

## Objectives

### Program Purpose

- Design a simple number guessing game
- The computer generates a hidden number between 1 and 25
- The user guesses whether it is even or odd
- A message displays whether the user guessed correctly or not

### Learning Goals

- Use of option buttons
- Use of a Boolean variable
- Use of comments in the code
- Use of 'If\_Then\_Else\_ElseIf\_End If' statements
- Use of a 'Randomize' statement

## Design Notes

The icon property of the form is used. This displays the small icon in the top-left corner of the form, besides the form's caption. These 'ico' files are usually installed in the following folder:  
*C:\Program Files\Microsoft Visual Studio\Common\Graphics\Icons\*.

Another property introduced here is the 'Startup Position' of the form. Setting this to 2 (centre screen) saves manually positioning the form.

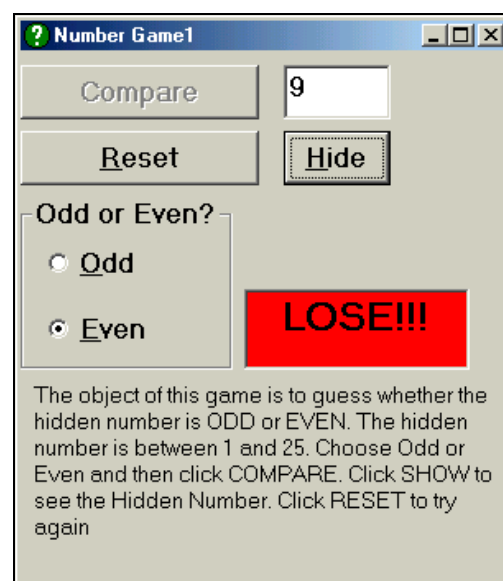
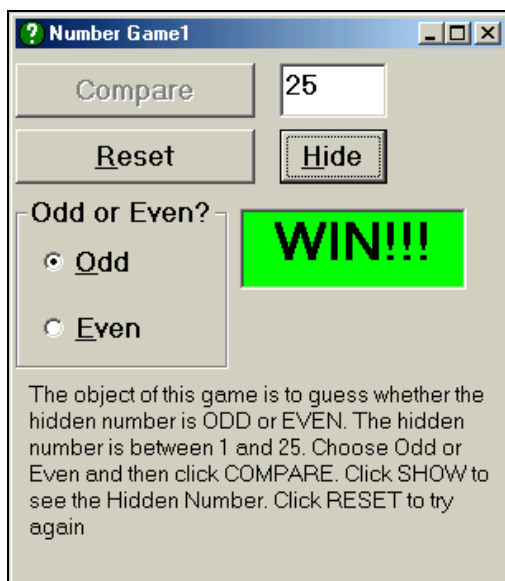
The initial properties are very important in this project. Pay particular attention to the 'Enabled', 'Visible' and 'Value' properties.

It is not necessary to name a label or frame if they serve no other purpose than displaying a message or to contain option buttons. Similarly, a form is only named if it is addressed in the code.

A variable is used to store information temporarily. A Boolean variable can have only 2 values (True or False). It is 'declared' with a DIM statement in the general section of the program.

## Interface

The Interface has 2 hidden labels, one of which is made visible (which one depends on the user's guess). The command button 'cmdHide' should have the caption '&Show'. This will change during the execution of the program.



## Names of Objects

Type of Object	Number	Names of Objects
Form	1	Form1
TextBox	1	txtDisplay
Command Button	3	cmdCompare, cmdReset, cmdHide
Frame	1	No name –Frame1
Option Button	2	optOdd optEven
Label	3	lblWin lblLose Label1

## Initial Properties of Objects

Object	Property	Initial Value
Form	Caption	Number Game 1
	Startup Position	2-Center Screen
	Icon	Browse for an "ico" file: see Design Notes above
txtDisplay	Text	""
	Visible	False
cmdCompare	Caption	&Compare
	Enabled	False
	Value	True
cmdReset	Caption	&Reset
	Enabled	False
cmdHide	Caption	&Show
	Enabled	False
optEven	Caption	&Even
	Value	True
optOdd	Caption	&Odd
	Value	True
Label1	Caption	The object of this game is to guess whether the hidden number is ODD or EVEN. The hidden number is between 1 and 25. Choose Odd or Even and then click COMPARE. Click SHOW to see the Hidden Number. Click RESET to try again
	Caption	WIN!!
	BackColor	GREEN
	Font	18 Bold
lblWin	Visible	False
	Caption	LOSE!!
	BackColor	RED
	Font	18 Bold
lblLose	Visible	False
	Font	18 Bold
Frame1	Caption	Odd or Even?

## Events – Code

### Private Sub cmdCompare\_Click()

```
'work out whether or not the number is even
'MOD gives the remainder after division
'VAL converts text to a number
'The next line declares a boolean variable
'This stores a value of true or false

Dim Even As Boolean

'first determine if the number is even or not

If Val(txtDisplay.Text) Mod 2 = 0 Then 'its even
    Even = True
Else
    Even = False
End If

'now compare the User's Guess with ..
'the value of the variable Even

If Even = True And optEven.Value = True Then
'we picked even and it was even

    lblWin.Visible = True
Elseif Even = False And optOdd.Value = True Then
'we picked odd and it was odd

    lblWin.Visible = True
Else
    lblLose.Visible = True
End If

'disable compare button so it can't be clicked until
'reset

cmdCompare.Enabled = False
cmdHide.Enabled = True

End Sub
```

### Private Sub cmdHide\_Click()

```
'if the text box is displayed it hides it
'otherwise it shows it

If txtDisplay.Visible = True Then 'Hide it!
    cmdHide.Caption = "&Show"
    txtDisplay.Visible = False
Else 'Show it
    cmdHide.Caption = "&Hide"
    txtDisplay.Visible = True
End If

End Sub

Private Sub cmdReset_Click()

'reset all the buttons and boxes

txtDisplay.Visible = False
cmdHide.Caption = "&Show"
lblWin.Visible = False
lblLose.Visible = False
cmdCompare.Enabled = True
cmdHide.Enabled = False

'generate a new random number
'randomly generates a number between 1 and 25
'and displays it in the textbox

Randomize
txtDisplay.Text = Int(Rnd * 25 + 1)

End Sub

Private Sub Form_Load()

Randomize
txtDisplay.Text = Int(Rnd * 25 + 1)

End Sub
```

## Further Design Notes

- Comments are used to display information for the programmer. The use of an apostrophe at the start of the line indicates it is a comment and not code. It will be ignored at runtime.
- The MOD operator gives a remainder. This is useful for checking if a number is even or not.
- The VAL function converts text to a number. Its use prevents odd results where strings are converted into their underlying ASCII value.
- Mistakes are easily made if variables are spelt or typed differently. To prevent this, the programmer should check 'Require Variable Declaration' under the 'Tools' – 'Options' menu.

**Modulus**

'MOD' is an operator that gives the remainder after integer division. 'Mod' comes from the word 'Modulus'.

12 MOD 5 = 2	13 MOD 5 = 3	14 MOD 5 = 4	0 MOD 4 = 4
12 MOD 4 = 0	13 MOD 3 = 1	15 MOD 7 = 1	10 MOD 7 = 3

**Random Numbers**

Random numbers are generated using the 'Randomize' statement and the 'Rnd' Function. The 'Int' function is used to convert a decimal to an integer (it actually rounds down to the next integer, so 25.99 becomes 25).

The use of the word **Randomize** ensures a different random number sequence is used each time the program is started. **Rnd** generates a random decimal between 0 and 1. By multiplying that random number by a given number, say 25, and adding 1 we can get a random number between 1 and 25.

```
Randomize
txtDisplay.Text = Int(Rnd * 25 + 1)
```

**Questions**

1. Work out the answers to these sums:
  - a. 10 MOD 6 =
  - b. 10 MOD 4 =
  - c. 4 MOD 10 =
  - d. 8 MOD 2 =
2. What are the 4 events we had to write code for in the 'Number Game 1' program?
3. What are comments used for?
4. How are comments put into a program?
5. Where should comments be put in a program?
6. What is a BOOLEAN variable used for? What are its 2 possible values?
7. How do we declare a BOOLEAN variable – give an example.
8. Write simple assignment (1 line code) statements to:
  - a. Disable a command button named 'cmdRain'
  - b. Enable a command button named 'cmdSnow'
  - c. Make visible a text box named 'txtEarth'
  - d. Make invisible a label named 'lblWeekend'
9. List 5 other events that code can be written for (e.g. Double\_Click).
10. Write code to generate a random integer between:
  - a. 1 and 50
  - b. 10 and 100
  - c. -5 and +5

**Extension Activity**

Open your 'Number Game 1' program again:

- a. Resave it as 'Number Game1a.vbp' (use 'File' – 'Save Project as').
- b. Resave the form file as well as Number Game 1a.frm (use 'File' – 'Save Form as')
- c. Modify the program to check whether the hidden number is
  - i. Between 1 and 10
  - ii. Between 11 and 20
  - iii. Between 21 and 30

You will need to modify the interface and some of the code in the 'cmdCompare' event and the 'Reset' and 'Form-Load' events.

1. Work out the answers to these sums:
  - a.  $10 \text{ MOD } 6 = 4$
  - b.  $10 \text{ MOD } 4 = 2$
  - c.  $4 \text{ MOD } 10 = 4$
  - d.  $8 \text{ MOD } 2 = 0$
2. What are the 4 events we had to write code for in the 'Number Game 1' program?  
**Form\_Load**  
**cmdReset\_Click**  
**cmdCompare\_Click**  
**cmdHide\_Click**
3. What are comments used for?  
**To explain code to the programmer or others who may read the code.**
4. How are comments put into a program?  
**By putting an apostrophe at the start of the line**
5. Where should comments be put in a program?  
**At the beginning of a section of code or wherever an explanation is helpful.**
6. What is a BOOLEAN variable used for? What are its 2 possible values?  
**To store a TRUE or FALSE value**
7. How do we declare a BOOLEAN variable – give an example.  
**The Declaration should be put in the General Section of Code.**  
**DIM Winner AS Boolean**
8. Write simple assignment (1 line code) statements to:
 

d. Disable a command button named cmdRain	<b>cmdRain.Enabled = False</b>
e. Enable a command button named cmdSnow	<b>cmdSnow.Enabled = True</b>
f. Make visible a text box named txtEarth	<b>txtEarth.Visible = True</b>
g. Make invisible a label named lblWeekend	<b>lblWeekend.Visible = False</b>
9. List 5 other events that code can be written for (e.g. Double\_Click).  
**Change, keyPress, MouseUp, MouseMove, Load (see the Code interface for more examples)**
10. Write code to generate a random integer between:
 

a. 1 and 50	<b>Int(Rnd*50+1)</b>
b. 10 and 100	<b>Int(Rnd*91+10)</b>
c. -5 and +5	<b>Int(Rnd*11-5)</b>

### Extension Activity

There are many ways to solve this programming problem. The following 'If\_Then\_ElseIf\_Else\_End If' structure would assist in determining if the number displayed in the textbox 'txtDisplay' is in the given ranges.

<b>If Val(txtDisplay.Text) &lt; 11 Then</b>	<b><i>'its less than 11</i></b>
<b>'display appropriate message</b>	
<b>ElseIf Val(txtDisplay.Text) &lt; 21 Then</b>	<b><i>'its greater than 10 and less than 21</i></b>
<b>'display appropriate message</b>	
<b>Else</b>	<b><i>'its greater than 20</i></b>
<b>'display appropriate message</b>	
<b>End If</b>	