

Teacher Designed Software
Declan O'Reilly.
University of Sheffield

Abstract

Teachers have always gone beyond the textbook or scheme to design worksheets or other materials geared to the-needs of their children. However, until recently, the parallel activity in a computer environment was not possible. This paper describes some research with year 5 children (ages 9 to 10) in which the medium of Boxer was used to introduce itself. Altogether, eight case-study children representing the full ability range were closely observed using three microworlds known as 'First-', 'Second-' and 'Third-Boxer'. However, the microworlds were also being used simultaneously by the remainder of the year 5 class in so far as time, hardware and other classroom limitations allowed. The feedback from these two sets of users formed an integral component of the iterative design process.

Introduction

The name 'Boxer' stems from the fact that programs are typed inside boxes on the screen. There are two major kinds of boxes: 'Doit' boxes and 'Data' boxes, which can be thought of as corresponding to verbs and nouns in the English language. Boxes can be opened or closed controlling the amount of detail which is visible on the screen at any one time. Figure 1a, for example, shows the 'First-Boxer' microworld whilst Figure 1b shows one of its constituent parts 'Shapes-1' partially opened. These are presented here to assist the reader. In reality, children tended to open up boxes as they needed them and close them when not in use.

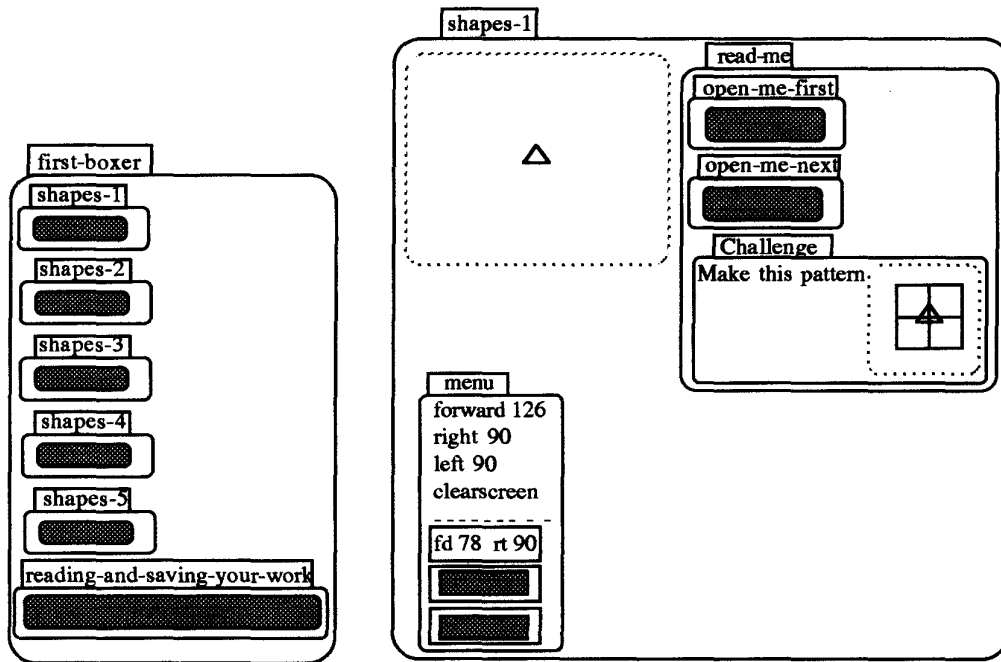


Figure 1a and 1b: First-Boxer and Shapes-1

Presentation and Communication

The first point that needs to be made here is the obvious one that communication was via the screen. This work was not supplemented by worksheets or by instructional manuals. The medium introduced itself. The text, the graphics and the programs were integrated on the same screen. Information overload was reduced by simply closing

boxes no longer in use. On the other hand, as it became cumulatively clear that children were having problems in interpreting the screen communication, the necessity for redesigning the microworld also became apparent.

Foremost among the design issues were those associated with reading. One of these was related to the technical jargon of programming. In the first version of this microworld, I used expressions such as 'Execute the commands' which were meaningless to the children. In a later version, the expression 'Try doing each of the commands in the menu.' proved to be little better since children needed to know what 'commands' were. This was replaced by the simpler 'Do the lines in the menu.' The notion of readability is well recognised in the world of traditional media (Shuard and Rothery, 1984). The findings of this part of the research suggests that it may be equally cogent in the world of the new visual media.

In one sense, the revisions to the on-screen text were no different to those that might be made to a traditional textbook or worksheet as a result of trialling. But, in another sense, there were significant differences. As the design evolved, it became apparent that with Boxer, the programming vocabulary itself was open change. Replacing the term 'procedure' by 'Doit box' was more than just a simplification of language. The new terminology was supported by the visual representations on screen. Doit boxes appeared as rectangular boxes on the screen, Le., as physical containers for programs. Variables were Data boxes which contained numbers or other data, and the fact that lines or unnamed Doit boxes could be executed in place meant that terms such as "Do the lines." or "Do the boxes. " conveyed a visual meaning.

Perhaps not surprisingly, I found that, if children were presented with a lot of writing, they didn't read at all. On the other hand, I observed that children were inclined to read text in boxes which they had opened. So, one of my first design changes to the presentation was to put the text inside boxes. But then, like diSessa (1990), I found that children frequently ignored closed boxes, and so a further modification was made by naming the boxes. I found that boxes with names such as 'read-me-first' were likely to be opened and the text inside read

The notion of on-screen challenges supported by a graphic of the goal is one of the many features which Boxer offers over and above Logo. However, the choice of the challenge itself was mine. As the research progressed, I made several changes to the challenges replacing some altogether. These changes were made usually in response to children's difficulties or in order to better match the challenge to the key aim of that section of the micro-world. Of course, the whole notion of on-screen challenges is itself up for challenge. In this research, some of the children ignored them completely, preferring to pursue their own goals. Others skipped everything else and began with the challenges.

Programming ConstnIcts

The manner in which commands were presented also evolved in the course of the research. The difficulties which children incurred in attempting tasks which required a recognition of state-transparency motivated me to reflect on the usual presentation of commands. In my early attempts at writing programs in Boxer, I carried over practices from my experiences with Logo. Only later, as children's difficulties arose did I begin to realise that some of the problems might stem from my presentation. Initially the commands were presented on a single line. Later, these lines were placed inside Doit boxes. The Doit box fulfilled two roles here: emphasis and function. Later still, each command was placed on a separate line inside Doit boxes. This last version meant that children could run each command, e.g., 'rt 90' on its own from within the box, or they could place the cursor outside the box and run a combination of commands.

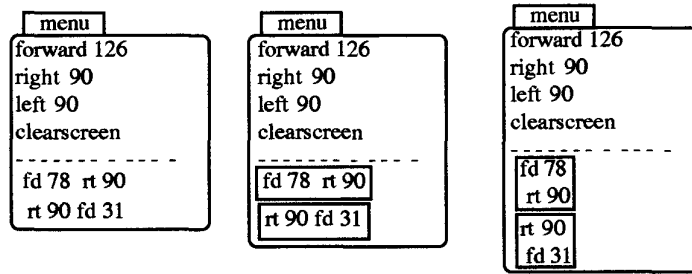
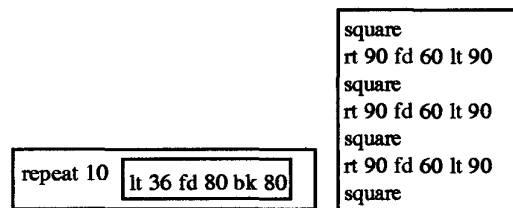


Figure 2: Changes to the presentation of commands

Throughout the whole of these microworlds, children were presented with models which made extensive use of structured programming. Moreover, I made design changes to the challenges to motivate them to consider modularity or the use of sub-procedures. The data from the latter parts of the research suggested that the children did adopt a modular form of programming, the modularity being related to functionality. Their use of modules to join separate procedures was inconsistent.

The use of un-named Doit boxes (Figure 3a) gave children a means of creating procedures that was half-way between programming by 'direct drive' and naming procedures (usually effected by moving into the editor). They put this to good use in creating super-procedures. It gave them a convenient way of stepping through more complex programs, and they often used this as a precursor to naming the super-procedure. It meant that debugging - usually seen as a post-programming repair activity - was built in to the program construction. Figure 3(b) shows an example of such a program written by Sara and Natalie which they later called 'malcom-x' (sic).



Figures 3a and 3b: Un-named Doit boxes, procedures and super-procedures

The initial findings from the early parts of 'First-Boxer' concerning the use of variables replicated earlier Logo research (Hoyle and Sutherland, 1989), e.g., children failing to recognise the significance of inputs. However, this raised an interesting question: was this phenomena being replicated in Boxer because I was writing programs in a Logo-like way? It is common practice in beginner's courses in Logo to write procedures without variables and at a later stage to introduce variables. In the course of designing the 'Second-Boxer' microworld, I found myself changing to a practice of always using variables, e.g., 'fd length' rather than 'fd 50' say.

Boxer permitted the creation of variable Data boxes (Figure 4) so this represented an intermediate stage between the two Logo-like ways of using inputs. Because the children had to directly change the contents of the Data box to effect a change on the screen, they were forced to consider the significance of inputs. Some evidence for this can be deduced from Joanna's remark about Data boxes: "They're called variables because you can vary the numbers in them."

On the other hand, repeatedly changing the contents of a local Data box led many children to ask if there was an 'easier way to do it'. Had I used either an external Data box or a general input line for 'square', then a running record in the form of an

unnamed Doit box could have been kept. Alternatively, the variable could have been changed under internal program control. At this stage of the research, I rejected each of these options as I thought that the physical act of directly changing the number in the Data box provided a concrete model of the notion of variable. But, the problem remained that there seemed to be no natural linkage¹ between internal Data boxes and those in the form of 'input x'.

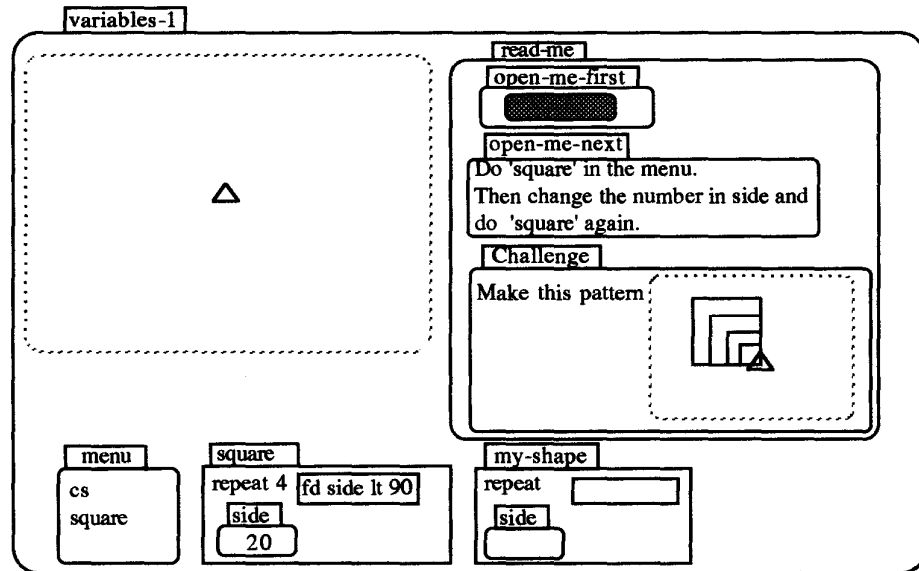


Figure 4: Variables-1

Conclusions

This paper has considered some of the design issues relating to communication, presentation and programming which arose through using the medium of Boxer to introduce itself. The changes which have been described here were made as part of a process of iterative design as the class teacher and I responded to the children's interactions. I have drawn a parallel with the practice of teacher's designing their own worksheets, and I suggest that Boxer opens up the possibility of teachers creating similar microworlds tuned to the needs of their pupils.

References

- diSessa (1990) Local Sciences: Viewing the Design of Human-Computer Systems as Cognitive Science. Boxer General Report No. G6. University of California, Berkeley.
- Hoyles and Noss (1987). Synthesising mathematical conceptions and their formalisations through the construction of a Logo-based school mathematics curriculum. *International Journal of Mathematics Education in Science and Technology*, 18 (4), 581 - 595.
- Hoyles, C. and Sutherland, R. (1989). *Logo Mathematics in the Classroom*. London: Routledge.
- Shuard, H. and Rothery, A. (1984). *Children Reading Mathematics*. London: John Murray.

¹ In current versions of Boxer, pressing 'Doit' on the line 'input side' will change the 'side' into a Data box with the name 'side', so this goes some way to making the link.