Objectives

Program Purpose

• Display single colours in a grid of shapes
• Display patterns of colours in a grid of shapes

Learning Goals

• Make and use a Control Array
• Use a fixed loop (FOR..NEXT) to manipulate properties of a control array
• Use the eight built-in Visual Basic Colour Constants
• Use simple Message Boxes and Message Box Constants
• Use the multiple branching structure: SELECT..CASE..END SELECT

Design Notes

This project is a simple introduction to the use of control arrays. A control array is a collection of objects of the same type, which have the same name but different index values. Using loops and variables, it very easy to manipulate the properties of a control array. Each element of the array is referenced by its unique ‘index value’.

In this example, we use the colour property of the shapes to demonstrate the ease of use of the control array. Visual Basic 6.0 has eight built-in colour constants: vbBlack, vbBlue, vbWhite, vbYellow, vbGreen, vbRed, vbCyan and vbMagenta. Note: Microsoft always use the American spelling of the word 'colour', i.e. 'color'.

When creating the squares, set the Font, BackStyle and BackColor for the first one, then copy and paste it. When copying and pasting, the application will ask the question: “Do you want to create a control array?” In this case, answer “Yes”.

The captions on the command buttons give clues as to the names of the objects. Always set the name properties before entering any code.

Interface

Create the interface as shown.
Use 1 form, 1 control array of 9 shapes and 6 command buttons.
Names and Initial Properties of Objects

<table>
<thead>
<tr>
<th>Type of Object</th>
<th>Number</th>
<th>Names of Objects</th>
<th>Simple Initial Properties of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>1</td>
<td>Form1</td>
<td>Caption - &quot;Control Arrays 1&quot;&lt;br&gt;Icon - Browse and choose any icon&lt;br&gt;Startup Position - Centre Screen</td>
</tr>
<tr>
<td>Control Array of Shapes</td>
<td>9</td>
<td>shpCell(0) to shpCell(8)</td>
<td>BackStyle - Opaque&lt;br&gt;BackColor - White</td>
</tr>
<tr>
<td>Command Buttons</td>
<td>6</td>
<td>cmdWhite, cmdBlack, cmdBlue, cmdRed, cmdPattern1, cmdPattern2</td>
<td>Font - Bold, 12&lt;br&gt;Captions - as per interface diagram</td>
</tr>
</tbody>
</table>

**Code**

**GENERAL SECTION**

Const NumButtons = 9<br>Const MsgColour = "The colour has been changed to "<br>Const MsgQuestion = "Do you really want to change the colour to "

Dim response As Integer, i As Integer

**EVENTS**

Private Sub cmdBlack_Click()

‘set all to black
response = MsgBox(MsgQuestion & "Black", vbYesNoCancel + vbQuestion, "Change Colour?")

If response = vbYes Then
   For i = 0 To NumButtons - 1
      shpCell(i).BackColor = vbBlack
   Next i
   MsgBox MsgColour & "Black", vbOKOnly + vbExclamation, "Colour Changed 
Else
   For i = 0 To NumButtons - 1
      shpCell(i).BackColor = vbBlue
   Next i
End If
End Sub

Private Sub cmdBlue_Click()

‘set all to blue
For i = 0 To NumButtons - 1
   shpCell(i).BackColor = vbBlue
Next i
End Sub
Private Sub cmdPattern1_Click()
'alternate green and cyan
For i = 0 To NumButtons - 1 Step 2
    shpCell(i).BackColor = vbGreen
    If i + 1 <= NumButtons - 1 Then
        shpCell(i + 1).BackColor = vbCyan
    End If
Next i
End Sub

Private Sub cmdPattern2_Click()
'diagonal of magenta
For i = 0 To NumButtons - 1
    Select Case i
        Case 2, 4, 6
            shpCell(i).BackColor = vbMagenta
        Case Else
            shpCell(i).BackColor = vbYellow
    End Select
Next i
End Sub

Private Sub cmdRed_Click()
'set all to red
For i = 0 To NumButtons - 1
    shpCell(i).BackColor = vbRed
Next i
End Sub

Private Sub cmdWhite_Click()
'set all to white
For i = 0 To NumButtons - 1
    shpCell(i).BackColor = vbWhite
Next i
End Sub

Consolidation and Extension

1. Which three lines of code set all the shapes to red?
2. Which two lines of code decide whether or not to set the colour of all the shapes to black?
3. The built-in Visual Basic constants 'vbYesNoCancel' and 'vbQuestion' determine the appearance of the MsgBox Form. Name three other possible combinations of constants that could be used.
   Hint: Start typing MsgBox "Prompt" in your project and the options will appear in a dropdown box.
4. What is the essential difference between the following two lines of code?
   LINE 1:  response = MsgBox(MsgQuestion & "Black", vbYesNoCancel + vbQuestion, "Change Colour?")
   LINE 2:  MsgBox MsgColour & "Black", vbOKOnly + vbExclamation, "Colour Changed"
5. We declare the Constant 'Numbuttons' as equal to 9, i.e.
   Const Numbutton = 9
   Why then do we then loop from 0 to 8 in the program when manipulating the shape colour properties? i.e.
   For i = 0 to NumButtons - 1 ....
6. Add two more command buttons to create the following patterns:
   a. Alternate blue and white.
   b. A pattern of red, green and yellow.
7. Extend the grid to 16 buttons. Modify the code to ensure that all the command buttons function correctly.