

TECHNICAL PAPER***USING VISUAL BASIC IN THE EXPERIMENTAL ANALYSIS OF HUMAN BEHAVIOR: A BRIEF INTRODUCTION***

Francisco Cabello, Dermot Barnes-Holmes, Denis O'Hora and Ian Stewart

NATIONAL UNIVERSITY OF IRELAND - MAYNOOTH

Computers have now become an essential part of the experimental psychology laboratory. In a previous paper, the advantages of using PsyScope, a graphical tool for designing psychology experiments were discussed (Roche, Stewart & Barnes-Holmes, 1999). The primary advantage identified was PsyScope's graphical user interface which is much more user-friendly than traditional programming languages. On balance, however, PsyScope is restricted to the Macintosh platform, is now no longer under development, and certain features such as drag & drop functions, are not available. As time passes, therefore, PsyScope may become increasingly obsolete. Fortunately, these issues can be addressed readily by turning to the use of Visual Basic (VB). This is a modern programming language that combines the use of a graphical user interface with the easy-to-learn BASIC programming language (see Bishop-Clark, 1998), and as part of Microsoft's ongoing development plans VB obsolescence is not an issue in the short to medium term. The key advantage to VB is that it provides the power and flexibility of PsyScope using the PC platform. Furthermore, VB can also be used on a Macintosh using standard Macintosh-PC emulators, whereas attempts in our laboratory to use PsyScope on a PC using a Macintosh emulator were unsuccessful.

The use of VB has been reported in a number of articles (Beatty, 1999; Dunham, Sherba & Rutherford, 1996; Pelechano & Darias, 2000; Saeki, Uchida & Ito, 1998). In the current paper, we will

provide some relevant examples that will illustrate some of the main characteristics of VB, that make it perhaps the most useful tool available to behavior-analytic researchers for developing computerized experimental control.¹

USING VISUAL BASIC

Those readers who remember the BASIC programming language will recall that the program consisted entirely of lines of symbolic instruction code (e.g. if $x=10$ then $y=0$). Any graphics or mouse control events required quite complex programming skills. In contrast, VB consists primarily of individual blank pages (called "forms"), upon which various common controls are presented, such as buttons, text boxes and picture frames. Stimuli can easily be placed on these and other types of controls, which may be then activated within the program to record either mouse clicks or button presses.

Example 1: Creating a Simple Matching-to-Sample Task

The first example will illustrate the process of writing a simple matching-to-sample task using VB. Figure 1 shows the typical arrangement of tools in the VB programming environment. Forms are displayed in the center of the screen, a tool bar containing all available controls is placed on the left side, and properties for each control can be found on the right side. For example, having placed a button on the form, the color of this button can be changed by selecting the appropriate property (see below).

In the first step of our example, "New Project" is selected from the "File" menu. A new project file is generated by the computer, including a blank form that will contain all the stimuli. The label control (used to display text) is

Preparation of this manuscript was supported by a scholarship awarded by the Irish Department of Education and Science to the first author, to spend a year at the National University of Ireland, Maynooth. Address correspondence to the first author at, francisco.cabelloluque@wanadoo.es.

Francisco Cabello is now at Universidad de la Rioja. Denis O'Hora is now at London Metropolitan University. Ian Stewart is now at National University of Ireland, Galway.

¹ More example programs and useful information are available from our VB website (www.may.ie/academic/psychology/VB/).

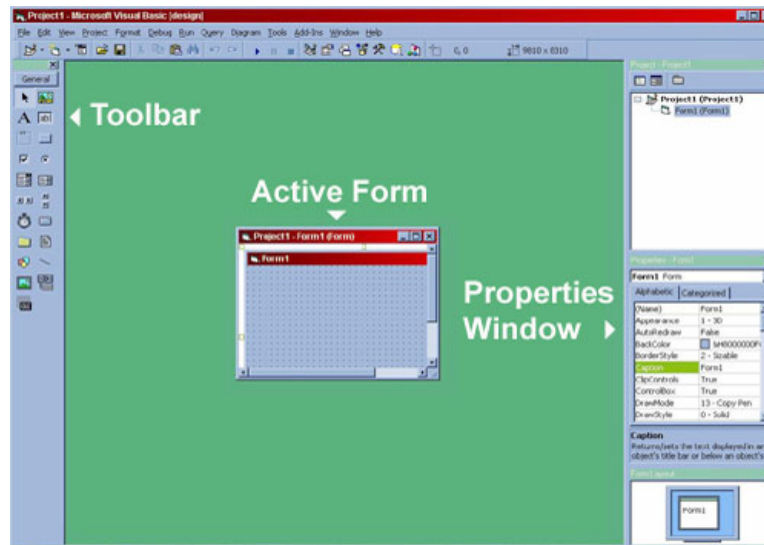


Figure 1

selected in the tool bar, and four labels are created and each placed in the desired position using the mouse (like in any drawing program): one on the top of screen, one in the center, and two at the bottom. The names of these labels will be changed using the Properties Window (by clicking on the *Name* property): the top label is named *Sample*, the center label *Feedback*, the left bottom label *ComparisonLeft* and the right bottom label *ComparisonRight*. Other properties that might be changed are the size and type of the font, the color of the text or the background, and so forth. Finally, the text displayed in each label is also changed using the Properties Window (by clicking on the *Caption* property). Text in the *Sample* label is set to "A1", text in the *ComparisonLeft* label to "B1", text in the *ComparisonRight* label to "B2", and text in the *Feedback* label to a blank space.

Thus far, all of the work has been done using the graphical user interface. The next step is to write the code that will control the program. The sample was A1, so *ComparisonLeft* (B1) is the correct comparison. To insert the code, the *ComparisonLeft* label is double-clicked, which opens the Code Window associated with this control. Every instruction written here will be executed when the *ComparisonLeft* label is activated by a mouse click (code for other events can be added, as in the case of pressing a key or moving the mouse over a control). Comments written after the inverted comma at the end of

each line explain what each line of code does within the program:

```
Sample.Caption=""
'clears text in the Sample label
ComparisonLeft.Caption=""
'clears text in the ComparisonLeft label
ComparisonRight.Caption=""
'clears text in the ComparisonRight label
Feedback.Caption="CORRECT"
'sets text in the Feedback label to
CORRECT
```

Finally, the *ComparisonRight* label is double-clicked, and the following code is added in the corresponding Code Window:

```
Sample.Caption=""
'clears text in the Sample label
ComparisonLeft.Caption=""
'clears text in the ComparisonLeft label
ComparisonRight.Caption=""
'clears text in the ComparisonRight label
Feedback.Caption="WRONG"
'sets text in the Feedback label to WRONG
```

If the program is executed, the screen will show the arrangement of stimuli; if the right comparison is clicked with the mouse, the message "CORRECT" appears ("WRONG" is displayed if the left comparison is clicked). Additional forms can be added to the project, each of which includes one type of task (e.g., presenting

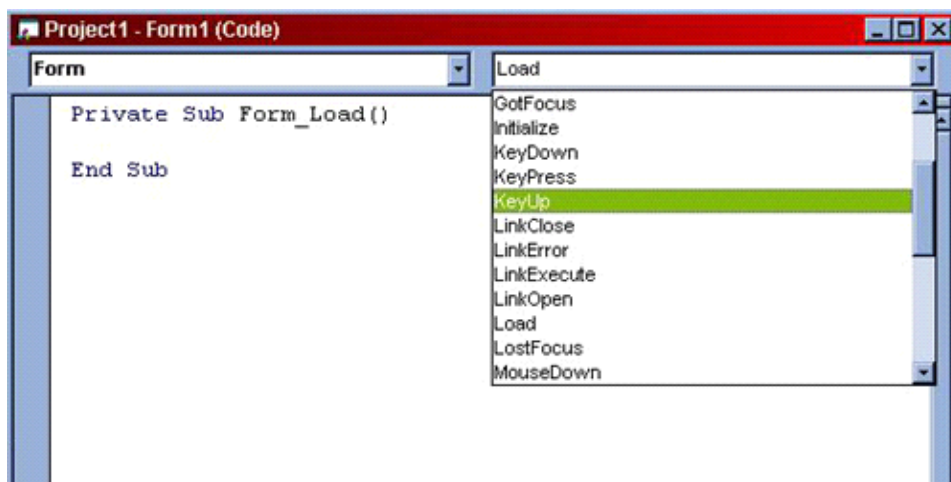


Figure 2

A2 as sample, B1 as left comparison and B2 as right comparison). Functions can be implemented to present these tasks in a random order or to record the performance of subjects.

Example 2: A Simple Reinforcement Schedule

The second example involves creating a program for controlling a Fixed-Ratio 8 schedule in which the space bar will function as the operandum. As in the previous example, selecting “New project” in the File menu creates a new project with a blank form. The label control is selected from the tool bar and one label is placed on the screen. Using the Properties Window, the name of the label is changed to *Score* (by clicking on the *Name* property), and the text displayed in the label is changed to “Points: 0” (by clicking on the *Caption* property).

Once the graphical work is done, the form is double-clicked and the Code Window will open. The following step involves creating two variables: *keypresses* (that will record the number of spacebar presses), and *points* (that will record the number of points gained by the subject). To do this, the following code is added in a special section of the Code Window, termed the Declarations section, which is placed at the very top of the Code Window (see below):

```
Dim keypresses
Dim points
```

The term “Dim” specifies the two variables (*keypresses* and *points*). Next, some code will be added in the KeyUp section of the form. To select

KeyUp, click the top right corner of the Code Window and select “KeyUp” (Figure 2 shows where to find the KeyUp section). The code written inside the KeyUp section will be executed when the space bar is pressed. Therefore, the following code is added:

```
If KeyCode=vbKeySpace Then
    ' when spacebar is pressed
    keypresses=keypresses+1
    ' adds one press
    If keypresses=8 Then
        ' if eight presses
        points=points+1
        ' adds one point
        Score.Caption="Points: " & points
        ' updates Score label
        Beep
        ' plays a sound
        keypresses=0
        ' starts schedule
    End If
End If
```

If the program is executed, the computer will display a screen with the message “Points: 0”. When the spacebar is pressed eight times, a point will be added to the points counter and a short sound will be played. Of course, this is a relatively simple program for just one simple schedule, but the programming principles involved in creating more complex experimental procedures do not differ greatly from those outlined in the current article.

CONCLUSION

These two simple examples illustrate how easy it is to use VB to create computer-controlled experimental procedures in behavioral studies, thus allowing for the rapid implementation of research ideas. Moreover, the tasks outlined here can be easily modified to include pictures, sounds, video clips, or to include drag & drop functions. Other VB controls can also be included in a project, that allow for the control of external electronic devices. Furthermore, Dixon (under review) suggests some of the possibilities of the use of VB in portable handheld devices for the collection of data. Finally, due to the widespread use of VB it is possible, at least in principle, for different behavioral laboratories to share their programs or sections of programs, thereby facilitating cooperation among different research groups and the replication of experimental procedures.

The emergence of the personal computer has transformed experimental psychology. Mastering programming skills provides the researcher with the ability to create his or her own highly flexible behavioral laboratory. However, mastering a programming language is often perceived to be an extremely time-consuming and intellectually burdensome task. Although computer programming does require some initial investment in time and energy, VB is particularly user-friendly for a novice programmer. Furthermore, once the basic principles of VB are understood, the ability to implement new research ideas easily, efficiently, and rapidly, provides a massive reinforcer for the initial investment.

Finally, there are many good textbooks out there for those wishing to teach themselves Visual Basic, but an excellent one for those completely fresh to this programming language is "Visual Basic for the Absolute Beginner" (Vine & Harris, 2001). Furthermore, a forthcoming book edited by Dixon and MacLin (in press) and published by Context Press provides an introduction and many examples of programs for behavioral researchers. As indicated above, the website www.may.ie/academic/psychology/VB/ contains useful information and programs that we have developed and used in our recent research.

REFERENCES

Beatty, J.C. (1999). The PowerChecker: a Visual Basic Program for Ensuring Data Integrity.

Behavior Research Methods, Instruments & Computers, **31**, 737-740.

Bishop-Clark, C. (1998). Comparing Understanding of Programming Design Concepts Using Visual Basic and Traditional Basic. *Journal of Educational Computing Research*, **18**, 37-47.

Dixon, M.R. (under review). Creating Your Own Portable Data Collection System with Microsoft Embedded Visual Tools for the Pocket PC. *Journal of Applied Behavior Analysis*.

Dixon, M.R. & MacLin, O.H. (in press). *Visual Basic for Behavioral Psychologists*. Reno, NV: Context Press.

Dunham, D.W., Sherba, M. & Rutherford, P. (1996). INFOBITS: A Visual Basic Program for Calculating Information Theoretic Statistics from Behavior Sequences. *Behavior Research Methods, Instruments & Computers*, **28**, 622-626.

Pelechano, V. & Darias, E. (2000). Computerized Personality Questionnaires in Microsoft Visual Basic 6.0 for Windows [Cuestionarios de Personalidad por Ordenador en Microsoft Visual Basic 6.0 para Windows]. *Psicothema*, **12**, 418-422.

Roche, B., Stewart, I. & Barnes-Holmes, D. (1999). PsyScope: an Easy-to-Use Graphic-Oriented Application for Designing and Controlling Computer-Based Research on Relational Responding. *Experimental Analysis of Human Behavior Bulletin*, **17**, 5-7.

Saeki, D., Uchida, Y. & Ito, M. (1998). A Visual Basic-based Control System for Behavioral Experiments. *Japanese Journal of Behavior Analysis*, **13**, 66-72.

Vine, M.A. & Harris, A. (2001). *Microsoft Visual Basic Programming for the Absolute Beginner*. Premier Pr.