Objectives

Program Purpose
- Display colours in a grid of shapes
- Display patterns of colours in a grid of shapes
- Randomly display a pattern of colours
- Mix and control the colours through three vertical scroll bars

Learning Goals
- To use a Control Array
- Use a FOR..NEXT loop to manipulate properties of a control array
- Use the VB Colour Constants, as well as user-defined colour constants
- Use a frame with option buttons
- Use the RGB function for colour control
- Use random numbers
- Use a parameter variable

Design Notes

This project follows on from Control Arrays 1. Remember that a control array is a collection of objects of the same type, which have the same name but different index values. Each element of the array is referenced by its unique 'index value'. 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”.

Set the name properties before entering any code. Labels that serve no purpose other than displaying text do not need particular names. For this reason, the labels in this project can be left with their default names (label1, label2 etc).

Interface

Create the interface as shown below.
Use 1 form, 1 control array of 20 shapes and a frame containing 8 option buttons, 4 labels, 3 vertical scroll bars and 1 command button.
Names & 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>
</table>
| Form                           | 1      | Form1                             | Caption - "Control Arrays 2"  
Icon - Browse and choose any icon  
Startup Position - Centre Screen                                                  |
| Control Array of Shapes        | 20     | Cell(0) to Cell(19)               | BackStyle - Opaque  
BackColor - Black  
Shape - 0 (Rectangle)                                                            |
| Frame                          | 1      | No name necessary                 | Caption - Colour                                                                  |
| Option Button                  | 8      | optBlack, optWhite, optGreen, optRed, optBlue, optYellow, optPurple, optOlive | Font - Bold, 12  
Captions - as per interface diagram                                                |
| Label                          | 4      | lblColor, label1, label2, label3  | LblColor: No caption; Borderstyle - Fixed;  
BackColor - White; BackStyle – Opaque.  
Other labels: Captions - Red, Green, Blue;  
Font - 12 Bold.                                                                   |
| Vertical Scroll Bar            | 3      | vsbRed, vsbGreen, vsbBlue         | Min - 1; Max - 255; LargeChange - 10.                                              |
| Command Button                 | 1      | cmdRandom                         | Font - Bold, 12  
Caption - &Random                                                                |

Code

**GENERAL SECTION**

```vbnet
Dim i As Integer  
Const NumButtons = 20

Sub Redraw(Color As Long)  
For i = 0 To NumButtons - 1  
   Cell(i).BackColor = Color  
Next i  
End Sub
```

**EVENTS**

```vbnet
Private Sub cmdRandom_Click()  
Randomize  
For i = 0 To NumButtons - 1  
   Cell(i).BackColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)  
Next i  
End Sub
```

```vbnet
Private Sub Form_Load()  
lblColor.BackColor = RGB(vsbRed, vsbGreen, vsbBlue)  
End Sub
```
EVENTS (continued)

Sub ChangeColors()
    lblColor.BackColor = RGB(vsbRed, vsbGreen, vsbBlue)
    Redraw lblColor.BackColor
End Sub

Private Sub lblColor_Click()
    Redraw lblColor.BackColor
End Sub

Private Sub optBlack_Click()
    Redraw vbBlack
End Sub

Private Sub optBlue_Click()
    Redraw vbBlue
End Sub

Private Sub optGreen_Click()
    Redraw vbGreen
End Sub

Private Sub optOlive_Click()
    Redraw &HC0C0&
End Sub

Private Sub optPurple_Click()
    Redraw &H800080
End Sub

Private Sub optRed_Click()
    Redraw vbRed
End Sub

Private Sub optWhite_Click()
    Redraw vbWhite
End Sub

Private Sub optYellow_Click()
    Redraw vbYellow
End Sub

Private Sub vsbBlue_Change()
    ChangeColors
End Sub

Private Sub vsbGreen_Change()
    ChangeColors
End Sub

Private Sub vsbRed_Change()
    ChangeColors
End Sub
Consolidation and Extension

1. Which line of code sets the BackColor property of the label lblColor? Which two events does this occur in?

2. Strictly speaking, ChangeColors is not an event. It is a procedure called from an event. What other procedures are used in this program?

3. What is the essential difference between the two procedures, Redraw and ChangeColors?

4. Which line of code declares the parameter variable in this program?

5. This program uses a control array for the pattern of shapes. Which other controls in this project could use a control array?

6. Try recreating the vertical scroll bars as a control array. Does it simplify the code?

7. Add option buttons for three more colours - Pink, Brown and Grey. Where do you get the values for these colours and what are they?

8. What is the difference between the SmallChange and the LargeChange Property? Try changing these values to 5 and 25 respectively.

9. The RGB function returns a colour based on the mixture of the 3 primary colours, Red, Green and Blue. What are the common colours represented by the following settings for the RGB function? (Hint: use the Vertical Scroll Bars to test).

   \[
   \begin{align*}
   &\text{RGB}(0, 0, 0) \\
   &\text{RGB}(255, 255, 255) \\
   &\text{RGB}(255, 255, 0) \\
   &\text{RGB}(0, 255, 255) \\
   &\text{RGB}(255, 0, 255)
   \end{align*}
   \]

10. Modify the project so that there are 40 shapes in the pattern. Modify the code in only one place to make the new shapes function.

11. Modify the array so that the shapes are no longer rectangles (e.g. change to circles or ovals etc).

12. Modify the BorderStyle and width properties of the shapes. What effect does this have?

13. What is the difference between FillColor and BackColor?

14. Comment out the Randomize statement in the CmdRandom_Click event.
   a. What effect does this have?
   b. Test and prove this by restarting the program five times and recording the first five colour values randomly generated (use MsgBox Cell(i).BackColor in the cmdRandom_Click event to capture the values).

15. There is a design flaw in this program. If an option button for a preset colour is clicked on (e.g. Red), the colour of the label, lblColor, doesn’t change. In addition, the vertical scroll bars for RGB do not reflect the colour actually displayed in the pattern. How could this be fixed?