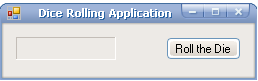
**A Simple Dice Rolling Application** or an introduction to random number creation.  
  
This tutorial will teach how to use the random number generation in VB to create a simple application to roll a 6-sided die. The steps in this can be further applied to include more than 1 die, different sizes of dice, or anything you want to think of that requires a random number generator. For simplicity's sake, I'll only be coding this into a single form, but the same techniques can be applied to custom classes as well.  
  
First things first, let's create a simple form.  
  
  
It doesn't take much for this application, just a button and an output label. Sure you can get fancy and add in images or a combo box to let them choose the number of sides, I'll even give you a screen shot and code snippet at the end of this tutorial showing you how, but for now let's keep it simple.  
  
***Note:****The name of my output label is lblDiceOutput and the name of my "Roll the Die" button is btnRoll*  
  
Okay, now that we have a form in place, let's move on to the simplistic code behind this app.

|  |  |  |
| --- | --- | --- |
| 1 | | Public Class frmD6 |
| 2 | Private Sub btnRoll\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnRoll.Click | | |

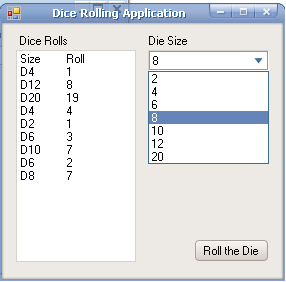
|  |  |
| --- | --- |
| 3 | Dim oDice As New Random |
| 4 | Dim iDiceResult As Integer = oDice.Next(1, 7) | |

|  |  |  |
| --- | --- | --- |
| 5 | lblDiceOutput.Text = iDiceResult.ToString | |
| 6 | End Sub |

|  |  |
| --- | --- |
| 7 | End Class |

This is a break down line by line for you so you understand what's going on.

1. Public Class frmD1 - This line is the opening of my form, frmD1 is what I decided to name it
2. Private Sub btnRoll\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRoll.Click - Starts the handler for the click event of btnRoll. Put simply, anything between this and End Sub will happen when you click the "Roll the Die" button.
3. Dim oDice As New Random - Creates a new object of the type random. oDice has now become our random number generator.
4. Dim iDiceResult As Integer = oDice.Next(1, 7) - This one can be a little hard to grasp at first. First I have to say this. Random is not entirely random. All computers can do in the end is perform math, and through math there are only a few limited ways to create "random" numbers, and then those go in a recurring pattern (it may only start recurring after 64,000 digits or so, but it is recurring). So this kept in mind, what this line does is it creates the iDiceResult variable as an Integer and then it sets it to the next random number in the sequence. 1 and 7 represent the lower and upper bounds respectively, remember when setting this that the second argument will always be greater than the random number generated. In this case we only want the die to go to 6, so the upperbound is 7.
5. lblDiceOutput.Text = iDiceResult.ToString - This simply sets the now "random" value stored iniDiceResult to the text property of lblDiceOutput to display what the "dice rolled."
6. End Sub - As I mentioned earlier, this closes the handler for the btnRoll.Click event.
7. End Class - And this ends my form code altogether.

So now you have a working digital 6-sided-die. Simple wasn't it? Now as I promised, here's the code for if you wanted to add a little flair to your form. I won't be adding images in this case for simplicities sake, but I will be adding a combobox to allow you to select how many sides the die has, and a list box to keep track of past rolls.  
  
The slightly more stylish form:  
  
  
The code behind it:

|  |  |  |
| --- | --- | --- |
| 01 | | Public Class frmD6 |
| 02 | Private Sub btnRoll\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnRoll.Click | | |

|  |  |
| --- | --- |
| 03 | If cmbDieSize.Text = String.Empty Then |
| 04 | MessageBox.Show("Please choose the size of die you wish to roll", "Error") 'Quick error handling in case the user tries to roll without selecting a die size | |

|  |  |
| --- | --- |
| 05 | Else |
| 06 | Dim oDice As New Random | |

|  |  |
| --- | --- |
| 07 | Dim iDieSize As Integer = CInt(cmbDieSize.Text) |
| 08 | Dim iDiceResult As Integer = oDice.Next(1, iDieSize + 1) | |

|  |  |  |
| --- | --- | --- |
| 09 | lstDiceRolls.Items.Add("D" & iDieSize.ToString & ControlChars.Tab & iDiceResult.ToString) | |
| 10 | End If |

|  |  |  |
| --- | --- | --- |
| 11 | End Sub | |
| 12 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 13 | Private Sub frmD6\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load | | |
| 14 | | lstDiceRolls.Items.Add("Size" & ControlChars.Tab & "Roll") |

|  |  |  |
| --- | --- | --- |
| 15 | End Sub | |
| 16 | End Class |

***Notes:****I kept my form name the same out of laziness. cmbDieSize is the combobox above the roll button containing the die sizes, it was populated in the design view. lstDiceRolls is the listbox on the left of the form that records the rolls.*  
  
Any and all corrections and feedback are welcome.