lớn hơn ô hiện hành 21 đơn vị. Và điều kiện kết thúc thuật toán là khi ô hiện hành nằm trong cột 19 (cột cuối cùng) hoặc hàng 19 (hàng cuối cùng) hoặc ô trống hay ô chứa ký tự khác với ký tự người chơi hiện hành.

8. Đếm số ô liên tiếp cùng dấu ở phía trên bên phải ô hiện hành

Để ý rằng vị trí ô ở phía trên bên phải nhỏ hơn ô hiện hành 19 đơn vị. Và điều kiện kết thúc thuật toán là khi ô hiện hành nằm trong cột 19 (cột cuối cùng) hoặc hàng 0 (hàng đầu tiên) hoặc ô trống hay ô chứa ký tự khác với ký tự người chơi hiện hành.

9. Đếm số ô liên tiếp cùng dấu ở phía dưới bên trái ô hiện hành

Để ý rằng vị trí ô ở phía dưới bên trái lớn hơn ô hiện hành 19 đơn vị. Và điều kiện kết thúc thuật toán là khi ô hiện hành nằm trong cột 0 (cột đầu tiên) hoặc hàng 19 (hàng cuối cùng) hoặc ô trống hoặc ô chứa ký tự khác với ký tự người chơi hiện hành.

10. Tính số ô liên tiếp cùng dấu bắt đầu từ vị trí ô vừa được đánh dấu

Để ý với mỗi ô vừa được đánh dấu, bạn cần tính số ô liên nhau theo bốn phương khác nhau:

- ♦ Phương ngang (từ trái sang phải) = số ô liên tiếp cùng dấu bên trái + 1 + số ô liên tiếp cùng dấu bên phải.
- ♦ Phương dọc (từ trên xuống dưới) = số ô liên tiếp cùng dấu ở trên + 1 + số ô liên tiếp cùng dấu ở dưới.
- ♦ Đường chéo thứ nhất (từ phía trên bên trái tới phía dưới bên phải) = số ô liên tiếp cùng dấu phía trên bên trái + 1 + số ô liên tiếp cùng dấu bên phải.
- ♦ Phương ngang (từ trái sang phải) = số ô liên tiếp cùng dấu bên trái + 1 + số ô liên tiếp cùng dấu bên phải.

Chỉ cần một trong bốn phương có số ô liên tiếp cùng dấu = 5 thì người chơi hiện hành (người vừa đánh dấu vào ô hiện hành) sẽ thắng cuộc.

11. Thuật toán TIMNUOCDI giúp tính nước đi tiếp theo cho máy tính trong trường hợp chơi giữa người và máy

Có thể nói đây là thuật toán quan trọng nhất trong trò chơi này. Một trò chơi carô có xứng đáng là đối thủ đáng gờm của những cao thủ chơi carô hay không (đối với chế độ chơi giữa người và máy) là hoàn toàn tùy vào mức độ “thông minh” của thuật toán này. Sau đây chỉ là một thuật toán mẫu, khá đơn giản để bạn làm quen.

Bước 1. Xác định phương có số lượng ô liên tiếp cùng dấu mà người chơi đã đánh dấu nhiều nhất, và có ít nhất một trong hai đầu máy tính chưa kịp chặn (*không xét các phương đã bị chặn hai đầu*).

Trong trường hợp có nhiều phương có cùng số lượng ô đã đánh bằng nhau, thì ưu tiên các phương theo thứ tự sau: “đường chéo thứ 2”, “đường chéo thứ nhất”, “phương đứng”, “phương ngang”.

Bước 2. Nếu cả bốn phương đều bị chặn cả hai đầu, thì đánh dấu vào một ô trống ngẫu nhiên. Còn như phương được chọn chỉ có một đầu chưa bị chặn, đánh dấu vào ô trống ở đầu chưa được chặn. Trường hợp phương được chọn chưa bị chặn cả hai đầu thì máy ưu tiên đánh dấu vào ô trống ở phía bên trái (phương ngang) hoặc ở trên (với những phương còn lại).

Nhận xét: Rõ ràng thuật toán này đơn giản chỉ là tìm chặn các đường tấn công của người chơi là chính, không cho người chơi kết thúc nhanh chóng ván chơi mà không có khuynh hướng tấn công. Vì thế máy tính sẽ dễ dàng bị hạ bởi các cao thủ carô.

III. Thiết kế chương trình

1. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start ➡ Programs ➡ Microsoft Visual Studio 6.0 ➡ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➡ New Project).

Bước 2. Visual Basic sẽ tạo một project mới, form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➡ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name: frmCaro
- Caption: Caro 2001
- WindowState: 0 – Normal

- ScaleMode: 1 - Twip
- Width: 11970
- Height: 9075
- BackColor: &H00FFC0C0&
- FillColor: &H00FF0000&
- Maxbutton: False

Chọn lệnh File ➤ Save Project, lưu form với tên *Caro.frm* và lưu project với tên *Caro.vbp* trong thư mục xác định, chẳng hạn *c:\ltgames*.

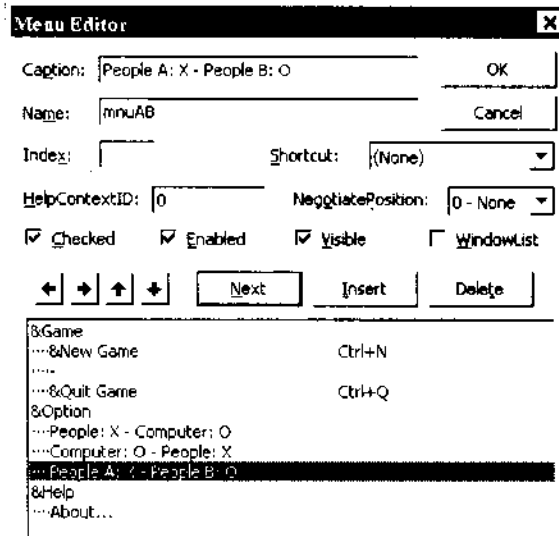
Bước 3. Thiết kế menu chương trình.

Mở form *frmCaro*. Xong chọn lệnh Tools ➤ Menu Editor.

Lần lượt định nghĩa các mục trên menu ngang và menu xổ xuống theo bảng sau:

Caption	Name	Checked	Shortcut
&Game	MnuGame		
..... New game	MnuNew		Ctrl+N
..... -	MnuSep1		
..... Quit game	MnuQuit		Ctrl+Q
&Option	MnuOptions		
..... People: X - Computer: O	MnuXO		
..... Computer: O - People: X	MnuOX		
..... People A: X - People B: O	MnuAB	True	
&Help	MnuHelp		
..... About	mnuAbout		

Kết quả của sổ Menu Designer sẽ như Hình 6.4.



Hình 6.4 Cửa sổ thiết kế menu.

Bước 4. Đặt các điều khiển vào form.

Đặt lên form một điều khiển Label và định lại các thuộc tính như sau:

Name = lblO, Alignment = 2 – Center, Index = 0, Visible = True
 BackColor = &H00C0FFFF&, BackStyle = 1 – Opaque
 AutoSize = False, BorderStyle = 1 – Fixed Single, Font = VNI-Centur,
 Height = 345, Width = 345, Left = 825, Top = 885

Sao chép điều khiển lblO thành 19 điều khiển nữa và định lại các thuộc tính sao cho 20 điều khiển sắp thành một hàng ngang:

1. Điều khiển Label thứ 2 (hàng 1 cột 2)	
Name : lblO	Index : 1

Top : 885	Left : 1155
2. Điều khiển Label thứ 3 (hàng 1 cột 3)	
Name : lblO	Index : 2
Top : 885	Left : 1485
3. Điều khiển Label thứ 4 (hàng 1 cột 4)	
Name : lblO	Index : 3
Top : 885	Left : 1815
4. Điều khiển Label thứ 5 (hàng 1 cột 5)	
Name : lblO	Index : 4
Top : 885	Left : 2130
5. Điều khiển Label thứ 6 (hàng 1 cột 6)	
Name : lblO	Index : 5
Top : 885	Left : 2460
6. Điều khiển Label thứ 7 (hàng 1 cột 7)	
Name : lblO	Index : 6
Top : 885	Left : 2790
7. Điều khiển Label thứ 8 (hàng 1 cột 8)	
Name : lblO	Index : 7
Top : 885	Left : 3120
8. Điều khiển Label thứ 9 (hàng 1 cột 9)	
Name : lblO	Index : 8
Top : 885	Left : 3450
9. Điều khiển Label thứ 10 (hàng 1 cột 10)	
Name : lblO	Index : 9

Top : 885	Left : 3765
10. Điều khiển Label thứ 11 (hàng 1 cột 11)	
Name : lblO	Index : 10
Top : 885	Left : 4095
11. Điều khiển Label thứ 12 (hàng 1 cột 12)	
Name : lblO	Index : 11
Top : 885	Left : 4425
12. Điều khiển Label thứ 13 (hàng 1 cột 13)	
Name : lblO	Index : 12
Top : 885	Left : 4755
13. Điều khiển Label thứ 14 (hàng 1 cột 14)	
Name : lblO	Index : 13
Top : 885	Left : 5085
14. Điều khiển Label thứ 15 (hàng 1 cột 15)	
Name : lblO	Index : 14
Top : 885	Left : 5400
15. Điều khiển Label thứ 16 (hàng 1 cột 16)	
Name : lblO	Index : 15
Top : 885	Left : 5730
16. Điều khiển Label thứ 17 (hàng 1 cột 17)	
Name : lblO	Index : 16
Top : 885	Left : 6060
17. Điều khiển Label thứ 18 (hàng 1 cột 18)	
Name : lblO	Index : 17

Top : 885	Left : 6390
18. Điều khiển Label thứ 19 (hàng 1 cột 19)	
Name : lblO	Index : 18
Top : 885	Left : 6720
19. Điều khiển Label thứ 20 (hàng 1 cột 20)	
Name : lblO	Index : 19
Top : 885	Left : 7035

Sau đó, bao vùng chọn tất cả 10 điều khiển *label lblO* trên hàng đầu tiên. Thực hiện lệnh Edit ➡ Copy, rồi Edit ➡ Paste để sao thành hàng thứ 2. Định lại thuộc tính Top = 1215 cho cả 10 điều khiển *lblO* vừa sao (hàng thứ 2 có chỉ số Index tăng dần từ 20 đến 39 tính từ trái sang phải).

Tương tự, bạn tiến hành sao thêm 18 hàng nữa. Như vậy, bạn đã có một mảng điều khiển *lblO* gồm 400 phần tử có chỉ số Index chạy từ 0 đến 399. Trong đó thuộc tính Top cho các điều khiển *lblO* trên từng hàng:

- Hàng 3: Top = 1545, chỉ số Index chạy từ 40 đến 59
- Hàng 4: Top = 1875, chỉ số Index chạy từ 60 đến 79
- Hàng 5: Top = 2190, chỉ số Index chạy từ 80 đến 99
- Hàng 6: Top = 2520, chỉ số Index chạy từ 100 đến 119
- Hàng 7: Top = 2825, chỉ số Index chạy từ 120 đến 139
- Hàng 8: Top = 3165, chỉ số Index chạy từ 140 đến 159
- Hàng 9: Top = 3495, chỉ số Index chạy từ 160 đến 179
- Hàng 10: Top = 3825, chỉ số Index chạy từ 180 đến 199

- Hàng 11: Top = 4155, chỉ số Index chạy từ 200 đến 219
- Hàng 12: Top = 4485, chỉ số Index chạy từ 220 đến 239
- Hàng 13: Top = 4815, chỉ số Index chạy từ 240 đến 259
- Hàng 14: Top = 5145, chỉ số Index chạy từ 260 đến 279
- Hàng 15: Top = 5460, chỉ số Index chạy từ 280 đến 299
- Hàng 16: Top = 5790, chỉ số Index chạy từ 300 đến 319
- Hàng 17: Top = 6105, chỉ số Index chạy từ 320 đến 339
- Hàng 18: Top = 6435, chỉ số Index chạy từ 340 đến 359
- Hàng 19: Top = 6765, chỉ số Index chạy từ 360 đến 379
- Hàng 20: Top = 7095, chỉ số Index chạy từ 380 đến 399

Đặt tiếp lên form một điều khiển Label nữa và định lại các thuộc tính như sau:

Name : lblTD	Caption : A
Alignment : 2 - Center	Index : 0
Width : 345	Left : 825
Height : 345	Top : 495
BackColor : &H80000005&	BackStyle : 0 – Transparent
BorderStyle : 0 – None	ForeColor : &H00800080&
Font : VNI-Times, Normal, 12	

Sao label lblTD thành 39 label nữa. Đổi thuộc tính Caption cho Label lần lượt thành B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20.

Xong sao chép 40 label trên thành 40 label nữa. Định lại các thuộc tính Top và Left cho thích hợp để được kết quả như Hình 6.1.

2. Viết mã lệnh

' Khai báo các biến toàn cục có tác dụng trong toàn form.

Option Explicit

Dim Banco(0 To 20, 0 To 20) As Integer

Dim CurPlayer, auto, vtcu

Dim diemmax1, diemmax2, diemmax3, diemmax4, otrong(4, 2)

' Khi form bắt đầu được nạp

Private Sub Form_Load()

Randomize *' khởi động chế độ chọn số ngẫu nhiên*

Dim i, j

' Nếu đang chọn chế độ hai người chơi với nhau

If mnuAB.Checked = True Then

auto = 1 *' chờ người chơi chọn ô*

CurPlayer = -1 *' Đến lượt người A chọn ô*

Else *' người chơi với máy*

auto = 0 *' máy tính có thể tự động chọn ô*

' Nếu người chơi chơi trước máy tính

If mnuXO.Checked = True Then

CurPlayer = -1 *' đến lượt người chơi chọn ô*

Else *' Nếu máy tính chơi trước người chơi*

CurPlayer = -2 *' đến lượt máy tính chọn ô*

End If

End If

For i = 0 To 20

For j = 0 To 20

```

Banco(i, j) = -50      ' giá trị cho biết ô tương ứng là ô trống
If i <> 20 And j <> 20 Then
    lblO(i * 20 + j).Caption = ""      ' Xóa trống các ô trên màn hình
End If
Next
Next
'Nếu đang ở chế độ chơi giữa người và máy, đồng thời máy tính chơi trước
If auto = 0 And CurPlayer = -2 Then
    ' Gọi thủ tục Danhdao, chọn một ô trống ngẫu nhiên trong 400 ô thuộc
    ' không gian chơi
    Danhdao (Fix(Rnd * 400))
End If
End Sub
' Khi người chơi nhấp mouse trên một ô
Private Sub lblO_MouseDown(Index As Integer, Button As Integer, Shift
As Integer, X As Single, Y As Single)
Dim ncell
If Button = 1 Then      ' nếu là nhấp trái mouse
                        ' và ô chưa được đánh dấu
If (lblO(Index).Caption <> "O") And (lblO(Index).Caption <> "X") Then
    Call Danhdao(Index)      ' Gọi thủ tục đánh dấu
    ' Nếu ở chế độ chơi giữa người và máy, và đến lượt máy tính đi
    ' thì tự động gọi thủ tục Timnuocdi để xác định nước đi kế tiếp
If auto = 0 And CurPlayer = -2 Then
    ncell = Timnuocdi(Index)
If ncell >= 0 Then
    Danhdao (ncell)      ' máy tính tự đánh dấu ô xác định được
End If

```

End If

End If

End If

End Sub

' Đánh dấu vào bàn cờ

Private Sub Danhdau(Index As Integer)

If CurPlayer = -1 Then *' Nếu là người chơi*

lblO(vtcu).BackColor = &HC0FFFF *' đặt lại màu vàng cho ô đã đánh dấu trước*

vtcu = Index *' lưu vị trí ô sẽ đánh dấu*

lblO(Index).BackColor = &HFF80FF *' đặt màu tím cho ô*

lblO(Index).ForeColor = &HFF& *' chữ đỏ*

lblO(Index).Caption = "X" *' đánh dấu bằng ký tự X*

Banco(Fix(Index / 20), Index Mod 20) = -1 *' ghi nhận ô đã đánh dấu*

' Gọi thủ tục Tinhdiem, xem có phương nào đủ năm ô liên tiếp hay chưa

Tinhdiem (Index)

' Chuyển quyền đánh dấu cho máy tính hoặc người chơi thứ 2

CurPlayer = -2

Else *' Nếu là người chơi thứ 2 hoặc đến lượt máy tính đánh dấu*

lblO(vtcu).BackColor = &HC0FFFF

vtcu = Index

lblO(Index).BackColor = &HFF80FF

lblO(Index).ForeColor = &HFF0000 *' đánh dấu bằng ký tự "O"*

lblO(Index).Caption = "O"

Banco(Fix(Index / 20), Index Mod 20) = -2 *' ghi nhận ô đã đánh dấu*

Tinhdiem (Index)

' Chuyển quyền đánh dấu trở lại cho người chơi thứ nhất

CurPlayer = -1

End If

End Sub

' Xác định nước đi tiếp theo, chỉ được gọi khi đến lượt máy tính chơi

Private Function Timnuocdi(ByVal Index As Integer) As Integer

Dim diemmax, direction, nchon, nstop

diemmax = diemmax1 *' Số ô được đánh dấu liên tiếp theo*

' phương ngang của đối thủ

direction = 1 *' mặc định chọn phương ngang làm phương thực hiện nước đi*

' Nếu phương dọc có nhiều ô đánh dấu liên tục hơn

' và ít nhất một trong hai đầu chưa bị chặn

If diemmax2 >= diemmax Then

If otrong(2, 1) >= 0 Or otrong(2, 2) >= 0 Then

diemmax = diemmax2 *' lưu lại vị trí số ô đánh dấu liên tiếp theo*

' phương đứng

direction = 2 *' chọn phương đứng làm phương thực hiện nước đi*

End If

End If

' Nếu phương theo đường chéo 1 có nhiều ô đánh dấu liên tục hơn '

' so với hai phương đứng và ngang. Đồng thời ít nhất một trong

' hai đầu chưa bị chặn

If diemmax3 >= diemmax Then

If otrong(3, 1) >= 0 Or otrong(3, 2) >= 0 Then

diemmax = diemmax3 *' lưu lại vị trí số ô đánh dấu liên tiếp theo*

' phương này

direction = 3 *' chọn phương đường chéo một làm*

' phương thực hiện nước đi

End If

End If

' Nếu phương theo đường chéo 2 có nhiều ô đánh dấu liên tục hơn '

' so với ba phương còn lại. Đồng thời ít nhất một trong hai đầu chưa bị chặn

If diemmax4 >= diemmax Then

```

If otrong(4, 1) >= 0 Or otrong(4, 2) >= 0 Then
    diemmax = diemmax4      ' lưu lại vị trí số ô đánh dấu liên tiếp theo
                             ' phương này
    direction = 4           ' chọn phương đường chéo 2 làm phương
                             ' thực hiện nước đi

End If
End If

' Nếu đầu trên của phương chưa bị chặn, thực hiện nước đi chặn trên
' ô trống bên trên
If otrong(direction, 1) >= 0 Then
    Timnuocdi = otrong(direction, 1)
Else
    ' Ngược lại nếu đầu trên của phương đã bị chặn, và đầu còn lại chưa bị chặn.
    ' Thực hiện nước đi chặn trên ô trống bên dưới
    If otrong(direction, 2) >= 0 Then
        Timnuocdi = otrong(direction, 2)
    Else
        ' Nếu cả hai đầu của phương đều bị chặn (nghĩa là đối thủ không thể
        ' phát triển tấn công theo phương này
        nstop = 0
        Do
            nchon = Fix(Rnd * 400)      ' Chọn một ô trống ngẫu nhiên bất kỳ
                                         ' còn lại trong
            nstop = nstop + 1           ' không gian chơi để thực hiện nước đi kế tiếp
            If nstop > 1000 Then        ' Nếu không gian chơi không còn ô trống
                MsgBox "Chúng ta hòa"  ' hiển thị thông báo Hòa sau 1000 lần chọn
                Form_Load              ' không thành công. Xong thực hiện lại trò chơi
            End If
        Loop Until Banco(Fix(nchon / 20), nchon Mod 20) = -50
    
```

Timnuocdi = nchon ' Trả lại vị trí ô mà máy tính sẽ thực hiện nước đi

End If

End If

End Function

' Kiểm tra xem đã có bên nào thắng hay chưa

Private Function Tinhdiem(ByVal Index As Integer) As Integer

Dim diem1, diem2, i, j

For i = 1 To 4 ' Xóa trống vị trí các ô trống máy tính dự định đi

For j = 1 To 2 ' nước tiếp theo

otrong(i, j) = -50

Next

Next

diem1 = LeftMark(Index) ' xác định số ô liên tiếp bên trái

diem2 = RightMark(Index) ' xác định số ô liên tiếp bên phải

diemmax1 = diem1 + diem2 + 1 ' số ô liên tiếp đã đánh dấu theo
' phương ngang

diem1 = UpMark(Index) ' xác định số ô liên tiếp ở phía trên

diem2 = DownMark(Index) ' xác định số ô liên tiếp ở phía dưới

diemmax2 = diem1 + diem2 + 1 ' số ô liên tiếp đã đánh dấu theo
' phương đứng

diem1 = LeftUp(Index) ' xác định số ô liên tiếp ở phía trên
' bên trái

diem2 = RightDown(Index) ' xác định số ô liên tiếp ở phía
' dưới bên phải

diemmax3 = diem1 + diem2 + 1 ' số ô liên tiếp đã đánh dấu theo
' đường chéo 1

diem1 = RightUp(Index) ' xác định số ô liên tiếp ở phía trên
' bên phải

diem2 = LeftDown(Index) ' xác định số ô liên tiếp ở phía
' dưới bên trái

```

diemmax4 = diem1 + diem2 + 1      ' số ô liên tiếp đã đánh dấu theo
                                   ' đường chéo 2
' Nếu một trong bốn phương có số ô liên tiếp đã đánh dấu bằng 5
If diemmax1 = 5 Or diemmax2 = 5 Or diemmax3 = 5 Or diemmax4 = 5 Then
    If CurPlayer = -1 Then      ' và người chơi là người vừa thực hiện nước đi
        MsgBox "You win"      ' Hiển thị thông báo "You win"
    Else ' ngược lại nếu máy tính hay người chơi 2 là người vừa thực hiện nước đi
        MsgBox "I win"        ' Hiển thị thông báo "I win"
    End If
    Form1_Load      ' Khởi tạo lại không gian chơi, chuẩn bị cho lần chơi tiếp theo
End If

```

End Function

' Đếm số dấu bên trái ô hiện hành

Private Function LeftMark(ByVal Index As Integer) As Integer

Dim dem, vt As Integer

dem = 0

Do

vt = Index Mod 20

If vt > 0 Then

Index = Index - 1

If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then

dem = dem + 1

Else

If Banco(Fix(Index / 20), Index Mod 20) = -50 Then

otrong(1, 1) = Index

End If

Exit Do

End If

```
End If
Loop Until vt = 0
LeftMark = dem
```

End Function

' Đếm số dấu bên phải ô hiện hành

Private Function RightMark(ByVal Index As Integer) As Integer

```
Dim dem, vt As Integer
dem = 0
Do
vt = Index Mod 20
If vt < 19 Then
Index = Index + 1
If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then
dem = dem + 1
Else
If Banco(Fix(Index / 20), Index Mod 20) = -50 Then
otrong(1, 2) = Index
End If
Exit Do
End If
End If
Loop Until vt = 19
RightMark = dem
```

End Function

' Đếm số dấu phía trên ô hiện hành

Private Function UpMark(ByVal Index As Integer) As Integer

```
Dim dem, vt As Integer
dem = 0
```

```
Do
    vt = Index - 20
    If vt >= 0 Then
        Index = Index - 20
        If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then
            dem = dem + 1
        Else
            If Banco(Fix(Index / 20), Index Mod 20) = -50 Then
                otrong(2, 1) = Index
            End If
        End Do
    End If
    End If
Loop Until vt <= 19
UpMark = dem
```

End Function

' Đếm số dấu phía dưới ô hiện hành

Private Function DownMark(ByVal Index As Integer) As Integer

```
Dim dem, vt As Integer
dem = 0
Do
    vt = Index + 20
    If vt <= 399 Then
        Index = Index + 20
        If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then
            dem = dem + 1
        Else
            If Banco(Fix(Index / 20), Index Mod 20) = -50 Then
```

```

        otrong(2, 2) = Index
    End If
    Exit Do
End If
End If

```

```

Loop Until vt >= 380
DownMark = dem

```

End Function

' Đếm số dấu ở trên bên trái ô hiện hành

Private Function LeftUp(ByVal Index As Integer) As Integer

Dim dem, vt As Integer

dem = 0

Do

vt = Index

If vt > 19 And vt Mod 20 > 0 Then

Index = Index - 21

If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then

dem = dem + 1

Else

If Banco(Fix(Index / 20), Index Mod 20) = -50 Then

otrong(3, 1) = Index

End If

Exit Do

End If

End If

Loop Until vt <= 19 Or vt Mod 20 = 0

LeftUp = dem

End Function

' Đếm số dấu ở dưới bên phải ô hiện hành

Private Function RightDown(ByVal Index As Integer) As Integer

Dim dem, vt As Integer

dem = 0

Do

vt = Index

If vt < 380 And vt Mod 20 < 19 Then

Index = Index + 21

If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then

dem = dem + 1

Else

If Banco(Fix(Index / 20), Index Mod 20) = -50 Then

otrong(3, 2) = Index

End If

Exit Do

End If

End If

Loop Until vt >= 380 Or vt Mod 20 = 19

RightDown = dem

End Function

' Đếm số dấu ở trên bên phải ô hiện hành

Private Function RightUp(ByVal Index As Integer) As Integer

Dim dem, vt As Integer

dem = 0

Do

vt = Index

If vt > 19 And vt Mod 20 < 19 Then

Index = Index - 19


```
If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then
    dem = dem + 1
Else
    If Banco(Fix(Index / 20), Index Mod 20) = -50 Then
        otrong(4, 1) = Index
    End If
Exit Do
End If
End If
Loop Until vt <= 19 Or vt Mod 20 = 19
RightUp = dem
```

End Function

' Đếm số dấu ở dưới bên trái ô hiện hành

Private Function LeftDown(ByVal Index As Integer) As Integer

```
Dim dem, vt As Integer
dem = 0
Do
    vt = Index
    If vt < 380 And vt Mod 20 > 0 Then
        Index = Index + 19
        If Banco(Fix(Index / 20), Index Mod 20) = CurPlayer Then
            dem = dem + 1
        Else
            If Banco(Fix(Index / 20), Index Mod 20) = -50 Then
                otrong(4, 2) = Index
            End If
        End If
    End Do
Exit Do
End If
```

End If

Loop Until vt >= 380 Or vt Mod 20 = 0

LeftDown = dem

End Function

' Khi chọn lệnh Game ➡ New Game hay nhấn tổ hợp phím CTRL+N

Private Sub mnuNew_Click()

Form_Load *' Gọi thủ tục biến cố Form_load*

End Sub

' Khi chọn lệnh Game ➡ Quit Game hay nhấn tổ hợp phím CTRL+Q

Private Sub mnuQuit_Click()

End *' Kết thúc chương trình*

End Sub

' Khi chọn tùy chọn người chơi với máy, người chơi đi trước

' "People: X – Computer: O"

Private Sub mnuXO_Click()

mnuXO.Checked = True *' Bật dấu chọn trước mục chọn*

mnuOX.Checked = False *' Tắt dấu chọn trước 2 mục chọn còn lại*

mnuAB.Checked = False *' trong menu Option*

Form_Load *' bắt đầu một ván chơi mới*

End Sub

' Khi chọn tùy chọn người chơi với máy, máy tính đi trước

' "Computer: O - People: X"

Private Sub mnuOX_Click()

mnuXO.Checked = False

mnuOX.Checked = True *' Bật dấu chọn trước mục chọn*

mnuAB.Checked = False *' Tắt dấu chọn trước hai mục chọn còn lại*

Form_Load *' bắt đầu một ván chơi mới*

End Sub

' Khi chọn tùy chọn hai người chơi với nhau "People A: X – People B: O"

Private Sub mnuAB_Click()

mnuXO.Checked = False ' Tắt dấu chọn trước hai mục chọn còn lại

mnuOX.Checked = False

mnuAB.Checked = True ' Bật dấu chọn trước mục chọn

Form_Load ' bắt đầu một ván chơi mới

End Sub

IV. Phân tích chương trình

1. Ý nghĩa các biến toàn cục đã khai báo

- ♦ Mảng **Banco**: Lưu thông tin các ô tương ứng trong không gian chơi. Cho biết ô còn trống hay đã được đánh dấu, và dấu đang chứa thuộc bên chơi nào.
- ♦ Mảng **oTrong**: Lưu vị trí của 8 ô trống chặn ở hai đầu khi mở rộng ô chơi hiện hành theo bốn hướng.
- ♦ Biến **Auto**: Xác định chương trình đang ở chế độ "hai người chơi với nhau" hay "người chơi với máy".
- ♦ Biến **vtcu**: Lưu vị trí ô đã được chọn trước khi chọn ô hiện hành.
- ♦ Biến **Curplayer**: Xác định người chơi hiện hành (người chơi A, B hay máy tính).
- ♦ Biến **diemmax1**: Xác định số ô liên tiếp đã được đánh dấu theo phương ngang.
- ♦ Biến **diemmax2**: Xác định số ô liên tiếp đã được đánh dấu theo phương đứng.
- ♦ Biến **diemmax3**: Xác định số ô liên tiếp đã được đánh dấu theo đường chéo 1.
- ♦ Biến **diemmax4**: Xác định số ô liên tiếp đã được đánh dấu theo đường chéo 2.

2. Giải thích kỹ thuật xác định hàng và cột cho ô vừa nhấp

Vị trí ô vừa nhấp được xác định bằng chỉ số mảng Index của điều khiển label lblO tương ứng. Nếu tính hàng đầu tiên là 0 và cột đầu tiên là 0 thì khi đó:

Vị trí hàng của ô: $\text{Fix}(\text{index}/20)$ (phần nguyên của thương $\text{index} / 20$)

Và vị trí cột của ô: $\text{Index mod } 20$ (phần dư của $\text{Index} / 20$)

Ví dụ: $\text{Index} = 29 \Rightarrow$ ô thuộc hàng 2, cột 9 (nghĩa là hàng thứ 3, cột thứ 10).

3. Kỹ thuật xác định ô đã đánh dấu

Sử dụng mảng Banco. Mỗi thành phần mảng tương ứng với một ô trong mảng điều khiển lblO. Giá trị của mỗi thành phần trong mảng *Banco* sẽ cho biết ô tương ứng là trống hay đã được đánh dấu. Cụ thể là:

- ♦ Giá trị -50: Cho biết ô còn trống
- ♦ Giá trị -1: Cho biết ô đã được đánh dấu "X" (ô của người chơi 1)
- ♦ Giá trị -2: Cho biết ô đã được đánh dấu "O" (ô của người chơi 2 trong chế độ "2 người chơi" hay máy tính trong chế độ chơi "giữa người và máy")

4. Kỹ thuật xác định người chơi hiện hành được quyền đánh dấu ô (nước đi tiếp theo sẽ thuộc bên chơi nào)

Sử dụng biến CurPlayer.

- ♦ Khi $\text{CurPlayer} = -1$: Luôn là người chơi A cho dù chế độ chơi giữa người và máy hay giữa hai người với nhau.
- ♦ Khi $\text{CurPlayer} = -2$: Là máy tính nếu chế độ chơi giữa người và máy, là người chơi B nếu chế độ chơi hai người với nhau.

5. Kỹ thuật xác định nước đi tiếp theo của máy tính

Mặc định máy tính sẽ chọn một trong bốn phương đi, sau đó sẽ chọn một trong hai ô trống ở hai đầu của dãy ô cờ đã đi của người chơi để chặn, máy tính sẽ bỏ qua không chọn các phương đã được máy tính chặn cả hai đầu dãy cờ của người chơi.

Trong trường hợp có nhiều phương chưa được chặn. Máy tính sẽ ưu tiên chặn các phương mà người chơi đã đi nhiều ô liên tiếp nhất.

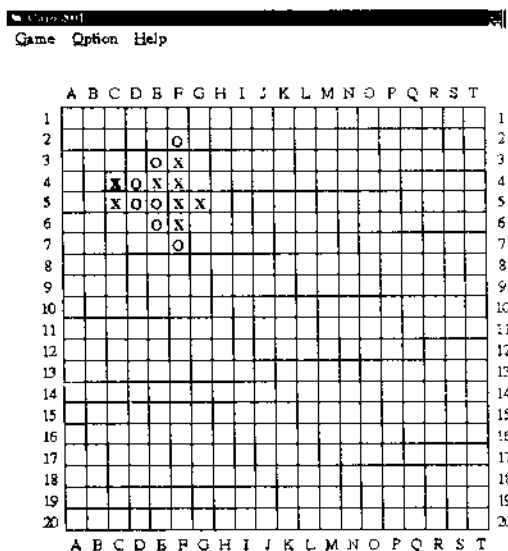
V. Bài tập thực hành

Bài 1. Như đã trình bày ở trên, thuật toán TIMDUONGDI còn khá nhiều hạn chế. Vì chỉ lo quan tâm đến việc phòng thủ, chặn đường thắng của người chơi là chính, mà chưa quan tâm đúng mức đến việc phát triển tấn công để thắng người chơi, cho nên, chỉ khi người chơi carô quá tệ hoặc sơ ý mới bị máy tính hạ. Giờ đây, bạn hãy cải tiến thuật toán sao cho máy tính trở nên "thông minh" hơn, chẳng những cải thiện được khả năng phòng thủ (có thể đối đầu với các cao thủ carô) mà còn biết chủ động tấn công để tìm chiến thắng.

Hướng dẫn: Để máy tính trở nên thông minh hơn, bạn cần lưu ý các điểm sau:

- ♦ Chỉ phòng thủ (chặn đầu người chơi) khi các phương có từ ba ô liên tiếp được người chơi đánh dấu trở lên.
- ♦ Nếu cả bốn phương đều có số lượng ô liên tiếp được đánh dấu nhỏ hơn hoặc bằng 2, thì chủ động tìm hướng tấn công lại người chơi.
- ♦ Trong quá trình tấn công, ưu tiên các hướng vừa có công dụng tấn công, vừa đánh chặn được người chơi.

Chẳng hạn, trong ví dụ Hình 6.5, khi người chơi đánh dấu "X" vào ô C4, thì phương đứng có số lượng ô liên tiếp nhiều nhất là 2. Nên lúc này nếu máy tính lo việc phòng thủ (đánh chặn ở ô C3 hay C6) là không hợp lý, vì sẽ bỏ mất dịp tấn công lại người chơi. Lúc này, máy tính cần chủ động tấn công bằng cách đánh vào một trong các ô sau: G8 (bắt buộc người chơi phải đánh chặn ở ô H9 trong nước đi kế tiếp để tránh thất bại), D3, D6 hoặc D7. Rõ ràng, trong trường hợp này, đánh vào vị trí ô D3 là thông minh nhất, vì không chỉ bắt người chơi phải lo đánh chặn phương (D3 D4 D5), mà còn chặn ngang hướng tấn công phối hợp tiềm ẩn của người chơi là (C2 C4 C5) và (C2 E4 F5). Vì nếu để người chơi có cơ hội đánh vào ô C2, trong khi máy tính chưa chặn đường D3, thì xem như máy tính vô phương cứu chữa ván chơi. Vì khi đó, nếu máy tính đánh chặn phương (C2 C4 C5) thì người chơi sẽ thắng bằng phương (C2 D3 E4 F5) và ngược lại nếu đánh chặn phương (C2 E4 F5) thì người chơi sẽ thắng bằng phương (C2 C3 C4 C5).



Hình 6.5 Xác định nước đi tối ưu tiếp theo của máy tính (ký tự "O").

BÀI 7.

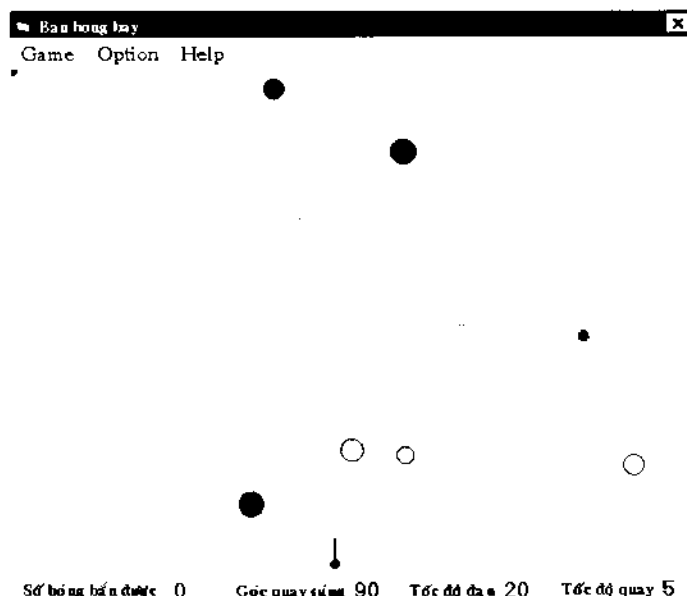
TRÒ CHƠI XẠ THỦ

I. Mô tả trò chơi

Trò chơi xạ thủ thực chất là trò chơi bắn bong bóng bay. Người chơi sẽ làm chủ một ụ súng, có thể xoay qua lại để bắn các bong bóng bay. Số bong bóng bay sẽ liên tục xuất hiện tại các vị trí ngẫu nhiên trên màn hình, di chuyển và biến mất. Người chơi sẽ thua khi không bắn đủ số bóng trong khoảng thời gian quy định hoặc để bất kỳ một quả bong bóng nào, bay đến và va chạm vào ụ súng.

Người chơi có thể sử dụng phím **Page Up** để điều chỉnh tăng tốc độ quay nòng súng, **Page Down** để điều chỉnh giảm tốc độ viên đạn, **→** để quay nòng súng sang phải, **←** để quay nòng súng sang trái, **↑** để điều chỉnh tăng tốc độ bay của viên đạn sẽ bắn ra, **↓** để điều chỉnh tốc độ bay của viên đạn sẽ bắn ra. Nhấn phím **Spacebar** để bắn.

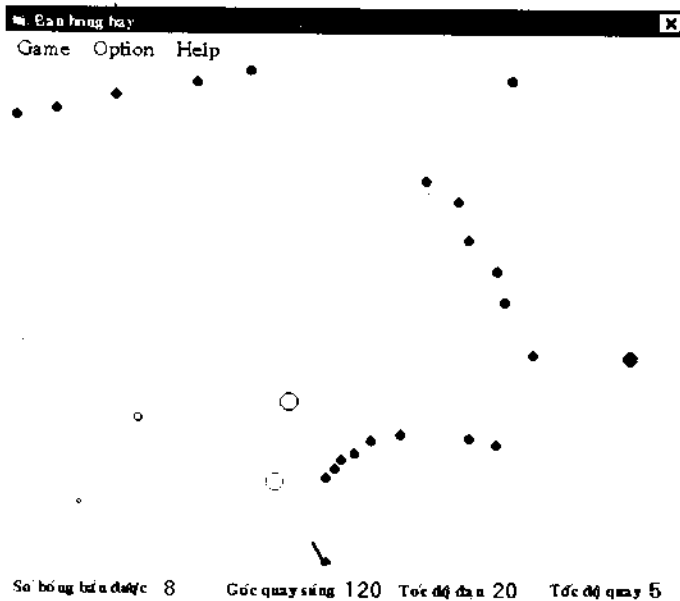
Trên màn hình trò chơi, các quả bóng di chuyển theo những hướng ngẫu nhiên và lớn dần lên cho đến khi chạm vào một trong bốn đường biên của không gian chơi hoặc bị bắn trúng thì biến mất. Một quả bóng khác sẽ tự động phát sinh để thay thế quả bóng vừa biến mất, sao cho luôn luôn đảm bảo số quả bóng di chuyển trên màn hình là 10. (Hình 7.1)



Hình 7.1 Số bóng tối đa xuất hiện và di chuyển trên màn hình luôn là 10.

Tương tự, người chơi sẽ bắn được mỗi lượt tối đa 20 viên đạn. Các viên đạn sẽ bay cho đến khi đụng phải một quả bóng hay gặp một trong bốn đường biên thì mới biến mất. Lúc đó, người chơi mới có thể tiếp tục bắn viên đạn mới. Nói cách khác, tại một thời điểm chơi bất kỳ, số viên đạn đang di chuyển trên màn hình tối đa là 20 (Hình 7.2).

Trò chơi chia làm 3 mức: mức 1 bắt buộc người chơi phải bắn trúng 50 quả bóng trong vòng 1 phút, khi đó trò chơi sẽ tự động chuyển sang mức 2. Mức 2, người chơi phải bắn trúng 100 quả bóng trong vòng 1 phút, khi đó trò chơi sẽ tự động chuyển sang mức 3 và ở mức 3, người chơi phải bắn trúng 150 quả bóng trong vòng 1 phút. Đối với những người chơi thành thạo, họ có thể chuyển ngay tới mức 3 để chơi bằng cách chọn trực tiếp tùy chọn mức trong menu Option.



Hình 7.2 Tại một thời điểm chỉ có thể có tối đa 20 viên đạn đang bay.

II. Thiết kế chương trình

1. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start ➡ Programs ➡ Microsoft Visual Studio 6.0 ➡ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➡ New Project).

Bước 2. Visual Basic sẽ tạo một project mới, form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➡ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name : frmXathu

- Caption : Bắn bóng bay
- WindowState : 0 – Normal
- ScaleMode : 1 - Twip
- Width : 7995
- Height : 7100
- BackColor : &H00FFC0C0&
- FillColor : &H00FF0000&
- Maxbutton : False
- Minbutton : False
- Moveable : True
- StartupPosition : 2 – CenterScreen

Chọn lệnh File ➡ Save Project, lưu form với tên *Xathu.frm* và lưu project với tên *Xathu.vbp* trong thư mục xác định, chẳng hạn *c:\ltgames*.

Bước 3. Thiết kế menu chương trình.

Mở form *frmXathu*. Xong chọn lệnh Tools ➡ Menu Editor.

Lần lượt định nghĩa các mục trên menu ngang và menu xổ xuống theo bảng sau:

Caption	Name	Checked	Shortcut
&Game	MnuGame		
..... New game	MnuNew		Ctrl+N
..... -	MnuSep1		
..... Quit game	MnuQuit		Ctrl+Q

&Option	MnuOptions		
..... Level 1	MnuL1	True	
..... Level 2	MnuL2		
..... Level 3	MnuL3		
&Help	MnuHelp		
..... About	MnuAbout		

Bước 4. Đặt các điều khiển vào form.

Đặt lên form ba điều khiển Timer, lần lượt định lại thuộc tính Internal cho từng Timer như sau:

- Điều khiển Timer 1 :
Name : Timer1, Interval = 5
- Điều khiển Timer 2 :
Name : Timer2, Interval = 5
- Điều khiển Timer 3 :
Name : Timer3, Interval = 60000

Đặt lên form một điều khiển Label và định lại các thuộc tính như sau:

Name = Label1, Alignment = 0 – Left, Visible = True
Caption: Số bóng bắn được, BackColor = &H80000009&
BackStyle = 1 – Opaque, AutoSize = False, BorderStyle = 0 – None,
Fontname = VNI-Times, FontSize = 10, FontStyle = Bold,
ForeColor = &H80000012&,
Height = 375, Width = 1605, Left = 90, Top = 5910

Sao chép điều khiển Label1 thành 7 Label nữa, định lại các thuộc tính và sắp chúng thành một hàng ngang:

1. Điều khiển Label thứ 2	
Name: Soquano	Top: 5910
Caption: (rỗng)	Left: 1815
2. Điều khiển Label thứ 3	
Name: label2	Top: 5910
Caption: Góc quay súng	Left: 2550
3. Điều khiển Label thứ 4	
Name: lblDegree	Top: 5910
Caption: (rỗng)	Left: 3900
4. Điều khiển Label thứ 5	
Name: label3	Top: 5910
Caption: Tốc độ đạn	Left: 4545
5. Điều khiển Label thứ 6	
Name: lblDodai	Top: 5910
Caption: (rỗng)	Left: 5625
6. Điều khiển Label thứ 7	
Name: Label 4	Top: 5910
Caption: Tốc độ quay	Left: 6300
7. Điều khiển Label thứ 8 (hàng 1 cột 8)	
Name: lblgocquay	Top: 5910
Caption: (rỗng)	Left: 7455

Lần lượt đặt tiếp lên form 8 điều khiển Shape nữa và định lại các thuộc tính như sau:

1. Điều khiển Shape thứ 1	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H008080FF&	Index: 0
Width: 105	Left: 3450
Height: 105	Top: 3180
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H000000FF&
FillStyle: 0 – Solid	Shape: 3 – Circle
2. Điều khiển Shape thứ 2	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H00404040&	Index: 1
Width: 105	Left: 6075
Height: 105	Top: 3150
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00C0FFC0&
FillStyle: 0 – Solid	Shape: 3 – Circle
3. Điều khiển Shape thứ 3	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H80000008&	Index: 2
Width: 105	Left: 1230
Height: 105	Top: 1770
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent

BorderWidth: 1	FillColor: &H00E0E0E9&
FillStyle: 0 – Solid	Shape: 3 – Circle
4. Điều khiển Shape thứ 4	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H000080FF&	Index: 3
Width: 105	Left: 4845
Height: 105	Top: 1545
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H000080FF&
FillStyle: 0 – Solid	Shape: 3 – Circle
5. Điều khiển Shape thứ 5	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H0080FF80&	Index: 4
Width: 105	Left: 1230
Height: 105	Top: 4050
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H0080FF80&
FillStyle: 0 – Solid	Shape: 3 – Circle
6. Điều khiển Shape thứ 6	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H0080C0FF&	Index: 5
Width: 105	Left: 4905
Height: 105	Top: 4350
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent

BorderWidth: 1	FillColor: &H0080C0FF&
FillStyle: 0 – Solid	Shape: 3 – Circle
7. Điều khiển Shape thứ 7	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H00008000&	Index: 6
Width: 105	Left: 2700
Height: 105	Top: 690
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00008000&
FillStyle: 0 – Solid	Shape: 3 – Circle
7. Điều khiển Shape thứ 8	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H80000008&	Index: 7
Width: 105	Left: 6795
Height: 105	Top: 540
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00FFFF00&
FillStyle: 0 – Solid	Shape: 3 – Circle
7. Điều khiển Shape thứ 9	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H80000008&	Index: 8
Width: 105	Left: 465
Height: 105	Top: 615
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent

BorderWidth: 1	FillColor: &H00FF0000&
FillStyle: 0 – Solid	Shape: 3 – Circle
7. Điều khiển Shape thứ 10	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H00404040&	Index: 9
Width: 105	Left: 4980
Height: 105	Top: 6810
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00FF00FF&
FillStyle: 0 – Solid	Shape: 3 – Circle

Sử dụng công cụ Line, vẽ một đoạn thẳng trên form và định lại các thuộc tính như sau:

Name: Line1, BoderColor: &H80000008&, BoderStyle: 6 – Inside Solid,
BorderWidth: 2, X1 : 3690, X2 : 3690, Y1 : 5490, Y2 : 5790

Đặt lên form thêm một điều khiển Shape mới và định lại các thuộc tính như sau:

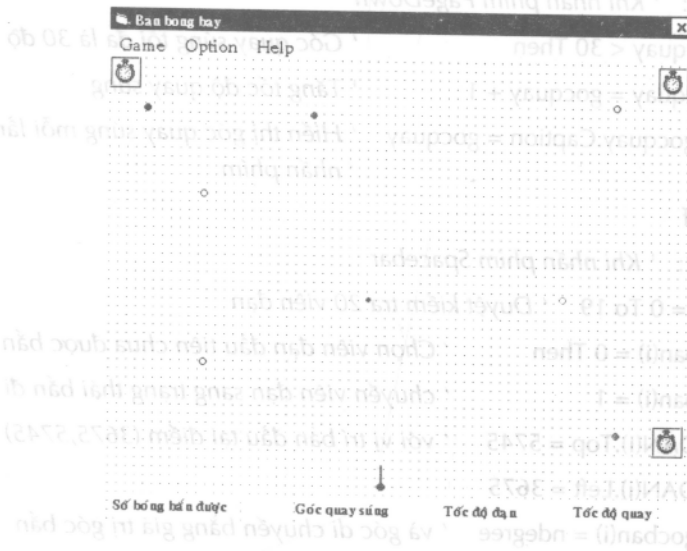
Name: DAN	BackColor: &H80000006&
BorderColor: &H80000008&	Index: 0
Width: 45	Left: 3675
Height: 45	Top: 5745
BorderStyle: 1 – Solid	BackStyle: 1 – Opaque
BorderWidth: 6	FillColor: &H00000000&
FillStyle: 1 – Transparent	Shape: 3 – Circle

Sao chép điều khiển DAN thành 19 điều khiển nữa để tạo mảng điều khiển DAN gồm 20 phần tử có chỉ số Index từ 0 đến 19.

Định lại thuộc tính Top và Left cho 20 điều khiển DAN là:

Top: 5745, Left: 3675

Kết quả màn hình thiết kế form *frmXathu* sẽ như Hình 7.3.



Hình 7.3 Màn hình thiết kế form *frmXathu*.

2. Viết mã lệnh

‘ Khai báo các biến toàn cục có tác dụng trong toàn form.

Dim direct1(10), direct2(10), left1(10), top1(10), step1h(10), step1v(10)

Dim ban(20), gocban(20), ndegree, nleft, ntop, gocquay, dodai, thoigian, nlevel

‘ Thủ tục được gọi khi người chơi nhấn một phím bất kỳ

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)

Select Case KeyCode

Case 34: ' Khi nhấn phím PageUp

```
If gocquay > 1 Then      ' Góc quay súng tối thiểu phải là 1 độ
    gocquay = gocquay - 1 ' giảm tốc độ quay súng
    lblgocquay.Caption = gocquay ' Hiển thị góc quay súng mỗi lần
                                ' nhấn phím
```

End If

Case 33: ' Khi nhấn phím PageDown

```
If gocquay < 30 Then      ' Góc quay súng tối đa là 30 độ
    gocquay = gocquay + 1  ' Tăng tốc độ quay súng
    lblgocquay.Caption = gocquay ' Hiển thị góc quay súng mỗi lần
                                ' nhấn phím
```

End If

Case 32: ' Khi nhấn phím Spacebar

```
For i = 0 To 19 ' Duyệt kiểm tra 20 viên đạn
    If ban(i) = 0 Then ' Chọn viên đạn đầu tiên chưa được bắn đi
        ban(i) = 1    ' chuyển viên đạn sang trạng thái bắn đi
        DAN(i).Top = 5745 ' với vị trí ban đầu tại điểm (3675,5745)
        DAN(i).Left = 3675
        gocban(i) = ndegree ' và góc di chuyển bằng giá trị góc bắn
                            ' hiện hành
```

Exit For

End If

Next

Case 38: ' Nếu nhấn phím Up arrow

```
If dodai < 50 Then      ' Tốc độ bay của viên đạn tối đa là 50 đơn vị
    dodai = dodai + 1    ' Tăng tốc độ bay của viên đạn
    lbldodai.Caption = dodai ' Hiển thị tốc độ bay
```

End If

Case 40: ' Nếu nhấn phím Down arrow

```
If dodai > 10 Then      ' Tốc độ bay của viên đạn tối thiểu là 10
    dodai = dodai - 1    ' Giảm tốc độ bay của viên đạn
    lblodai.Caption = dodai    ' Hiển thị tốc độ bay
End If
```

Case 37: ' Nếu nhấn phím Left arrow

```
ndegree = ndegree + gocquay    ' Quay nòng súng sang trái
If ndegree > 180 Then          ' Tối đa là 180 độ
    ndegree = 180
End If
lbldegree.Caption = ndegree    ' Hiển thị góc quay hiện hành của
                                ' nòng súng
' Tính độ dời sang trái và lên trên (nếu góc quay từ 0 đến 90 độ) hoặc
' xuống dưới (nếu góc quay từ 90 đến 180 độ) của nòng súng
ntop = 300 * Sin(ndegree * 3.141592653589 / 180)
nleft = 300 * Cos(ndegree * 3.141592653589 / 180)
Line1.Y1 = Line1.Y2 - ntop    ' Hiển thị vị trí mới của nòng súng
Line1.X1 = Line1.X2 + nleft
```

Case 39: ' Khi nhấn phím Right arrow

```
ndegree = ndegree - gocquay    ' Quay nòng súng sang phải
If ndegree < 0 Then            ' Tối đa là 0 độ
    ndegree = 0
End If
lbldegree.Caption = ndegree
' Tính độ dời sang phải và xuống dưới (nếu góc quay giảm từ 180 đến
' 90 độ) hoặc lên trên (nếu góc quay giảm từ 90 đến 0 độ) của
' nòng súng
ntop = 300 * Sin(ndegree * 3.141592653589 / 180)
nleft = 300 * Cos(ndegree * 3.141592653589 / 180)
Line1.Y1 = Line1.Y2 - ntop    ' Hiển thị vị trí mới của nòng súng
```

```
Line1.X1 = Line1.X2 + nleft
```

```
End Select
```

```
End Sub
```

```
' Thủ tục: được gọi khi form vừa nạp hay khi bắt đầu một lượt chơi mới
```

```
' Thiết đặt lại các giá trị ban đầu cho lượt chơi
```

```
Private Sub Form_Load()
```

```
Randomize ' Khởi động chế độ chọn số ngẫu nhiên
```

```
soquano.Caption = 0 ' Số quả bóng bắn được ban đầu là 0
```

```
nlevel = 1 ' Mặc định trình độ chơi là 1
```

```
For i = 0 To 9
```

```
    If Rnd > 0.5 Then ' Chọn hướng di chuyển ngẫu nhiên cho 10 quả bóng
```

```
        direct1(i) = 1 ' theo chiều đứng và chiều ngang
```

```
    Else
```

```
        direct1(i) = -1
```

```
    End If
```

```
    If Rnd > 0.5 Then
```

```
        direct2(i) = 1
```

```
    Else
```

```
        direct2(i) = -1
```

```
    End If
```

```
    left1(i) = Shape1(i).Left ' Xác định vị trí xuất hiện của mỗi quả bóng
```

```
    top1(i) = Shape1(i).Top
```

```
    step1h(i) = Int(10 * Rnd) + 25 ' và độ dài mỗi bước dịch chuyển
```

```
    step1v(i) = Int(10 * Rnd) + 25
```

```
Next
```

```
ntop = 0
```

```
nleft = 0
```

```
ndegree = 90 ' Hướng súng mặc định là 90 độ (hướng thẳng lên)
```

```

gocquay = 5      ' Mặc định góc quay súng mỗi lần là 5 độ
dodai = 20       ' đoạn đường đi của viên đạn mỗi lần dịch chuyển là 20
' Hiển thị giá trị hướng súng hiện hành, tốc độ đạn và tốc độ quay nòng súng
lblgocquay.Caption = gocquay
lbldodai.Caption = dodai
lbldegree.Caption = ndegree
For i = 0 To 20  ' Góc bắn ban đầu của mỗi viên đạn đều bằng 90 độ
    ban(i) = 0   ' và trạng thái ban đầu của đạn đều chưa được bắn đi
    gocban(i) = ndegree
Next
thoigian = 0     ' Thời gian chơi ban đầu là 0
End Sub

' Khi nhấp tùy chọn Level 1 trong menu Option
Private Sub mnul1_Click()
nlevel = 1      ' đặt mức độ chơi về 1
mnul1.Checked = True    ' Bật dấu chọn trước mục chọn
mnul2.Checked = False   ' Tắt dấu chọn trước 2 mục chọn còn lại
mnul3.Checked = False   ' trong menu Option
End Sub

' Khi nhấp tùy chọn Level 2 trong menu Option
Private Sub mnul2_Click()
nlevel = 2      ' đặt mức độ chơi về 2
mnul2.Checked = True    ' Bật dấu chọn trước mục chọn
mnul1.Checked = False   ' Tắt dấu chọn trước 2 mục chọn còn lại
mnul3.Checked = False   ' trong menu Option
End Sub

' Khi nhấp tùy chọn Level 3 trong menu Option
Private Sub mnul3_Click()

```

```

nlevel = 3          ' đặt mức độ chơi về 3
mnu12.Checked = False ' Tất dấu chọn trước 2 mục chọn còn lại
mnu11.Checked = False ' trong menu Option
mnu13.Checked = True  ' Bật dấu chọn trước mục chọn
End Sub

' Thủ tục được gọi sau mỗi khoảng thời gian 5 mili giây
' Dùng để điều khiển việc chuyển động của các quả bóng
' và kiểm tra tình trạng đạn có bắn trúng bóng hay không
Private Sub Timer1_Timer()
' Duyệt qua tất cả 10 quả bóng, kiểm tra tình trạng của từng quả bóng
For i = 0 To 9
    ' Nếu quả bóng di chuyển chạm đến một trong bốn đường biên của form
    If (Shape1(i).Top < 0) Or _
        (Shape1(i).Top > frmXathu.Height - Shape1(i).Height) Or _
        (Shape1(i).Left < 0) Or _
        (Shape1(i).Left > frmXathu.Width - Shape1(i).Width) Then
        Shape1(i).Height = 50      ' đặt lại kích thước ban đầu cho bóng
        Shape1(i).Width = 50
        ' Chọn hướng di chuyển mới theo chiều ngang cho quả bóng
        If Rnd > 0.5 Then
            direct1(i) = 1      ' di chuyển sang phải
        Else
            direct1(i) = -1     ' di chuyển sang trái
        End If
        ' Chọn hướng di chuyển mới theo chiều dọc cho quả bóng
        If Rnd > 0.5 Then
            direct2(i) = 1      ' di chuyển hướng xuống
        Else

```

direct2(i) = -1 ' di chuyển hướng lên

End If

' Xác định vị trí xuất hiện ngẫu nhiên mới của quả bóng

left1(i) = Int(frmXathu.Width * Rnd)

top1(i) = Int(frmXathu.Height * Rnd)

' Xác định ngẫu nhiên khoảng cách theo chiều ngang và chiều dọc

' mà quả bóng sẽ dịch chuyển mỗi lần

step1h(i) = Int(20 * Rnd) + 25

step1v(i) = Int(20 * Rnd) + 25

End If

' Tự động tăng kích thước quả bóng sau mỗi lần dịch chuyển

Shape1(i).Height = Shape1(i).Height + 2

Shape1(i).Width = Shape1(i).Width + 2

' Tính độ dời quả bóng theo các hướng đã xác định

left1(i) = left1(i) + step1h(i) * direct1(i)

top1(i) = top1(i) + step1v(i) * direct2(i)

' dịch chuyển quả bóng đến vị trí mới

Shape1(i).Move left1(i), top1(i)

' Goc thủ tục Trungdich để kiểm tra xem có viên đạn nào bắn trúng bóng
' hay không

trungdich (i)

Next I

End Sub

' Thủ tục được gọi sau mỗi khoảng thời gian 5 mili giây

' Dùng để điều khiển việc chuyển động của các viên đạn

Private Sub Timer2_Timer()

' Duyệt qua tất cả 20 viên đạn, kiểm tra tình trạng của từng viên đạn

For i = 0 To 19

' Nếu viên đạn đã được bắn đi (đang di chuyển)

If ban(i) = 1 Then

 ' Tính khoảng cách dịch chuyển của viên đạn theo chiều đứng và
 ' chiều ngang

 ntop = dodai * Sin(gocban(i) * 3.141592653589 / 180)

 nleft = dodai * Cos(gocban(i) * 3.141592653589 / 180)

 ' Dịch chuyển viên đạn tới vị trí mới

 DAN(i).Top = DAN(i).Top - ntop

 DAN(i).Left = DAN(i).Left + nleft

End If

 ' Điều kiện dừng viên đạn (chuyển viên đạn trở lại trạng thái chưa được
 ' bắn đi) là khi thấy viên đạn di chuyển vượt qua một trong bốn
 ' đường biên của form

 If DAN(i).Top < 0 Or DAN(i).Left < 0 Or _

 DAN(i).Top > frmXathu.ScaleHeight Or _

 DAN(i).Left > frmXathu.ScaleWidth Then

 DAN(i).Top = 5745 ' đặt viên đạn trở lại vị trí chuẩn bị bắn đi

 DAN(i).Left = 3675

 ban(i) = 0 ' đặt trạng thái viên đạn trở lại trạng thái
 ' chờ bắn

End If

Next

End Sub

 ' Thủ tục được gọi để kiểm tra xem có viên đạn nào

 ' (trong số các viên đạn đã được bắn) chạm trúng một trong số 10 quả bóng
 ' đang chuyển động

Sub trungdich(ByVal index)

 ' Duyệt qua tất cả 20 viên đạn, kiểm tra tình trạng của từng viên đạn

 For i = 0 To 19

 ' Nếu viên đạn đã được bắn đi (đang di chuyển)

 If ban(i) = 1 Then


```

' và viên đạn chạm trúng quả bóng đang được duyệt
If DAN(i).Left >= Shape1(index).Left And _
    DAN(i).Left <= Shape1(index).Left + Shape1(index).Width And _
    DAN(i).Top >= Shape1(index).Top And _
    DAN(i).Top <= Shape1(index).Top + Shape1(index).Height Then
' Tăng số lượng bóng bị bắn trúng lên 1
soquano.Caption = soquano.Caption + 1
' Viên đạn sẽ biến mất bằng cách
' đặt viên đạn trở lại vị trí xuất phát và chuyển về trạng thái chờ
' được bắn đi
DAN(i).Top = 5745
DAN(i).Left = 3675
ban(i) = 0
' quả bóng sẽ nổ và xuất hiện quả bóng mới bằng cách
' đặt quả bóng trở lại kích thước ban đầu
Shape1(index).Height = 50
Shape1(index).Width = 50
' xác định ngẫu nhiên hướng dịch chuyển mới cho quả bóng
If Rnd > 0.5 Then
    direct1(index) = 1      ' Sang phải
Else
    direct1(index) = -1     ' Sang trái
End If
If Rnd > 0.5 Then
    direct2(index) = 1      ' Hướng xuống
Else
    direct2(index) = -1     ' Hướng lên
End If

```

' xác định vị trí xuất hiện mới của quả bóng một cách ngẫu nhiên

$\text{left1}(\text{index}) = \text{Int}(\text{frmXathu.Width} * \text{Rnd})$

$\text{top1}(\text{index}) = \text{Int}(\text{frmXathu.Height} * \text{Rnd})$

' xác định khoảng cách di chuyển mới của quả bóng

' theo chiều ngang và dọc một cách ngẫu nhiên

$\text{step1h}(\text{index}) = \text{Int}(20 * \text{Rnd}) + 25$

$\text{step1v}(\text{index}) = \text{Int}(20 * \text{Rnd}) + 25$

' Nếu quả bóng đang được duyệt đã bị bắn trúng

' thì không cần duyệt các viên đạn còn lại mà kết thúc thủ tục

' trung dịch ngay

Exit For

End If

End If

Next

End Sub

' Thủ tục được gọi sau mỗi khoảng thời gian 1 phút

' Dùng để đếm thời gian đã chơi và kiểm tra kết quả của người chơi

Private Sub Timer3_Timer()

' Cộng vào số thời gian đã chơi thêm 1 phút

thoigian = thoigian + 1

If thoigian = 1 Then *' nếu người chơi đã chơi đủ 1 phút*

Select Case nlevel

Case 1: *' với mức độ chơi đang chọn là Level 1*

If soquano >= 50 Then *' và số quả bóng bắn được từ 50 quả trở lên*

MsgBox "Bạn thắng rồi !" *' hiển thị thông báo chiến thắng*

Else *' nếu số quả bóng bắn được ít hơn 50*

MsgBox "Bạn thua rồi !" *' hiển thị thông báo thất bại*

End If

```

Case 2: ' với mức độ chơi đang chọn là Level 2
    If soquano >= 100 Then ' và số quả bóng bắn được từ 100 quả trở lên
        MsgBox "Bạn thắng rồi !" ' hiển thị thông báo chiến thắng
    Else ' nếu số quả bóng bắn được ít hơn 100
        MsgBox "Bạn thua rồi !" ' hiển thị thông báo thất bại
    End If
Case 3: ' với mức độ chơi đang chọn là Level 3
    If soquano >= 150 Then ' và số quả bóng bắn được từ 150 quả trở lên
        MsgBox "Bạn thắng rồi !" ' hiển thị thông báo chiến thắng
    Else ' nếu số quả bóng bắn được ít hơn 150
        MsgBox "Bạn thua rồi !" ' hiển thị thông báo thất bại
    End If
End Select

Form_Load ' bắt đầu một lượt chơi mới và bắt đầu đếm lại thời gian chơi
End If

End Sub

' Khi chọn lệnh Game ➡ New Game hay nhấn tổ hợp phím CTRL+N
' bắt đầu lượt chơi mới với mức độ chơi đang chọn

Private Sub mnuNew_Click()
    Noldlevel = nlevel ' lưu lại mức độ chơi đang chọn
    Form_Load ' Gọi thủ tục biến cố Form_load
    nlevel = noldlevel ' bắt đầu lượt chơi mới với mức độ chơi đang chọn
End Sub

' Khi chọn lệnh Game ➡ Quit Game hay nhấn tổ hợp phím CTRL+Q

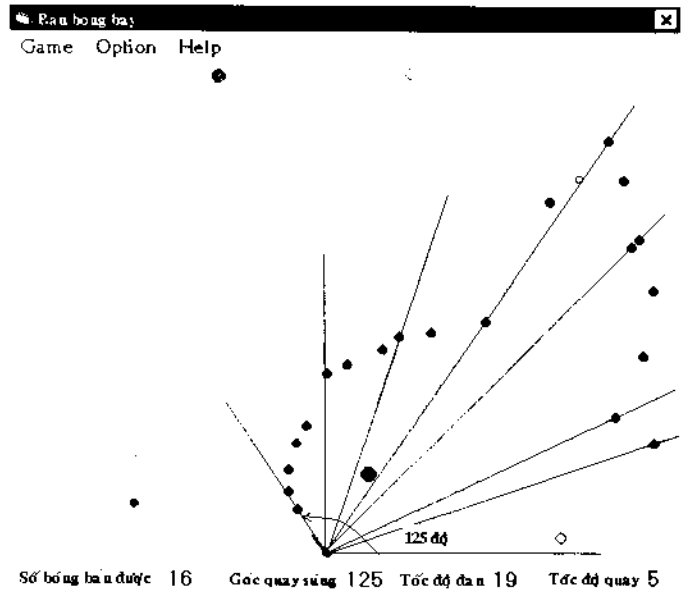
Private Sub mnuQuit_Click()
    End ' Kết thúc chương trình
End Sub

```

III. Phân tích chương trình và một số thuật toán sử dụng

1. Ý nghĩa các biến toàn cục đã khai báo

- ◆ **Mảng điều khiển *Dan***: Lưu vị trí, kích thước và màu sắc các viên đạn trong không gian chơi. Có tổng cộng 20 viên đạn.
- ◆ **Mảng *Ban***: Lưu trạng thái hiện hành của từng viên đạn (đang chờ bắn – giá trị 0 hay đã bắn và đang di chuyển – giá trị 1).
- ◆ **Mảng điều khiển *Shape1***: Lưu vị trí, kích thước và màu sắc các quả bóng trong không gian chơi. Có tổng cộng 10 quả bóng.
- ◆ **Mảng *Gocban***: Lưu góc bắn của từng viên đạn. Để ý rằng góc bắn cũng chính là hướng di chuyển của từng viên đạn đã được bắn đi so với phương nằm ngang, tính ngược chiều kim đồng hồ. Mặt khác, hướng di chuyển của các viên đạn có thể giống hoặc khác nhau (Hình 7.4 – Hướng di chuyển của các viên đạn là những đường nối từ tâm ụ súng đến tâm viên đạn, dễ thấy có ba cặp đạn di chuyển cùng hướng, còn lại đều có góc bắn khác nhau, trong đó hai viên đạn bắn sau cùng, có hướng di chuyển tạo với phương nằm ngang một góc 125 độ tính ngược chiều kim đồng hồ). Do đó, cần lưu lại hướng di chuyển của riêng từng viên đạn.
- ◆ **Mảng *Direct1***: Tương tự mỗi quả bóng cũng có hướng dịch chuyển khác nhau. Mảng *Direct1* lưu hướng dịch chuyển theo phương ngang của từng quả bóng (giá trị 1 – dịch sang phải, giá trị -1 – dịch sang trái).
- ◆ **Mảng *Direct2***: Lưu hướng dịch chuyển theo phương dọc của từng quả bóng (giá trị 1 – dịch xuống dưới, giá trị -1 – dịch lên trên).



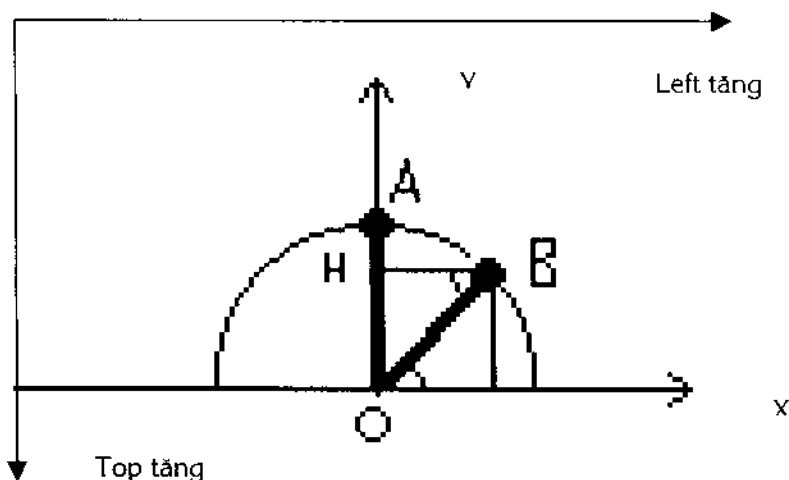
Hình 7.4 Các viên đạn đã bắn đi di chuyển theo nhiều hướng khác nhau.

- ♦ Mảng **Left1** và **Top1**: Lưu vị trí hiện hành của từng quả bóng trong không gian chơi.
- ♦ Mảng **Step1h**: Lưu khoảng cách dịch chuyển theo chiều ngang của từng quả bóng.
- ♦ Mảng **Step1v**: Lưu khoảng cách dịch chuyển theo chiều dọc của từng quả bóng.
- ♦ Biến **ndegree**: Lưu giá trị góc xác định hướng súng hiện hành (góc hợp với phương nằm ngang tính theo ngược chiều kim đồng hồ).
- ♦ Biến **Gocquay**: Xác định góc quay hiện hành mỗi lần thay đổi hướng súng.
- ♦ Biến **nlevel**: Xác định mức độ chơi hiện hành (có 3 mức độ chơi là 1, 2 hoặc 3).

- ♦ Biến **thoigian**: Xác định thời gian chơi đã trôi qua (tính bằng phút).
- ♦ Biến **nleft** và **nTop**: Xác định khoảng cách dịch chuyển theo chiều ngang và chiều dọc của đỉnh nòng súng hay viên đạn hiện hành.
- ♦ Biến **Dodat**: Xác định tốc độ bay hiện hành của tất cả các viên đạn.

2. Giải thích kỹ thuật quay nòng súng theo đúng một góc xác định

Để ý rằng độ dài nòng súng là không thay đổi. Theo thiết đặt lúc thiết kế điều khiển Line1 (X1 : 3690, X2 : 3690, Y1 : 5490, Y2 : 5790), dễ thấy độ dài nòng súng có giá trị cố định là $5790 - 5490 = 300$ Twip. Điều đó có nghĩa nòng súng sẽ quay trên một đường tròn bán kính 300 Twip với tâm (X2, Y2) (Hình 7.5).



Hình 7.5 Quay nòng súng từ vị trí cũ OA đến vị trí mới OB.

- * **Xét trường hợp 1:** Vị trí mới của nòng súng hợp với phương ngang OX một góc nhỏ hơn 90 độ (tính ngược chiều kim đồng hồ).

Giả sử, bạn muốn quay súng từ vị trí 90 độ (OA) đến vị trí mới q độ (OB). Thực chất là xác định tọa độ điểm B trong hệ tọa độ màn hình sao cho thỏa các điều kiện sau:

- Góc XOB = q độ (với $q \leq 90$ độ)
- Đoạn OB = 300 Twip

Để ý rằng, bạn đã biết tọa độ điểm O là (X2,Y2), trong trường hợp này là (3690,5790). Do vậy, bạn có thể dễ dàng tính được tọa độ điểm B, nếu xác định được độ dài các đoạn OH và HB.

Tam giác OHB là tam giác vuông có góc HBO = XOB = 45 độ và OB = 300 Twip là cạnh huyền.

$$OH = \sin(HOB) * OB = \sin(XOB) * 300$$

$$HB = \cos(HOB) * OB = \cos(XOB) * 300$$

Khi đó, tọa độ điểm B tính trong hệ trục tọa độ Decart XOY sẽ là:

$$Y1 = Y2 + OH$$

$$X1 = X2 + HB$$

Tuy nhiên, trong hệ tọa độ màn hình, góc tọa độ bắt đầu từ góc trên bên trái màn hình. Điều đó có nghĩa càng xuống dưới giá trị TOP càng tăng. Do vậy, trong hệ tọa độ màn hình, công thức để tính tọa độ điểm B(X1,Y1) sẽ là:

$$Y1 = Y2 - OH = Y2 - \sin(XOB) * 300$$

$$X1 = X2 + HB = X2 + \cos(XOB) * 300$$

Do các hàm Sin và Cos trong Visual Basic nhận những đối số có dạng Radian, trong khi bạn đang tính góc XOB theo độ, vì thế, bạn cần chuyển đổi chúng thành Radian:

$$XOB \text{ (radian)} = XOB \text{ (độ)} * 3.141592653589 / 180$$

Tóm lại, công thức tính tọa độ đỉnh nòng súng - điểm B (X1,Y1) trong hệ tọa độ màn hình khi biết góc quay (tính theo độ), độ dài súng và vị trí ụ súng O (X2,Y2):

$$Y1 = Y2 - \sin(XOB * 3.141592653589 / 180) * 300 \quad (*)$$

$$X1 = X2 + \cos(XOB * 3.141592653589 / 180) * 300 \quad (**)$$

- * **Xét trường hợp 2:** Vị trí mới của nòng súng hợp với phương ngang OX một góc lớn hơn 90 độ (tính ngược chiều kim đồng hồ).

Bạn có thể dễ dàng chứng minh được công thức (*) và (**) vẫn áp dụng đúng trong trường hợp này.

3. Kỹ thuật dịch chuyển viên đạn theo đúng hướng đã bắn ra với tốc độ xác định

Tương tự như kỹ thuật quay nòng súng. Điểm khác biệt ở đây là:

- Góc XOB thay vì có thể thay đổi, giờ đây lại cố định với giá trị *gocban(i)*.
- Đoạn OB thay vì cố định 300 Twip, giờ đây lại có thể thay đổi theo giá trị biến *dodai*.

Kết quả, vị trí mới của viên đạn thứ *i* sẽ được tính theo công thức sau:

$$Top_{mới} = Top_{cũ} - dodai * \sin(gocban(i) * 3.141592653589 / 180)$$

$$Left_{mới} = Left_{cũ} + dodai * \cos(gocban(i) * 3.141592653589 / 180)$$

4. Kỹ thuật dịch chuyển các quả bóng

Kỹ thuật dịch chuyển những viên đạn và các quả bóng thực chất đều là những dịch chuyển có hướng xác định (góc và hướng dịch chuyển kể từ khi viên đạn bắn ra hay quả bóng phát sinh cho đến khi

biến mất là không thay đổi) và khoảng cách giữa các lần dịch chuyển đều cố định. Có thể nói, đây là một phương án dịch chuyển khác mà thay vì xác định chính xác trước góc dịch chuyển, lập trình viên lại xác định chính xác trước khoảng cách và hướng dịch chuyển theo chiều ngang và chiều dọc.

$$\text{Top}_{\text{mới}} = \text{Top}_{\text{cũ}} + \text{dodai} * \text{huong1}$$

$$\text{Left}_{\text{mới}} = \text{Left}_{\text{cũ}} + \text{dodai} * \text{huong2}$$

Trong đó:

Huong1 xác định hướng dịch chuyển dọc (lên / xuống).

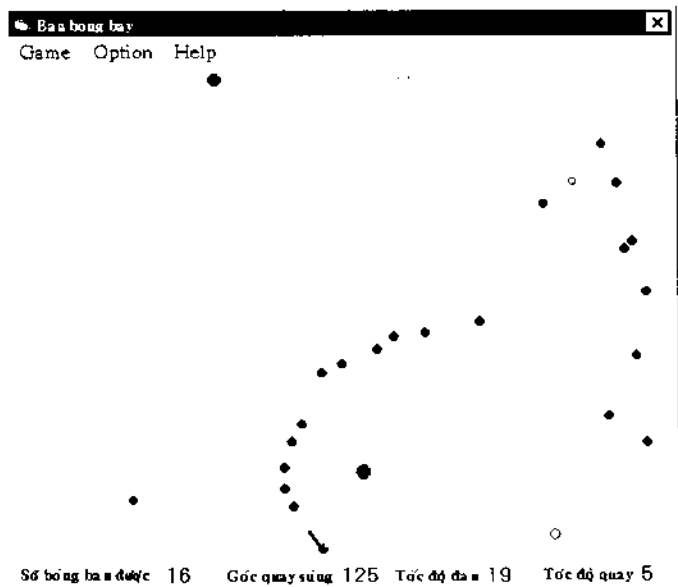
Huong2 xác định hướng dịch chuyển ngang (trái / phải).

Ví dụ: Trở lại Hình 7.5, để viên đạn luôn dịch chuyển theo hướng OB, bạn cần đảm bảo một trong hai điều kiện sau trong mỗi lần dịch chuyển:

- Hoặc góc dịch chuyển luôn bằng BOX.
- Hoặc khoảng cách dịch chuyển theo chiều ngang luôn bằng HB và khoảng cách dịch chuyển theo chiều đứng luôn bằng OH.

IV. Bài tập thực hành

Bài 1. Rõ ràng với chương trình đã xây dựng, người chơi chỉ thua khi bắn không đủ số bóng yêu cầu sau thời gian 1 phút. Để tăng sức hấp dẫn và độ khó đối với người chơi, bạn hãy phát triển chương trình sao cho trong quá trình chơi, khi có bất kỳ một quả bóng nào di chuyển chạm phải nòng súng, người chơi sẽ thua và trò chơi kết thúc.



Hình 7.6 Trò chơi kết thúc do người chơi để quả bóng màu cam dịch chuyển chạm vào nòng súng.

BÀI 8.

TRÒ CHƠI KHÔNG CHIẾN

I. Mô tả trò chơi

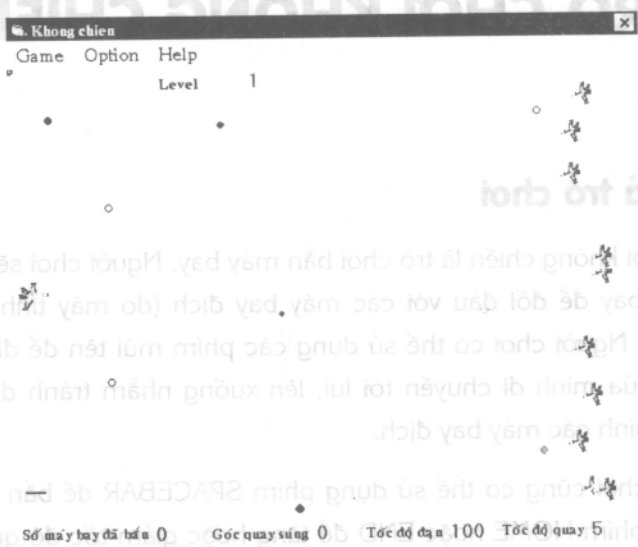
Trò chơi không chiến là trò chơi bắn máy bay. Người chơi sẽ sử dụng một máy bay để đối đầu với các máy bay địch (do máy tính tự động phát sinh). Người chơi có thể sử dụng các phím mũi tên để điều chỉnh máy bay của mình di chuyển tới lui, lên xuống nhằm tránh đạn phải đạn hay chính các máy bay địch.

Người chơi cũng có thể sử dụng phím SPACEBAR để bắn máy bay địch. Các phím HOME hoặc END để tăng hoặc giảm tốc độ quay súng trước khi bắn. Các phím PAGEUP hoặc PAGEDOWN để điều chỉnh góc bắn. Các phím + xám hoặc – xám để tăng hoặc giảm tốc độ bay của đạn trong giới hạn cho phép.

Trò chơi sẽ chia làm ba mức, thời gian chơi mỗi mức là 3 phút. Bắt đầu từ mức 1.

- ♦ *Mức 1:* 10 máy bay địch dàn hàng ngang bay về phía máy bay của người chơi. Khi một trong các máy bay địch bị bắn rơi hoặc bay khuất sau những đường biên form, một máy bay khác lập tức sẽ cất cánh để thay thế (Hình 8.1).

- ♦ **Mức 2:** Các máy bay địch sẽ hướng về máy bay của người chơi, nhưng theo đường chéo bất kỳ. Tốc độ bay của các máy bay địch cũng nhanh hơn mức 1.
- ♦ **Mức 3:** Các máy bay địch sẽ bay hướng về máy bay của người chơi, nhưng liên tục đổi hướng theo các đường zig zắc, đồng thời bắn trả lại máy bay người chơi.



Hình 8.1 Các máy bay địch ban đầu dàn hàng ngang bay về phía máy bay người chơi.

Trong vòng 3 phút nếu máy bay của người chơi không đụng phải máy bay địch và không bị máy bay địch bắn hạ thì chương trình sẽ tự động chuyển sang mức cao hơn.

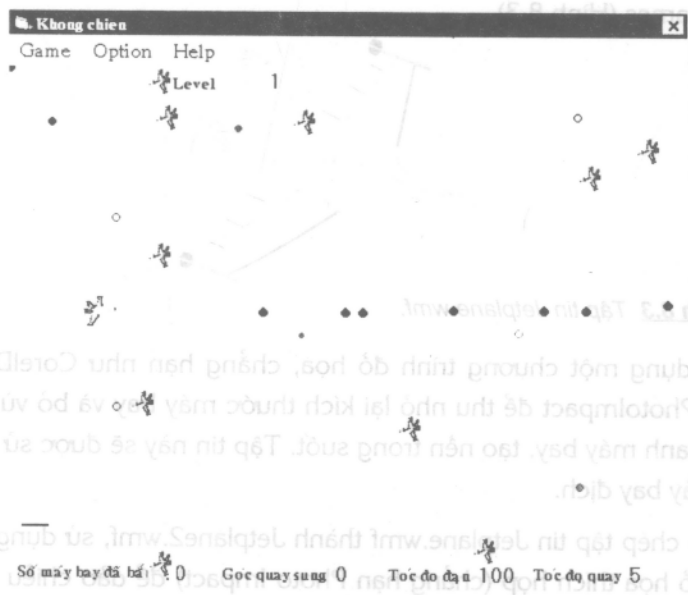
Trò chơi sẽ kết thúc với một trong những trường hợp sau:

- ♦ Người chơi hoàn thành sắc cả ba mức chơi với tổng thời gian chơi là 9 phút mà máy bay không bị bắn hạ. Xem như người chơi đã chiến thắng.

- ◆ Khi chiếc máy bay của người chơi va chạm với một trong các máy bay địch.
- ◆ Khi máy bay của người chơi bị bắn trúng đạn của các máy bay địch ở mức chơi 3.

Trên màn hình trò chơi, bạn có thể thêm các quả bóng hay đám mây làm màn hình đẹp mắt, đồng thời tăng độ khó của trò chơi vì nó làm phân tán sự tập trung của người chơi trong quá trình bay.

Người chơi chỉ bắn tối đa được mỗi lượt 20 viên đạn. Các viên đạn sẽ bay cho đến khi đụng trúng một máy bay địch hay bay khuất khỏi những đường biên form. Lúc đó, người chơi mới có thể tiếp tục bắn loạt đạn mới. Nói cách khác, tại một thời điểm chơi bất kỳ, số viên đạn đang di chuyển trên màn hình tối đa là 20 (xem Hình 8.2).



Hình 8.2 Máy bay người chơi đang tiến lên và chặn bắn máy bay địch.

Nhận xét: Nếu đã hiểu kỹ bài 7, bạn sẽ nhanh chóng phát hiện trò chơi không chiến thực chất là một biến dạng của trò chơi xạ thủ, nhưng đã được cải tiến ở mức độ cao hơn. Chính vì vậy, mà chúng tôi sẽ không giải thích lại các giải thuật, mà chỉ tập trung trình bày cách thiết kế và cách viết mã lệnh chương trình.

II. Thiết kế chương trình

1. Chuẩn bị hình ảnh:

Tối thiểu bạn phải chuẩn bị hai hình: máy bay ta và máy bay địch. Ngoài ra, nếu có thể, bạn có thể tạo thêm hình các đám mây.

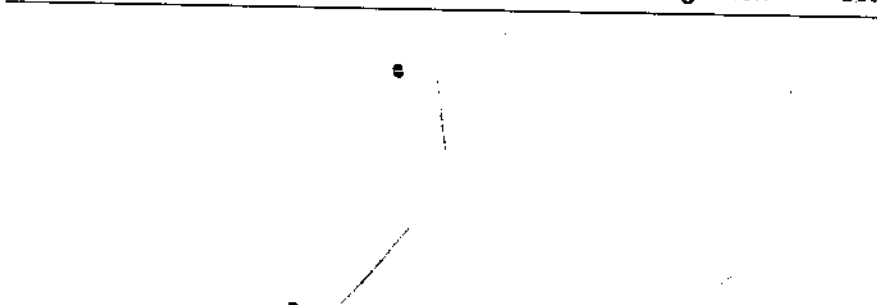
Nếu máy của bạn đã cài Microsoft Office, vào thư mục C:\Program Files\Microsoft Office\Clipart, copy tập tin Jetplane.wmf về thư mục C:\LTGames (Hình 8.3).



Hình 8.3 Tập tin Jetplane.wmf.

Sử dụng một chương trình đồ họa, chẳng hạn như CorelDraw hay Ulead PhotoImpact để thu nhỏ lại kích thước máy bay và bỏ vùng trắng bao quanh máy bay, tạo nền trong suốt. Tập tin này sẽ được sử dụng để làm máy bay địch.

Sao chép tập tin Jetplane.wmf thành Jetplane2.wmf, sử dụng chương trình đồ họa thích hợp (chẳng hạn Photo Impact) để đảo chiều máy bay như Hình 8.4. Tập tin này sẽ dùng làm máy bay của người chơi.



Hình 8.4 Tập tin Jetplane2.wmf.

2. Thiết kế giao diện

Bước 1. Khởi động Visual Basic bằng cách chọn Start ➤ Programs ➤ Microsoft Visual Studio 6.0 ➤ Microsoft Visual Basic 6.0. Trong cửa sổ New Project, nhấp đúp biểu tượng Standard Exe (tương đương với lệnh File ➤ New Project).

Bước 2. Visual Basic sẽ tạo một project mới, form mặc định là form1. Nhấn F4 (tương đương với lệnh View ➤ Properties Window) để kích hoạt cửa sổ Properties, thay đổi các thuộc tính form như sau:

- Name: frmKhongchien
- Caption: Không chiến
- WindowState: 0 – Normal
- ScaleMode: 1 - Twip
- Width: 8295
- Height: 7155
- BackColor: &H00FFC0C0&
- FillColor: &H00FF0000&

- Maxbutton: False
- Minbutton: False
- Moveable: True
- StartupPosition: 2 – CenterScreen

Chọn lệnh File ➡ Save Project, lưu form với tên *Khongchien.frm* và lưu project với tên *Khongchien.vbp* trong thư mục xác định, chẳng hạn c:\ltgames.

Bước 3. Thiết kế menu chương trình.

Mở form *frmKhongchien*. Xong chọn lệnh Tools ➡ Menu Editor.

Lần lượt định nghĩa các mục trên menu ngang và menu xổ xuống theo bảng sau:

Caption	Name	Checked	Shortcut
&Game	MnuGame		
..... New game	MnuNew		Ctrl+N
..... -	MnuSep1		
..... Quit game	MnuQuit		Ctrl+Q
&Option	MnuOptions		
..... Level 1	MnuL1	True	
..... Level 2	MnuL2		
..... Level 3	MnuL3		
&Help	MnuHelp		
..... About	MnuAbout		

Bước 4. Đặt các điều khiển vào form.

Đặt lên form 3 điều khiển Timer, lần lượt định lại thuộc tính Internal cho từng Timer như sau:

- Điều khiển Timer 1:
Name: Timer1, Interval = 5
- Điều khiển Timer 2:
Name: Timer2, Interval = 5
- Điều khiển Timer 3:
Name: Timer3, Interval = 60000

Đặt lên form 1 điều khiển Label và định lại các thuộc tính như sau:

Name = Label1, Alignment = 0 – Left, Visible = True
Caption: Số máy bay đã bắn, BackColor = &H80000009&
BackStyle = 1 – Opaque, AutoSize = False, BorderStyle = 0 – None,
Fontname = VNI-Times, FontSize = 10, FontStyle = Bold,
ForeColor = &H80000012&,
Height = 375, Width = 1710, Left = 90, Top = 5910

Sao chép điều khiển Label1 thành 10 Label nữa, định lại các thuộc tính và sắp chúng thành hai hàng ngang (một hàng phía trên và một hàng bên dưới) như Hình 8.5:

1. Điều khiển Label thứ 2	
Name: Somaybay	Top: 5910
Caption: (rỗng)	Left: 1815
2. Điều khiển Label thứ 3	
Name: label2	Top: 5910
Caption: Góc quay súng	Left: 2550

3. Điều khiển Label thứ 4	
Name: lblDegree	Top: 5910
Caption: (rỗng)	Left: 3900
4. Điều khiển Label thứ 5	
Name: label3	Top: 5910
Caption: Tốc độ đạn	Left: 4545
5. Điều khiển Label thứ 6	
Name: lblDodai	Top: 5910
Caption: (rỗng)	Left: 5625
6. Điều khiển Label thứ 7	
Name: Label 4	Top: 5910
Caption: Tốc độ quay	Left: 6300
7. Điều khiển Label thứ 8	
Name: lblgocquay	Top: 5910
Caption: (rỗng)	Left: 7455
8. Điều khiển Label thứ 9	
Name: label5	Top: 30
Caption: Level	Left: 1965
9. Điều khiển Label thứ 10	
Name: lblLevel	Top: 30
Caption: (rỗng)	Left: 3120
10. Điều khiển Label thứ 11	
Name: label6	Top: 30
Caption: Hướng đạn	Left: 4155

Để tạo thêm các quả bóng có màu sắc khác nhau cho đẹp mắt, lần lượt đặt tiếp lên form 10 điều khiển Shape nữa với cùng tên Shape1, tạo thành mảng điều khiển 10 phần tử có chỉ số Index từ 0 đến 9 và định lại các thuộc tính như sau:

1. Điều khiển Shape thứ 1	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H008080FF&	Index: 0
Width: 105	Left: 3450
Height: 105	Top: 3180
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H000000FF&
FillStyle: 0 – Solid	Shape: 3 – Circle
2. Điều khiển Shape thứ 2	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H00404040&	Index: 1
Width: 105	Left: 6075
Height: 105	Top: 3150
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00C0FFC0&
FillStyle: 0 – Solid	Shape: 3 – Circle
3. Điều khiển Shape thứ 3	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H80000008&	Index: 2
Width: 105	Left: 1230

Height: 105	Top: 1770
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00E0E0E0&
FillStyle: 0 – Solid	Shape: 3 – Circle
4. Điều khiển Shape thứ 4	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H000080FF&	Index: 3
Width: 105	Left: 4845
Height: 105	Top: 1545
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H000080FF&
FillStyle: 0 – Solid	Shape: 3 – Circle
5. Điều khiển Shape thứ 5	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H0080FF80&	Index: 4
Width: 105	Left: 1230
Height: 105	Top: 4050
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H0080FF80&
FillStyle: 0 – Solid	Shape: 3 – Circle
6. Điều khiển Shape thứ 6	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H0080C0FF&	Index: 5

Width: 105	Left: 4905
Height: 105	Top: 4350
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H0080C0FF&
FillStyle: 0 – Solid	Shape: 3 – Circle
7. Điều khiển Shape thứ 7	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H00008000&	Index: 6
Width: 105	Left: 2700
Height: 105	Top: 690
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00008000&
FillStyle: 0 – Solid	Shape: 3 – Circle
8. Điều khiển Shape thứ 8	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H80000008&	Index: 7
Width: 105	Left: 6795
Height: 105	Top: 540
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00FFFF00&
FillStyle: 0 – Solid	Shape: 3 – Circle
9. Điều khiển Shape thứ 9	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H80000008&	Index: 8

Width: 105	Left: 465
Height: 105	Top: 615
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00FF0000&
FillStyle: 0 – Solid	Shape: 3 – Circle
10. Điều khiển Shape thứ 10	
Name: Shape1	BackColor: &H80000005&
BorderColor: &H00404040&	Index: 9
Width: 105	Left: 4980
Height: 105	Top: 6810
BorderStyle: 1 – Solid	BackStyle: 0 – Transparent
BorderWidth: 1	FillColor: &H00FF00FF&
FillStyle: 0 – Solid	Shape: 3 – Circle

Sử dụng công cụ Line, vẽ một đoạn thẳng trên form và định lại các thuộc tính như sau:

Name: Line1, BoderColor: &H80000008&, BoderStyle: 6 – Inside Solid,
BorderWidth: 2, X1 : 5775, X2 : 5535, Y1 : 240, Y2 : 240

Đặt lên form thêm 1 điều khiển Shape mới và định lại các thuộc tính như sau:

Name: DAN	BackColor: &H80000006&
BorderColor: &H80000008&	Index: 0
Width: 45	Left: 0
Height: 45	Top: 0
BorderStyle: 1 – Solid	BackStyle: 1 – Opaque

BorderWidth: 6	FillColor: &H00000000&
FillStyle: 1 – Transparent	Shape: 3 – Circle

Sao chép điều khiển DAN thành 19 điều khiển nữa để tạo mảng điều khiển DAN gồm 20 phần tử có chỉ số Index từ 0 đến 19.

Định lại thuộc tính Top và Left cho 20 điều khiển DAN là:

Top: 0, Left: 0

- ◆ Bổ sung các máy bay vào form:

Đặt lên form một điều khiển Image và định lại các thuộc tính như sau:

Name: mbta	Stretch: True
BorderStyle: 0 – None	Picture: chọn tập tin Jetplane2.wmf
Width: 510	Left: 15
Height: 480	Top: 2715

Đặt lên form một điều khiển Image khác và định lại các thuộc tính như sau:

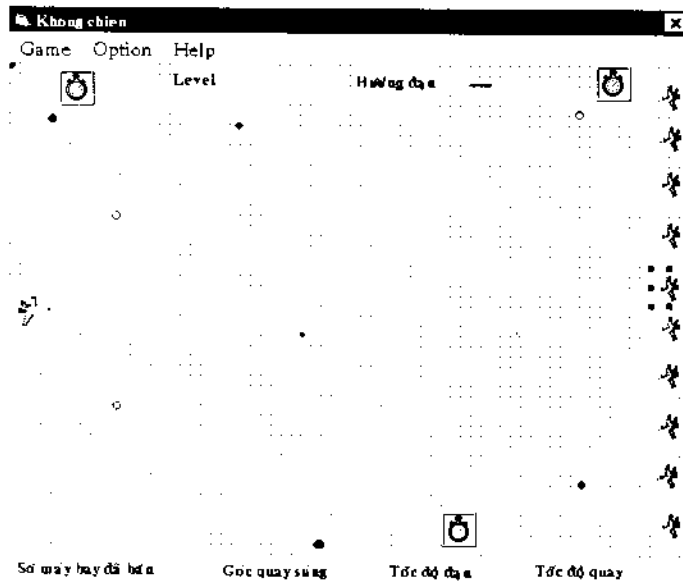
Name: mbdich	Stretch: True
BorderStyle: 0 – None	Picture: chọn tập tin Jetplane1.wmf
Width: 350	Left: 7770
Height: 350	Top: 1230

Sao chép điều khiển Image mbdich thành 9 điều khiển nữa để tạo mảng điều khiển 10 phần tử có chỉ số Index từ 0 đến 9.

Định lại vị trí xuất phát của 10 máy bay địch theo bảng sau:

Name: mbdich	Index: 0
Top: 1230	Left: 7770
Name: mbdich	Index: 1
Top: 195	Left: 7770
Name: mbdich	Index: 2
Top: 690	Left: 7770
Name: mbdich	Index: 3
Top: 1845	Left: 7770
Name: mbdich	Index: 4
Top: 2475	Left: 7770
Name: mbdich	Index: 5
Top: 2955	Left: 7770
Name: mbdich	Index: 6
Top: 3525	Left: 7770
Name: mbdich	Index: 7
Top: 4140	Left: 7770
Name: mbdich	Index: 8
Top: 4740	Left: 7770
Name: mbdich	Index: 9
Top: 5340	Left: 7770

Kết quả màn hình thiết kế form *frmKhongchien* sẽ như Hình 8.5.



Hình 8.5 Màn hình thiết kế form frmKhongchien.

3. Viết mã lệnh

' Khởi tạo các biến toàn cục

Dim direct1(10), direct2(10), left1(10), top1(10), step1h(10), step1v(10)

Dim ban(20), gochan(20), ndegree, nleft, ntop, gocquay, huong, dodai, thoigian, nlevel

' Khi người chơi nhấn một phím bất kỳ

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)

Select Case KeyCode

Case 35: *' nhấn phím End*

If gocquay > 1 Then

gocquay = gocquay - 1 *' giảm góc quay súng*

lblgocquay.Caption = gocquay

End If

Case 36: ' *nhấn phím Home*

If gocquay < 30 Then

gocquay = gocquay + 1 ' *tăng góc quay súng*

lblgocquay.Caption = gocquay

End If

Case 34: ' *nhấn phím PageUp*

' *tăng dần góc bắn đạn*

ndegree = ndegree + gocquay

' *Hướng đạn bắn ra thay đổi trong phạm vi từ -90 đến 90 độ (Hình 8.6)*

If ndegree > 90 Then

ndegree = 90

End If

If ndegree < -90 Then

ndegree = -90

End If

lbldegree.Caption = ndegree

ntop = 300 * Sin(ndegree * 3.141592653589 / 180)

nleft = 300 * Cos(ndegree * 3.141592653589 / 180)

Line1.Y1 = Line1.Y2 - ntop

Line1.X1 = Line1.X2 + nleft

Case 33: ' *nhấn phím PageDown*

' *Giảm dần góc bắn đạn*

ndegree = ndegree - gocquay

' *Hướng đạn bắn ra thay đổi trong phạm vi từ 90 đến - 90 độ*

If ndegree > 90 Then

ndegree = 90

```

End If
If ndegree < -90 Then
    ndegree = -90
End If
lbldegree.Caption = ndegree
ntop = 300 * Sin(ndegree * 3.141592653589 / 180)
nleft = 300 * Cos(ndegree * 3.141592653589 / 180)
Line1.Y1 = Line1.Y2 - ntop
Line1.X1 = Line1.X2 + nleft
Line1.X1 = Line1.X2 + nleft

```

Case 32: ' Nhấn phím Spacebar

```

For i = 0 To 19
    If ban(i) = 0 Then
        ban(i) = 1
        DAN(i).Top = mbta.Top + mbta.Height / 2
        DAN(i).Left = mbta.Left + mbta.Width
        gochan(i) = ndegree
    Exit For
End If
Next

```

Case 38: ' Nhấn phím ↑

```

mbta.Top = mbta.Top - 20
If mbta.Top <= 0 Then
    mbta.Top = 0
End If

```

Case 40: ' Nhấn phím ↓

```

mbta.Top = mbta.Top + 20

```

```
If mbta.Top >= frmkhongchien.ScaleHeight - 1000 Then
```

```
    mbta.Top = frmkhongchien.ScaleHeight - 1000
```

```
End If
```

```
Case 37: ' Nhấn phím ←
```

```
    mbta.Left = mbta.Left - 20
```

```
If mbta.Left <= 0 Then
```

```
    mbta.Left = 0
```

```
End If
```

```
Case 39: ' Nhấn phím →
```

```
    mbta.Left = mbta.Left + 20
```

```
If mbta.Left >= frmkhongchien.ScaleWidth - 2000 Then
```

```
    mbta.Left = frmkhongchien.ScaleWidth - 2000
```

```
End If
```

```
Case 107: ' Nhấn phím + xám (bên phải bàn phím)
```

```
If dodai < 500 Then ' Tốc độ bay của viên đạn tối đa là 500 đơn vị
```

```
    dodai = dodai + 10 ' Tăng tốc độ bay của viên đạn
```

```
    lbldodai.Caption = dodai ' Hiển thị tốc độ bay
```

```
End If
```

```
Case 109: ' Nếu nhấn phím - xám (bên phải bàn phím)
```

```
If dodai > 100 Then ' Tốc độ bay của viên đạn tối thiểu là 10
```

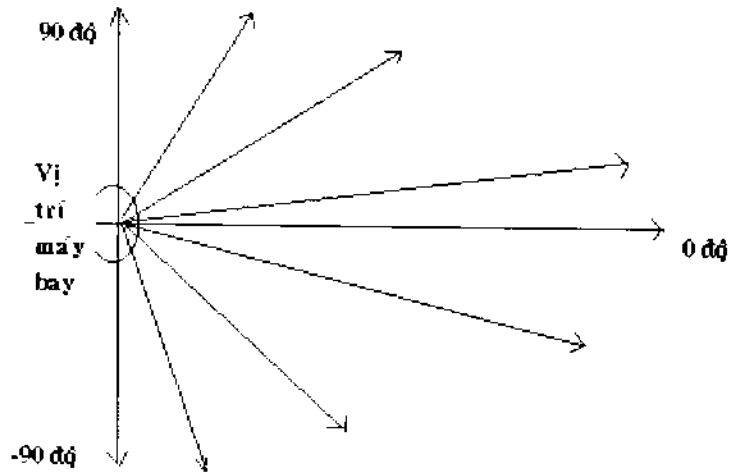
```
    dodai = dodai - 10 ' Giảm tốc độ bay của viên đạn
```

```
    lbldodai.Caption = dodai ' Hiển thị tốc độ bay
```

```
End If
```

```
End Select
```

```
End Sub
```



Hình 8.6 Người chơi có thể thay đổi hướng đạn bắn ra nhưng phải luôn thuộc cùng phần tư thứ nhất hoặc thứ tư.

Thủ tục được gọi khi bắt đầu chương trình hay khi bắt đầu một lượt chơi mới

Private Sub Form_Load()

somaybay.Caption = 0

nlevel = 1

Randomize

For i = 0 To 9

 If Rnd > 0.5 Then

 direct2(i) = 1

 Else

 direct2(i) = -1

 End If

 direct1(i) = -1

 left1(i) = frmkhongchien.Width - Int(1000 * Rnd) + 50

 top1(i) = mbdich(i).Top

```

step1h(i) = 50
step1v(index) = 0
Next
ntop = 0
nleft = 0
ndegree = 0 ' Góc bắn đạn mặc định là nằm ngang
gocquay = 5 ' Mỗi lần điều chỉnh tăng hoặc giảm góc bắn đạn là 5 độ
dodai = 100 ' đoạn đường dịch chuyển mỗi lần của viên đạn
lblgocquay.Caption = gocquay
lbldodai.Caption = dodai
lbldegree.Caption = ndegree
For i = 0 To 20
    ban(i) = 0
    gocban(i) = ndegree
Next
thoigian = 0
lbllevel = nlevel
End Sub
' Khi người chơi chọn lệnh Option ➡ Level 1
Private Sub mnul1_Click()
nlevel = 1
mnul2.Checked = False
mnul1.Checked = True
mnul3.Checked = False
End Sub
' Khi người chơi chọn lệnh Option ➡ Level 2
Private Sub mnul2_Click()
nlevel = 2

```

```
mnul2.Checked = True  
mnul1.Checked = False  
mnul3.Checked = False
```

End Sub

' Khi người chơi chọn lệnh Option ➡ Level 3

Private Sub mnul3_Click()

```
nlevel = 3  
mnul2.Checked = False  
mnul1.Checked = False  
mnul3.Checked = True
```

End Sub

' Thủ tục được gọi tự động sau mỗi 5 mili giây 1 lần

Private Sub Timer1_Timer()

```
for i = 0 To 9  
    If (mbdich(i).Top < 0) Or _  
        (mbdich(i).Top > frmkhongchien.Height - mbdich(i).Height) Or _  
        (mbdich(i).Left < 0) Or _  
        (mbdich(i).Left > frmkhongchien.Width - mbdich(i).Width) Then  
        mbdich(i).Height = 350  
        mbdich(i).Width = 350  
        If Rnd > 0.5 Then  
            direct2(i) = 1  
        Else  
            direct2(i) = -1  
        End If  
        direct1(i) = -1  
        left1(i) = frmkhongchien.Width - 500  
        top1(i) = Int(frmkhongchien.Height * Rnd)
```

```
step1h(i) = 50
```

```
Select Case nlevel
```

```
Case 1:
```

```
step1v(index) = 0 ' Máy bay di chuyển theo chiều ngang
```

```
Case 2:
```

```
step1v(index) = Int(10 * Rnd) ' Máy bay di chuyển trên các đường xiên
```

```
Case 3:
```

```
' Máy bay di chuyển trên các đường có độ dốc cao hơn mức 2
```

```
step1v(index) = Int(20 * Rnd)
```

```
End Select
```

```
End If
```

```
left1(i) = left1(i) + step1h(i) * direct1(i)
```

```
top1(i) = top1(i) + step1v(i) * direct2(i)
```

```
mbdich(i).Move left1(i), top1(i)
```

```
trungdich (i)
```

```
' Kiểm tra xem máy bay của người chơi có đụng phải các máy bay địch
```

```
If mbta.Left + mbta.Height - 10 >= mbdich(i).Left And _
```

```
mbta.Left + mbta.Height - 10 <= mbdich(i).Left + mbdich(i).Width And _
```

```
mbta.Top + mbta.Width / 2 >= mbdich(i).Top And _
```

```
mbta.Top + mbta.Width / 2 <= mbdich(i).Top + mbdich(i).Height Then
```

```
MsgBox "Bạn thua rồi"
```

```
Form_Load ' Bắt đầu lại lượt chơi mới
```

```
End If
```

```
Next i
```

```
End Sub
```

```
' Thủ tục được gọi tự động sau mỗi 5 mili giây 1 lần
```

```
Private Sub Timer2_Timer()
```

```
For i = 0 To 19
```


If ban(i) = 1 Then

ntop = dodai * Sin(gocban(i) * 3.141592653589 / 180)

nleft = dodai * Cos(gocban(i) * 3.141592653589 / 180)

DAN(i).Top = DAN(i).Top - ntop

If gocban(i) < 90 Then

DAN(i).Left = DAN(i).Left + nleft

Else

DAN(i).Left = DAN(i).Left + nleft

End If

End If

' Điều kiện dừng viên đạn

If DAN(i).Top < 0 Or DAN(i).Left < 0 Or _

DAN(i).Top > frmkhongchien.ScaleHeight Or _

DAN(i).Left > frmkhongchien.ScaleWidth Then

DAN(i).Top = 5745

DAN(i).Left = 3675

ban(i) = 0

End If

Next

End Sub

' Kiểm tra xem có máy bay địch nào bị trúng đạn hay không

Sub trungdich(ByVal index)

For i = 0 To 19

If ban(i) = 1 Then

If DAN(i).Left >= mbdich(index).Left And _

DAN(i).Left <= mbdich(index).Left + mbdich(index).Width And _

DAN(i).Top >= mbdich(index).Top And _

DAN(i).Top <= mbdich(index).Top + mbdich(index).Height Then

```
somaybay.Caption = somaybay.Caption + 1
' Viên đạn bắn trúng máy bay địch sẽ biến mất
DAN(i).Top = 5745
DAN(i).Left = 3675
ban(i) = 0
' Và máy bay địch bị bắn trúng sẽ bị rơi
direct1(index) = -1
If Rnd > 0.5 Then
    direct2(index) = 1
Else
    direct2(index) = -1
End If
left1(index) = frmkhongchien.Width - 500
top1(index) = Int(frmkhongchien.Height * Rnd)
step1h(index) = 50
Select Case nlevel
    Case 1:
        step1v(index) = 0
    Case 2:
        step1v(index) = Int(10 * Rnd)
    Case 3:
        step1v(index) = Int(20 * Rnd)
End Select
Exit For
End If
End If
Next
End Sub
```

' Thủ tục được gọi tự động sau mỗi phút 1 lần

Private Sub Timer3_Timer()

' Tăng biến đếm thời gian chơi thêm 1 giây

thoigian = thoigian + 1

If thoigian = 3 Then *' Nếu đã đủ 3 phút*

Select Case nlevel

Case 1: *' và đang ở mức chơi 1*

nlevel = nlevel + 1 *' thì chuyển sang mức chơi 2*

Case 2: *' và đang ở mức chơi 2*

nlevel = nlevel + 1 *' thì chuyển sang mức chơi 3*

Case 3: *' và đang ở mức chơi 3*

MsgBox "Bạn thắng rồi !" *' hiển thị thông báo chiến thắng*

nlevel = 1 *' bắt đầu lượt chơi mới với mức thấp nhất*

End Select

noldlevel = nlevel

Form_Load *' bắt đầu lượt chơi mới*

nlevel = noldlevel

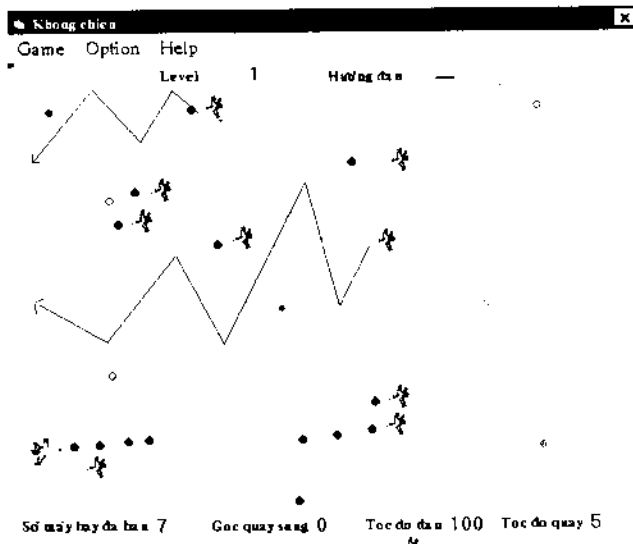
lbllevel = nlevel

End If

End Sub

III. Bài tập nâng cao

Bài 1. Trong phần mã lệnh đã trình bày, câu lệnh `step1v(index) = Int(20 * Rnd)` ở mức chơi chỉ giúp các máy bay địch có khả năng di chuyển trên những đường có độ dốc cao hơn mức 2. Hãy phát triển đoạn mã lệnh này để máy bay địch có thể bắn lại máy bay người chơi. Và di chuyển ngẫu nhiên trên các đường zic zắc (Hình 8.7).



Hình 8.7 Máy bay địch vừa bắn lại, vừa di chuyển ngẫu nhiên trên các đường zig zắc.

Bài 2. Rõ ràng với chương trình đã cải tiến trong bài tập nâng cao 1, người chơi sẽ rất khó hoàn thành mức chơi 3. Để khuyến khích người chơi, giờ đây bạn hãy tăng thêm số lượng máy bay của người chơi thành 3 chiếc. Người chơi chỉ thua cuộc khi nào cả 3 máy bay đều bị bắn hạ.

Lưu ý rằng mỗi thời điểm người chơi chỉ điều khiển duy nhất một máy bay. Chỉ khi máy bay đang điều khiển bị bắn hạ, thì máy bay thứ 2 mới cất cánh để tiếp tục cuộc chơi. Thời gian chơi vẫn tiếp tục cộng dồn mà không phải chơi lại từ đầu.

MỤC LỤC

Bài 1. Trò chơi sắp số	7
Bài 2. Trò chơi dò mìn	25
Bài 3. Trò chơi xếp gạch	65
Bài 4. Trò chơi tìm cặp	149
Bài 5. Trò chơi tìm cặp hai	189
Bài 6. Trò chơi carô	245
Bài 7. Trò chơi xạ thủ	277
Bài 8. Trò chơi không chiến	305

Lập trình Game bằng Visual Basic

-VN Guide-

Chịu trách nhiệm xuất bản :
CÁT VÂN THÀNH

Biên tập : BAN BIÊN TẬP TỔNG HỢP
Trình bày : THIÊN TRUNG
Vẽ bìa : THIÊN ÂN
Sửa bản in : NGUYỄN THỦY

NHÀ XUẤT BẢN THỐNG KÊ
98 Thụy Khuê - Tây Hồ - Hà Nội
CN : 16 Nguyễn Huệ, Quận 1, TP. Hồ Chí Minh
ĐT : 8290047

Liên kết xuất bản :

CTY VĂN HÓA MINH TRÍ - NS. VĂN LANG
25 Nguyễn Thị Minh Khai, Q.1, TPHCM
ĐT : 8.242157 - 8233022 - Fax : 84.8.235079

In 1000 cuốn khổ 14.5x20.5cm tại Xưởng in CN Trung Tâm Hội
Chợ Triển Lãm Việt Nam. Giấy chấp nhận đăng ký KHXB số 205/
XB-QLXB Cục xuất bản cấp ngày 3.3.2003. Trích ngang kế hoạch
xuất bản số 118-205/XB-QLXB Nhà xuất bản Thống Kê cấp ngày
4.4.2003. In xong và nộp lưu chiểu quý 2 năm 2003.

LẬP TRÌNH GAME BẰNG

VISUAL BASIC

- Trò chơi sắp số đòi hỏi khả năng tính toán và sự sắp xếp khéo léo của người chơi.
- Trò chơi dò mìn khá đơn giản, chỉ cần người chơi chịu khó tính toán và một chút may mắn là thành công.
- Trò chơi xếp gạch có thể chơi bằng máy chơi game, trên điện thoại di động hoặc bằng máy tính cá nhân.
- Trò chơi tìm cặp là trò chơi lật ảnh để tìm cặp với số lần lật càng ít càng tốt.
- Trò chơi caro là một trong những trò chơi phổ biến mà hầu hết các bạn đều đã biết luật chơi.
- Trò chơi xạ thủ thực chất là trò chơi bắn bong bóng bay.
- Trò chơi không chiến là một biến dạng của trò chơi xạ thủ nhưng đã được cải tiến ở mức độ cao hơn.

 **Minh Trí**
25 Nguyễn Thị Minh Khai, Q.1, TP.HCM
ĐT: 8242157 - 8233022 - FAX: 8235079
9 Phan Đăng Lưu, Q.Đ.T, TP.HCM - ĐT: 8413306
Email: minhtricom@hcm.vnn.vn

It game bằng visual basic



Giá: 40.000đ