

Nate Barnes
MBA 614
Final Project
4/20/2014

EXECUTIVE SUMMARY

I'm working with a friend of mine on a business idea. Our business is based on the idea that social media is simply a social engagement tool, or a method to increase or smoothen human social-interactions. On that premise, we're developing BracketCity. BracketCity is a simple method for people to share and complete brackets with their friends. But the brackets behind BracketCity are meant to go beyond what you typically see during March madness. Using the BracketCity app, you can create a bracket of *anything* you want to share with your friends. The idea is users can generate brackets which appeal uniquely to their friend-group, like which friend is most likely to be a millionaire or who's the best looking girl at your school, etc.

For this project, I have built a prototype, which will be used for testing the marketability and usability of our bracket business. BracketCity relies heavily on creative user-generated content and our main focus with this prototype is testing out several brackets, which we are putting extra effort into. Essentially, we're testing whether or not great content will get higher response rates and attract more users. Additionally, this prototype will be used as a minimum viable product, we'll ask for feedback from users on the format, layout and flow before programming a prototype online or in the IOS/Andriod environment.

OVERVIEW OF SOLUTION

The BracketCity excel file is meant to be distributed to multiple users as a method for collecting results from a large sample of potential users. This file is self-sufficient and does not require any additional external files to function properly. The basic layout of the "Bracket" is as follows:

SELECTIONS	
Round 1	
Round 1	
Round 1	
Round 1	
Round 1	
Round 1	
Round 1	
Round 1	
Round 2	
Round 2	
Round 2	
Round 2	
Round 3	
Round 3	
Round 3	
Round 3	

The macro-programming behind the bracket guides the user through 4 rounds of “matchups.” During each round the following steps were taken:

1. Drop-down menus are generated to sort through the matchups.
2. The user is prompted with message boxes.
3. The user selects winners for each matchup using the drop-downs.
4. Buttons are used to direct the program forward to the next round of matchups.
5. Results are recorded.

Each matchup’s winner is picked by the user via a drop-down (also known as a ComboBox). This ensures that the user cannot generate their own solution, but it’s restricted to those specified in the drop-down. As each round is decided upon by the user and finalized, the selections are recorded in a table on the right-hand side of the sheet. These selections are recorded in a unified manner so the user can email the selections to me for aggregation.

The resulting sheet should look as follows:

					Name:	nate@gmail.com	SELECTIONS	
No. 1 seed							Round 1	No. 1 seed
No. 16 seed	No. 1 seed						Round 1	No. 8 seed
		No. 8 seed					Round 1	No. 12 seed
No. 8 seed							Round 1	No. 13 seed
No. 9 seed	No. 8 seed						Round 1	No. 6 seed
			No. 8 seed				Round 1	No. 3 seed
No. 5 seed							Round 1	No. 7 seed
No. 12 seed	No. 12 seed						Round 1	No. 2 seed
		No. 12 seed					Round 2	No. 8 seed
No. 4 seed							Round 2	No. 12 seed
No. 13 seed	No. 13 seed						Round 2	No. 6 seed
							Round 2	No. 2 seed
No. 6 seed							Round 3	No. 8 seed
No. 11 seed	No. 6 seed						Round 3	No. 2 seed
No. 3 seed		No. 6 seed					Winner	No. 8 seed
No. 14 seed	No. 3 seed							
No. 7 seed								
No. 10 seed	No. 7 seed							
		No. 2 seed						
No. 2 seed								
No. 15 seed	No. 2 seed							

Upon completing the fourth and final round of the brackets, the user is prompted to email the results to nate.a.barnes@gmail.com for aggregation.

DETAILED SOLUTION

Upon opening the file, users will see 16 items, seeded from 1 to 16, in the typical “bracket” fashion. This mimics what users will see when we have developed our own prototype of a bracket, seeded from 1-16.

The “Fill Out Bracket” button on the customized “BracketCity” tab starts the process of completing a bracket. Once the “Fill Out Bracket” button is pushed, the following code is executed:

```

Public Sub Callback1(control As IRibbonControl)
    insertComboBoxes
    initializeValues
    showIntro
        MsgBox "OK, now use the drop-down boxes to select your winners for round #1. Once
        You've made all of your 1st round selections, continue to round 2 by pressing the
        'Continue to Round 2' button on the right"
    insertRound1Button
    insertAllResetButton
End Sub

```

This code runs several function on the backend and several visible to the user. First, we'll talk about the user prompts. The user is prompted by two consecutive pop-up boxes. The first is an InputBox (from the showIntro sub procedure), which asks the user to input a name and recording that name on the top of the "Bracket" sheet. The second is a message box which prompts the user to make first round selections by using the drop-down (combo-boxes). These prompts are used to help direct the user. Filling out a bracket is pretty straight forward, but directions are always helpful when dealing with an aggregated population.

Of course, on the back-end the program is running separate macros to alter the display. Also tied to the "Fill Out Bracket" button (in the code above) is the following 4-step process:

- Step 1** DropDown menus are created for each of the round 1 matchups (insertComboBoxes).
- Step 2** DropDown menus are initialized with the seeds from round 1 (initializeValues).
- Step 3** A Button is created for the user to continue to the next round (insertRound1Button).
- Step 4** Selections are recorded.

STEP 1: The drop-down menus must first be created before populated. There was some consideration of just leaving them pre-formed in the excel file before sending. It was decided to have them created in the startup code for several reasons 1.) It made it easier to debug as re-creating new drop-downs was written into the code and 2.) This makes it easy for the user to reset if they too mess something up.

The following code was used to insert the drop-downs into the brackets:

```

With Range("C3")
    Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name = "ComboBox1"
End With

```

Using the "With Range("C3")" object, the dropdowns would automatically adjust their size to fit each respective cell. This same code was used for each of the 8 initial dropdown or ComboBoxes.

STEP 2: After the drop-downs are created, they are populated. This is simply recording the values held in the first round of the bracket and putting them as options in the drop-downs. This is done with the following code:

```
Sub initializeValues()
  With Sheet1.Shapes("ComboBox1").ControlFormat
    .AddItem Sheets("Bracket").Cells(2, 2)
    .AddItem Sheets("Bracket").Cells(4, 2)
  End With
```

I wrote "initializeValues" as a separate sub procedure outside of the main "Callback1" trigger connected to the "Fill Out Bracket" button so as to be able to debug the function separately. This procedure uses the .AddItem method to add values to the list of the dropdown (ComboBox). And within this "initializeValues" sub, there were 8 different sections adding values to the 8 separate drop-downs.

STEP 3: The "Continue" buttons control the flow of the program and the bracket. They act as gate-keepers for each round. Once all selections have been made, the user has the option of hitting the "Continue to Round 2" button.

Name:				SELE
			Round 1	
			Round 1	
			Round 1	
			Round 1	
			Round 1	
			Round 1	
		Continue to Round 2	Round 1	
			Round 1	
			Round 1	
			Round 1	

Upon hitting this button, a similar code is triggered through the .OnAction method of the button. The user will cycle to the next round and a near-exact 3-step process. Below is the code used to cycle to round 2 from round 1:

```
Sub insertRound1Button()

  With Range("H7:H8")
    Worksheets("Bracket").Buttons.Add(.Left, .Top, .Width, .Height).Select
    Selection.Name = "round1Complete"
    Selection.OnAction = "round2"
  With Selection
```

```

.Characters.Text = "Continue to Round 2"
    With .Font
        .Name = "Arial"
        .Size = 12
    End With
End With
End With
End Sub

```

Step 4 Lastly, once the selections have been made and the button is pushed, the selections are recorded near the top right-hand side of the sheet.

SELECTIONS	
Round 1	No. 1 seed
Round 1	No. 9 seed
Round 1	No. 5 seed
Round 1	No. 4 seed
Round 1	No. 6 seed
Round 1	No. 3 seed
Round 1	No. 7 seed
Round 1	No. 2 seed

The table is used to collect the selections but selections are also inputted into the bracket. The following code is used to record the selections:

```

With Worksheets("Bracket").Shapes("ComboBox1").ControlFormat

    Worksheets("Bracket").Range("K2").Value = .List(.ListIndex)

End With

Worksheets("Bracket").Range("C3").Value = Worksheets("Bracket").Range("K2").Value

```

Along with this 4-Step process of 1.)Creating drop-downs, 2.)Populating drop-downs 3.)Creating buttons and 4.)Recording selections there are minor secondary elements to the code.

- Drop-downs from previous rounds are deleted.
- “Continue” buttons from previous rounds are deleted.

One additional step is unique to the first round, a button created for resetting the brackets (insertAllResetButton). This button remains in place throughout the program giving the user an option to reset the bracket completely and start from scratch.



Upon pressing the reset button, all selections will be deleted from the entire bracket and the user can start again from round 1. The following code was associated with the ResetAll button:

```
Sub resetAll()  
    ActiveSheet.DropDowns.Delete  
    ActiveSheet.Range("K2:K16").ClearContents  
    ActiveSheet.Range("J2:J16").ClearContents  
    ActiveSheet.Buttons.Delete  
    ActiveSheet.Range("C3:C31").ClearContents  
    ActiveSheet.Range("D3:D31").ClearContents  
    ActiveSheet.Range("E3:E31").ClearContents  
    ActiveSheet.Range("F17").ClearContents  
End Sub
```

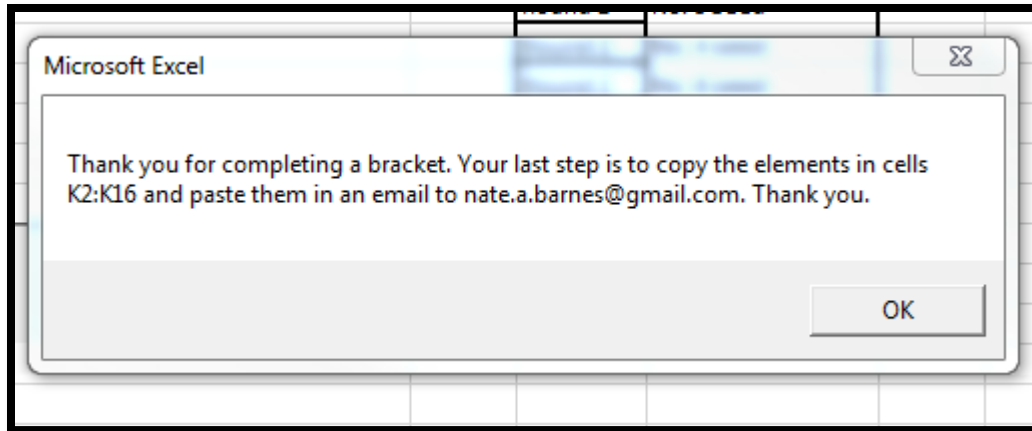
SUBSEQUENT ROUNDS

Each subsequent round follows a similar process. After the user has seen a prompt with directions, the code follows a similar 4-Step process:

- Step 1** DropDown menus are created for each of the round 1 matchups.
- Step 2** DropDown menus are initialized with the seeds from round 1.
- Step 3** A Button is created for the user to continue to the next round.
- Step 4** Selections are recorded.

All of these steps are very similar to the first round, with adjustments made for the positioning of the drop-downs, number of drop-downs, linkage of drop-downs, etc.

Finally, on the last round of the brackets, the user is prompted by a message box which is different from the previous rounds. After selecting the champion for the bracket and pressing the "Continue" button for the final round, the user is prompted with the following message box:



At this point, the bracket is complete and the selections have been organized neatly on the top sheet. All the user needs to do is email the results to be aggregated.

DISCUSSION & CONCLUSIONS

To be clear, my project was built completely from scratch. There was no template used in building this and no assistance was asked for or taken. Much of what I figured out came from the internet and searching for key terms associated with VBA. And much of the difficulty came from figuring out which key terms to search for.

Frankly, I thought the most difficult aspect to coding in VBA is figuring out the structure of the language. The conceptual build-out of the sheet was not that complicated and I had a good vision for what it was supposed to look like but things got messy when the code wasn't clean. As an example, I struggled with differentiating between the various types of ComboBoxes which could be used. I settled on a "form control" ComboBox because it seemed to be easier to manipulate at first, but in the end it was a pain. Apparently the .Value property of the ComboBoxes was not the selected value from the box but the index value. So, in order to return the selected values from the ComboBoxes, I had to do the following each time:

```
With Worksheets("Bracket").Shapes("ComboBox1").ControlFormat
    Worksheets("Bracket").Range("K2").Value = .List(.ListIndex)
End With
```

This was a pain to code and it took me hours to figure out what I needed to do to get the selected values. In hindsight, the "ActiveX" type was likely easier.

Another issue I constantly ran into came about when dealing with the various properties associated with each object. It seems that VBA gives a ton of control and leaves the user with many options to toggle when setting up the program. But all of that freedom makes it difficult to search for and find answers. The VBA world is an absolute ocean of knowledge online but when you don't know which part of the ocean you need to be in, it's very difficult to come up with the relevant answers.

The process of coding this project taught me a lot and I'm glad I finally completed it. Additionally, I think this prototype bracket along with subsequent others will help in identifying and understanding the

market pain we're going after with BracketCity. I feel confident that this is a good working prototype which I can send to others to record information and get feedback.

APPENDIX

Below is all of the code used for this project:

```
Public Sub Callback1(control As IRibbonControl)

    insertComboBoxes

    initializeValues

    showIntro

    MsgBox "OK, now use the drop-down boxes to select your winners for round #1.
    Once You've made all of your 1st round selections, continue to round 2 by pressing the
    'Continue to Round 2' button on the right"

    insertRound1Button

    insertAllResetButton

End Sub
Sub round2()

    ' Here we record the selected values from teh ComboBoxes

    With Worksheets("Bracket").Shapes("ComboBox1").ControlFormat
        Worksheets("Bracket").Range("K2").Value = .List(.ListIndex)
    End With
    With Worksheets("Bracket").Shapes("ComboBox2").ControlFormat
        Worksheets("Bracket").Range("K3").Value = .List(.ListIndex)
    End With
    With Worksheets("Bracket").Shapes("ComboBox3").ControlFormat
        Worksheets("Bracket").Range("K4").Value = .List(.ListIndex)
    End With
    With Worksheets("Bracket").Shapes("ComboBox4").ControlFormat
        Worksheets("Bracket").Range("K5").Value = .List(.ListIndex)
    End With
End Sub
```



```
End With
With Worksheets("Bracket").Shapes("ComboBox5").ControlFormat
    Worksheets("Bracket").Range("K6").Value = .List(.ListIndex)
End With
With Worksheets("Bracket").Shapes("ComboBox6").ControlFormat
    Worksheets("Bracket").Range("K7").Value = .List(.ListIndex)
End With
With Worksheets("Bracket").Shapes("ComboBox7").ControlFormat
    Worksheets("Bracket").Range("K8").Value = .List(.ListIndex)
End With
With Worksheets("Bracket").Shapes("ComboBox8").ControlFormat
    Worksheets("Bracket").Range("K9").Value = .List(.ListIndex)
End With
```

```
'Remove rnd1 comboboxes
```

```
ActiveSheet.DropDowns("ComboBox1").Delete
ActiveSheet.DropDowns("ComboBox2").Delete
ActiveSheet.DropDowns("ComboBox3").Delete
ActiveSheet.DropDowns("ComboBox4").Delete
ActiveSheet.DropDowns("ComboBox5").Delete
ActiveSheet.DropDowns("ComboBox6").Delete
ActiveSheet.DropDowns("ComboBox7").Delete
ActiveSheet.DropDowns("ComboBox8").Delete
```

```
'place values from k2:k9 where previous comboboxes were
```

```
Worksheets("Bracket").Range("C3").Value = Worksheets("Bracket").Range("K2").Value
Worksheets("Bracket").Range("C7").Value = Worksheets("Bracket").Range("K3").Value
Worksheets("Bracket").Range("C11").Value = Worksheets("Bracket").Range("K4").Value
Worksheets("Bracket").Range("C15").Value = Worksheets("Bracket").Range("K5").Value
Worksheets("Bracket").Range("C19").Value = Worksheets("Bracket").Range("K6").Value
Worksheets("Bracket").Range("C23").Value = Worksheets("Bracket").Range("K7").Value
Worksheets("Bracket").Range("C27").Value = Worksheets("Bracket").Range("K8").Value
Worksheets("Bracket").Range("C31").Value = Worksheets("Bracket").Range("K9").Value
```

```
'Delete continue to rnd 2 button
```

```
ActiveSheet.Buttons("round1Complete").Delete
```

```
insertComboBoxes2
initializeValues2
```

MsgBox "OK, now use the drop-down boxes to select your winners for round #2.
Once You've made all of your 2nd round selections, continue to round 3 by pressing the
'Continue to Round 3' button on the right"

```
insertRound2Button
```

```
End Sub
```

```
Sub round3()
```

```
' Here we record the selected values from teh ComboBoxes
With Worksheets("Bracket").Shapes("ComboBox9").ControlFormat
    Worksheets("Bracket").Range("K10").Value = .List(.ListIndex)
End With
With Worksheets("Bracket").Shapes("ComboBox10").ControlFormat
    Worksheets("Bracket").Range("K11").Value = .List(.ListIndex)
End With
With Worksheets("Bracket").Shapes("ComboBox11").ControlFormat
    Worksheets("Bracket").Range("K12").Value = .List(.ListIndex)
End With
With Worksheets("Bracket").Shapes("ComboBox12").ControlFormat
    Worksheets("Bracket").Range("K13").Value = .List(.ListIndex)
End With
```

```
'Remove rnd1 comboboxes
ActiveSheet.DropDowns("ComboBox9").Delete
ActiveSheet.DropDowns("ComboBox10").Delete
ActiveSheet.DropDowns("ComboBox11").Delete
ActiveSheet.DropDowns("ComboBox12").Delete
```

```
'place values from k10:k13 where previous comboboxes were
Worksheets("Bracket").Range("D5").Value = Worksheets("Bracket").Range("K10").Value
Worksheets("Bracket").Range("D13").Value = Worksheets("Bracket").Range("K11").Value
Worksheets("Bracket").Range("D21").Value = Worksheets("Bracket").Range("K12").Value
Worksheets("Bracket").Range("D29").Value = Worksheets("Bracket").Range("K13").Value
```

```
'Delete continue to rnd 2 button  
ActiveSheet.Buttons("round2Complete").Delete
```

```
insertComboBoxes3  
initializeValues3
```

```
MsgBox "OK, now use the drop-down boxes to select your winners for round #3.  
Once You've made all of your 3rd round selections, continue to round 4 by pressing the  
'Continue to Round 4' button on the right"
```

```
insertRound3Button
```

```
End Sub
```

```
Sub round4()
```

```
' Here we record the selected values from teh ComboBoxes  
With Worksheets("Bracket").Shapes("ComboBox13").ControlFormat  
    Worksheets("Bracket").Range("K14").Value = .List(.ListIndex)  
End With  
With Worksheets("Bracket").Shapes("ComboBox14").ControlFormat  
    Worksheets("Bracket").Range("K15").Value = .List(.ListIndex)  
End With
```

```
'Remove rnd1 comboboxes  
ActiveSheet.DropDowns("ComboBox13").Delete  
ActiveSheet.DropDowns("ComboBox14").Delete
```

```
'place values from k10:k13 where previous comboboxes were  
Worksheets("Bracket").Range("E9").Value = Worksheets("Bracket").Range("K14").Value  
Worksheets("Bracket").Range("E25").Value = Worksheets("Bracket").Range("K15").Value
```

```
'Delete continue to rnd 2 button  
ActiveSheet.Buttons("round3Complete").Delete
```

```
insertComboBoxes4  
initializeValues4
```

```
MsgBox "OK, now onto the final round. I hope you're ready for this."
```

```
insertSubmissionButton
```

```
End Sub
```

```
Sub insertAllResetButton()
```

```
With Range("H35:H36")
```

```
Worksheets("Bracket").Buttons.Add(.Left, .Top, .Width, .Height).Select
```

```
Selection.Name = "resetButton"
```

```
Selection.OnAction = "resetAll"
```

```
With Selection
```

```
.Characters.Text = "Reset All"
```

```
With .Font
```

```
.Name = "Arial"
```

```
.FontStyle = "Bold"
```

```
.Size = 14
```

```
End With
```

```
End With
```

```
End With
```

```
Sheet1.Range("H1").Select
```

```
End Sub
```

```
Sub resetAll()
```

```
ActiveSheet.DropDowns.Delete
```

```
ActiveSheet.Range("K2:K16").ClearContents
```

```
ActiveSheet.Range("J2:J16").ClearContents
```

```
ActiveSheet.Buttons.Delete
```

```
ActiveSheet.Range("C3:C31").ClearContents
```

```
ActiveSheet.Range("D3:D31").ClearContents
```

```
ActiveSheet.Range("E3:E31").ClearContents
```

```
ActiveSheet.Range("F17").ClearContents
```

```
End Sub
```

```
Sub insertRound1Button()
```

```
    With Range("H7:H8")
```

```
        Worksheets("Bracket").Buttons.Add(.Left, .Top, .Width, .Height).Select
```

```
        Selection.Name = "round1Complete"
```

```
        Selection.OnAction = "round2"
```

```
        With Selection
```

```
            .Characters.Text = "Continue to Round 2"
```

```
            With .Font
```

```
                .Name = "Arial"
```

```
                .Size = 12
```

```
            End With
```

```
        End With
```

```
    End With
```

```
End Sub
```

```
Sub initializeValues()
```

```
    With Sheet1.Shapes("ComboBox1").ControlFormat
```

```
        .AddItem Sheets("Bracket").Cells(2, 2)
```

```
        .AddItem Sheets("Bracket").Cells(4, 2)
```

```
    End With
```

```
    With Sheet1.Shapes("ComboBox2").ControlFormat
```

```
        .AddItem Sheets("Bracket").Cells(6, 2)
```

```
        .AddItem Sheets("Bracket").Cells(8, 2)
```

```
    End With
```

```
With Sheet1.Shapes("ComboBox3").ControlFormat
    .AddItem Sheets("Bracket").Cells(10, 2)
    .AddItem Sheets("Bracket").Cells(12, 2)
End With
```

```
With Sheet1.Shapes("ComboBox4").ControlFormat
    .AddItem Sheets("Bracket").Cells(14, 2)
    .AddItem Sheets("Bracket").Cells(16, 2)
End With
```

```
With Sheet1.Shapes("ComboBox5").ControlFormat
    .AddItem Sheets("Bracket").Cells(18, 2)
    .AddItem Sheets("Bracket").Cells(20, 2)
End With
```

```
With Sheet1.Shapes("ComboBox6").ControlFormat
    .AddItem Sheets("Bracket").Cells(22, 2)
    .AddItem Sheets("Bracket").Cells(24, 2)
End With
```

```
With Sheet1.Shapes("ComboBox7").ControlFormat
    .AddItem Sheets("Bracket").Cells(26, 2)
    .AddItem Sheets("Bracket").Cells(28, 2)
End With
```

```
With Sheet1.Shapes("ComboBox8").ControlFormat
    .AddItem Sheets("Bracket").Cells(30, 2)
    .AddItem Sheets("Bracket").Cells(32, 2)
End With
```

```
End Sub
```

```
Public Sub Callback2(control As IRibbonControl)
```

```
    MsgBox "Hey, Congratulations! Your Bracket is 100% incorrect"
```

```
End Sub
```

```
Sub showIntro()
```

```
Dim str1 As String
```

```
    str1 = InputBox(Prompt:="Hello welcome to bracket city, let's fill out a bracket!  
But first please enter your first and last name", _
```

```
    Title:="", Default:="")
```

```
    Sheets("Bracket").Cells(1, 7) = "Name:"
```

```
    Sheets("Bracket").Cells(1, 8) = str1
```

```
End Sub
```

```
Sub insertComboBoxes()
```

```
    With Range("C3")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =  
"ComboBox1"
```

```
    End With
```

```
    With Range("C7")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =  
"ComboBox2"
```

```
    End With
```

```
    With Range("C11")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =  
"ComboBox3"
```

```
    End With
```

```
    With Range("C15")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =  
"ComboBox4"
```

```
    End With
```

```
    With Range("C19")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =  
"ComboBox5"
```

```
    End With
```

```
With Range("C23")
    Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =
"ComboBox6"
End With
```

```
With Range("C27")
    Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =
"ComboBox7"
End With
```

```
With Range("C31")
    Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =
"ComboBox8"
End With
```

'Next we insert a table to record the selected values

```
Worksheets("Bracket").Range("K1") = "SELECTIONS"
Worksheets("Bracket").Range("J2:J9") = "Round 1"
```

```
End Sub
```

```
Sub insertRound2Button()
```

```
With Range("H10:H11")
    Worksheets("Bracket").Buttons.Add(.Left, .Top, .Width, .Height).Select
    Selection.Name = "round2Complete"
    Selection.OnAction = "round3"
    With Selection
        .Characters.Text = "Continue to Round 3"
        With .Font
            .Name = "Arial"
            .Size = 12
        End With
    End With
End With
```

```
Sheet1.Range("H1").Select
```



```
End Sub
```

```
Sub insertRound3Button()
```

```
    With Range("H13:H14")
```

```
        Worksheets("Bracket").Buttons.Add(.Left, .Top, .Width, .Height).Select
```

```
        Selection.Name = "round3Complete"
```

```
        Selection.OnAction = "round4"
```

```
        With Selection
```

```
            .Characters.Text = "Continue to Round 4"
```

```
                With .Font
```

```
                    .Name = "Arial"
```

```
                    .Size = 12
```

```
                End With
```

```
            End With
```

```
        End With
```

```
Sheet1.Range("H1").Select
```

```
End Sub
```

```
Sub insertComboBoxes2()
```

```
    With Range("D5")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name = "ComboBox9"
```

```
    End With
```

```
    With Range("D13")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name = "ComboBox10"
```

```
    End With
```

```
    With Range("D21")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name = "ComboBox11"
```

```
    End With
```

```
    With Range("D29")
```

```
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name = "ComboBox12"
```

```
End With
```

```
'Next we insert a table to record the selected values
```

```
Worksheets("Bracket").Range("J10:J13") = "Round 2"
```

```
End Sub
```

```
Sub initializeValues2()
```

```
With Sheet1.Shapes("ComboBox9").ControlFormat
```

```
.AddItem Sheets("Bracket").Range("C3")
```

```
.AddItem Sheets("Bracket").Range("C7")
```

```
End With
```

```
With Sheet1.Shapes("ComboBox10").ControlFormat
```

```
.AddItem Sheets("Bracket").Range("C11")
```

```
.AddItem Sheets("Bracket").Range("C15")
```

```
End With
```

```
With Sheet1.Shapes("ComboBox11").ControlFormat
```

```
.AddItem Sheets("Bracket").Range("C19")
```

```
.AddItem Sheets("Bracket").Range("C23")
```

```
End With
```

```
With Sheet1.Shapes("ComboBox12").ControlFormat
```

```
.AddItem Sheets("Bracket").Range("C27")
```

```
.AddItem Sheets("Bracket").Range("C31")
```

```
End With
```

```
End Sub
```

```
Sub insertComboBoxes3()
```

```
With Range("E9")
    Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =
"ComboBox13"
End With
```

```
With Range("E25")
    Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =
"ComboBox14"
End With
```

'Next we insert a table to record the selected values

```
Worksheets("Bracket").Range("J14:J15") = "Round 3"
```

```
End Sub
```

```
Sub initializeValues3()
```

```
With Sheet1.Shapes("ComboBox13").ControlFormat
    .AddItem Sheets("Bracket").Range("D5")
    .AddItem Sheets("Bracket").Range("D13")
```

```
End With
```

```
With Sheet1.Shapes("ComboBox14").ControlFormat
    .AddItem Sheets("Bracket").Range("D21")
    .AddItem Sheets("Bracket").Range("D29")
End With
```

```
End Sub
```

```

Sub insertComboBoxes4()

    With Range("F17")
        Worksheets("Bracket").DropDowns.Add(.Left, .Top, .Width, .Height).Name =
"ComboBox15"
    End With

    'Next we insert a table to record the selected values

    Worksheets("Bracket").Range("J16") = "Winner"

End Sub

Sub initializeValues4()

    With Sheet1.Shapes("ComboBox15").ControlFormat
        .AddItem Sheets("Bracket").Range("E9")
        .AddItem Sheets("Bracket").Range("E25")

    End With

End Sub

Sub insertSubmissionButton()

    With Range("H16:H18")
        Worksheets("Bracket").Buttons.Add(.Left, .Top, .Width, .Height).Select
        Selection.Name = "bracketComplete"
        Selection.OnAction = "submitStuff"
        With Selection
            .Characters.Text = "Submit Bracket"
            With .Font
                .Name = "Arial"
                .Size = 12
                .FontStyle = "Bold"
            End With
        End With
    End With
End Sub

```

```

End With

Sheet1.Range("H1").Select
End Sub

Sub submitStuff()

' Here we record the selected values from teh ComboBoxes
  With Worksheets("Bracket").Shapes("ComboBox15").ControlFormat
    Worksheets("Bracket").Range("K16").Value = .List(.ListIndex)
  End With

'Remove rnd1 comboboxes
  ActiveSheet.DropDowns("ComboBox15").Delete

'place values from k10:k13 where previous comboboxes were
  Worksheets("Bracket").Range("F17").Value = Worksheets("Bracket").Range("K16").Value

  MsgBox "Thank you for completing a bracket. Your last step is to copy the elements in
  cells K2:K16 and paste them in an email to nate.a.barnes@gmail.com. Thank you."

  'Delete continue to rnd 2 button
  ActiveSheet.Buttons("bracketComplete").Delete

End Sub

Sub warning()
  MsgBox "You need to complete all entries for this round before proceeding."
End Sub

```