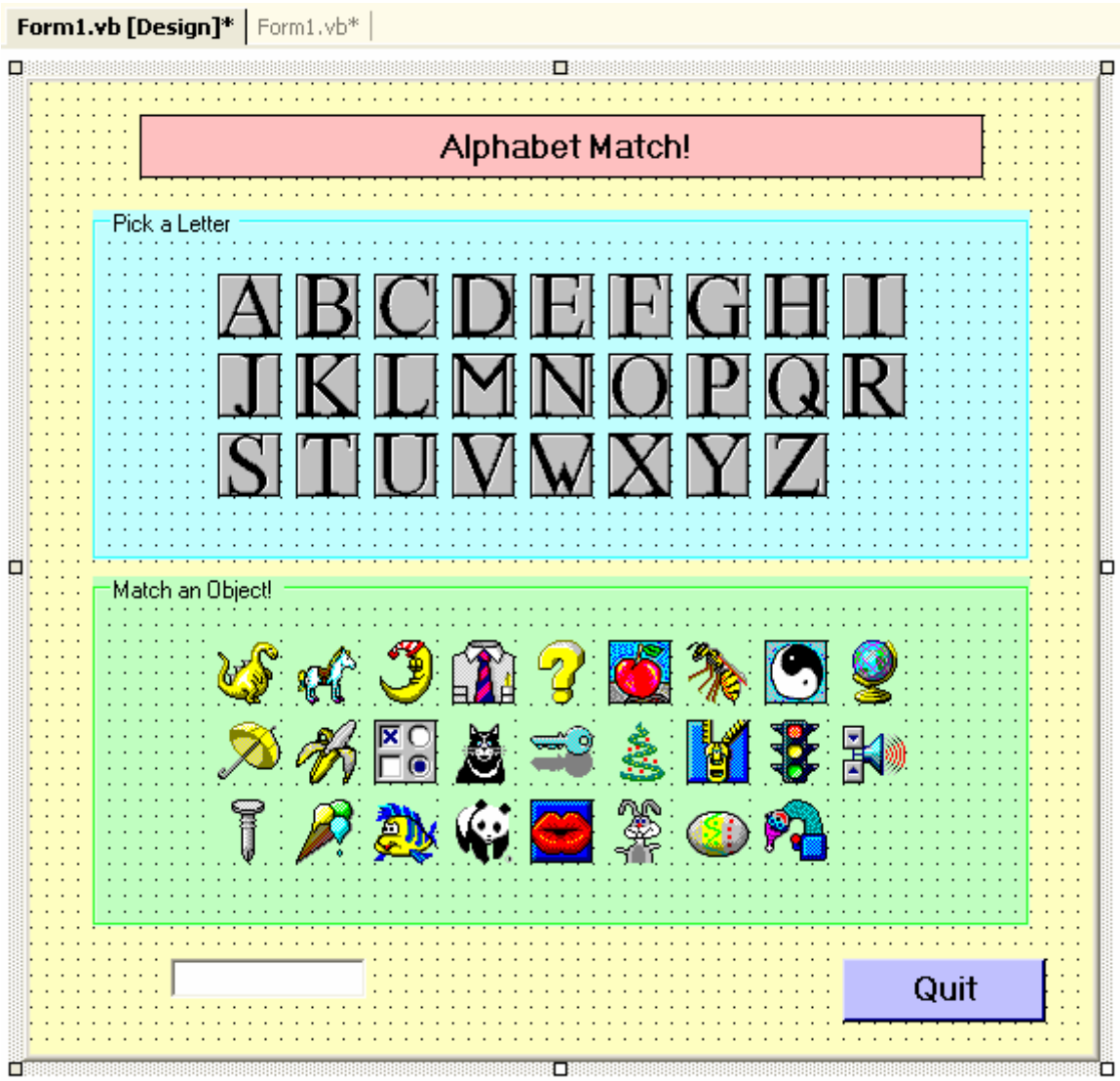
Early Learning Tool

Problem Description: Design a program that can be used by a toddler (aged 2-4 years). To provide the greatest appeal, it should be easy to use, colourful and provide instant feedback. In addition to this, the program should also be educational. You may choose to base the program around the alphabet, counting, colours, shapes, animals or any early learning theme you wish, as long as what is required is not too complex.

Skills Covered: (Novice) Form properties, grouping objects, Tag property, variable declarations.

This problem can be approached in a variety of different ways. What follows is a suggestion for one program that focuses on the letters of the alphabet. You could create this program and then add other learning elements using a similar coding style if you wish.



The program asks the user to click on a letter and then to try to pick the object which begins with the same letter. If an incorrect choice is made, the program allows the user to pick again. When a correct match is made, a congratulatory message is displayed.

Setting up the objects for our program

Follow these steps to create the program.

1. Start a new Visual Basic.NET program called 'EarlyLearn'.

2. Select the Label tool from the toolbox. Use it to draw a rectangular box in the center and at the top of the form. This will be our main heading. Name it 'lblTitle'.

3. Use the Button tool to draw a button in the bottom right hand corner of the form. Name the button 'btnQuit'. This button will be used to end the program.

4. Use the GroupBox tool to draw two large group boxes on the form. These will be used to hold all of the letters of the alphabet and the images of the objects which the user will be trying to match. Make sure the group boxes are large enough so that we can fit in all of the objects that we need to. Name the first group box 'grpLetters' and the second 'grpObjects'.

5. Use the PictureBox tool to create a single rectangular box in the first group box called 'grpLetters'. Even though there are a large number of picture boxes that we are going to use, we are only going to create one to start with. By doing this, we can set it up and copy it as many times as we need to, and save ourselves some time. Set the SizeMode property of the picture box to AutoSize. Click on the Image property of the picture box and then browse to the 'Icons' folder in the same folder as this document. Locate the file named 'A.ico' and select this. The letter A should now be displayed on the picture box. Copy and paste the picture box to make up all the letters of the alphabet, and locate each letter image in the 'Icons' folder. Rename all of the picture boxes appropriately. For example, 'picA', 'picB', etc.

6. Use the PictureBox tool to create another rectangular box, but this time in the second group box called 'grpObjects'. As in the previous step, set the SizeMode property of the picture box to AutoSize. Click on the Image property of the picture box and then browse to the 'Icons' folder in the same folder as this document. Locate the file named 'Apple.ico' and select this. An apple should now be displayed on the picture box. Copy and paste the picture box 26 times. Clicking on the picture boxes in a random fashion, select all of them in turn and set their Image property to be one of the object image files located in the 'Icons' folder. There is one object image per letter of the alphabet. Rename all of the picture boxes appropriately. For example, 'picObjA', 'picObjB', etc.

7. Use the TextBox tool to add a small text box at the bottom of the form. Name it 'txtSelectedLetter'. This text box will be used to hold the letter that has been clicked on by the user.

Changing the properties of the objects

Click on each object in turn and set the following properties:

Form1

StartPosition: CenterScreen
Text: *Blank*
MaximizeBox: False
MinimizeBox: False
ControlBox: False

By setting the form's properties in this way, we can create a borderless form that appears on the screen that cannot be moved, closed or resized by the user. Given that this program will be used by small children, doing this ensures that the program is not ended prematurely. This method can also be used to create forms of a different shape. You could design a background for the form in a paint program and place this in the BackGroundImage property. If the transparent colour of the form is set to the colour that is surrounding your background, the form will appear to have a different shape.

lblTitle

Text: Alphabet Match!
TextAlign: MiddleCenter
BorderStyle: FixedSingle
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

btnQuit

Text: Quit
TextAlign: MiddleCenter
Set the Font and BackColor properties yourself, so that the button is colourful and easy to read.

grpLetters

Text: Pick a Letter
BackColor: A colour of your choosing.

grpObjects

Text: Match an Object!
BackColor: A colour of your choosing, different from the colour chosen for grpLetters.

Placing objects inside a group box can be very beneficial. During design time, all of the objects inside the group box can be moved around simply by moving the group box on the form. During the execution of a program, if the group box is made invisible, all the objects within the group box are also invisible. Likewise, if the group box is disabled, all the objects inside the group box will act as if they are also disabled. This can save a great deal of time.

As you will also see when we begin coding our program, it is very easy to access all of the objects within a group box and change their properties via programming code.

picA – picZ

Image: The revelant image file from the 'Icons' folder.
SizeMode: AutoSize
Tag: The respective letter of the alphabet. For example 'A', 'B', etc.

The Tag property provides a method by which information can be stored in the object for future reference. In this case, by storing the corresponding letter in the Tag property of the picture box, we can easily refer to the property in code to work out which letter has been clicked on by the user.

picObjA – picObjZ

Image: The revelant image file from the 'Icons' folder.
SizeMode: AutoSize

txtSelectedLetter

Text: *Blank*.
Visible: False

Adding code to our program

1. When the user runs the program, they will first click on one of the letters and then try to match this with one of the objects. In order to keep track of the letter that the user has clicked on, we need to declare a variable at the top of the form. This variable will be of type Object, which means we can use it to store any type of object that we choose. Move to the top of the form, and just underneath the 'Inherits' statement, type the following variable declaration:

```
Dim Letter As Object
```

2. From the Class Name drop down box, select the picture box called 'picA'. This is the picture box in 'grpLetters' which has the letter 'A' on it. When the user clicks on this picture box, we want all of the other letters to disappear and this letter to become larger. In addition to this, we want to place the letter that is stored in the Tag property into the text box called 'txtSelectedLetter'. To ensure that we can refer to this object later on, we will store it in the variable 'Letter'. Type in the following code which performs these functions:

```
Private Sub picA_Click(ByVal sender As System.Object, ByVal e_
      As System.EventArgs) Handles picA.Click
      If txtSelectedLetter.Text = "" Then
            picA.Visible = True
            picB.Visible = False
            picC.Visible = False
            picD.Visible = False
```

```
            picE.Visible = False
            picF.Visible = False
            picG.Visible = False
            picH.Visible = False
            picI.Visible = False
            picJ.Visible = False
            picK.Visible = False
            picL.Visible = False
            picM.Visible = False
            picN.Visible = False
            picO.Visible = False
            picP.Visible = False
            picQ.Visible = False
            picR.Visible = False
            picS.Visible = False
            picT.Visible = False
            picU.Visible = False
            picV.Visible = False
            picW.Visible = False
            picX.Visible = False
            picY.Visible = False
            picZ.Visible = False
            picA.SizeMode = PictureBoxSizeMode.StretchImage
            picA.Height = 64
            picA.Width = 64
            picA.Left = picA.Left - 16
            picA.Top = picA.Top - 16
            txtSelectedLetter.Text = picA.Tag
            Letter = picA
      End If
End Sub
```

3. This code will set up the letter ‘A’ so that it works correctly, but all of the other letters also need to be set up in a similar fashion. Copy and paste this code into the Click event of the other letters, changing the relevant parts of the code.

4. We now need to code the corresponding objects so that when they are clicked, we make all of the others invisible and check to see whether the user has selected the correct pair. If the user has done this, a congratulatory message should be displayed. All of the letters and objects should then be reset back to their starting positions. Select ‘picObjA’ from the Class Name drop down box and type in the following code:

```
Private Sub picObjA_Click(ByVal sender As System.Object, ByVal _
      e As System.EventArgs) Handles picObjA.Click
      If txtSelectedLetter.Text <> "" Then
            picObjA.Visible = True
            picObjB.Visible = False
            picObjC.Visible = False
            picObjD.Visible = False
            picObjE.Visible = False
            picObjF.Visible = False
            picObjG.Visible = False
            picObjH.Visible = False
            picObjI.Visible = False
            picObjJ.Visible = False
            picObjK.Visible = False
            picObjL.Visible = False
            picObjM.Visible = False
            picObjN.Visible = False
            picObjO.Visible = False
```

```
picObjP.Visible = False
picObjQ.Visible = False
picObjR.Visible = False
picObjS.Visible = False
picObjT.Visible = False
picObjU.Visible = False
picObjV.Visible = False
picObjW.Visible = False
picObjX.Visible = False
picObjY.Visible = False
picObjZ.Visible = False
picObjA.SizeMode = PictureBoxSizeMode.StretchImage
picObjA.Height = 64
picObjA.Width = 64
picObjA.Left = picObjA.Left - 16
picObjA.Top = picObjA.Top - 16
If txtSelectedLetter.Text = _
      Microsoft.VisualBasic.Right(picObjA.Name, 1) Then
      MsgBox("Correct!!", MsgBoxStyle.OKOnly + _
            MsgBoxStyle.Exclamation)
Else
      MsgBox("Bad luck! Try again!", MsgBoxStyle.OKOnly _
            + MsgBoxStyle.Exclamation)
End If
picObjA.SizeMode = PictureBoxSizeMode.AutoSize
picObjA.Left = picObjA.Left + 16
picObjA.Top = picObjA.Top + 16
picObjA.Visible = True
picObjB.Visible = True
picObjC.Visible = True
picObjD.Visible = True
picObjE.Visible = True
picObjF.Visible = True
picObjG.Visible = True
picObjH.Visible = True
picObjI.Visible = True
picObjJ.Visible = True
picObjK.Visible = True
picObjL.Visible = True
picObjM.Visible = True
picObjN.Visible = True
picObjO.Visible = True
picObjP.Visible = True
picObjQ.Visible = True
picObjR.Visible = True
picObjS.Visible = True
picObjT.Visible = True
picObjU.Visible = True
picObjV.Visible = True
picObjW.Visible = True
picObjX.Visible = True
picObjY.Visible = True
picObjZ.Visible = True
picA.Visible = True
picB.Visible = True
picC.Visible = True
picD.Visible = True
picE.Visible = True
picF.Visible = True
picG.Visible = True
picH.Visible = True
picI.Visible = True
picJ.Visible = True
picK.Visible = True
```

```
            picL.Visible = True
            picM.Visible = True
            picN.Visible = True
            picO.Visible = True
            picP.Visible = True
            picQ.Visible = True
            picR.Visible = True
            picS.Visible = True
            picT.Visible = True
            picU.Visible = True
            picV.Visible = True
            picW.Visible = True
            picX.Visible = True
            picY.Visible = True
            picZ.Visible = True
            Letter.Height = 32
            Letter.Width = 32
            Letter.Left = Letter.Left + 16
            Letter.Top = Letter.Top + 16
            txtSelectedLetter.Text = ""
        End If
    End Sub
```

5. This code will set up the object ‘A’ so that it works correctly, but all of the other objects also need to be set up in a similar fashion. Copy and paste this code into the Click event of the other objects, changing the relevant parts of the code.

6. Type the following code into the Click event of the button ‘btnQuit’, which will end the program:

```
Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e _
   As System.EventArgs) Handles btnQuit.Click
   End
End Sub
```

Testing the program

Run and test the program and ensure that it behaves as expected. Test that it is possible to match two picture boxes and receive the congratulatory message. Test that the alternative message is displayed when the two picture boxes do not match. Also check to see that all of the picture boxes are reset once the selections have been made.

Further ideas to develop

- Incorporate sound effects and animations to make the game more interesting.
- Create a program which uses the same principles based around the numbers 1-20, basic shapes, farm animals or a related theme.
- As each pair is selected, make them invisible so that the user can see what pairs are still remaining. When all the pairs have been selected, display a congratulatory message and give the user the option of resetting the program so that they can start again.