Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Education 46 (2006) 206–220
COMPUTERS & EDUCATION
www.elsevier.com/locate/compedu

Why are macros not used? A brief review and an approach for improving training

Eldad Yechiam *

Department of Psychology, Indiana University, 1101 East Tenth Street, Bloomington, IN 47405, USA

Received 12 January 2004; accepted 2 August 2004

Abstract

Macros are programming scripts that can be generated by recording users' actions. It appears that despite their potential for reducing monotonous work, they are under-used by non-programmers. The present paper reviews the literature on the use of macros. Included is an original study that examines the effect of adding the "search and replace" option as an alternative to using macros in a hypothetical task. It is concluded that the problem of not using macros is due, in part, to motivational biases. Motivational biases emerge even after macro training because simpler alternatives that "compete" with macros are more attractive to users. These biases can be addressed through the training process. Some training principles for a strategically competitive real-world environment are presented.

Keywords: Teaching/learning strategies; Programming and programming languages; Human–computer interface

1. Introduction

At the beginning of the 1990s the use of macros, a "pseudo-programming language" (Richardson, 1990) was an exciting avenue for researchers in the field of computers and education (e.g., Litecky, 1990; Ramadurai, 1991; Richardson, 1990; Valliere, 1986; Woodrow, 1988, 1989). While there are many implementations of macros, they are typically used to re-execute user actions. This

* Tel.: +1 812 856 4678; fax: +1 812 855 4691.
E-mail address: eyehiam@indiana.edu.

0360-1315/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2004.08.009

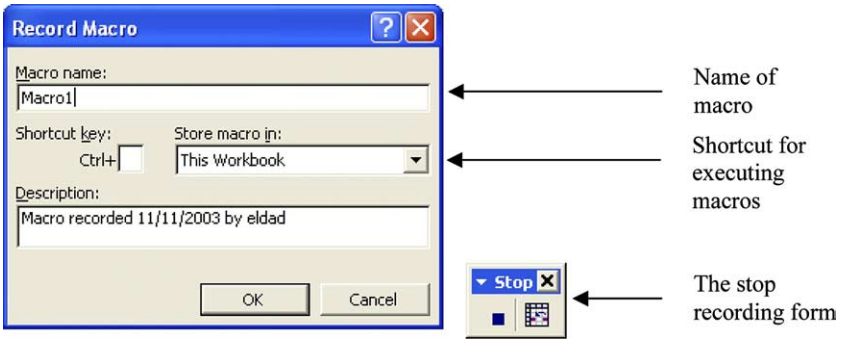
can reduce repeated performance of monotonous actions. One way of building a macro that has recently become popular is the recording of user actions, which can then be re-executed at will. Box 1 demonstrates this simple use of macros in Microsoft Excel. A more advanced method of generating macros is writing code.

Box 1: An example of the use of macros in Microsoft Excel

The following example demonstrates the use of macros in Microsoft Excel (its use is identical in all other Microsoft Office applications).

The task: The user wishes to repeatedly open a file (with a fixed name but different content), sort it into five fixed areas, and print it.

The solution: After opening the file, the users chooses: Tools ⇒ Macros ⇒ Record new macro. This presents the "Record macro" form, where the user can enter additional information (see figure). After pressing "ok", the "stop recording" form appears.



The user then performs the required sorting and printing operations, and presses the stop button in the "stop recording" form to end the macro recording. To execute the macro in a new file, the user opens a new file and chooses: Tools ⇒ Macros ⇒ Macros. The user selects the macro to be run from a list ("macro 1"). The sorting and printing actions are now performed in the new file.

This simple macro operates by a code generated from a recording of the user's actions. For example, the code for selecting cells in Excel is:

```
Range("F9:I10").Select
```

Note that the code is a simple translation of the operation on a given range of Excel cells. A more advanced way to create macros is to write code directly.

Macros are potentially beneficial for educators as well as computer users. For users, the employment of macros has the potential of relieving the need to perform repetitive series of actions. Macros also introduce more punctuality in action repetition, since repetitions are performed by the machine (Yechiam, Erev, & Parush, 2004a). For educators, macros constitute a programming environment that is easy to teach. This environment may be suitable for young children or for people with cognitive impairments. It can be taught for purposes of future use, or as a gateway to other programming languages.

One advantage of learning macros over ordinary programming languages is in memory requirements. Since code can be generated by users' actions, it is often not necessary to write code from