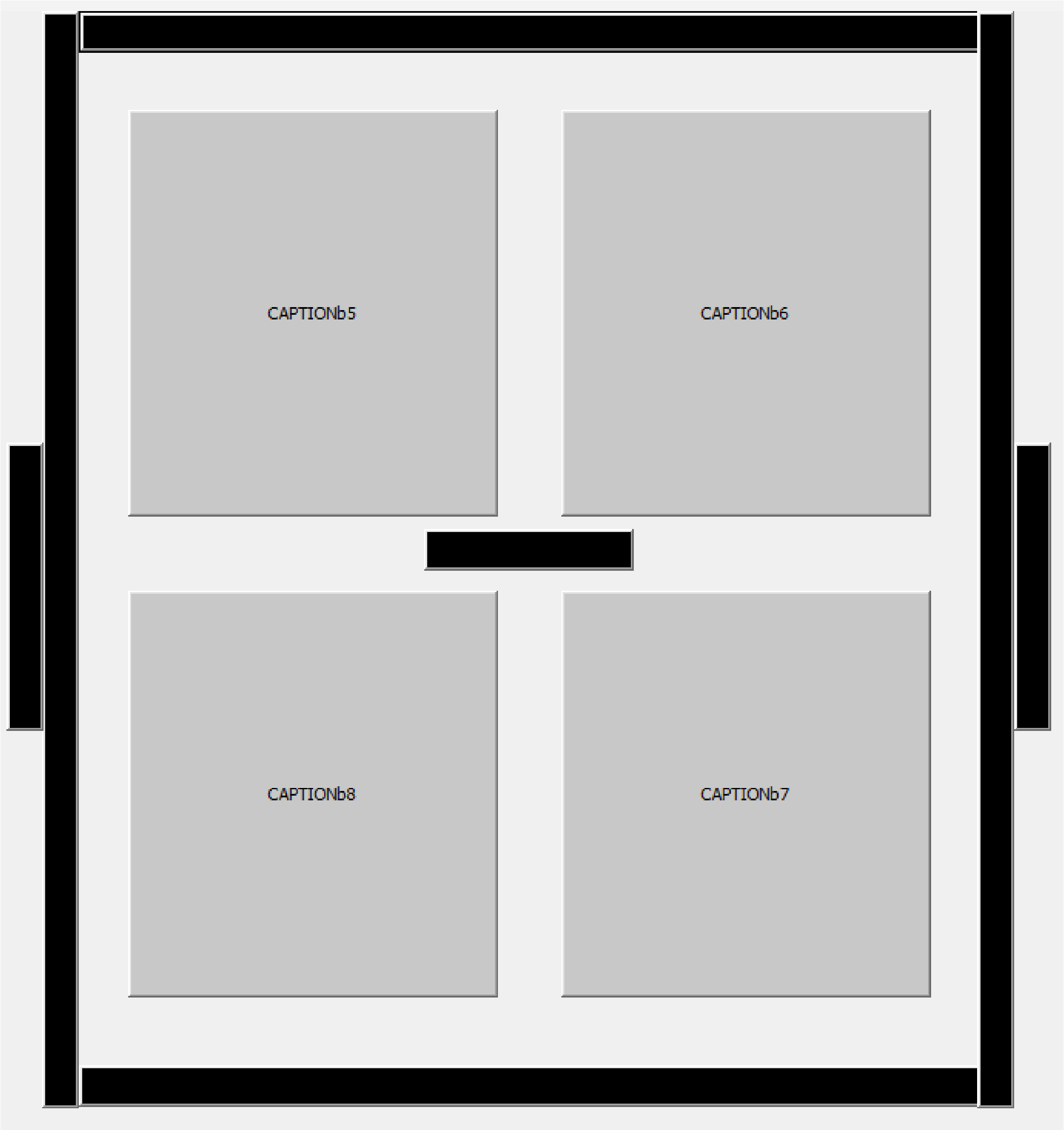
VBA SCRIPT FOR ADDING COMMAND BUTTONS TO A USERFORM LIKE POSITIONING BRICKS ON A 350 SQUARE METERS LAND TO BUILD A HOUSE

By
Hossou Zounffa

2/24/2024

Output Generated by the Welder Subroutine



INTRODUCTION

Just as bricks are laid one at a time to erect walls and define the house's layout, command buttons are positioned and customized individually to assemble the UserForm, making it functional and tailored to create a digital welded sheet metal door.

Making the Bricks

The AddCommandButton subroutine functions as the primary method for generating "bricks," where each "brick" symbolizes a command button on the UserForm. This subroutine defines the properties of these command buttons, including their name, caption, positional coordinates (left and top), dimensions (width and height), and optionally, their height and back color. Through specifying these attributes, the subroutine facilitates the precise and customized construction of the UserForm's interface, ensuring each button is positioned and tailored to fulfill its designated role within the form's overall architecture.

```
Sub AddCommandButton(USERFORMb As Object, ByVal Name As String, ByVal Caption As String, _
ByVal Left As Single, ByVal Top As Single, ByVal Width As Single, _
Optional ByVal Height As Single = 20, Optional ByVal BackColor As Long)

Dim CommandButtonb As Object

Set CommandButtonb = USERFORMb.Designer.Controls.Add("Forms.CommandButton.1")

CommandButtonb.Name = Name
CommandButtonb.Caption = Caption
CommandButtonb.Left = Left
CommandButtonb.Top = Top
CommandButtonb.Width = Width
CommandButtonb.Height = Height

If Not IsMissing(BackColor) Then CommandButtonb.BackColor = BackColor

End Sub
```

Construction Process

- **Adding Button # 1**

We introduce the initial command button. Positioned at the top left corner, this button marks the beginning of our form's interface layout, serving as a cornerstone upon which additional elements will be methodically added. This strategic placement sets the precedent for the organization and design of the entire form.

```
Sub TheWelder()
Const Leftb As Integer = 78
Const WIDTHb As Integer = 455
Const Topb As Integer = 0
Const SPACINGb As Integer = 30

Dim USERFORMb As Object
Set USERFORMb = ThisWorkbook.VBProject.VBComponents.Add(3) ' 3 denotes a UserForm

USERFORMb.Properties("Width") = 622
USERFORMb.Properties("Height") = 623

AddCommandButton USERFORMb, "NAMEb1", "CAPTIONb1", Leftb, Topb, WIDTHb
AddCommandButton USERFORMb, "NAMEb2", "CAPTIONb2", Leftb, Topb + 513, WIDTHb

VBA.UserForms.Add(USERFORMb.name).Show

End Sub
```

- **Adding Button # 2**

We proceed to position the second button. This second button is placed further down on the UserForm.

```
Sub TheWelder ()
Const Leftb As Integer = 78
Const WIDTHb As Integer = 455
Const Topb As Integer = 0
Const SPACINGb As Integer = 30

Dim USERFORMb As Object
Set USERFORMb = ThisWorkbook.VBProject.VBComponents.Add(3) ' 3 denotes a UserForm

USERFORMb.Properties("Width") = 622
USERFORMb.Properties("Height") = 623

AddCommandButton USERFORMb, "NAMEb1", "CAPTIONb1", Leftb, Topb, WIDTHb
AddCommandButton USERFORMb, "NAMEb2", "CAPTIONb2", Leftb, Topb + 513, WIDTHb
VBA.UserForms.Add(USERFORMb.name).Show

End Sub
```

- **Adding Button # 3**

We continue our UserForm's construction, echoing the methodical process of laying the third brick in our structural analogy.

```
Sub TheWelder ()
Const Leftb As Integer = 78
Const WIDTHb As Integer = 455
Const Topb As Integer = 0
Const SPACINGb As Integer = 30

Dim USERFORMb As Object
Set USERFORMb = ThisWorkbook.VBProject.VBComponents.Add(3) ' 3 denotes a UserForm

USERFORMb.Properties("Width") = 622
USERFORMb.Properties("Height") = 623

AddCommandButton USERFORMb, "NAMEb1", "CAPTIONb1", Leftb, Topb, WIDTHb
AddCommandButton USERFORMb, "NAMEb2", "CAPTIONb2", Leftb, Topb + 513, WIDTHb
AddCommandButton USERFORMb, "NAMEb3", "CAPTIONb3", Leftb - 18, Topb, WIDTHb - 437, 534

VBA.UserForms.Add(USERFORMb.name).Show

End Sub
```

- **Adding Button # 4 and others up to buttons # 11**

We place the fourth button, extending the analogy of laying bricks to build a structure. We will proceed in this manner, adding buttons up to button #11, each tailored and placed to enhance the UserForm's interactivity and user experience.

```
Sub TheWelder ()
Const Leftb As Integer = 78
Const WIDTHb As Integer = 455
Const Topb As Integer = 0
Const SPACINGb As Integer = 30

Dim USERFORMb As Object
Set USERFORMb = ThisWorkbook.VBProject.VBComponents.Add(3) ' 3 denotes a UserForm
USERFORMb.Properties("Width") = 622
USERFORMb.Properties("Height") = 623
AddCommandButton USERFORMb, "NAMEb1", "CAPTIONb1", Leftb, Topb, WIDTHb
AddCommandButton USERFORMb, "NAMEb2", "CAPTIONb2", Leftb, Topb + 513, WIDTHb
AddCommandButton USERFORMb, "NAMEb3", "CAPTIONb3", Leftb - 18, Topb, WIDTHb - 437, 534
AddCommandButton USERFORMb, "NAMEb4", "CAPTIONb4", Leftb - 18 + WIDTHb, Topb,
WIDTHb - 437, 534
AddCommandButton USERFORMb, "NAMEb5", "CAPTIONb5", Leftb + 24, Topb + 48, WIDTHb -
275, 198, RGB(200, 200, 200)
AddCommandButton USERFORMb, "NAMEb6", "CAPTIONb6", Leftb + 235, Topb + 48, WIDTHb -
275, 198, RGB(200, 200, 200)
AddCommandButton USERFORMb, "NAMEb7", "CAPTIONb7", Leftb + 235, Topb + 282, WIDTHb -
275, 198, RGB(200, 200, 200)
AddCommandButton USERFORMb, "NAMEb8", "CAPTIONb8", Leftb + 24, Topb + 282, WIDTHb -
275, 198, RGB(200, 200, 200)
AddCommandButton USERFORMb, "NAMEb9", "CAPTIONb9", Leftb - 35, Topb + 210, WIDTHb -
437, 140
AddCommandButton USERFORMb, "NAMEb10", "CAPTIONb10", Leftb + 455, Topb + 210, WIDTHb
- 437, 140
AddCommandButton USERFORMb, "NAMEb11", "CAPTIONb11", Leftb + 168, Topb + 252, WIDTHb
- 353

VBA.UserForms.Add(USERFORMb.name).Show

End Sub
```

CONCLUSION

In this article, the process of building a house brick by brick is used to illustrate the structured approach in VBA development for creating and customizing UserForms. This analogy emphasizes the importance of planning, where each command button represents a brick, contributing to the overall functionality and design of the UserForm.

It's now time fitting to showcase the final product. To this end, the following code snippet provides a   method for printing the outcome:

```
Private Sub CommandButton1_Click()
   UserForm1.PrintForm
End Sub
```