Case Study: The Shell Game

The Shell Game displays pictures of three shells. Under one shell is a “hidden” pearl. The user guesses which shell is hiding the pearl by clicking a shell. The hidden pearl is then displayed along with a message telling the player if a correct guess was made. The pearl is hidden again after each try so that the game can be played again and again. An algorithm for implementing this kind of guessing game is:

1. Generate a random number between 1 and 3. Use this number to determine which shell is “hiding” the pearl.
2. Show a pearl picture below the shell that corresponds to the random number.
3. Using one procedure that handles click events for all three shells, determine if the user clicked the shell that corresponds to the random number.
4. Display a message to the player. The player won if the shell clicked corresponds to the generated random number.
5. Make the pearl picture no longer visible so that the game can be played again without quitting and running the application again.

**CREATE A NEW PROJECT:**

Create a Windows application named Shell Game.

**CREATE THE INTERFACE:**

Using the illustration and table below set the objects and their properties.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Object** | **Name** | **Text** | **SizeMode** | **Visible** | **Tag** | **Size** |
| Form1 |  | Shell Game |  |  |  |  |
| PictureBox 1 | picShell 1 |  | AutoSize | True | 1 | 110, 110 |
| PictureBox 2 | picPearl 1 |  | Stretch Image | False | 1 | 32, 32 |
| PictureBox 3 | picShell 2 |  | AutoSize | True | 2 | 110, 110 |
| PictureBox 4 | picPearl 2 |  | Stretch Image | False | 2 | 32, 32 |
| PictureBox 5 | picShell 3 |  | AutoSize | True | 3 | 110, 110 |
| PictureBox 6 | picPearl 3 |  | Stretch Image | False | 3 | 32, 32 |

**IMPORT IMAGE RESOURCES AND ADD TO INTERFACE**

1. Select the picShell1 picture box.
2. In the Properties window, select the Image property. The little box with three little dots in it, button is displayed.
3. Click the little box with the dots, The Select Resources dialog box is displayed.
   1. Select Project resource file and then select Import. The Open dialog box is displayed.
   2. Navigate to the location of the shell.gif file, a data file for this text.
   3. Import the pearl.gif file, a data file for this text.
   4. In the files list, select Shell to display the shell image in the picture box.
   5. Select OK. The dialog box is removed.
4. Display the shell image in the picShell2 and picShell3 picture boxes.
5. Display the pearl image in the picPearl1, picPearl2, and picPearl3 picture boxes.

**WRITE THE APPLICATION CODE**

1. Display the Code window.
2. Add comments that include your name, assignment, and today’s date.
3. Create a picShell1\_Cleick event procedure.

Public Class Form1

'Determines which sehll was clicked and displays a message if shell clicked is the sameas a randomly chosen shell.

'pre: Shell picture objects have valid Tag properties

'post: The hidden pearl has been shown and a message box has been displayed.

Private Sub picShell1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles picShell1.Click, picShell2.Click, picShell3.Click

Dim picShellClicked As PictureBox = sender

Dim shellClicked As Integer = Val(picShellClicked.Tag)

Randomize()

Dim shellWithPearl As Integer = Int(3 \* Rnd()) + 1

Select Case shellWithPearl

Case 1

Me.picPearl1.Visible = True

Case 2

Me.picPearl2.Visible = True

Case 3

Me.picPearl3.Visible = True

End Select

If shellClicked = shellWithPearl Then

MessageBox.Show("Sorry, you lose.")

End If

Select Case shellWithPearl

Case 1

Me.picPearl1.Visible = False

Case 2

Me.picPearl2.Visible = False

Case 3

Me.picPearl3.Visible = False

End Select

End Sub

End Class

Run the application.