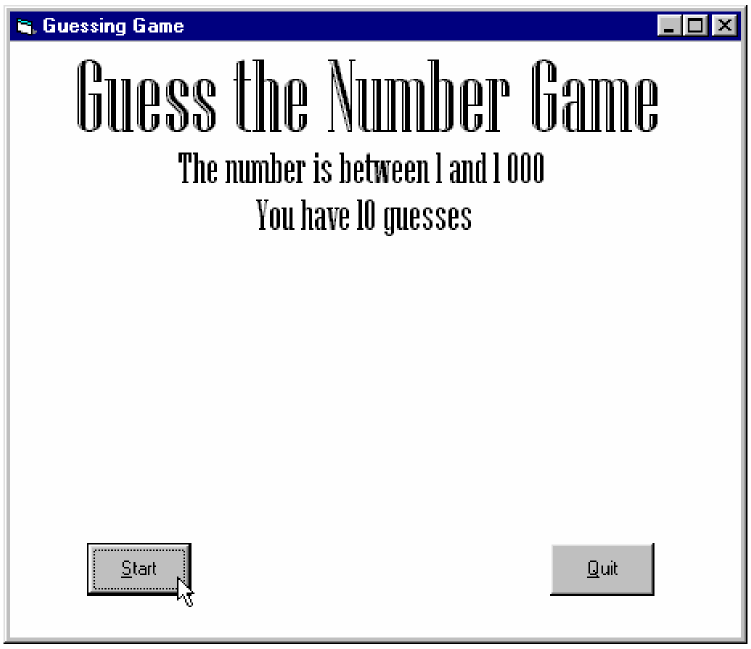
Project 11 – Guess the number

Task: Give the user ten guesses at a random number between 1 and 1 000.

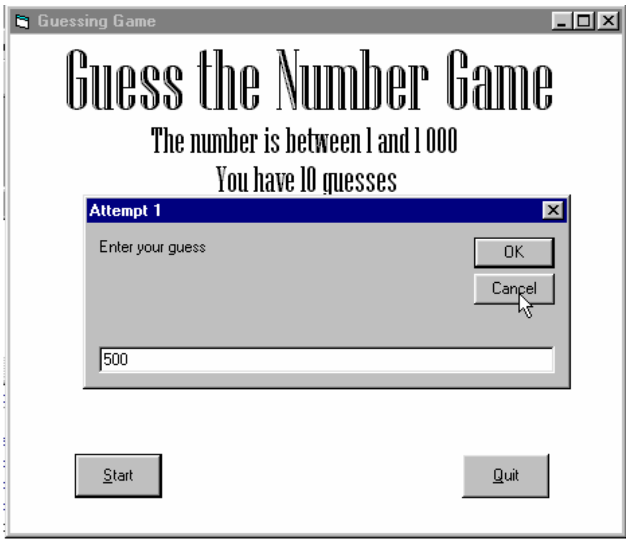
Objects: Title label and instructions label, dialog and message boxes, buttons to start and exit the game.

Events: Mouse click on command buttons, and on dialog box buttons.

1. Start a new project and create the following form:



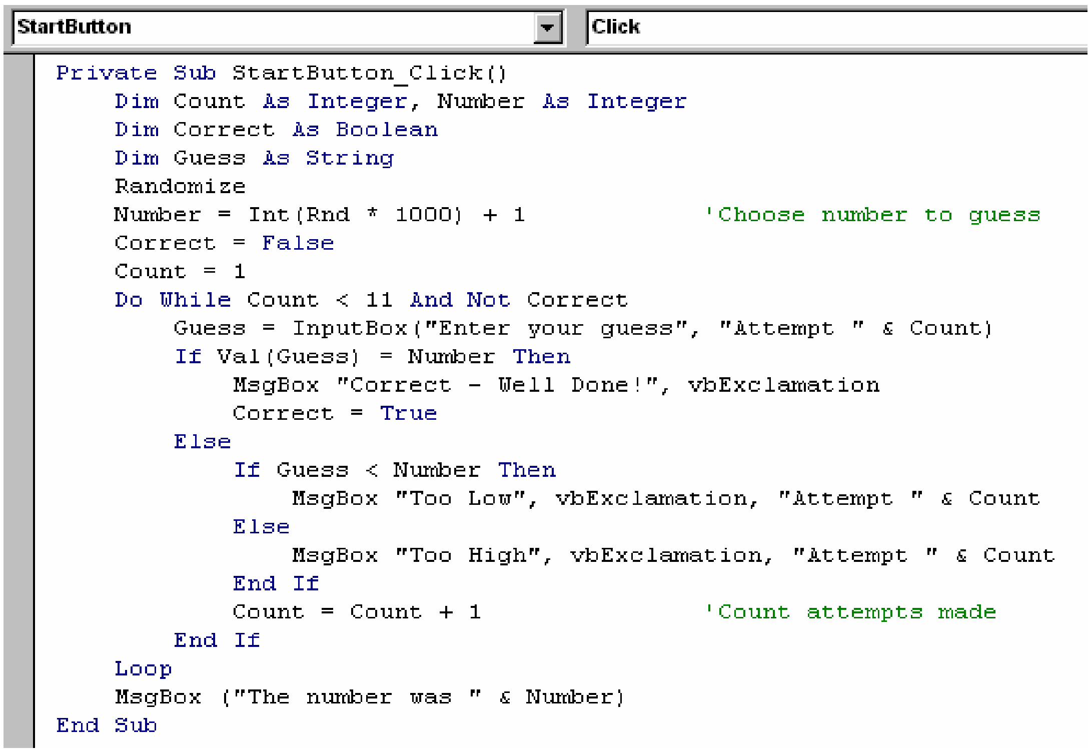
Leave a gap in the middle as indicated as we will be placing *dialog boxes* there as the program runs.



2. a Name the *start* and *quit* buttons.

   b Add code to the *QuitButton*.

   c The code for the *StartButton* is:
   (Hint: the use of the *tab* and the *backspace* keys will assist with effective indentation.)



StartButton | Click

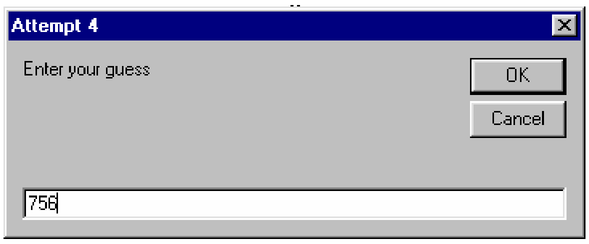
```
Private Sub StartButton_Click()
    Dim Count As Integer, Number As Integer
    Dim Correct As Boolean
    Dim Guess As String
    Randomize
    Number = Int(Rnd * 1000) + 1            'Choose number to guess
    Correct = False
    Count = 1
    Do While Count < 11 And Not Correct
        Guess = InputBox("Enter your guess", "Attempt " & Count)
        If Val(Guess) = Number Then
            MsgBox "Correct - Well Done!", vbExclamation
            Correct = True
        Else
            If Guess < Number Then
                MsgBox "Too Low", vbExclamation, "Attempt " & Count
            Else
                MsgBox "Too High", vbExclamation, "Attempt " & Count
            End If
            Count = Count + 1               'Count attempts made
        End If
    Loop
    MsgBox ("The number was " & Number)
End Sub
```

(This will all be explained shortly.)

3. a Run the program (saving with appropriate file names).

   b Make any improvements you wish (e.g. adjust placement of the window on screen, or add tooltips).

There are two new devices used in the project, an *InputBox* and a *MsgBox*. These are examples of *dialog boxes.*



Attempt 4

Enter your guess

756

The line:

```
Guess = InputBox("Enter your guess", "Attempt " & Count)
```

uses an *input dialog box* to get the user to enter their next guess. An input dialog box must be used in this way i.e. assigned to a variable. Note also that the value that is input is a string. Because of this, in the first *if* statement we have to use the *Val* function to convert the text from the input box into a numeric value so we can compare it with the integer variable *Number*.

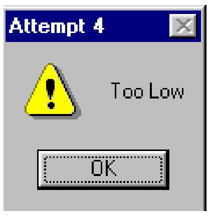
The first text in quotes (`"Enter your guess"`) is the message to the user. The second part (`"Attempt " & Count`) is the caption on the dialog box. (If their is no second set of quotes the project name is displayed.) In this case we also wished to show the attempt number. We did this by joining the variable *Count* to the text using the *&* symbol.

4. If we wish we can also stipulate the default value for the input box. Try the following:

   ```
   Guess = InputBox("Enter your guess","Attempt " & Count, "500")
   ```

   Separate the text in quotes with commas. The third value is the initial value to be displayed.

5. The *MsgBox dialog* is used to display messages.

   Too Low

   Re-run the program with the following changes for *vbExclamation* to change the graphic displayed in the dialog box:

   Change from: `MsgBox "Too low", vbExclamation`
   to: `MsgBox "Too low", vbQuestion`
   or: `MsgBox "Too low", vbInformation`

   You can also see the effect of replacing *vbExclamation* with any number between 0 to 5:

   e.g. `MsgBox "Too low", 2`

   For an explanation of what each of these refer to check out *MsgBox Function* in Help.

   Note also we have used & to join the variable *Count* to the caption in each of the three message boxes as we did with the input box.

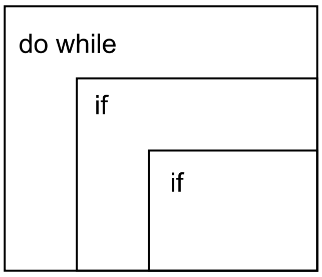
Do while

The pseudocode for the program in project 11 is:

```
NUMBER = random 1000
CORRECT = false
COUNT = 1
do while COUNT < 10 and not CORRECT
    read GUESS
    if GUESS = NUMBER then
        write "Correct – Well Done"
        CORRECT = true
    else
        if GUESS < NUMBER then
            write "Too low"
```

```
            else
                write “Too high”
            end if
            COUNT = COUNT + 1
        end if
    loop
    write “The number was” NUMBER
```

In this pseudocode we have a nested *if* selection inside of a *while* loop:



The variable *Number* holds the random value to be guessed, *Correct* records if the user has the right value, *Guess* holds their current attempt, and *count* records the number of attempts.

The part of the program that does the work is the *while loop*. This is controlled by two conditions. *Count* has to stay under 11 and *Correct* has to stay false for the loop to continue.

Before the loop starts *Count* (how many guesses the user has taken) is set to 1, and then every time the loop executes *Count* is incremented (`Count = Count + 1`). When it reaches 11 the loop stops whether the user has guessed correctly or not. This is how the user is given 10 goes to guess the number. (The setting of *Count* to 1 before the loop starts is known as *initialising* the loop.)

The variable *Correct* is used to stop the loop if the user does guess the number before *count* gets over 10. Before the loop starts *Correct* is set to *False*; it only becomes *True* if the user guesses the number.

This condition has been written as:

```
Do While Count < 11 And Not Correct
```

but it could also have been written as:

```
Do While Count < 11 And Correct = False
```

The two conditions in the *while* statement are joined with the reserved word *And*. This means the loop will only run if *both* conditions are true.

Notice again the use of *indentation* in the above code. Everything that is to be repeated as part of the *while loop* is indented to make it easier to see that that section of code belongs together. In the same way the pseudocode belonging to the *then* and *else* parts of the *if* statements are also indented.

Forms of iteration

The *while loop* is a *pre-tested indefinite iteration*. Pre-tested means the loop is tested before it runs – if either condition is false the code in the *while loop* will not be executed. It is *indefinite* in that