

**Visualising Number:  
A Study of Children's Developing Sense of Number  
in the Computational Medium of Boxer.**

**Declan O'Reilly**

**Thesis Submitted in fulfilment of the requirement for the  
Ph.D. degree of the University of London**

**Institute of Education, University of London  
October, 1995**



## Abstract

The aim of this study is to investigate children's developing sense of number in the computational medium of Boxer. Boxer's combination of graphical and symbolic elements afforded the opportunity for children to visualise numbers in an operational way while simultaneously offering insights into how this operational approach mediated their thinking. There were three inter-related aspects to the study, with visualisation being the common feature of all three.

- (i) How does the visual structure of Boxer influence students' (aged 9 - 11) ability to program?
- (ii) What interpretations do students place on a number system extended beyond the natural numbers and how did they choose to represent these?
- (iii) How can the learning environment of Boxer be exploited as a context for developing students' sense of number?

Following an exploratory study, pedagogical models for investigating issues (i) and (iii) were developed. For issue (i), Boxer was exploited as a means of introducing itself. This, in turn, meant documenting the issues involved in a process of iterative design. For issue (iii), a model of learning was developed which proposed that the children should construct their own microworlds. Following an off-computer investigation of issue (ii), the model was refined to that of children constructing *operational* computational objects, and the research aim broadened to include an investigation of how these objects mediated their expression of number. This part of the research consisted of a longitudinal study lasting two years. It entailed case studies with four pairs of children, while the rest of the class learned Boxer independently. None of the children had previous Boxer or Logo experience. The research setting was a normal classroom in an inner-London primary school. Data for issues (i) and (iii) was obtained by means of video recordings and annotated print-outs, while data for issue (ii) was obtained by written records and audio recordings.

Evidence from the research suggests that students' programming is significantly more structured in Boxer relative to Logo, and this structure is directly related to the visual nature of Boxer. Moreover, data from the number studies suggests that this visual structure was also instrumental in providing students with the means to connect number processes with number concepts, thus enabling them to engage with number ideas which might otherwise have been beyond their reach.

## **Acknowledgements**

I would like to express my thanks to my supervisor Dr. Richard Noss for all his help and advice throughout the writing of this thesis, and to Richard and Professor Celia Hoyles for their support and friendship during this period.

I would also like to express my thanks to the staff and pupils at Thornhill Primary School. In particular, I want to thank Sue Seifert the Headteacher for allowing me to work in her school, and to Denise Homer and Mel Ahmet for welcoming me to their classrooms. I learnt a lot of good things about teaching from my visits and it was not all about Boxer.

# Table of Contents

<b>Table of Contents .....</b>	<b>4</b>
<b>List of Tables .....</b>	<b>9</b>
<b>List of Figures .....</b>	<b>10</b>
<b>Chapter 1: Introduction .....</b>	<b>13</b>
1.1: Background to the Research .....	13
1.2: An Overview of the Research .....	16
1.2.1: Theoretical Perspective.....	17
1.2.2: Methodology .....	18
1.2.3: The Exploratory study.....	19
1.2.4: The Longitudinal study .....	19
1.2.5: Summary and Implications .....	21
<b>Chapter 2: Learning Mathematics and Learning about Number .....</b>	<b>22</b>
2.0: Overview .....	22
2.1: Learning Mathematics – A Theoretical Perspective.....	22
2.1.1: Epistemology .....	22
2.1.2: Constructivism .....	24
2.1.3: Contextualising Constructivism .....	26
2.1.4: Concepts.....	28
2.1.5: Intuitions .....	29
2.1.6: Constructing Learning Environments .....	30
2.2: Learning Number – A Research Perspective .....	32
2.2.1: Introduction.....	32
2.2.2: Number Sense .....	33
2.2.3: Directed Numbers .....	34
2.2.4: Place Value and Decimals .....	36
2.2.5: Conceptual and Procedural knowledge.....	39
2.3: Summary .....	42
<b>Chapter 3: Visualisation .....</b>	<b>45</b>
3.0: Overview .....	45
3.1: Visual Thinking .....	45
3.1.1: Visual Thinking in Mathematics .....	45

3.1.2: Visual Thinking in Computer Environments .....	48
3.2: Visual Representations.....	50
3.2.1: Visual Representations in Mathematics .....	50
3.2.2: Visual Representations Computer Environments .....	53
3.3: Programming and Boxer .....	56
3.3.1: Visualisation and Programming.....	56
3.3.2: Visualisation and Boxer.....	58
3.4: Summary .....	70
<b>Chapter 4: Research Methodology .....</b>	<b>72</b>
4.0: Introduction .....	72
4.1: Issues for Research Methodology .....	72
4.1.1: Theoretical Perspective.....	72
4.1.2: Pre-conceptions .....	74
4.1.3: Research Setting.....	74
4.1.4: Field Relations: .....	75
4.2: Data Collection .....	76
4.2.1: Research Timetable.....	77
4.2.2: The Exploratory Study .....	78
4.2.3: The Longitudinal Study .....	79
4.3: Case study students .....	82
4.4: Data Analysis .....	83
4.4.1: The Interviews .....	83
4.4.2: Triangulation.....	84
4.4.3: The process of Analysis .....	85
4.5: Summary .....	86
<b>Chapter 5: Exploring the Medium .....</b>	<b>88</b>
5.1: Introduction .....	88
5.2: Exploring Programming in Boxer.....	88
5.2.1: Working at a Syntactical Level.....	88
5.2.2: ‘Making Sense Of’ .....	90
5.2.3: Goal-directed Activity .....	93
5.3: Exploring Number in Boxer.....	95
5.3.1: Number Issues arising from the ‘line-numbers’ representation .....	95

5.3.2: Design Issues arising from the ‘line-numbers’ representation .....	100
5.4: Developing a Pedagogical Approach.....	101
5.5: Summary and Implications .....	102
5.5.1: Programming Issues.....	102
5.5.2: Number Issues.....	102
5.5.3: Pedagogical Approach .....	103
5.5.4: Implications for the remainder of the thesis .....	103
<b>Chapter 6: Exploiting the Medium.....</b>	<b>104</b>
6.1: Introduction and Overview .....	104
6.1.1: Introduction .....	104
6.1.2: Overview .....	104
6.2: Programming.....	106
6.2.1: Developing a sense of command .....	106
6.2.2: Procedure as an entity .....	109
6.2.3: Structured Programming .....	111
6.2.4: Variables .....	113
6.3: Number .....	117
6.3.1: Investigating Number in the context of Area.....	117
6.3.2: Investigating Number in the context of Co-ordinates .....	120
6.3.3: Investigating Number in the context of a screen Counter.....	123
6.3.4: Investigating Number in the context of List Processing .....	127
6.4: Design Issues .....	129
6.4.1: Presentation.....	129
6.4.2: Programming.....	132
6.5: Summary and Implications .....	136
6.5.1: Design Issues: .....	136
6.5.2: Programming Issues:.....	137
6.5.3: Number Issues:.....	138
6.5.4: Implications for the rest of the thesis .....	138
<b>Chapter 7: Students' Interpretations and Representations of Numbers .....</b>	<b>139</b>
7.1: Introduction and Overview .....	139
7.1.1: Introduction .....	139
7.1.2: Overview .....	140
7.2: Students' Interpretations of Number .....	140

7.2.1: Natural Numbers .....	140
7.2.2: Rational Numbers .....	141
7.2.2: Directed Numbers .....	143
7.3: Students' Representations of Numbers .....	146
7.3.1: Natural Numbers .....	146
7.3.2: Rational Numbers .....	149
7.3.3: Directed Numbers .....	151
7.4: Summary and Implications .....	152
7.4.1: Students' Interpretations of Numbers .....	152
7.4.2: Students' Representations of Numbers .....	153
7.4.3: Implications for the Remainder of the Thesis .....	154
<b>Chapter 8: The Number Line Study .....</b>	<b>156</b>
8.1: Introduction and Overview .....	156
8.1.1: Introduction .....	156
8.1.2: Overview .....	157
8.2: Constructing the Boxer Number Line .....	157
8.2.1: Constructing the Number line 1 .....	157
8.2.2: Constructing the Number line 2 .....	162
8.2.3: Linking Programming and Number .....	165
8.3: Interacting with the Boxer Number Line .....	165
8.3.1: Visualising Natural Numbers .....	166
8.3.2: Visualising Directed Numbers .....	169
8.3.3: Visualising Decimals .....	172
8.4: Summary and Implications .....	185
8.4.1: Programming Findings .....	186
8.4.2: Number Findings .....	187
8.4.3: Implications for the Remainder of the Thesis .....	189
<b>Chapter 9: The Function Machine Study .....</b>	<b>190</b>
9.1: Introduction and Overview .....	190
9.1.1: Introduction .....	190
9.1.2: Overview .....	191
9.2: Constructing the Boxer Function Machine .....	191
9.2.1: Students' Programming in Boxer .....	191
9.2.2: Linking Programming and Number .....	207
9.3: Interacting with the Boxer Function Machine .....	212

9.3.1: Directed Numbers .....	212
9.3.2: Decimals .....	220
9.4: Conclusions and Implications for Research .....	230
9.4.1: Programming Findings.....	230
9.4.2: Number Findings .....	234
9.4.3: Implications for the Thesis as a whole.....	237
<b>Chapter 10: Overall Conclusions and Implications of Research.....</b>	<b>238</b>
10.1: Programming.....	238
10.1.1: The General Programming Environment .....	238
10.1.2: Procedures .....	240
10.1.3: Variables .....	242
10.1.4: Discussion of Programming Issues .....	243
10.2: Number .....	245
10.2.1: The Boxer Number Line Object .....	245
10.2.2: The Boxer Function Machine Object.....	247
10.2.3: Discussion of Number Issues.....	250
10.3: Pedagogy .....	251
10.3.1: Developing a Pedagogy for Introductory Programming .....	251
10.3.2: Creating a Constructionist Pedagogy for Number .....	253
10.4: Implications for Research .....	254
<b>References .....</b>	<b>256</b>
<b>Appendices .....</b>	<b>275</b>
Appendix 1: The Line-Numbers Representation .....	275
Appendix 2: First Boxer .....	279
Appendix 3: Second Boxer .....	287
Appendix 4: Third Boxer .....	295
Appendix 5: Students' Representations .....	302

## List of Tables

4.1: Timetable for the Research .....	77
4.2: Timetable of Data Collection .....	77
4.3: Task Domains for Phase 1 of Exploratory Research .....	78
4.4: Video Record for Programming Phase .....	80
4.5: Year 6 Students interviewed on Number .....	81
4.6: Video Record for Number Phase .....	82
4.7: Composition of the Case Study Pairs (September 1991) .....	82
5.1: Results from the on-screen Comparison Questions .....	96
5.2: Leonie and Jermaine's Number Challenges .....	96
5.3: Ella and Steven's Number Challenges .....	97
5.4: James and Sian's Number Challenges .....	99
6.1: Programming and Number Organisation of First- to Third-Boxer .....	105
7.1: Students' choice of whole numbers.....	141
7.2: Students' choice of non-integers .....	142
7.3: Students' knowledge of Decimals .....	143
7.4: Students' choice of smallest number .....	144
7.5: Students' comparison of Directed Numbers.....	145
7.6: Students' interpretation of the Real Number continuum .....	146

## List of Figures

1.1: Theoretical areas for this research .....	13
1.2: A diagrammatic representation of the research .....	18
3.1: Placing rational numbers on a number line.....	52
3.2: Students' number line representations .....	52
3.3: Common representations of functions .....	53
3.4: Programming languages and generality .....	56
3.5: The boxes of Boxer .....	58
4.1: Theoretical social roles for fieldwork .....	76
4.2: A typical print-out with annotations .....	79
5.1: The 'saw-tooth' pattern.....	89
5.2: Finneas and Tariq's cup doit box and drawing .....	90
5.3: Tom and Tavis's dodecagon pattern .....	92
5.4: Anoushka and Natasha's house .....	94
5.5: The 'line-numbers' representation .....	95
5.6: Controlling the 'scale' and 'start-number' .....	100
6.1: First-Boxer .....	105
6.2: Shapes-1 partially opened .....	106
6.3: Shapes-1 fully opened showing the 'window' challenge .....	107
6.4: Shapes-2 fully opened showing the 'staircase' challenge.....	109
6.5: Shapes-5 fully opened showing the 'fence' challenge.....	112
6.6: Nico and Joanna's first attempts at the 'fence' challenge.....	112
6.7: Variables-1 .....	113
6.8: Variables-2 .....	115
6.9: Variables-3 .....	117
6.10: Variables-4 .....	118
6.11: Lines.....	120
6.12: Writing .....	121
6.13: Joanna and Faye's use of 'move' for writing.....	123
6.14: Build-it .....	124
6.15: Lots-at-once .....	128
6.16: Changes to the presentation mode .....	131
6.17: Changes to the presentation of commands.....	132
6.18: Combining 'repeat' with doit boxes .....	133
6.19: Two ways of doing the nested squares .....	134

6.20: The two versions of ‘counter’ .....	135
7.1: Students’ representations to compare whole numbers .....	147
7.2: Students’ representations to compare rational numbers .....	149
7.3: Students’ representations to compare directed numbers .....	151
8.1: The ‘T’ challenge .....	158
8.2: Jessie and Kathleen’s buggy procedure ‘green’ .....	159
8.3: Stage 1 of Kathleen and Jessie’s Number Line .....	160
8.4: Stage 1 of Liam and Jason’s Number Line .....	162
8.5: Liam and Jason’s ‘op-5’, ‘op-6’ and ‘ear’ procedures .....	163
8.6: Stage 2 of Liam and Jason’s Number Line .....	164
8.7: Expressing Patterns with Kathleen and Jessie’s Number Line .....	166
8.8: Visualising the 12 time table with Courtney and Laura’s Number Line .....	168
8.9: Visualising Directed Numbers with Kathleen and Jessie’s Number Line .....	170
8.10: Visualising Decimals with Courtney and Laura’s Number Line .....	173
8.11: Visualising the Number Continuum through Liam and Jason’s Number Line .....	175
8.12: Visualising the Number Continuum through Kathleen and Jessie’s Number Line .....	178
8.13: Expressing small decimal quantities with Joanna and Nico’s Number Line .....	180
8.14: Expressing small decimal quantities with Liam and Jason’s Number Line .....	181
8.15: Expressing small decimal quantities with Kathleen and Jessie’s Number Line .....	184
9.1: A typical cardboard model of a Function Machine .....	190
9.2: Laura and Courtney’s cardboard face function model for $2x - 1$ .....	193
9.3: Laura and Courtney’s ‘cryu’ procedure .....	193
9.4: Laura and Courtney’s ‘semi-modular’ programming .....	194
9.5: Liam and Jason’s cardboard function model for $7x - 11$ .....	195
9.6: Liam and Jason’s ‘make’ procedure and the first bug .....	195
9.7: Liam and Jason’s ‘live’ procedure and the second bug .....	196
9.8: Liam and Jason’s third bug .....	197
9.9: Naomi’s cardboard spider function model for $3.5x - 6$ .....	198
9.10: Naomi’s Spider Function Machine .....	199
9.11: Modularity and Complexity in Naomi’s Spider Machine.....	200
9.12: Nico and Joanna’s cardboard function model for $7x - 5$ .....	201
9.13: Joanna and Nico’s Function Machine under construction .....	202

9.14: Rosie and Chloe's cardboard function model for $x/2 + 3$ .....	203
9.15: Rosie and Chloe's named sub-procedures 'in' and 'out' inside an unnamed super-procedure .....	204
9.16: Sara and Natalie's cardboard Nintendo Gameboy function model for $8x + 94$ .....	205
9.17: Sara and Natalie's named sub-procedure 'rect' inside a named super-procedure 'gameboy' .....	205
9.18: Sara and Natalie's developing use of structured programming .....	206
9.19: Changing the Domain and Range for Liam and Jason's Function Machine.....	208
9.20: Articulating informal algebraic expressions through Rosie and Chloe's function doit boxes.....	210
9.21: Expressing functions through informal algebra with Liam and Jason's k-5 doit box .....	211
9.22: Approaching Directed numbers through subtraction with Naomi's Function Machine .....	213
9.23: Approaching Directed Numbers through functions with Joanna and Nico's Function Machine .....	215
9.24: Approaching Directed Numbers through inverse functions with Sara and Natalie's Function Machine .....	217
9.25: Approaching Directed Numbers through functions with Liam and Jason's Function Machine.....	218
9.26: Approaching Decimals through functions with Liam and Jason's Function Machine .....	220
9.27: Approaching Decimals through division with Naomi's Function Machine.....	223
9.28: Relating Decimals through inverse functions with Sara and Natalie's Function Machine .....	226
9.29: Relating Decimals through inverse functions with Liam and Jason's Function Machine .....	228
9.30: Ways of constructing super-procedures in Boxer .....	233

# Chapter 1: Introduction

Arithmetic, even its elementary aspects, deals with very important mathematical concepts. To establish the link between ordinary mathematical situations and the relevant mathematical concepts is probably the most challenging question in mathematical education. (Vergnaud, 1978, p. 344)

## 1.1: Background to the Research

This thesis has been undertaken to investigate children's developing sense of number in the computational medium of Boxer. It asserts that the critical factor in developing students' understandings is their construction of meaning for number, and it presents evidence to show how programming in Boxer offers a powerful way for constructing such meaning.

It draws together three distinct knowledge domains: theory and research relating the learning of programming with the learning of mathematics in general; theory and research concerned with the learning of number; and theory and research linking each of these to visualisation (Figure 1.1). These domains do not stand alone but are located in more general theories of learning.

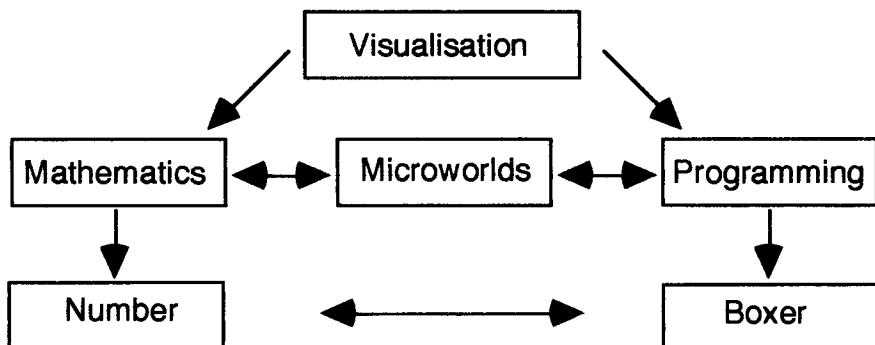


Figure 1.1: Theoretical areas for this research

The study reported here took place within a broadly constructivist paradigm taking as one of its main strands the theoretical framework established by Logo researchers in mathematics over the past decade or more (Papert, 1980; Noss, 1985; Sutherland, 1988; Hoyles and Sutherland, 1989; Hoyles and Noss, 1992a), particularly in relation to the construction of mathematical environments. It then supplements this strand by drawing on the emerging literature relating to constructionism (Harel, 1991; Papert, 1993b). The other source of theoretical insights stems from the increasingly sophisticated literature (Hiebert and Wearne, 1986; Wearne, 1990; Dubinsky, 1991;

Sfard, 1991; Gray and Tall, 1994) pertaining to conceptual and procedural knowledge in mathematics. These two strands are bound together by the twin threads of visualisation and ‘connectedness’, particularly in relation to number.

In the first case, advances in technology have led to a renaissance in visualisation in mathematics (Zimmermann and Cunningham, 1993), in mathematics education (Bishop, 1989), and in the design of computer interfaces themselves (Shneiderman, 1983). In the second case, the notion of ‘connectedness’ arises both with regards to reevaluating the concrete/abstract dichotomy (Wilensky, 1993; Ackermann, 1993), but also in the number sense literature where meaning is seen to be attained through the establishment of multiple links between different nodes of knowledge (Sowder, 1989). Hoyles and Noss (in press) put forward a case for learning as the construction of a web of connections and have located as a central question for learning and teaching: “... how to assist students in building on their own meanings in ways which are concurrent with mathematical ways of knowing by interactions within computational microworlds.” (p. 5.1) an expression which articulates my Boxer endeavours with respect to number.

The findings of major research studies in this country such as the C.S.M.S. (Hart et al., 1981), the A.P.U. (Foxman et al, 1985), and the N.C.T.M. (Carpenter et al., 1981) in the United States leave little doubt as to the difficulties which many students experience in dealing with number, difficulties which persist well into the secondary school. Moreover, these difficulties cannot be simply reduced to the mathematical properties of the task.

Problems that have the same logical structure and that call for the same mathematical operations can be handled quite differently by young children. Changes in the content or context of a task can have a marked effect on task difficulty. (Bergerons and Herscovics, 1990; Küchemann, 1981). Both the CSMS research (Hart, op cit.) and the APU research (Foxman, op cit.) found that changes in the numerical elements, for example, from positive to negative numbers or from whole numbers to fractions or decimals, had a pronounced affect on facility levels. These changes, Küchemann (op cit.) notes, can in each case be described as a shift towards elements that are less ‘intuitible’ or ‘concrete’.

On the other hand, Johnson (1989) observes that for at least 20 years, the training of primary school teachers has been dominated by the belief that, if children are within

the Piagetian concrete level stage, then: "... their learning should be through 'concrete' experiences." (p. 3), the implicit assumption being that these concrete experiences will form the basis for more formal learning of the same ideas. However, he reports the results of a research project (Children's Mathematical Frameworks) which appears to show that, even when teaching is designed to foster the transition between learning mathematics from practical or concrete work to using formalised versions of the same mathematical ideas, children do not connect the two sets of experiences. This is expressed in stark terms by one of the participants, Hart (1987), who writes: "The results showed overwhelmingly that the recipients of a series of practical experiences, leading up to a formalisation did not appreciate that the latter was a synthesis of the former." (p. 410).

Such insights serve to alert us to the dangers of an over-simplification of the link between practical activities and formal understanding, and points to the need to create contexts which help students develop meanings for mathematical entities. As Vergnaud (1988a) argues:

It is our job to organise didactic situations and to experiment with them, both with the short-term objective of enabling students to develop new competencies and conceptions for immediate use and with the long-term perspective of offering a basis for concepts that will be essential a few years later. For instance, it may be interesting to introduce negative and positive numbers in elementary school not only to help students to deal immediately with the combination of directed transformations but also to prevent them from sticking to a narrow conception of numbers as positive quantities, a conception that later will be an obstacle to the learning of algebra. (Vergnaud, 1988a, p. 143)

In this research, pedagogical situations were organised in which primary school students (aged 9 – 11) encountered directed numbers and decimals in the course of interacting with computational objects in Boxer. The concern was not to measure or improve facility levels, but to investigate how this interaction might affect students' construction of meaning.

diSessa (1987c) emphasises that: "... computers have a character that better adapts them to some areas than others – in particular, to areas where the essential structure is visual, geometric and dynamic." (p. 74). The challenge for this study was to see if the traditionally non-graphical area of number might also be approached in a visual way.

These twin programming and number aims can be summarised as follows:

- to investigate how students' programmed in Boxer;
- to investigate how Boxer mediated students' sense of number.

However, as the research progressed, other subsidiary questions arose. Following an exploratory study in which students with a Logo background were observed, a key question which emerged for the longitudinal study was how students with no programming experience might be introduced to Boxer? This in turn evolved into a question of whether the medium could be exploited to introduce itself, i.e., of whether the visual facilities of Boxer could be utilised to construct an introductory course in programming, and if so, what issues might this involve? These then became subsidiary aims for the programming phase of the study.

As a result of a series of off-computer interviews with Boxer and non-Boxer students, the notion of using Boxer to construct meaning for number was refined to one of using Boxer to construct *operational* meanings for number, i.e., it was hypothesised that students would acquire meaning for directed numbers and decimals if they saw these numbers as a *consequence* of operations. This posed several intermediate problems. Firstly, there was the problem of identifying which Boxer objects might best lend themselves to an operational approach and secondly, there was the question of the programming demands that the construction of such objects set for the students. The number research question now became refined to one of investigating how different Boxer objects mediated students' expression of number.

At this point, it is perhaps worth clarifying my use of the term 'mediate'. Following Vygotsky (1978), the view taken here is that the tools at our disposal mediate the ideas we can express. Thus expressing a number sequence with a spreadsheet, for example, will summon up different kinds of knowledge than that for pen-and-paper. The task for this research was to investigate on the one hand how Boxer's visual constructs mediated students' learning of programming, and on the other how the objects of their programming mediated their sense of number.

## 1.2: An Overview of the Research

This section presents an overview of the chapters which form this thesis, the intention being to orientate the reader, and to relate the separate components of the thesis to the main aims of the study as outlined above.

### **1.2.1: Theoretical Perspective**

In chapter 2, ‘Learning Mathematics and Learning Number’, theoretical issues that relate to learning mathematics are discussed in general. Issues of particular relevance to this research were those relating to: the origins and growth of students’ knowledge; the need for the child to be an active participant in her or his learning; how such learning might be affected by contexts; and how situations might be created to build bridges from intuitive to formal knowledge. Thus, this section ranges over issues of epistemology, constructivism, the nature of mathematical concepts and the effects of mathematical intuition.

It then considers how the insights derived from these theoretical perspectives might affect practice, and it finds exemplification in the accumulated body of Logo research in ways that are especially apt for this study. This section addresses how Piagetian learning might be mediated by programming, and it draws on Papert’s notion of ‘constructionism’ as a means of analysis. It re-examines the relationship between the concrete and the abstract and sees considerable merit in the idea of ‘connectedness’.

In the second half of chapter 2, the discussion focuses on number. It considers the meaning of the expression ‘number sense’, how this relates to ‘connectedness’, and what these might mean for students’ developing sense of number. This is then followed by a review of the research literature relating to those aspects of Decimals and Directed Numbers which the students met in this study. This review reveals two facts: students encounter considerable difficulties with number concepts, and there is an accumulating literature suggesting why this occurs and giving a broad framework for its resolution. The chapter ends by examining what seems to be the writings of most relevance to this research, namely that pertaining to the distinction between procedural and conceptual knowledge.

Chapter 3 ‘Visualisation, Mathematics and Boxer’ continues the theoretical discussion, with the emphasis on visualisation, its relationship to the learning of mathematics and how the computer affects this relationship. In particular, the literature concerning the changing visual nature of the human-computer interface is discussed in terms of what it means to program. In the second part of the chapter, the discussion of programming issues narrows the focus in on Boxer. The point is made that Boxer is more than a programming language: it is an ‘integrated computational medium’ designed with explicit pedagogical intentions. What this means and what

these intentions are then considered in terms of Boxer's design and implementation. Moving from a general discussion of structural and functional modelling, the chapter outlines the underlying models of Boxer's design and the two central principles ('naive reality' and 'the 'spatial metaphor') which guided that design. Finally, the chapter reviews currently available Boxer research.

### **1.2.2: Methodology**

Chapter 4 'Methodology' details the research methodology employed: the nature of the data collected, the methodology for collection, and describes the classroom setting and the role of the researcher. It also records how these changed during the course of the research both as a consequence of the findings of the exploratory study and in order to adapt to the modified classroom conditions from the exploratory to the longitudinal stage. The empirical stages of the research form the subjects of chapters 5 to 9, as depicted in Figure 1.2.

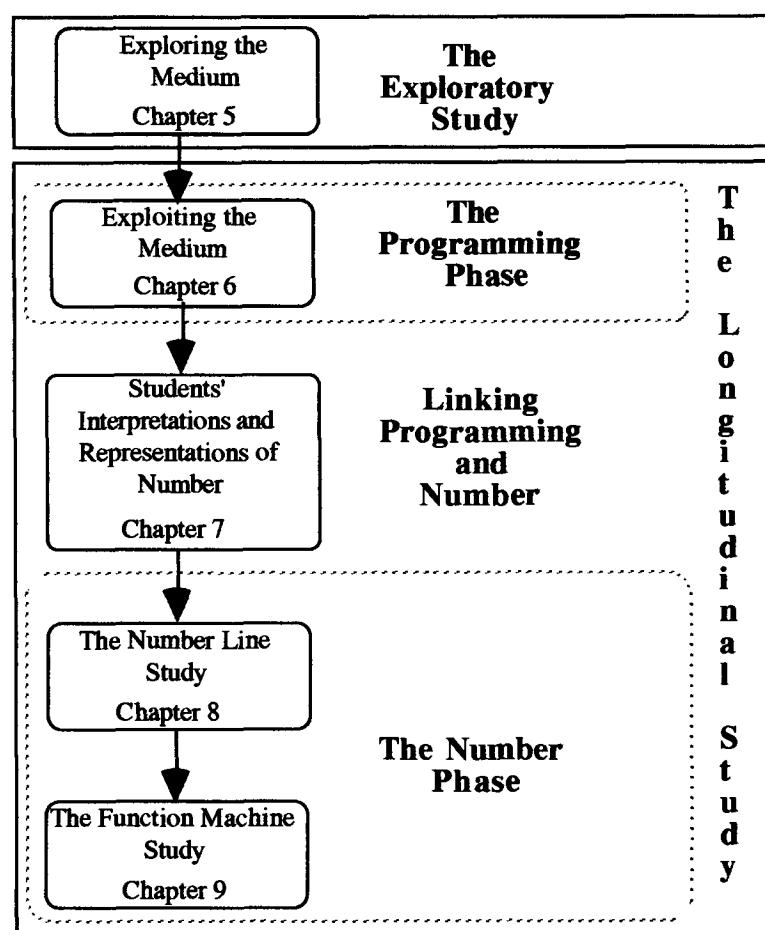


Figure 1.2: A diagrammatic representation of the research

Writing on the role of intervention in learning, Hoyles and Noss (1992b, p. 32) describe “a critical tension between the need to allow students room for manoeuvre and exploration on the one hand, and our own intentions and structuring on the other”. The methodology of this thesis can be seen as attempting to resolve this tension over the course of longitudinal study. From an initial structured setting which aimed to establish a baseline of programming knowledge, it devolved increasing degrees of freedom to the learner as the research progressed to and through the number phase. Each of these phases were characterised by their own distinctive methodology: the programming phase by the exploitation of the medium to introduce itself; the interview phase by its aim of integrating students’ knowledge into the research design, and the number phase by its constructionist nature.

### **1.2.3: The Exploratory study**

Chapter 5 ‘Exploring the medium’ describes the exploratory stage of the research whose major aim was to identify and refine issues for the longitudinal study. It gives an account of how students who had previous Logo experience adapted their knowledge in learning to program in Boxer, and it relates how a vector representation was used to explore students’ understanding of number. In terms of the thesis as a whole, it shows how ethical considerations combined with the aim of establishing a foundation of programming knowledge over the long term led to the iterative design of the programming phase of longitudinal study (chapter 6).

### **1.2.4: The Longitudinal study**

#### **1.2.4.1: the programming phase**

Chapter 6 ‘Exploiting the medium’ describes the programming phase of the longitudinal study. It gives an account of how the aims outlined above were put into practice by using the medium to introduce itself. Data was obtained by closely observing four representative pairs of case study students using three micro-worlds: ‘First-Boxer’, Second-Boxer’ and ‘Third-Boxer’. This data was combined with that obtained from the teacher’s observations of the rest of the class, who were independently using these same micro-worlds, to effect the process called ‘iterative design’. The chapter documents this ‘iterative design’ process with the dual aim of showing how the students learnt to program in Boxer and to highlight the issues involved in using the medium to introduce itself.

#### 1.2.4.2: the semi-structured interviews

Chapter 7 ‘Students’ Interpretations and Representations of Number’ gives an account of the semi-structured interviews carried out with students from the Boxer group and those from a parallel year six class. Just as the exploratory study had refined issues for the programming phase, this study helped to refine issues for the number phase. It clarified the ways in which these students thought of number in general, and of directed numbers and decimals in particular. In particular, it highlighted their action-based meaning for fractions and led directly to the ‘operational’ approach adopted in the two number studies. Moreover, it began to establish some baseline knowledge of the kinds of representations that they used and found useful for depicting these numbers. By comparing the Boxer students with those from a parallel class, it accentuated differences which pointed to medium and thus helped to shape the next phase of the research.

#### 1.2.4.3: The Number Line Study

Chapter 8 ‘The Number Line Study’ recounts the first of two number studies whose design was influenced by the earlier exploratory stage (chapter 5), the number investigations of the programming phase (chapter 6), and the results of the semi-structured interviews (chapter 7). It describes how the case-study students utilised their programming knowledge to construct an ‘operational’ number line representation in Boxer, and relates how the insight gained through this construction influenced their subsequent interactions with it.

The study shows that, within the medium of the Boxer students displayed a facility for manipulating directed numbers and decimals which was at odds with research with older students in conventional media – differences all the more remarkable since no direct teaching was involved. Chapter 8 acknowledges the situated nature of such learning, but it gives specific instances of the ways that the number line device mediated students’ expression of number, and asserts that the evidence points to the students’ own constructs of the Boxer number line as being a powerful mathematical object through which they constructed meaning for number.

#### 1.2.4.3: The Function Machine Study

Chapter 9 ‘The Function Machine Study’ describes the second of the two number studies. It gives an account of how in replicating a classroom project (making cardboard models of function machines) in Boxer, the students utilised and extended

their programming knowledge. It shows how the insight gained through this construction influenced their subsequent interactions with it. It then analyses the ways this ‘operational’ Boxer object mediated their expression of number.

In doing so, it both reinforces and differs from the findings reported in chapter 8. In reinforces them to the extend that it demonstrates again how students developed an extraordinary fluency in dealing with directed numbers and decimals within the medium. It differs in the manner that the function machine mediated their expression of number. Whereas the Boxer number line seemed to dispose students to attend to what might be categorised as geometrical aspects of number, such as those relating to the number continuum, the function machine seemed to push them towards algebraic aspects. The function machine context entailed approaching number through algebra (in opposition to the usual school approach) and student activities and descriptions reflect that approach. They sought and adapted functions to map one set of numbers to another, and they constructed meaning in terms of operations or inverse operations.

### **1.2.5: Summary and Implications**

Finally, Chapter 10 summarises the research. The results are discussed from the perspective of the theoretical models developed in earlier chapters. The conclusions, including implications for future research, are considered both in relation to the design of the longitudinal study and in terms of the interactional nature of the computational objects through which the students developed their sense of number.

## **Chapter 2: Learning Mathematics and Learning about Number**

### **2.0: Overview**

This chapter is about learning mathematics generally, about learning number in particular, and about how learning environments can be created through programming. It draws on theory and research relating to each of these. From a theoretical perspective, it discusses issues relating epistemology, constructivism, social constructivism, concept analysis and intuition. From a research perspective, it considers the body of knowledge that has been built up by Logo researchers over the years, and explicates Papert's notion of constructionism and concreteness. It then narrows the focus to number, discusses the literature relating to 'number sense' and recounts the research findings relating to the learning of decimals and directed numbers. Finally, it considers the relationship between conceptual and procedural knowledge.

### **2.1: Learning Mathematics – A Theoretical Perspective**

Theory is essential to organise our knowledge on mathematics education in coherent systems of description and in powerful concepts. In order to understand how individuals develop or fail to develop knowledge, it is essential to discuss alternative interpretations of constructivism and other theoretical frameworks. (Vergnaud, 1987a p. 43)

#### **2.1.1: Epistemology**

Like many theses in mathematics, this one begins with Piaget. The Piaget of stage theory is, of course, well known. But this Piaget is, in Papert's words: "essentially conservative, almost reactionary, in emphasising what children cannot do." He, by contrast, identifies: "more revolutionary Piaget, one whose epistemological ideas might expand the known bounds of the human mind." (1980, p. 157). Other writers have uncovered similar ideas in Piaget, but use different phrases to express what they have found, e.g., 'revolutionary constructivist epistemology' (Cobb, 1986) 'dialectical constructivist' (Higginson, 1980; Sinclair, 1987)

Higginson (op cit.) identifies two fundamental characteristics of the theory. Firstly, the child actively constructs her/his own reality. Secondly, this construction process is dialectical. The critical analysis of existing structures yields new structures which incorporate the old ones. The image of the child as a dialectical constructivist, he

suggests, is compatible with views of ‘the child as scientist’, and he draws on the work of Ginsburg (1977) as evidence of the ingenuity of children’s attempts to make sense of certain concepts and processes.

Traditionally, epistemology has been used to describe the study of the conditions of validity of knowledge. Papert (1980) writes that Piaget’s epistemology is concerned not with validity of knowledge but with its origin and growth. When Piaget writes about the developing child, he is in fact writing about the development of knowledge. This statement allows a contrast to be drawn between epistemological and psychological ways of understanding learning. In a purely psychological perspective, the focus is on the laws that govern the learner rather than on what is being learned. “For Piaget, the separation between the learning process and what is being learned is a mistake: the way to understand how children learn number is through a deeper understanding of what number is” (Papert, op cit. p. 158).

Vergnaud (1990) distinguishes three interweaving strands of knowledge: epistemology of mathematics, epistemology of psychology, and epistemology of mathematics education. There are, he suggests, several approaches to the epistemology of mathematics. Firstly, there is the spontaneous reflection by mathematicians themselves on the nature of their knowledge and on the nature of the invention and discovery process. (cf. Poincaré, 1913, 1920; Hadamard, 1949). Secondly, there is the historical approach. Its aim is to understand the scientific and social environment in which new mathematical concepts and techniques have emerged and developed (cf. Davies and Hersh, 1981). Thirdly, there is the mathematical-philosophical debate on the foundations of mathematics, e.g., logicism, intuitionism, formalism, constructivism etc.

The epistemology of psychology, he writes, has different roots. The first of these relates to the nature of objects that psychology is concerned with as a science: behaviour, consciousness, unconsciousness, elementary modules of action, perception, and memory, complex organisations of behaviour, complex representations? Secondly, the kinds of models that can be used to give an account of psychological phenomena: associations, inborn mechanisms, general biological processes like adaption, neuro-physiological models, linguistics, computer science models. Finally, he says, the epistemology of mathematics education inherits questions from both fields and new ones because: “Mathematics education takes place in a certain society, a certain institution, a certain classroom, with such different aims

as the education of future mathematicians and the education of rank-and-file citizens.” (Vergnaud et al., 1990, p. 15).

### 2.1.2: Constructivism

Piaget, according to Vergnaud (1987a) was, in his day, the most systematic theorist of constructivism. But, what exactly is constructivism, and what are its implications for the teaching and learning of mathematics? One answer to the first of these two questions is given by Novak (1986) who sets constructivism in opposition to behaviourism and positivism:

As long as we were saddled with behaviourism and positivism, ... there was no need to discuss the *concept* of number, line, point, slope, and so on. And few worried where the concepts came from, who *created* them, and how concept meanings evolve over time, or change with context. But cognitive psychologies and constructive epistemologies tell us that we *must* concern ourselves with the latter issues. (Novak, 1986, p. 181)

A more definitive approach is taken by Kilpatrick (1987) who writes that the constructivist view involves two principles: (1) Knowledge is actively constructed by the cognizing subject, not passively received from the environment. (2) Coming to know is an adaptive process that organises one's experiential world; it does not discover an independent pre-existing world outside of the knower. He observes that the first principle is much more acceptable than the second by many who call themselves constructivists. He writes (cf. Davis and Mason, 1986) that this second principle separates 'simple' constructivism from 'radical' constructivism.

Von Glaserfeld (1987) refers to this latter principle as 'viability', i.e., a concept works or is viable to the extent that it does what we need it to do. Confrey (1995) points out that radical constructivism is itself subject to this principle, which she refers to as 'recursive fidelity'. Cobb (1986) uses the expressions 'empiricist-oriented constructivists' and 'radical constructivists' to express the same divide. Vergnaud (1987a) sees two independent ideas in the second principle: the adaptive process, and the radical constructivist's denial of an independent pre-existing world for the other. He does not see the necessity to accept the latter of these in order to be a radical constructivist.

One dilemma that would appear to arise from the adoption of a radical constructivist position is how does one person know if their construction of reality is the same as another's? Confrey (1985) stresses that, for the constructivist, there can be no *certain* knowledge that another person's construct is identical to one's own, but she also

points to the role of others in the construction process. This corresponds with Piaget's view, according to Sinclair (1987) who says that the quality of knowledge gathered through our actions on the real world is determined by the ways in which reality reacts to our interventions and by its correspondence to the knowledge that other people have constructed. She quotes (op cit., p. 32) Piaget as saying "Objective knowledge is only attained when it has been discussed and checked by others". This seems to accord well with Vergnaud's position who writes:

I see constructivism as the best way to consider the process of appropriation by which a student makes mathematics his own knowledge. Rather than a pure and lonely construction, the learning of mathematics is for me the difficult appropriation of a social knowledge.  
(Vergnaud, 1987a, p. 53)

Kilpatrick (op cit.) cites Von Glaserfeld (1983) as setting out the pedagogical implications of constructivism as follows:

- (i) teaching is sharply distinguished from training (the former is aimed at understanding, the latter at repetition);
- (ii) processes inferred as inside the student's head become more interesting than overt behaviour;
- (iii) linguistic communication becomes a process for guiding a student's learning, not a process for transferring knowledge;
- (iv) students' deviations from the teacher's expectations become means for understanding their efforts to understand;
- (v) teaching interviews become attempts not only to infer cognitive structures but also to modify them.

But, he points out that such consequences fit other philosophical positions as well. Creating a dichotomy between teaching and learning ignores contexts in which the two terms are used interchangeably. The contrast between internal and overt processes may simply be one of focus. The behaviourist teacher attempts to see *in* the overt action; the constructivist teacher attempts to see *through* it. The ensuing teaching actions, however, may not be any different. The metaphor of knowledge being constructed as opposed to knowledge being transferred is, he argues, only a metaphor. For Kilpatrick, the attention paid by constructivists to student errors is one of its most attractive aspects, but again he observes that "One need not be a constructivist to be interested in, or to study, the errors that students make that are contrary to the teacher's expectations" (op cit., p. 16). Finally, he notes that history, not logical necessity, links the teaching interview to constructivism.

Constructivists, according to Kilpatrick, need to clarify and develop their ontological commitments. They need to think through and spell out more clearly than they have done thus far the relationships between constructivism and mathematics as a discipline and mathematics as a school subject. Finally, he says that constructivism needs to become connected to reality: the reality of everyday scientific activity, mathematical investigation, and classroom practice.

Confrey (1994) acknowledges limitations in constructivism, amongst which she lists its assumption that: “a theory of learning provides an adequate theory of instruction.” (p. 7) and its heavy dependence on the autonomy of the individual. Simon (1995) contends that constructivism: “does not tell us how to teach; that is it does not stipulate a particular model” (p. 114), an interpretation accepted by Steffe and D’Ambrosia (1995) who reply: “We expect different mathematics educators working in a constructivist epistemology to generate different specimens of teaching” (p. 147). Dubinsky (1991), while not using the term ‘constructivism,’ nevertheless emphasises the importance of the constructive process. Moreover, he goes some way in setting out an instructional approach and indicates the importance of computers in fostering ‘reflective abstraction’, a concept which he writes was: “... introduced by Piaget to describe the construction of logico-mathematical structures by an individual during the course of cognitive development.” (p. 95).

### **2.1.3: Contextualising Constructivism**

One theorist who attempts to connect constructivism with pedagogy is Bruner. Bruner’s theory of cognitive development is marked by his attention to representation, language and the influence of the social setting. He writes that the child first knows her world principally through the action she directs towards it. Later, she is able to form images of objects in their absence and this helps to free her from purely action-guided representation. Finally, she can convert both action and image into symbols such as words. Bruner (1968) calls these three modes of representation: enactive representation, iconic representation and symbolic representation. They appear in that order and all three remain with us throughout our lives.

In his view, cognition is influenced as much from the outside in (culture) as from the inside out (mental structures). Bruner (1980) believes that cognitive growth is dependent on our abilities to link ourselves to what he calls: “culturally transmitted amplifiers of motor, sensory and ratiocinative capacities” (p. 326). These ‘amplifiers’ can range from a primitive cutting-tool to a modern microscope, and include theory,

myths and language. In contrast to Piaget, he stresses the importance of language. He argues (1980) that: “language provides a means not only for representing experience but also for transforming it.” (p. 330).

Bruner, again departs from Piaget, in that he is concerned with pedagogical issues. The teacher, he argues, should not slavishly follow the course of cognitive growth. Rather, he says that the aim should be that of tempting the child into the next stage of development. We can see in Bruner’s theories the unmistakable influence of Vygotsky with respect to culture, with respect to language and with respect to the ‘zone of proximal development’.

According to Confrey (1995), the attention to the role of tools in mediating knowledge is a clear focus of Vygotsky theory. This focus, she writes: “led him to propose that language is itself a form of psychological tool” (p. 41). Forman and Cazden (1985) stress that Vygotsky is not simply claiming that social interaction leads to the development of the child’s abilities. “Rather, he is saying that the very means (especially speech) used in social interaction are taken over by the individual child and internalised.” (p. 323).

They write that, whereas for Piaget social experiences derive their importance from their potential for cognitive conflict, Vygotsky’s ideas offer insights in situations where mutual guidance and support are evident. They say that he acknowledged the discrepancy between solitary and social problem solving when he developed his notion of the zone of proximal development. He defined this zone as “the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers” (1978, p. 86). According to Vygotsky, development proceeds when inter-psychological regulation is transformed into intra-psychological regulation.

The notion of a zone of proximal development, he writes, “enables us to propound a new formula, namely that the only ‘good learning’ is that which is advance of development” (op cit., p. 89). Bruner (1985) writes of an apparent contradiction between these two proposals for on the one hand the zone of proximal development has to do with “consciousness and control”, but to imply that “good learning” is in advance of development seems to negate this. He resolves it with the following formulation:

If the child is enabled to advance by being under the tutelage of an adult or a more competent peer, then the tutor or aiding peer serves the learner as a vicarious form of consciousness until such time as the learner is able to master his own action through his own consciousness and control. When the child achieves that conscious control over a new function or conceptual system, it is then able to use it as a tool. Up to that point, the tutor in effect performs the critical function of "scaffolding" the learning task to make it possible for the child, in Vygotsky's word, to internalise external knowledge and convert it into a tool for conscious control. (Bruner, 1985, p. 24)

Vygotskian perspectives have served as a starting point for many studies. Some have focused on the teacher-pupil relationship whilst others have chosen the pupil-pupil as their focus. Jaworski (1990) has studied the "acts a teacher undertakes to facilitate a pupil's mathematical progress" (p. 94) while Forman and Cazden (1985) have explored instead the value of peer-interaction. The notion of 'scaffolding' has also been extended from that of adults and capable peers to include the computer (Hoyle and Noss 1993c; Noss and Hoyle, 1987).

However, Hoyle and Noss (1993c) explicate differences between their formulation of scaffolding and that of Bruner and his colleagues. Firstly, they place the computer in the role of the human tutor, yet seek to do so in such a way that the learner retains control. Secondly, they observe that the traditional view of scaffolding is 'universal' in that it is independent of context. Their notion, by contrast, is domain contingent. Finally, they write that their notion is concerned with scaffolding of *concepts* rather than skills.

#### 2.1.4: Concepts

Another theorist who attempts to connect the constructivism of Piaget with the realities of pedagogy is Vergnaud. He (1987a) equates Piaget's 'normative facts' (Sinclair, 1987) with his own 'invariants', or 'theorems-in-action'. But, whereas Piaget analysed tasks in terms of their logical structure, Vergnaud begins with an analysis of concepts. He writes (1984, p. 19) that a concept can be considered as a triplet of three sets:  $C = \{S, I, \zeta\}$  where:

- S: the set of situations that make the concept meaningful;
- I: the set of invariants (or 'theorems-in-action') that constitute the concept and insure its stability under certain classes of transformations and variations;
- $\zeta$ : the set of symbolic representations that may represent the concept and the situations in which it is involved.

In other words, he calls S the referent, I the signified and  $\zeta$  the signifier.

He argues that the core of cognition lies in the operational invariants that students gradually discover or appropriate. He proposes that a large part of our knowledge is only implicit: we take up information with the help of invariants without expressing or even being able to express these invariants. This is especially visible in students' mathematical behaviour. Very often, they choose the right thing to do without being able to articulate the reasons for it. The cognitive analysis of such behaviours very often reveals the existence of powerful implicit mathematical concepts and theorems, which Vergnaud calls concepts-in-action or theorems-in-action.

The situation, he says, is made more complex by the fact that not only does a concept refer to a variety of situations, but each single situation cannot usually be analysed with only one concept. He uses the expression 'conceptual-field' to describe such entities, and adds (1983a) that conceptual fields develop over a long period of time. Elsewhere, Vergnaud (1988a) writes that the main reason that mathematics researchers should study such comprehensive systems as conceptual fields is to understand the filiations and jumps in students' acquisition of knowledge; while theorems-in-action are a way for us to analyse students' intuitive strategies and to help them in transforming intuitive knowledge into explicit knowledge.

### 2.1.5: Intuitions

This latter appeal begs the question: "What exactly do we mean by intuitive knowledge?" Like visualisation (chapter 3), it is very often taken as a primitive concept, i.e., one which requires no further explanation. But, as Fischbein (1978) writes, it is not enough that we use the term 'intuition' to describes certain forms of thinking. Rather we must characterise and analyse the nature of intuitions:

Intuition has an important impact on mathematical thinking, whether positive or negative. So it cannot be neglected by mathematical education. Both in trying to use its beneficial qualities and in striving to eliminate intuition from pupils' mathematical reasoning, we have to take into account its possible effects. (Fischbein, 1978, p. 151)

He writes that great mathematicians such as Felix Klein and Henry Poincaré accepted that intuitions played an active role in mathematical invention and mathematical education. But he contends that it is philosophers rather than psychologists who have seen in intuition a fundamental source of knowledge (op cit.). He aims to remedy this deficit and to reveal the implications of his analysis for mathematical education. He describes different characteristics of intuitions:

- (i) They seem to be self-evident. No arguments appear to be necessary to convince us of their truth, and this explains their great impact on the course of thinking. In educational terms, students may not see the necessity to prove what is obvious.
- (ii) As a consequence of their self-evidence, intuitions exert a coercive effect on the processes of conjecturing, explaining, and interpreting various facts. Several educational results of this can be seen. Firstly, the intuition may be so strong that it dominates any alternative explanation. Secondly, two opposite interpretations may co-exist for a long time without the subject being aware of the contradiction.
- (iii) Intuitions are marked by their capacity to extrapolate. Whereas logical thinking makes predictions on the basis of rigorously and explicitly established arguments, intuition represents a leap which cannot be completely justified.
- (iv) Intuitive judgements are inclined to be global. They are also frequently expressed by a visual symbolisation.

He proposes a classification of intuitions and analyses certain mechanisms like over-confidence, premature closure (a halt to the search for new information), and the primacy effect (giving privilege to first interpretations). Fischbein gives examples of correct mathematical intuitions (transitivity of equivalence) and over-generalised intuitions (multiplication makes larger, division makes smaller). He also gives examples of primary intuitions (the more space, the more things), and secondary or constructed intuitions (conservation of quantity). He observes that intuitions change in the course of cognitive development: the intuition of subtraction is different for a 5-year-old and a 16-year-old, as is the intuition of similarity. The feeling of immediacy that we have with our intuitions is therefore misleading and needs to be analysed as the result of accommodation to experience and as a change in implicit knowledge.

Fischbein's analysis of intuitions, Vergnaud's analysis of concepts, and Bruner's theory of cognitive development with its constituent themes of representation, language and social interaction together form a backdrop to the work of mathematics educators who have explored the creation of learning environments through Logo.

### **2.1.6: Constructing Learning Environments**

There is now a considerable literature on the use of Logo and the learning of mathematics (Hoyles and Noss, 1987a, Hoyles and Sutherland, 1989; Hoyles and Noss, 1992a). Likewise, a sophisticated pedagogy has been built up concerning the nature of mathematical microworlds (Hoyles, 1987; Hoyles and Noss, 1987a, 1992b, 1993a). Summarising the literature, Hoyles and Noss (1987a) distinguish two distinct strands in the Logo research: those that have taken a content-related approach and those that have adopted a heuristic approach, neither of which have been particularly

successful. Elsewhere (1993b), they conclude that Logo is best viewed as a ‘conceptual framework’ with which children learn mathematics. The critical question, they write, is: “whether and how the computer might offer support structures which fulfil a scaffolding role” (p. 418). Their tentative conclusion is that the computer can indeed fulfil this function by providing an environment through which intuitions can be made explicit, tested, and changed if necessary, thus making robust mathematical ideas built on intuitions.

They consider such environments in relation to the role of concrete materials in the learning of mathematics, and observe (cf. Resnick, 1981; Hart, 1981) that the connection in the child’s mind between the concrete and the abstract may not be at all clear. The problem, they say, is to lay connections between the concrete and the abstract in such a way that the child creates meaningful links between the two. The way to do this, they argue, “lies not in the use of concrete activity and then its formalisation, but in the continued use of concrete embodiments – moving backwards and forwards between the concrete and the abstract, between the general and the particular, and between pupil’s conceptions and their formalisations” (1987, p. 586).

Papert gives theoretical underpinnings to this conclusion through his notion of constructionism:

Constructionism ... shares constructivism’s connotation of learning as “building knowledge structures” irrespective of the circumstances of the learning. It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity ... (Papert, 1993b, p. 1)

Papert puts constructionism in opposition to instructionism, but then writes that this simply looks like a split about strategies for knowing: two ways of thinking about the transmission of knowledge. The differences, he argues, are more profound. They are not merely about the acquisition of knowledge: they are about the nature of knowledge and the nature of knowing, and these are epistemological issues. Traditional epistemology, he writes, gives a privileged position to knowledge that is abstract, impersonal, and detached from the knower and treats other forms of knowledge as inferior. “But, feminist scholars have argued that many women prefer working with more personal, less detached knowledge and do so very successfully.” (op cit., p. 10). Elsewhere, Turkle and Papert (1993a) analyse the abstract-concrete dichotomy in epistemology and argue the case for a ‘revaluation of the concrete’.

Wilensky (1993) takes this as his starting point. He writes that concreteness is not a property of an object but rather a property of a person's relationship to an object. The more connections we make between an object and other objects, the more concrete it becomes for us. Once we see this, he continues, it is not difficult to go further and see that any object/concept can become concrete for someone.

The pivotal point on which the determination of concreteness turns is not some intensive examination of the object, but rather an examination of the modes of interaction and the models which the person uses to understand the object. This view will lead us to allow objects not mediated by the senses, objects which are usually considered abstract – such as mathematical objects – to be concrete; provided that we have multiple modes of engagement with them and a sufficiently rich collection of models to represent them.

(Wilensky, 1993, p. 199)

This notion of concreteness as a relational property, he writes, turns the standard Piagetian view on its head. "The lesson we take from Piaget is not that the child develops by leaving behind the primitive world of concrete operations and leaping into the enlightened world of adult formal operations. Rather ... the child concretises his or her world by engaging in multiple and complex relationships with it" (p. 199).

How this might occur in practice is the theme of Harel's (1991) research. She describes how fourth grade children designed a piece of software with the aim of teaching fractions. In doing so, they were learning about Logo and learning about fractions. She writes that whereas conventional approaches to teaching fractions have focused on manipulative aids, i.e., helping the child to move from the concrete to the abstract, her research design provided students with an environment in which to explore relationships between different representational modes.

## **2.2: Learning Number – A Research Perspective**

To understand how a child learns number, we have to study number. And we have to study number in a particular way: We have to study the structure of number, a mathematically serious undertaking. (Papert, 1980, p. 158)

### **2.2.1: Introduction**

The term 'number' has a simplicity about it which can be misleading. On the one hand, it would seem that everyone knows about number. On the other hand, 'number' encompasses such a wide field that researchers have generally felt obliged to restrict their inquiries to one aspect of it. Hence, they have tended to examine children's understanding of Decimals as distinct from children's understanding of Fractions, as distinct from their understanding of Directed Numbers and so on. This choice of categories reflects, to a large extent, the manner in which schools deal with these

areas, with the hierarchy: Whole Numbers --> Fractions --> Decimals --> Directed Numbers reflecting the order in which children are taught these topics.

Moreover, there is a depth to the acquisition of number concepts that further defines research in this field. Researchers such as Piaget (1965) have alerted us to the complexity involved in the acquisition of number concepts by very young children; for example, in discriminating between ordinal and cardinal aspects. Post-Piagetian researchers (Gelman and Gallistel, 1978; Donaldson, 1987, Ackermann, 1993) have shown us that the situation is even more complex than Piaget thought, by demonstrating the powerful influences of language and context. In research with primary school children aged 9 - 11, a review of the number literature pertaining to early childhood was deemed inappropriate. On the other hand, my desire to explore children's 'developing sense of number' in a Boxer environment clearly pointed to the need to consider context. As Vergnaud (1978) points out:

It is hopeless to try to understand the acquisition of arithmetical concepts and to propose better conditions for the child to understand them if one does not make the effort to analyse the tasks through which these concepts are made meaningful and useful to the child.

(Vergnaud, 1978, p. 347)

A third approach to number is that usually adopted by researchers interested in topics such as estimation and is labelled 'number sense'. Thus, Sowder (1989) writes that "... number sense refers to a well-organised conceptual network that enables one to relate number and number operations, and to solve number problems in flexible and creative ways." (p. 381). These three categories are not intended to be exhaustive of all number research. Rather, they identify three trends which intersect in the present research. The review of the literature will reflect that intersection.

### **2.2.2: Number Sense**

Sowder (op cit.) draws attention to Marshall (1989), who suggests that we think of number sense in terms of the 'connectedness' of a vast network of nodes of mathematical knowledge. For example, students may have knowledge about whole number properties and about addition of whole numbers, but these topics may be encoded in memory without being connected. Thinking of number sense as the connectedness of number knowledge implies that number sense is not a body of knowledge to be taught, but rather that teaching should focus on the development of links between nodes of number knowledge.

McIntosh, Reys and Reys (1992) trace the origins of the phrase ‘number sense’ to a desire to replace the word ‘numeracy’ by one which does not have its abstract ring or its association with a conservative and sterile view of mathematical needs. But, they remark that the phrase itself is open to different interpretations. For them:

Number sense refers to a person’s general understanding of number and operations along with the ability and inclination to use this understanding in flexible ways to make mathematical judgements and to develop useful strategies for handling numbers and operations.  
(McIntosh, Reys and Reys, 1992, p. 3)

Other characteristics which they say typify ‘number sense’ include:

- the theme of learning mathematics as a sense-making activity;
- its acquisition as a gradual, evolutionary process, beginning long before the start of formal schooling;
- growing older does not necessarily ensure either its development or utilisation;
- it is triggered by the context in which the mathematics evolves;
- it is highly personalised and is related to what ideas about numbers have been established and also how these ideas were established

However, they acknowledge that merely listing the components of number sense or the attributes of the students who possess it is insufficient in itself. They agree with Greeno (1991) who suggests that ‘number sense’ is a term that requires theoretical analysis, rather than a definition. Often it seems that ‘number sense’ is apparent by its absence, and, as the following section indicates, its absence is frequently marked by the lack of connectedness between number concepts and number algorithms.

### 2.2.3: Directed Numbers

In this country, children are not usually taught directed numbers until the secondary school. One reason for this seems to be the general recognition of the difficulties which children have in this area (Bell, Costello and Küchemann, 1983). In this, they are in good company, for, as the history of mathematics shows (Vergnaud, 1990; Hefendehl-Hebeker, 1991), negative numbers provided intellectual obstacles which many great mathematicians failed to overcome. However, acknowledging difficulty need not imply avoidance or even postponement:

Studies on the acquisition of negative numbers reveal long-lasting obstacles in 15- to 16-year old students, especially when they have to multiply a negative by a negative or when they come to a negative solution. Paradoxically, there are some aspects of negative numbers that can easily be understood by primary school students: a negative transformation (decrease, loss, consumption, backward displacement), a negative relationship (less than, debt), or even a negative abscissa (below the ground floor).  
(Vergnaud, 1990, p. 25)

With respect to operations on directed numbers, both the CSMS and APU studies found that addition was considerably easier than subtraction. The approach used in each was different with only the former employing explicit models, and so a direct comparison is not possible. In the CSMS research (Hart et al., op cit.), the facilities for these operations were typically of the order 80 - 90% and 36 - 77% respectively, whilst in the APU research (Foxman et al., op cit.), they were of the order 65% and 50% respectively. Küchemann (1981) reports on the misapplication of the rule 'two minuses make a plus in the case of subtraction but not multiplication. Arcavi and Bruckheimer (1981), in a comparison of different strategies for introducing the latter, found surprisingly little differences between them and the rote method which they characterised by reproducing the old adage: "Minus time minus equals plus - the reason for this we need not discuss" (p. 31).

Küchemann (op cit.) suggests that there are two main approaches to the teaching of directed numbers; an abstract axiomatic approach and one in which various models are used to elucidate the operation being used. The first of these, he argues, has advantages in terms of mathematical consistency but he doubts whether such an approach would be accessible to all but a minority of secondary school children. This conclusion, he claims, is supported by the work of Collis (1975, 1978) who argues that the ability to work consistently within an abstract mathematical system requires formal operational thought.

On the other hand, he says that the concrete approach suffers the disadvantage that the models are not consistent across operations. Thus shifts along the number line appears to succeed for addition but not for subtraction. He suggests abandoning the number line model completely in favour of one in which the positive numbers are seen in terms of cancelling out the negative numbers. Rowland (1982) reports some success using such a model.

Küchemann's argument raises an interesting question: namely, whether a model needs be consistent over operations, or indeed different numbers? In a different context, diSessa (1986b) addresses this question. He uses the term 'patchwork' to describe the manner in which we employ different models as explanatory devices to account for different aspects of complex phenomena. The trade-off, he suggests, is between doing this or using ever more complicated – if consistent – models.

#### **2.2.4: Place Value and Decimals**

It is above all clear that the learning of whole numbers and decimals is not just a matter of recalling some place-names and a few rules of computation, as it often appears from the textbooks. Indeed the children who did rely blindly on rules more often misapplied them than not. Instead, it involves internalising a whole chain of relationships and connections, some within place-structure itself (e.g., 0.9 is equivalent to 0.90), some linking to other concepts like those of fractions (e.g., the notion of one hundredth and its relationship to one tenth), some visual correspondences and some connecting to applications in the ‘real’ world.

(Brown, 1981, p. 64)

In the APU surveys (Foxman et al., 1985), the most difficult whole number items for 11 year-olds were those concerned with the relationship between the column values of a number. For example, children presented with the numbers 107, 71, 7, 710 and asked to ‘Put a ring round the number in which the 7 stands for 7 tens’, only 69% chose the correct answer (*ibid.*, p. 42). The facility level for the item ‘The number which is one less than 2010 is’ was similar: 71% (*ibid.*, p. 42). Another item which asked ‘Which number is ten times 1000?’ received a correct response of only 62% (*ibid.*, P. 44). The attainment band analyses suggested that the bottom band, in particular, at age 11 had considerable difficulty with items which relate to place value.

This finding is corroborated by Brown (1981) who reports that, the weakest children when interviewed showed only a superficial knowledge of place value. For example, 8030 was read as ‘eight hundred and thirty’. Conversely, the correct oral answer was given in response to ‘add ten to 3597’ but recorded as 367. She noted that items containing numbers over a thousand exposed uncertainties even among children of average attainment.

Carpenter et al. (1981) note that in order to construct an understanding of decimals, students needed to have an appreciation of place value and a firm grasp of the relationship between decimals and common fractions. Their results suggest an absence of each. They say that, since the 9-year-olds in their research had not yet studied decimals, it was not surprising that they ignored the decimal and interpreted the number as a whole number. They also noted that fewer than 1 percent of the 9-year-olds gave the correct decimal equivalent for a fraction expressed in tenths. 55 percent of 13-year-olds could undertake this latter task, but “as many chose 0.5 for the decimal equivalent of  $\frac{1}{5}$  as chose the correct response 0.2.” (p. 35).

In the APU surveys, items were used which involved the direct examination of the value of particular columns, decimal equivalents of fractions which were tenths or hundredths, or ordering of decimals. Facilities were less than half those for

of places to the right of the decimal point." (p. 33). Certainly, in the present research, children did consider 0.25 to be bigger than 0.3, but since they also thought 0.5 and 0.50 to be different numbers, Grossman's explanation may be inadequate in itself.

Nesher (*op cit.*) observes that when asked to write the rational number  $3/4$  in decimal form, some of the students wrote 3.4, 0.3 or 0.34. She goes on to claim that children are consistent in the rules that they apply, each of which emerges from previous knowledge: the DPI error from the domain of whole numbers and the LS error from the domain of rational numbers. This theory of a 'retrieval of incorrect frames' (Davies, 1984) is supported by Zazkis and Khoury's (1993) study of pre-service teachers' understanding of rational numbers in bases other than ten. They record similar examples of students over-generalising and ignoring the 'separating point'. How schooling influences such errors is considered by Resnick et al. (1989) who conclude that: "different curriculum sequences produce different patterns of rule invention." (p. 25).

On the other hand, Fischbein et al. (1985) in their research with Italian children (grades 5, 7 and 9) appear to confirm the universality of certain strategies. They hypothesise that: "Each fundamental operation of arithmetic generally remains linked to an implicit, unconscious, and primitive model." (p. 4), and write that: "A next step in this line of research would be to attempt to provide learners with efficient mental strategies that would enable them to control the impact of these primitive models." (p. 16). The widely reported 'multiplication makes it bigger' and 'division makes it smaller' syndrome is a case in point (Brown, 1981; Foxman et al., 1985). They argue that the misconceptions follow from the model of multiplication as repeated addition. Greer and Mangan (1985) report changing the multiplier from an integer to a decimal less than one lowered performance by about 46% while Graeber and Tirosh (1990) report that such beliefs are held by students as early as the fourth and fifth grade.

Vance (1986) identifies two characteristics of the rational numbers which makes them more difficult to learn than whole numbers: "(1) each rational number has many different names, and (2) the rational numbers are dense – between any two numbers is another number." (p. 56). In his research with grade 6 and 7 classes, many students expressed frustration in trying to find a number between 0.23 and 0.24 or between  $3/5$  and  $4/5$ , and he writes that the answers  $0.23\frac{1}{2}$  and  $3\frac{1}{2}/5$  were often tentatively proposed. He concludes: "If students are to use decimals and fractions in problem solving and estimation situations, they must be able to visualize the numbers

comparable whole number tasks. Typical results, at age eleven, for giving decimal equivalents of 3/100 or 4/100 were 40 to 45%. Tasks which involved ordering or selecting the largest or smallest of decimals less than one saw an even more dramatic fall in facility to 20% (*ibid.*, p. 48).

When asked to "Put these decimals: 0.07, 0.23, 0.1 in order of size, smallest first.", the correct response was given by 23% of the children. On the other hand, if each item contained the same number of decimal places, the facility level increased substantially. Asked to "Put these decimals: 0.3, 0.1, 0.7, 0.6 in order of size, smallest first.", the responses increased to 75%. The large discrepancy in the success rates of these two items was attributed in Primary Report No 2 to the greater complexity of the first item in that only one decimal place was involved in the latter. The 1985 Report recognised two distinctive errors which occurred: 'Decimal Point Ignored (DPI)' and 'After the decimal point, largest is smallest (LS).' The APU authors made the observation with respect to these errors: "The DPI and LS rules are empirical descriptions which fit the results so far obtained; they do not indicate how individual pupils who make these errors think about decimals." (Foxman et al., p. 59). Brown (*op cit.*) gave one pupil's explanation of the first of these errors (in comparing 0.75 and 0.8): "This is nothing before and seventy-five; this is nothing before and just eight" (p. 52).

Brown writes that the major difficulty which the weaker children seemed to have was in understanding that the figures after the point indicated that part of the number which was less than one unit. "Instead, children seemed to think that the figures after the point represented a 'different' number which also had tens, units etc." (*op cit.*, p. 51), a finding corroborated by Carpenter et al (1981). In the present study, this way of thinking was often revealed by the way the students articulated decimals, e.g., 'Two point forty three.' instead of say: 'Two point four three.' However, (Klein, 1990) suggests that the latter way of verbalising decimals may not be much better. He argues that a number such as 2.43 should be expressed instead by: "the more meaningful two and forty-three hundredths" (p. 31).

With respect to the LS error, Nesher (1985) reports: "In this case children claim that 3.2 is bigger than 3.47, because the shorter number has tenths and the longer number has hundredths and tenths are bigger than hundredths." (p. 335). This hypothesis is supported by Grossman (1983) who attributes its cause to students "not having learned to annex enough zeros to each decimal to give the decimals the same number

represented by the symbols and understand the elements of the representational systems.” (op cit., p. 58)

Students’ difficulties with the number continuum have been reported elsewhere. On being asked: “How many different numbers can you get which lie between 0.41 and 0.42?”, (Hart, 1981) reports that the most popular answer was 8, 9 or 10. Similarly, when children (aged 14 – 15) were asked how many fractions came between  $\frac{1}{4}$  and  $\frac{1}{2}$ , over 50% gave a number less than 20, whilst 30% gave the answer ‘one’. This latter answer, she attributes to their thinking in terms of unit fractions so that the only possible answer then is  $\frac{1}{3}$ . (Hart, 1981). A follow-up study on fractions with 13 to 14 year olds (Kerslake, 1986) found a related result. When asked to plot three numbers between 1 and 2 on a number line, ten children (out of a total of 59) said that there were none, and a further nine children made no response. She records two limitations in students’ conceptions of fractions: “... that children find it difficult to think of a fraction as a number, and that they do not make a connection between  $\frac{a}{b}$  and  $a \div b$ .” (p. 98).

Markovits and Sowder (1991), in their research with grade 6 students, record similar misconceptions. They conclude: “It would seem that instruction that focuses on the meaning of fractions and of decimals form a basis on which to build understanding of the relationship.” (p. 11). Goldenberg’s (1991) experiment would seem to support this conclusion. He relates how students’ initial uncertainty about the existence of decimal quantities could be transformed by a carefully thought-out problem.

### **2.2.5: Conceptual and Procedural knowledge**

Some writers attribute a lack of understanding to the nature of the relationship between conceptual and algorithmic approaches to number. Zawojewski (1983) makes a plea for helping children understand the concept of decimals prior to their use in computation. Owen (1987) notes how, pupils who could successfully compute with decimals, were unable to rationalise the processes. Wearne (1990) observes that students compensate for their lack of conceptual knowledge by resorting to memorising rules and procedures. Thipkong and Davis (1991) found that such conceptual difficulties were not confined to children. Many of the pre-service teachers in their study exhibited similar problems. They note that such teachers: “should first understand the concepts of decimal numbers. If they do not, decimals are likely to be taught as algorithms to be memorised rather than concepts to be understood” (p. 98).

Hiebert and Wearne (1986) observe that extending concepts of whole numbers into referents that are appropriate for decimal fraction symbols is a delicate process. Students must recognise the features of whole numbers that are similar to decimal fractions and those that are unique to whole numbers. As they say: “With some important exceptions, it is the conceptual features that generalise to decimals and the syntactic features that do not . The trick for students is to sort out which are which.” (p. 204). Wearne and Hiebert (1988) conclude that: “it is difficult for semantic processes to penetrate routinised procedures.” (p. 380). Thompson (1992), who used a computer-based microworld in his approach to decimals, relates how difficult it was to graft conceptual onto procedural thinking. He writes: “If students memorise a procedure meaninglessly, it is extremely difficult to get them to change it, even with extended, meaningful remediation.” (p. 144)

The distinction between different forms of knowledge in mathematics is an old one, as Hiebert and Lefevre (1986) make clear, frequently being expressed in the dichotomy of skills versus understanding. However, they say, there are differences between current discussions and those of the past. Present writers are more concerned with description as opposed to prescription. Secondly, the focus is no longer solely on school-based knowledge. Lastly, more attention is now being paid to relationships between concepts and procedures. This is a theme picked up by Silver (1986) who contends that when knowledge is used dynamically, it is the relationships *among*, and not the distinctions *between*, concepts and procedures that become of primary importance.

Steinbring (1988) addresses this relationship in the context of arithmetic. He states that the four fundamental operations of arithmetic with decimals are presented in textbooks as direct ‘generalisations’ of the operations with natural numbers. “The only thing to do – this is often stated in textbooks – is to define exactly the correct use of the decimal point in the representation of the new numbers; apart from that, the pupil may calculate in the same way as with natural numbers” (p. 25). Furthermore, he asserts, that in the opinion of many mathematics teachers, the process of developing mathematical knowledge rests upon well-defined operations and algorithmic procedures and upon clear-cut formal definitions of concepts. He contrasts this to the research findings which indicate the important conceptual difficulties and misconceptions that pupils have regarding decimals. He raises the question (p. 30): “Does conceptual knowledge take priority over procedural

knowledge in the frame of learning and understanding, or is procedural knowledge a ‘precursor’ of conceptual knowledge?”

Carpenter et al (1981) argue that conceptual understanding should precede computational proficiency: “We should be spending more time having children become familiar with decimals, their meanings and uses, before rushing directly to decimal computation.” (p. 37). This issue is also addressed by Nesher (1986a) who questions whether teaching for understanding necessarily improves algorithmic performance. It is possible, she argues, that the opposite is true, i.e., “knowing the algorithms and procedures contributes to the student’s understanding” (p. 8). Sfard (1991, p. 10) postulates: “... maybe even a certain degree in mastery in performing these processes, should be viewed as a *basis* for understanding such concepts rather than as its *outcome*, and she reproduces the following quote from Kilpatrick (1988):

Why is it that so many intelligent, well-trained, well-intentioned teachers put such a premium on developing students’ skill in the routines of arithmetic and algebra despite decades of advice to the contrary from so-called experts? What is it that teachers know that others do not? (Sfard, 1991, p. 10 cf. Kilpatrick, 1988)

Noting the traditional dichotomy between conceptual and procedural knowledge, Sfard (op cit.) stresses instead the dual nature of mathematical conceptions. Treating mathematical notions as if they refer to some abstract objects, and characterising these notions as processes, she argues, are two sides of the one coin. She refers to these as *structural and operational* conceptions respectively.

Seeing a mathematical entity as an object means being capable of referring to it as if it was a real thing – a static structure, existing somewhere in space and time. It also means to recognise the idea ‘at a glance’ and to manipulate it as a whole, without going into details.

...  
In contrast, interpreting a notion as a process implies regarding it as a potential rather than actual entity, which comes into existence upon request in a sequence of actions.

(Sfard, 1991, p. 4)

She argues that ability of seeing function or a number both as a process and as an object is indispensable for a deep understanding of mathematics, and she draws on both historical and psychological evidence to support her thesis.

The history of numbers, she writes, can be seen as a long chain of transitions from operational to structural conceptions: “... again and again, processes performed on already accepted abstract objects have been converted into compact wholes, or reified (p. 14). Psychologically, her contention is that the acquisition of mathematical ideas follow a similar path: from an initial operational view of the notion, which in turn go

through the stages of ‘interiorisation, condensation, and reification’, to a structural view. However, she notes a ‘vicious circle of reification’ in that a person must be quite skilful at performing algorithms in order to attain a good idea of the ‘objects’ involved, yet one must also have a conception of these objects since without them the processes are meaningless. She asks: “How can anything be a process and an object at the same time?” (1991, p. 151).

The answer, according to Gray and Tall (1994), lies in the inherent ambiguity of mathematical symbolism. Mathematics is pervaded, they write, by the use of the same notation to represent both a process and the product of that process. They put forward the notion of *procept* to refer to the amalgam of concept and process represented by the same symbol, and they write of proceptual thinking as being characterised by: “... the ability to compress stages of symbol manipulation to the point where the symbols are viewed as objects that can be decomposed and recomposed in flexible ways.” (p. 132)

This proceptual encapsulation occurs at various stages throughout mathematics: repeated counting becomes addition, repeated addition becomes multiplication, and so on. Similar problems, they write, characterise other parts of the curriculum. For example, pupils perceiving the symbolism  $2 + 3x$  as representing only a procedure that cannot be carried out because the value of  $x$  is not known. However, they also argue that the use of the computer to carry out the process enables the learner to concentrate on the product and significantly improves the learning experience. My contention in this thesis is that this depends very much on the nature of the interaction between the computer and the user, i. e., on the means by which the medium enables learners to express mathematical ideas.

### 2.3: Summary

This chapter began by stating the case for theory. It will end by putting the case for theory in context. Theories are no use, according to Balacheff (1990), unless they are related to precise problems. “Theories are tools to solve problems or to clarify them and to improve their formulation. Inversely, to solve research problems very often leads to the improvement of theories” (p. 258). My research problem was to investigate children’s developing sense of number in the computational environment of Boxer. Section 2.1 discussed the theoretical and research foundations for

constructing such learning environments, while section 2.2 considered the problems surrounding the learning of number.

In section 2.1, different notions of constructivism were elucidated, and their implications for teaching and learning considered. The notion of the child as a constructor of her or his own reality encapsulates an appealing rhetoric, but interpreting this doctrine in terms of classroom practice poses problems. Indeed, Papert (1980) doubts whether the kind of Piagetian learning that the constructivists have in mind could have taken place in the past. In his view, it is only now with the advent of the computer that it is possible to do so.

Moving beyond the cognitive, this section also attempted to take account of the setting in which learning takes place. The Vygotskian idea of a zone of proximal development, Bruner's idea of a tool-rich culture, together with the notion of scaffolding were discussed, along with Vergnaud's appeal to examine concepts in context and Fischbein's analysis of intuitions. I have then drawn on the accumulated body of Logo research to elucidate how learning environments which embrace these notions can be constructed.

My research aim involved students constructing their own number microworlds. However, they first needed to learn to program, and this became a subsidiary but significant part of the research. It involved the notion of designing microworlds in which the medium would introduce itself. These microworlds were subject to a process of 'iterative design' in which changes to the microworlds were being made as they were being used by myself and the class teacher independently. It therefore embraced a constructionist approach from the perspective of the teacher, from my position as a researcher and from the students' role as builders of their own learning environments.

In section 2.2, I examined the research findings pertaining to certain aspects of number as they affected this study. The literature leaves little doubt that students here and elsewhere have persistent difficulties with concepts of number. There was less agreement about why such misunderstandings might arise or what might be done to overcome them. Some identified causes in early intuitions, others in over-generalising from previous learning, while still others located the source of problems in the relationship between conceptual and procedural learning.

The ‘number sense’ literature threw up the idea of ‘connected knowing’ which tied in well with Papert and Wilensky’s revaluation of the concrete. Sfard’s account of how the processes of mathematics at one level become reified into objects on which processes act at another level offered another insight. Her argument for the need of seeing a number both as a process and as an object was fleshed out by Gray and Tall’s notion of a procept, in which the ambiguity of symbolism played a prominent part. Their suggestion that the computer offers a means for promoting proceptual thinking gives broad support to the research design employed in this study.

Further support can be found in Harel’s (1991) research, which in many respects sets a precedent for this thesis. Its theoretical foundations in constructivism and Logo research are similar to that outlined above. It was also concerned with number and, in it, students designed their own software. But, there were important differences too. Her students designed instructional programs with Logo ostensibly to be used by other students. The students in my study designed Boxer mathematical objects (a number line and a function machine) for their own use. Whereas, she focused on fractions, these formed only a peripheral part of this thesis. My research was concerned with the rational numbers in its broader sense, i.e., decimals and directed numbers. In this respect, the two studies complement each other and can be connected through the two notions expressed in the symbol  $a/b$ . On the one hand, it denotes the result of an operation ( $a$  lots of  $b$ ), but on the other, it expresses an operation to be performed  $a \div b$ . Harel’s study was concerned with the former, my research with the latter. Finally, visualisation was central to each of our studies.

In chapter 3, the role of visualisation in mathematics is analysed from historical as well as a psychological perspective. Some of the issues raised in the present chapter will be revisited, but this time in relation to visual representations and their role in promoting or indeed interfering with understanding. Changes to computer hardware and software have led to very significant changes to the part that visual thinking can be called upon in mathematical thinking. The relationship between mathematics and programming has always been a close one, but the cognitive obstacles needed to become accomplished at the latter meant that few but the experts could make the link. The visual nature of a programming language like Boxer now opens up the possibility that we are reaching the stage where such power can be put in the hands of children. Chapter 3 discusses those visual qualities, relates them to existing research and the aims of the present study.

# Chapter 3: Visualisation

If present trends are any indication, it seems that mathematics will evolve in a direction which will make visualisation even more important in the future than it is now. At the same time, the evolution of technology will make even more powerful visualisation tools available. (Zimmermann and Cunningham, 1991, p. 7).

## 3.0: Overview

This chapter reviews the literature on visualisation in relation to mathematics, and in relation to computers and programming. It identifies two common themes in the mathematical literature: visual thinking and visual representations, and it considers the impact of computers on each of these. It then focuses in on the influence of visualisation on programming before introducing the computational medium of Boxer.

This chapter complements chapter 2 in that issues of epistemology, constructivism, the contextualisation of concepts and the role of intuition are re-visited, but the extra dimension of visualisation gives a new perspective to that discussion. In introducing Boxer, my intention is to go beyond simply describing the software to consider how the beliefs of its designers concerning these issues influenced its design, and to show how these design features of Boxer make it a suitable computational medium for the kind of visually-mediated learning proposed in this research.

## 3.1: Visual Thinking

### 3.1.1: Visual Thinking in Mathematics

The literature on visualisation is replete with terms such as ‘spatial ability’, ‘visual thinking’, ‘spatial intuition’, ‘visual understanding’ and so on. Often, these descriptions are contrasted to analytical or symbolic modes of thought as though the two forms of representation were discrete entities. The dangers inherent in such an approach have been clearly set out by Zimmerman and Cunningham (1991) who write:

Mathematical visualisation is not ‘math appreciation through pictures’. The intuition which mathematical visualisation seeks is not a vague kind of intuition, a superficial substitute for understanding, but the kind of intuition which penetrates to the heart of an idea. It gives depth and meaning to understanding, serves as a reliable guide to problem solving, and inspires creative discoveries. To achieve this kind of understanding, visualisation cannot be isolated from the rest of mathematics. Visual thinking and graphical representation must be linked to other modes of mathematical thinking and other forms of representation. One must learn how ideas can be represented symbolically, numerically, and graphically, and to move back and forth among these modes. (1991, p. 3).

The assertion that visualisation will necessarily aid understanding is not universally agreed. Eisenberg and Dreyfus (1986), for example, write:

It is certainly true that spatial visualization plays a role in mathematical thinking in general, and in concept acquisition and problem-solving in particular. But the influence of spatial visualization abilities may well be highly overrated. (1986, p. 153)

They observe that for many students visual thinking and analytical thinking seem to be dichotomous modes, and advance the hypothesis that analytical thinking overrides visual thinking even in experts. More recently, Campbell, Collis and Watson (1995) write that iconic support may be particularly helpful when conceptual knowledge is weak, but "... tends to disappear once the problem is well understood and the mathematical procedures for its solution well practised." (p. 192).

Some writers have examined the practice of mathematics to account for the role of visual thinking. Hadamard (1954), for example, studied the working practices of eminent mathematicians to determine the part played by visual thinking in their work. He reported that mathematicians frequently used images and these images were often of a geometrical nature. Eisenberg and Dreyfus (*op cit.*), however, observe that there appears to be a difference between methods of processing information used by mathematicians in their own work, which are very often visual, and those they teach, in which the visual element is relegated to an illustrative role if it is used at all. Dreyfus (1991) asks why, if diagrams are essential for mathematical thinking, do mathematicians hide their visualisations and the arguments based on them? A possible answer is advanced by Barwise and Etchemendy (1991):

The main reason for the low repute of diagrams and other forms of visual representation in logic is the awareness of a variety of ill-understood mistakes one can make using them: witness the fallacies that have arisen from the misuse of diagrams in geometry. By contrast, it is felt that we have a fairly sophisticated semantic analysis of linguistic based reasoning.

...  
Our counter to this is two-fold. First, we note the obvious fact that a wide variety of mistaken proofs and fallacious inferences do not use visual information. Our second reply is to point out that although one can make mistakes using various forms of visual representation, it is also possible to give perfectly valid proofs using them. (1991, p. 16)

Eisenberg and Dreyfus (1991) observe similar behaviour with many students. Whenever possible, they say, students seem to choose a symbolic framework to process mathematical information rather than a visual one. While this is also true for many teachers, it does not seem to hold for professional mathematicians. "For them, the choice of representation in which to solve a problem seems to depend as much on the problem itself as on personal preferences (p. 26)." But, as Tall (1991) relates: "The visual ideas often considered intuitive by an experienced mathematician are not

necessarily intuitive to an inexperienced student." (p. 105). Elsewhere (1985), he notes that the major difference between the mature mathematician and the learner is that the former already has a global picture of the concept.

Eisenberg and Dreyfus (*op cit.*) propose three explanations for the pre-disposition of students for non-visual arguments: a cognitive one (visual reasoning is more difficult); a sociological one (visual reasoning is harder to teach); and one related to beliefs about the nature of mathematics (visual reasoning is not mathematical). They quote Chevallard (1985) who uses the term 'didactical transposition' to define the change that scientific, academic knowledge undergoes as it changes to instructional knowledge as taught in school. Analytic processing uses sentential representations, in which the information forms a sequence of expressions; visual processing uses diagrammatic representations. Chevallard claims that school knowledge is necessarily sequential; it is thus represented sequentially, not diagrammatically. As a consequence, teachers prefer to use sentential rather than diagrammatical representations.

Kruteskii (1976), in his study of gifted children, found that although there was a small 'geometric' group who appeared to use spatial representations in solving problems, and a small 'analytic' group who rarely used spatial reasoning, the majority varied their approach depending on the context of the problem. Dickson et al (1988), in their review of the literature on spatial thinking, recount how some writers (cf. Pask 1976, Sharma, 1979) have identified language functions and spatial processing with the left right and halves of the brain. Pask (1976) also categorises learners as 'serialists' or 'holists' and associates these with left and right hemisphere preference respectively. However, they conclude that: "The two aspects of mathematics represented, on the one hand by language and symbols, and on the other hand, by spatial representations, are entirely complementary in nature, and each should receive a reasonable share of attention in any mathematics curriculum." (p. 9).

Larkin and Simon (1987) further claim that the processing of these two types of representations is necessarily different. The fundamental difference between the two is that diagrammatic representations preserve explicitly the information about the topological and geometric relations among the components of the problem whilst sentential representations do not. Tall and Thomas (1989) observe that the activities that are prized in mathematics are usually symbolic and logical. "Far less often do we emphasise the visual and holistic" (p. 117).

Presmeg (1986) studied the effects of different teaching styles on the learning of mathematics by visualisers. She identified a visual, a middle and a non-visual group amongst the teachers. She found that non-visual teaching had the effect of leading visualisers to believe that success in mathematics depended on rote memorisation of rules and formulae. Teachers in the middle group often used visual methods but stressed abstraction and generalisation, and this aspect aided visualisers in overcoming some of the difficulties associated with the one-case concreteness of an image or a diagram. The visual teachers were unanimously positive in their attitudes towards visual methods, but they were not always able to lead visualisers to overcome the limitations, and to make optimal use of the strengths, of visual processing.

Earlier work on gender seemed to suggest that the under-achievement of girls in mathematics might partially be explained by differences in ‘spatial ability’. But, as Shuard (1982) writes:

The question of biological differences in either spatial visualisation or mathematical ability is not fully understood, but in view of the fact that the differences between the sexes in mathematical attainment are more marked in some countries than in others, there would seem to be factors other than differences in spatial visualisation which influence differences in mathematical attainment. (Shuard, 1982, pp. 279 - 280)

It is clear from this discussion so far that there are strong social factors which mitigate against the use of visual methods in mathematics. However, it should be remembered that these findings relate to doing mathematics in a pen-and-paper environment. Working in a computer environment may well alter the methods favoured by students as well as the valuation of those methods by their teachers.

### 3.1.2: Visual Thinking in Computer Environments

The idea that the computer can enable humans to exploit a visual capacity which they already possess is a central theme for several writers (diSessa, 1988a, 1989; Tall and West, 1986). Tall and West, for example write:

The human brain is powerfully equipped to process visual information. By using computer graphics, it is possible to tap this power to help students gain a greater understanding of many mathematical concepts. Furthermore, *dynamic* representations of mathematical processes furnish a degree of psychological reality that enables the mind to manipulate them in a far more fruitful way than could ever be achieved starting from static texts and pictures in a book. (Tall and West, 1986, p. 107).

This is also a core idea in Turkle and Papert’s (1990) thesis which links visualisation and gender with different styles of interacting with computers. There is, they argue, a tradition of scientific epistemology which sees the essence of science in objectivity and

the objectivity in a distanced relationship with the object of study. But girls may have access to a different form of reasoning which allows them to imagine themselves ‘inside the system’. In their thesis, the computer is seen as bridging the gap between the world of formal systems and physical things. They describe these two ways of working as ‘hard’ and ‘soft’ mastery, which they characterise as follows:

Hard mastery	Soft mastery
• a distanced stance	• closeness to objects.
• abstract thinking	• concrete reasoning
• systematic planning	• negotiative approach
• scientific method	• scientific practice

Traditional computer courses, they say, emphasise only the ‘hard’ way to interact with a computer; that is control through structure and planning, whereas ‘soft’ is often associated with feminine and unscientific approaches. But they argue: “Soft is a good word for a flexible and non-hierarchical style, open to experience of a close connection with the object of study. Using it goes along with insisting on negotiation, relationship and attachment as cognitive virtues” (p. 349). Central to fostering this way of working is what they term ‘transitional objects’:

An object moving on a computer screen might be defined by the most formal of rules and so be like a construct in pure mathematics; but at the same time it is visible, almost tangible, and allows a sense of direct manipulation that only the encultured mathematician can feel in traditional formal systems. (*ibid.*, p. 346)

Goldenberg (1989) also sees the potential of computer animated graphics as that of enabling students to *do* what mathematicians do. The means of doing so, he argues, lies in the use of properly designed computer-supported environments. These can provide, through their concreteness, a scaffold for reasoning and can foster the development and use of qualitative, visually based, reasoning styles to augment the traditionally taught symbolic-deductive methods. Such environments, he argues, can also open up previously inaccessible mathematical domains, allowing students to investigate topics that are often thought too advanced for them.

Some evidence for this assertion can be found in Tall and Thomas (1989) who assert that advances in mathematical thinking require the learner to gain an overall view of a topic or task. It is precisely this form of thinking, they suggest, that the computer can support. They report that the students in their experiment: “were found to be performing in global/holistic terms at a level comparable with more highly qualified university students with two years additional experience ... ” (p. 120).



Support for taking a holistic perspective can also be found in Mason (1993) who writes that the experience of screen images guarantees neither construal of nor abstraction from these images, and may even require more work than text to see through the particulars to the general. On the other hand, he says:

Experience in other media suggests that a more fruitful approach may be to learn from a Gestalt perspective, providing a minimum of screen detail which will sufficiently intrigue the viewer to try to make sense of what is seen, through filling in of details, and so initiate a sense-making action which will develop into accounting-for that succession of images.

(Mason, 1993, p. 371)

The question of how different representations mediate learning applies to traditional forms of media as well as to computers, and it is to these that the discussion now turns.

### **3.2: Visual Representations**

#### **3.2.1: Visual Representations in Mathematics**

Skemp (1971, p. 111) links visual thinking with visual symbols by contrasting and comparing the properties of visual and non-visual symbol as follows:

<b>Visual</b>	<b>Verbal-algebraic</b>
Abstracts spatial properties, such as shape, position.	Abstracts properties which are independent of spatial configuration, such as number.
Harder to communicate.	Easier to communicate.
May represent more individual thinking.	May represent more socialized thinking.
Integrative, showing structure.	Analytic, showing detail.
Simultaneous.	Sequential.
Intuitive.	Logical.

Dreyfus (1991) draws attention to the wide variety of visual representations which are used in mathematics: geometrical figures, tree diagrams, Venn diagrams, number lines and Cartesian planes, often with graphs of functions or scatter diagrams, sketches, diagrams and figures created locally for dealing with a specific problem may be even more important. But, he continues, there are major differences between these visual representations. Some are pictorial: they directly represent objects, e.g., geometrical ones. Others represent mathematical objects in a symbolic stylised way. Some are static, others are dynamic or at least imply the idea of a dynamic transformation. The same mathematical object may take different shapes in different visualisations. Hence, he argues, the patterns of reasoning (and possible problems) may differ for these different representations.

Plunkett (1979) adopts a similar theme. He points out that spatial matters involved in the teaching of mathematics are of two basic types: those to do with the real world and those concerning representations of the real world. Spatial representations involve pictures and diagrams. Plunkett uses the term ‘diagram’ for a spatial means of representing non-spatial ideas, for example, the number-line to represent the sequence of numbers, Venn diagrams to represent sets, etc. In contrast, he uses the term ‘picture’ for a spatial means of representing spatial ideas such as geometrical drawings, 3-d models, maps etc. Inevitably, he writes, these kinds of representations involve varying degrees of distortion of the real world.

Bishop (1979), in his research into the learning of mathematics by students in Papua New Guinea, observed that the picture conventions that are so prevalent in Western textbooks are by no means immediately recognisable by those from non-Western cultures. For example, he reports that the representation of a three-dimensional object by means of a two dimensional diagram demands considerable conventionalising. This, he argues needs to be considered in making judgements about spatial skills:

It has often been reported that students from non-Western cultures are poor at spatial skills, but it is often forgotten that ‘pictorial’ spatial tests invariably involve conventions. We are so familiar with these that we take their knowledge for granted and assume a universality of understanding which is quite erroneous. (Bishop, 1979, p. 138)

The observation that unfamiliarity with visual conventions may be the cause of a failure to understand some part of mathematics is, of course, not restricted to non-Western societies. There is a considerable body of research in this country (e.g., Hart et al, 1981 and Foxman et al, 1985) which highlight students’ failure to appreciate certain visual conventions in mathematics. Kerslake (1981), in her investigation of students’ understanding of graphs, found that many of the students had incorrect perceptual interpretations. Some thought of the lines on a distance-time graph as representing a journey up or down a hill. Elsewhere (1986), she makes the point that some representations can have a detrimental effect: “... if a fraction is seen in geometric, ‘part of a whole’ terms, then a major readjustment is required if a fraction is also to be thought of a number.” (p. 95). On the other hand, changing the representation may cause other problems.

The number line representation, a Boxer version of which formed the basis of the number line study in this research (chapter 8), may not be as understandable for students as its familiarity might suggest. Hasemann (1981) notes that it is not easy to put fractions in order of size on a number line, while Driscoll (1984, cf. Larson, 1980)

reports that in one study, less than 20 percent of seventh graders were able to locate the fraction  $\frac{1}{5}$  on the following number line:

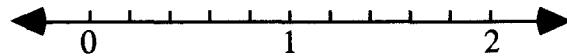


Figure 3.1: Placing rational numbers on a number line

Carr and Katterns (1984) write that: "Children fall into the trap of counting numerals, and overlook, therefore the importance of the spaces on the number line." (p. 33). It is not correct, they argue, to assume that a number line per se is a concrete illustration of measurement. Rather, they assert it is a symbolic representation, a point reinforced by Bright et al (1988), who observe that the number line requires the integration of two forms of information, visual and symbolic. This integration, they say, does not seem essential with other models. They hypothesise that it is translations between and within modes of representations which facilitate learning.

Vergnaud (1983a) writes that because 7- or 8-year-olds are already able to reproduce the graduation of their ruler upon a line, teachers may have the illusion that they understand the number line properties. However, he asks: 'which properties?' (p. 16). He writes that in order to understand the conventional number line representation, it is necessary that they synthesise the order properties of dots and the length properties of segments, and the concept of origin accompanies this synthesis. "Space as a signifier is then used to represent both order and magnitudes." (p. 17). He says that 13 year-olds may have two different ideas: either they use the order properties of space or they use measures, refusing the principle of inclusion.

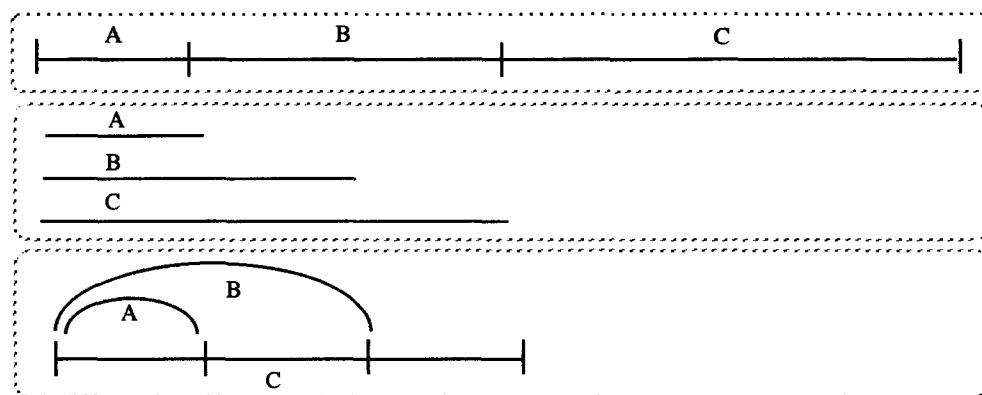


Figure 3.2: Students' number line representations; (Vergnaud and Errecalde, 1980, p. 288 – 289)

Vergnaud and Errecalde (1980) write that the status of intervals is not fixed for many students at this age, with the inclusive case being particularly difficult, so that some children separate diagrams, whilst others feel the need to explain. They illustrate three of the methods used by students (aged 10 - 13) to represent data on a line (Figure 3.2).

This close connection between representations and the concept represented has been addressed by Pimm (1987) who writes of how ‘extra-mathematical metaphors’ are imported from the real-world to illuminate mathematical concepts, a function as a machine with inputs and outputs being one of these. These are so institutionalised, he notes, that teachers may not accept that they are metaphors at all. Such usage, he suggests, is acceptable in an introductory setting where the metaphor is being used as a conceptual bridge, but he cautions (p. 98): “It is essential for pupils ultimately to be aware that a function is not a machine.” Elsewhere (1995), he observes:

‘The’ notion of function is actually subtly different, depending on whether it is accessed through algebraic forms, graphs or numerical tables. Because the generative direction *from representation to ‘object’* is seen as the reverse, the central role of representations in conditioning mathematics is insufficiently appreciated. (Pimm, 1995, p. 102)

Sutherland (1988) reproduced the following three common representations of functions and drew attention to the close affinity between the mathematical notion of a function and the functional structure of Logo. A subsidiary aim of her functions materials was “to provoke pupils to use decimals and negative numbers” (p. 205).

(a) Logo Notation	(b) Mapping Diagram	(c) Algebraic Notation								
<pre>ADDFOUR "X OUTPUT ADD :X 4 END</pre>	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>IN</th><th>OUT</th></tr> </thead> <tbody> <tr> <td>3</td><td>7</td></tr> <tr> <td>-2</td><td>2</td></tr> <tr> <td>1.5</td><td>5.5</td></tr> </tbody> </table>	IN	OUT	3	7	-2	2	1.5	5.5	$F(X) = X + 4$ or $X \rightarrow X + 4$
IN	OUT									
3	7									
-2	2									
1.5	5.5									

**Figure 3.3: Common representations of functions (Sutherland, 1987, p. 203)**

In this, she set a precedent for the function machine study (chapter 9) of this research, but whereas she employed representation (a) above, which is essentially textual, I used a function machine representation which is essentially visual.

### 3.2.2: Visual Representations in Computer Environments

Thompson and Dreyfus’s (1988) research shared some characteristics of one of the number studies in this research (chapter 8) in that it employed a computerised number line. In their study, a model of integers as transformations was presented with a turtle

walking up and down the number line according to the operation proposed. They conclude that sixth graders can conceive of integers as transformations in non-trivial ways, but they also comment on the instability of students' conceptualisations, and point to diSessa's (1983) notion of phenomenological primitives for a theory of explanation.

Goldenberg (1988) remarks that although students spend years perceiving and drawing visual forms before they encounter algebraic symbols, graphs have conventions and ambiguities of their own. To interpret graphs correctly, he says, we need mathematical knowledge and expectations, not just perceptual experience. He goes on to discuss issues of partial views, scale, the role of the student vis-a-vis the computer and problems of presentation.

Two conclusions stand out: simplistic software design or thoughtless use of computer graphing in classrooms may further obscure some of what we already find very difficult to teach. On the other hand, thoughtful design and use of graphing software presents new opportunities to focus on challenging and important mathematical issues that were always important to our students but that were never so accessible before.

(Goldenberg, 1988, p. 135)

Yerushalmy and Chazan (1991) identify three obstacles, culled from previous research, which students must overcome when examining and interpreting diagrams: firstly, diagrams are particular; secondly, common usage confuses certain standard diagrams with the classes of objects to which they belong; thirdly, a single diagram can be viewed and described in different ways. They outline the resources of 'Supposer', a set of microcomputer software tools which were designed to aid students in overcoming these obstacles.

Kaput (1986) argues for the creation of multiple representations of the same concept in the belief that this multiplicity of views will somehow enable the learner to see what it is that is central to that concept. He claims that such an environment renders three very important goals of mathematics far more attainable than previously. The first of these has three sub-goals: by presenting a family of iconic representations varying in concreteness, it increases the chances of linking with the student's existing cognitive structures; it provides schematic representations for different problem situations; and it exposes the deeper mathematical commonalities. Secondly, by making visually explicit the relationships between different representations and the ways that actions in one have consequences in the others, the most difficult pedagogical and curricular problem of building cognitive links between them becomes more tractable. Finally, by providing a series of carefully chosen representations that begin with the student's primitive and

inflexible ones and ramp upward to ever more powerful and abstract mathematical ones, a new level of longitudinal coherence in the mathematics curriculum is possible.

In a later paper, (Kaput, 1989), he again emphasises the visual linking of different representations: “Our effort is also marked by a deliberate attempt to tie visually concrete and enactive operations on objects (in this case, objects on a computer screen) with more formal and abstract representations of these operations.” (ibid., p. 35). However, he acknowledges some of the difficulties which might occur “Just as the visual experience is a powerful organiser of appropriate cognitive structure in some cases, when it acts in other cases to produce a structure contradictory to an existing one, it leads to debilitating confusion.” (ibid., p. 42).

Goldenberg (1988) also considers multiple representations. He begins by asserting that “Common sense supports the notion that multiple representation will aid understanding.” (p. 136), but then goes on to say that very little is known about what actually takes place when multiple linked representation software is used extensively in the classroom: “While potentially reducing ambiguity, multiple representation also presents a student with more places to look and is potentially complicated and distracting” (ibid., p. 137).

A different form of a ‘manipulable multiple representation’ approach is the software ‘Math. Path’ as described by Kaput (1986): “With this, the student can build, modify, and otherwise pedagogically engage a machine-based representation of arithmetic and algebraic expressions. An input-output device is connected by wires. When one inputs numerical values, they travel along the wires as a mouse in a snake, passing through the various components and eventually landing in an output bin.” (ibid., p. 202). diSessa (1986a) writes that such ‘device programming’ is attractive because it has such a simple and graphic method of combining elements to make compound things. “There is reason to believe it can have some intuitive accessibility that the hidden data flow and complex sequencing of pure procedural languages does not.” (p. 138).

Laborde (1993) writes that: “The graphical and computing possibilities of software allow now a reification of abstract objects and in particular of mathematical objects ... ” (p. 40). Cuoco (in press) is more specific. He compares how different computational media affect students’ conceptions of function along a continuum of action --> process --> object. *Function Machines* (cf. Goldenberg, 1985), he says, help students to iterorize actions into processes, but Logo helps them iterorize processes into objects. In chapter 9, an

account is given of how one of the computational number objects of this research, the function-machine object, embraced some elements of both forms of media, and how these qualities helped to provide the ‘scaffolding’ to support the *encapsulation* (Dubinsky, 1991) of number processes into number objects.

### 3.3: Programming and Boxer

#### 3.3.1: Visualisation and Programming

The manipulation of icons on a screen, along with menu-driven choices mark out in many respects a region in which the question of what constitutes programming becomes increasingly blurred. When computers were first introduced, the only interface to them was at the machine language level. Since that time, designers have attempted to simplify that interface by introducing ‘higher level’ languages. However, as the level of a language increases, its domain of generality decreases. For example, most menu systems are easy to use but the user has limited choices. Conventional programming, on the other hand, might seem to offer unlimited choice or control, but the difficulties in learning these languages are such as to afford only limited access.

Hoyles and Noss (1992a) are critical of systems whose major design consideration is ease of access, arguing that such systems limit users to interpretation rather than construction. Similarly, they believe that ‘authoring systems’ constrain what can be written and close down the flexibility of a true programming environment. They write:

Given that a primary aim of mathematical activity is for students to interact with mathematical objects and processes, it is often insufficient for students to see what happens, they need to know how it happens - and this is why we see programming as an important component of computer-systems. (Hoyles and Noss, 1992a, p. 3)

Zloof (1984) illustrates the relationship between the level of programming and its generality by the following diagram:

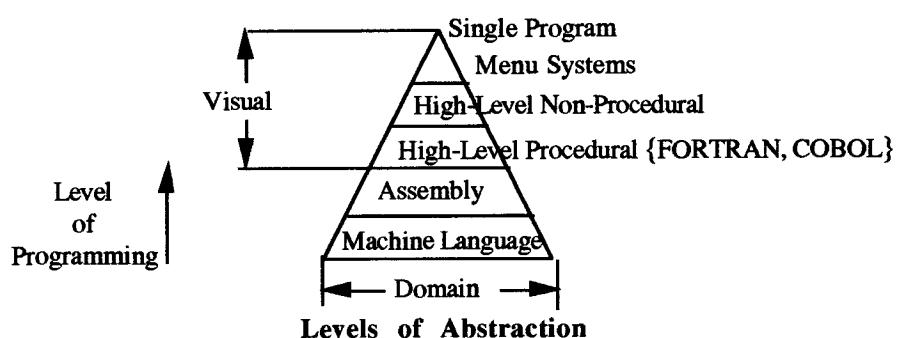


Figure 3.4: Programming languages and generality (Zloof, p. 232)

Programming that allows visual manipulation would appear to offer both choice and access, so it is questionable whether Zloof's diagram can accommodate such languages. Indeed, as I hope to make clear in what follows, it is the intention of designers of visual programming environments (including Boxer), to shorten the distance between programmers and users or between experts and novices.

Myers (1986) attributes the growth of visual or graphic programming to the widespread use of personal computers and the recognised difficulties in learning conventional programming languages. But he writes that another motivation for using graphics is that it tends to be a higher level description of the desired actions (often de-emphasising issues of syntax and providing a higher level of abstraction) and may therefore make the programming task easier for professional programmers. This, he says may be especially true during debugging.

Raeder (1985) writes that traditional computer programming languages and environments are usually rather difficult to use because designers have not come very far in adapting them to the user. This lack of adaptation, he says, exists both at the representational level (that is the graphical layout seen by the user) and at the conceptual level (which is to say, in the computational model). Because traditional computer systems show very little of their internal state to the user, he or she must remember and manipulate a large, complex, and abstract system structure with only hints from the computer about whether his or her conjectures regarding the state are correct. Screen editors, spreadsheets, and electronic desktops attempt to remedy this situation by making use of graphics to show as much of the state as possible.

Shneiderman (1983) found that certain interactive systems generated glowing enthusiasm amongst these users – in marked contrast with the more common reaction of grudging acceptance or outright hostility. He reports:

As I talked with these enthusiasts and examined the systems they used, I began to develop a model of the features that produced such delight. The central ideas seemed to be the visibility of the object of interest; rapid, reversible, incremental actions; and replacement of complex command language syntax by direct manipulation of the object of interest - hence the name 'direct manipulation'. (Shneiderman, 1983, p. 57)

However, he also notes that direct manipulation systems are not devoid of problems. He records that using spatial or graphic representations of a problem does not necessarily improve performance and suggests that the content of a graphic

representations is a critical determinant of their utility. As he says “Choosing the right representations and operations is not easy. Simple metaphors, analogies, or models with a minimal set of concepts seem most appropriate.” (p. 64).

### 3.3.2: Visualisation and Boxer

#### 3.3.2.1: The boxes of Boxer

Perhaps the first thing one should say about Boxer is that it is a computational environment in which the central metaphors are visual, yet the programming language itself is based on text processing. As diSessa (1986a) writes: “In comparison to iconic programming or programming by direct manipulation, Boxer’s text orientation may seem conservative, if not reactionary. But the main reason we chose to move in that direction was to promote a synergy between the written word, including all the incremental advantage it offers, and the advantage that programming and programming structure offers to written communication” (p. 141).

In contrast to the range of objects in most computer systems such as lists, character strings, numbers, variables and so on, in Boxer there is a single entity: the box. 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. Figure 3.5 shows examples of data boxes (rounded corners) and doit boxes (square corners) as well as a graphics box (dashed outline).

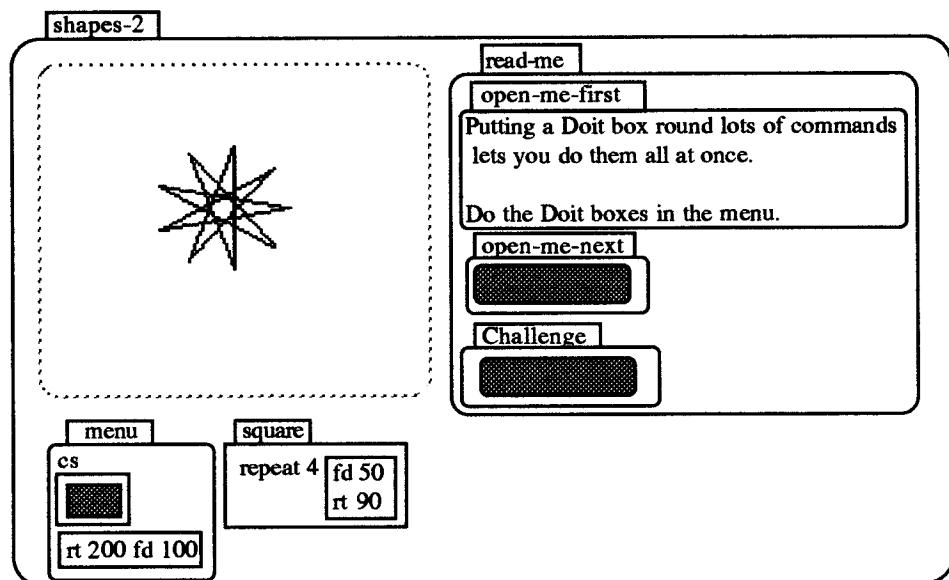


Figure 3.5: The boxes of Boxer

Data boxes combine things so that they can be accessed in an organised way, named, and operated upon by programs. A data box identifies its contents as a data object, that is, a unit that can be named, passed as an argument to a procedure, or returned as a value by a procedure. The contents of data boxes can be changed either by typing in the box on the screen or by using the command ‘change’. Doit boxes (or procedure boxes) group expressions to form programs. Besides evaluating an entire doit box as a unit, users can also step through the procedure line-by-line. Boxes can also be named, and evaluating the name will evaluate the lines in the doit box.

There are also three subsidiary boxes: graphics boxes, closets, and ports. A graphics box is an area on the screen on which drawings can be made. Graphics boxes can be inhabited by sprites, which are objects that can draw and have local data, procedures and touch-sensitive capabilities. Associated with each ordinary box is a transparent box, the closets which is usually invisible. It can be made visible to put tools in it, then hid again. Ports allow data and programs to be shared among different parts of a Boxer environment. Ports give a ‘view of’ an ordinary box located somewhere else. Changing the port (by typing in it or by using the ‘change’ command), also changes the original box.

There is thus a tremendous compression of diversity to simplify structural comprehension. The diversity needed in programming is provided in two basic ways. Firstly, the internal structure of the box can be varied. Secondly, any box may be named, and naming accomplishes the transition of a data or do-it box into a variable or defined procedure. Any box may be saved to disk (with or without a name), making it a file, and also making the surrounding box structure implicitly into a hierarchical file system. A record with fields is simply a data box with named data sub-boxes. A local variable or sub-procedure is simply a data or doit box located inside some other procedure.

Boxes can be closed, opened or expanded to full screen size. Novices can simply use the procedures (doit boxes) as given: they can open shrunken boxes to examine details of procedures or execute pieces of them to see how they work. As they gain in experience, they can of course write the procedures themselves. Since any aspect of Boxer, can be inspected, microworlds can be designed to be used at different levels.

Beyond just controlling how things are displayed, the arrangement of boxes forms a structure that organises how programs and data are used. The nesting of boxes within

boxes organises a Boxer environment by providing a concrete representation for block structure. In general, the Boxer evaluator will ‘see’ the names of boxes that are in the same box as the expression being evaluated. But it cannot look inside sub-boxes to see the names of their internal boxes. If it does not find a box of that name then it looks in the next outermost box, then the next outermost, and so on (referred to as *shadowing* ). This means that a single name can be used to mean different things in different parts of a system without confusion. Another way to say this is that a name is *local* to the box in which it is defined. On the other hand, should the sharing of names be desired, then this is accomplished by nesting boxes within boxes.

The internal programming language of Boxer is essentially that of Logo, but there are important differences. There is, for example, no distinction between the editor and the workspace; the system is always at editor ‘top-level’. The syntactical devices of quotes and colons used to mark variables have gone, so that the first line of a procedure is of the form ‘input x’. There are also changes with respect to the list processing facilities. The data box takes on the role of the list and has its own data operators, and variables are simply named data boxes.

The motivation for these design decisions can be understood firstly by reference to the Boxer group’s overall aims and secondly to their views on epistemology, pedagogy and programming.

The Boxer Project is aiming to test the feasibility of filling a rather grand and still hypothetical social niche, that for a computational medium. We are trying to produce a prototype of a system that extends with computational capabilities the role now played in our culture by written text. We wish to change the common infrastructure of knowledge presentation, manipulation and development. We want a general purpose system to serve the needs of students, teachers and curriculum developers, something that is so useful for such a broad range of activities that the community as a whole will judge it valuable enough to warrant the effort of learning a new and extended literacy. (diSessa, 1990b, p. 303)

diSessa (1983) sets out the beginnings of a theory of knowledge that traces the origins of scientific understanding in simple abstractions of everyday events (Phenomenological primitives). diSessa (1987) argues that many of the behaviours that Piaget attributed to pre-formal stages of reasoning can be seen to be a result of these naive conceptions, while diSessa (1988) sees the fundamental question as being that of the transition from intuitive understanding to scientific understanding. diSessa (1989) challenges the assumption that intelligence is an invariant that does not change substantially from context to context, and is impervious from instruction and learning. He writes instead of what can be achieved in tandem with computers:

There are important other ways to think about intelligence. In particular, consider the possibility of *symbiotic intelligence*, that individuals in concert with computers (via our new medium) simply are able to accomplish much more in intellectual tasks than individuals alone, no matter how well trained, or what general skills they have.

(diSessa, 1989, p. 7)

A central image for the Boxer group is that of a tool-rich culture where individuals borrow, share and invent a broad range of tools to serve their purposes. A computational medium, they believe, can support a new tool-rich culture in schools. They argue that computational tools can take over the menial tasks, say arithmetical or graphical ones (such as plotting points), to allow students to work at the more important higher levels of deciding on approach and interpreting results.

The Boxer group begin with the premise that, in the future, most computer users will be people who are not computer specialists. Rather, they are likely to be people who wish to use the computer as a tool to serve their own needs without requiring inordinate computational sophistication or effort. As diSessa, Abelson and Ploger (1991) put it: "Students and teachers can shape an expressive computational medium for their own ends" (p. 3). Amongst these needs are: text-editing, using and modifying pre-written programs, writing programs, searching and manipulating databases and producing graphics with a flexible programming graphics facility.

They believe that these needs can best be served by providing an integrated computational environment that can accomplish these tasks via common facilities rather than by means of special sub-systems. They argue that even if novices need not deal with all of the above facilities, there are advantages in having a system in which learning in one area could be carried over to another. They use the phrase *synergistic effect* to refer to this carrying over of competences. Focusing in what might be required for the successful implementation of such a system, diSessa (1990a) sets out parameters.

On the high end, we are not interested in expertise as defined by current professional programming. The things people will be doing with computational medium are not the same things professional programmers do. The scale of complexity in programming will not approach that of 50,000 line programs. The capability to design, comprehend and modify diverse, small but non-trivial programs is more central. My personal estimate is that our target lies largely within the range of programming that can be done in a day by an expert programmer (presuming an excellent, tool-rich programming environment).

(diSessa, 1990a, p. 17).

The design of such an integrated computational environment for novices raises many issues. In particular, there is the issue of understandability of such a system as perceived by its user. It is therefore necessary to try to comprehend the mental models

that people form of such complex systems. Towards this end, the distinction between structure and function is helpful. This distinction focuses on the characteristics of an object or action which are independent of specific use (structure) or alternatively, characteristics which have to do with specific use (function). diSessa (1986b) approaches understanding by reifying three kinds of models: the *structural* model, the *functional* model and the *distributed* model, and argues that "All of these models can and should be cultivated simultaneously in the learner" (p. 201)

The expression *mental models* has come to denote a very rich and well-developed knowledge system that serves as a means for understanding less accessible knowledge. An example of this sort of model is Papert's (1980) 'little man' model of Logo. diSessa (1986b, p. 202) notes that Young (1983) uses the term 'surrogate models' but he prefers the description *structural model* for the same thing.

Typically these are abstract dealing with a system independently of how it is used. Although, structural models may be very good for debugging, they can suffer from a number of problems when it comes to understandability. Firstly, since structural models aim at being uniform views of rather complex systems, they themselves tend to be complex which leads to a related problem in that they are slow to run and hard to learn. Incremental learning is sacrificed for the sake of uniformity and completeness. Furthermore, functional understanding is not well supported by a structural model.

The characteristics of functional models are in many ways complementary to those of structural models. With functional models, the constructs of a system are understood according to what they do. diSessa (1986b) refers to Young's (1983) study of students' understanding of how a calculator works as an archetypal functional model. Keying in  $3 + 5$  and pressing '=' gives no hint as to the structural model of the algebraic calculator, but it provides a perfectly adequate functional model which is based on the user's previous knowledge of arithmetic. Functional models have advantages because they allow sufficient understanding of a machine to get something done. They are also useful when it comes to designing and building something. On the other hand, functional models are often not comprehensive enough. They are good at the early stages of understanding but they will need to be supplemented by other kinds of models (structural models or the equivalent) at later stages.

diSessa (op cit.) identifies a third class of model (*distributed or patchwork*) models by noting how students explain to themselves early constructs of languages. These

descriptions, he writes, typically fall far short of the intended structural models, with much more diversity in their sources and unreliability in their transfer to other situations. Distributed models depend on specific prior knowledge that users have and may bring to the system. diSessa (1985c) gives the example of FORWARD 100 from beginner's learning Logo. Initially, he says, this is interpreted simply as an abbreviation for the English sentence "Go forward 100 steps." The need for input to the command FORWARD is not, therefore, understood structurally, but according to the semantic need to complete the sentence FORWARD <how far?>. "Linguistic and semantic function here come first and provide a preliminary model of the structure *command plus input.*" (p. 11).

He asserts that the long-term effects of particular functional models need to be considered. Some will fade away and be replaced by structural models. Others will be integrated as special cases. The Boxer group believe that in designing complex systems, it is vital to take into account the distributed models that learners use in coming to understand such systems.

The design of Boxer relies critically on human spatial/visual capabilities as the basis for both structural and distributed models. Two important principles guided Boxer's design towards achieving this. The first, *naive realism*, is an extension of the "what you see is what you get" concept that has become commonplace in text editors and spreadsheets, but not for programming languages. In Boxer, users can usefully pretend that what they see on the screen is their computational world in their entirety. In particular, any text that appears on the screen can be edited or evaluated.

The second principle, *spatial metaphor*, uses people's intuition of space as the basis for programming the entire computational system. Whereas in many hypertext systems, relations are expressed abstractly as links between nodes, in Boxer, almost all relations are expressed by the place the box appears. Because of the spatial metaphor, Boxer needs no separate browser. One simply moves the cursor about in Boxer's space. Spatial arrangements of objects on the screen are meaningful and permanent until the user changes them directly or indirectly (through programs). Unlike window-based hypertext systems, spatial arrangements are created almost exclusively to express relatively long-term semantic relationships rather than for transient screen-management.

Visual understanding alone is insufficient to explain the choices made by Boxer's designers. So, whilst recognising the importance of icons and visual analogues of real

world objects (desk tops, file folders, trash cans) as important to comprehensibility, they rejected these functionally-oriented aids and chose instead visual devices that relate directly to the underlying structure of Boxer programming. Similarly, purely graphically programming in Boxer was avoided because the choice of that kind of structure moved too far away from the uses intended for Boxer, and it would thus afford too little functional modelling support. The most prominent decision in this respect was to use a text-based surface representational form. The choice of text derives both from the desire to import familiar language and to pave the way towards incorporating a text-editing facility as part of an integrated computational environment.

At the time of this research, the balance between functional, structural and distributed models was part of an on-going debate within the Boxer community – a debate which was informed by the experimental results, which form the subject of the next section.

### 3.3.2.2: Experimental Results

In this section, the research evidence relating to Boxer is reviewed. Three categories of research are identified: that concerned with novel aspects of Boxer; that concerned with programming; and that which relates to Boxer representation. It should be emphasised that few studies fall completely into one or other of these categories, and my accounts vary proportionately to what I see as their relevance to this study.

Adams' (1989) study belongs in the first of these categories. He documents a case study of students' knowledge of dinosaurs. Adams illustrates his discussion with images from the Boxer dinosaur microworld. However, there is no research evidence presented in support of his claims for databases in Boxer generally, or the dinosaur database in particular. Nevertheless, the study is valuable both in terms of showing how the full integrated components of Boxer can be exploited to create and interrogate a database, and in the way it indicates how Boxer might be used to build on students' knowledge and interests.

I have located three quite different studies in the category of Boxer as a programming language. The first of these was undertaken by a graduate student Chuen-Tsai Sun but reported in diSessa (1990a). There were five students involved aged 11 - 13 years old, from a private and academically oriented school in Oakland, California. The students' experience with Boxer ranged from a few months to a little over a year. They were judged to have understood the notions of data, variable and procedure in Boxer, but none of the students had any instruction in Boxer scoping.

The researchers found that transfer of scoping rules to procedure variables located inside a procedure initially unproblematic. “Students understood intuitively and immediately that local variables would be accessed on execution of that variable in place of more global ones.” (p. 48). However, they observed that no notion of copy was needed up to this point. The copy-and-execute model, they report, was appropriated with some false steps and trepidation on the part of the students, but in the end all of them applied it successfully to novel problems without help. All in all, this study was that of validating design principles. Distributed models were found to be adequate to the early stages of learning and were of the form of visual-spatial type that were expected. Clearly, there are limits to this study in that the teachers were all ‘experts’ and the students probably atypical. But, the designers write that they “were extraordinarily pleased with this formative study.” (p. 53).

Visual-spatial considerations were central to a second study by Schweiker and Muthig (1987) also. In recent years, they note, a great deal of work has been devoted to reducing the cognitive load on naive users of computers. In particular, a much greater emphasis has been placed on visual programming. However, despite this emphasis, they write that (cf. Gorny and Tauber, 1987) empirical evidence on the relative pros and cons of visual aids in programming are lacking.

They conjecture that this seems to be due to the difficulties in isolating the relative effects of the visual aids provided, since programming languages which differ in this respect, often also differ with respect to their inherent structure. Thus, in order to evaluate the relative efficiency of visual aids in programming, languages have to be compared which are basically similar with respect to their inherent structure but differ from each other with respect to presentation mode. Logo and Boxer were adjudged to satisfy these criteria.

To study performance in programming, they observe (cf. Brooks, 1980) that measures could be used for each of the following tasks: program construction (time to write a program), debugging and program modification (number of bugs identified, time to identify a certain bug), memorisation-recall and reconstruction (speed and accuracy in reproducing a program), question answering (number of correct answers, time to answer), and hand execution (speed and accuracy of the execution). However, since they had no implementation of Boxer available at the time of the study, their investigation was restricted to debugging and program modification.

The task domain for the research concentrated on list related operations, since these operations form an important part of many Logo applications. In order to apply the respective list operations on actual lists, it was decided to use lists which contain two pairs of randomly selected letters each. By using these this kind of artificial lists, it was hoped that possible familiarity effects might be prevented. Twenty-three students (13 university undergraduates and 10 student-workers from IBM) volunteered to participate in the experiment, and were randomly assigned to either a Logo or Boxer group. Although most of the subjects had expertise in at least one programming language, none of them had any prior experience with Logo or Boxer.

The researchers found that, on average, significantly less time was needed to evaluate a proposed solution in Boxer (11.44 seconds.) than in Logo (29.5 seconds). They surmised that this might be due to the simple step-wise comparison of the respective outcomes with the desired result which can be performed in Boxer; whereas, in Logo, intricate checks of correctness are required. They concluded that, overall, the visual aids provided by Boxer proved to be efficient in reducing the cognitive demands for solving these kind of problems and in providing a more concrete problem space for solution. They also found that mean error rates in evaluating the proposed solutions were considerably smaller in Boxer (1.04%) than in Logo (11.36%). The results of their experiment indicate, they argue, that Boxer must not be regarded as only a modernised update of Logo, but as a real improvement over Logo - at least with respect to the features which were the subject of the investigation.

A very different approach to the Heidleberg research was that undertaken by Leonard (1990). This study focused on how two 8th grade girls learned recursion in Boxer. The theoretical perspective taken is that students learn by constructing new knowledge to overcome *impasses* in their problem solving. The particular knowledge with which the research is concerned is structural knowledge of programming. Since Boxer was designed to make structure more visible and understandable to students, and given that an understanding of recursion requires a fairly deep understanding of structure, the researcher set out to test this hypothesis.

Most work on learning computer programming, he observes has focused on plans for writing programs. This work is limited, he suggests, since firstly, it does not explicitly consider structural or strategic knowledge, and secondly, it considers the products of learning, but provides no models of how actual students acquire these plans. By

contrast, he says his study provides a detailed model of how one of the students acquired structural knowledge. His research focused on two of the misconceptions identified by Kurland and Pea (1989): namely “mental model of embedded recursion as looping” and “decontextualised interpretation of commands”.

Leonard concludes that many aspects of Boxer were well understood by the student. In this sense, he says, the study is compatible with that of Schweiker and Muthig’s controlled study given above. It showed also how impasse-analysis could be used for assessing students’ knowledge and it demonstrated how Boxer helped to overcome the aforementioned misconceptions.

Outside of programming, Boxer research appears to have been conducted in three main knowledge domains: biology, physics and mathematics. Several studies (Ploger and Lay, 1990; Ploger, 1990a, 1990b, 1991) have testified to the efficacy of dynamic Boxer representations in the field of Biology. Other studies in physics (diSessa, 1989; Adams and diSessa, 1990) have emphasised the constructivist roots of the Boxer group pointing to how visual representations have fostered child expertise and continuity of activities. They observe how programming in Boxer replaced algebra thus enabling young students to deal with concepts (like relative motion). These twin notions of visual representation and visual thinking were also central to the Boxer studies in mathematics which I shall look at in more detail.

Picciotto’s study (1989b) was with a group of twelve students aged 12 to 16 in the context of a six week summer course as part of the University of California’s Academic Talent Program. Some had neither experience of programming nor algebra, whilst others had considerable experience of both. In addition, he writes of having too few computers and of bugs and inexplicable crashes with the implemented version of Boxer. He decided to teach the concept of sampling, and designed a discovery-based curriculum to assist the theoretical understanding. To simulate the random sampling of a yes-no population, dice, marbles and spinners were used. The students were then asked to simulate each process in Boxer. He recalls what happened next:

The transition from a real world experiment to a Boxer version was remarkably easy as compared with doing the same thing in Logo or BASIC, perhaps because variables and graphics are visible at all times. However, this SAMPLER program still proved to be much too difficult for beginners to write, and two (out of four) teams failed to complete it. For them, this was rather frustrating, and it became clear that while Boxer was easier to learn than other languages, programming is a difficult mental discipline, and cannot be picked up on the fly in a couple of weeks. (Picciotto, 1989b, p. 12)

As a result, he decided to de-emphasise programming. He re-arranged the student Boxer teams to be more homogeneous and asked the students to create Boxer projects at their own level of programming expertise. They were free to borrow procedures from teacher-written programs and from one another. He suggested four topics and found, to his surprise, that all four were implemented, each one by a different group. He reports that he was not surprised that the experienced programmers were successful, but adds that he was stunned by the achievements of group of three students who were beginners with very limited exposure to programming (one had done a little Logo, one had done some BASIC and the third had never used a computer).

Following Picciotto's suggestion, these students designed a random walk for a turtle in a graphics box. This would go straight down or diagonally, depending on the outcome of asking a random member of a simulated population for a yes-no answer. This was done repeatedly, creating paths that follow the design of Pascal's triangle and provide a dynamic visual model of sampling. The students implemented a suggestion of another researcher (Don Ploger) that the turtle drop a 'marble' at the end of its walk with the result that a histogram was drawn which dramatically displayed the distribution of the samples.

He writes that, while the key ideas behind the program were suggested by teachers, and the most difficult pieces of code were borrowed from the work of more experienced programmers, the scope of the project and every step of its implementation were determined by the students themselves. He concludes that a program of this sophistication could not have been achieved in this short a time in any other language, by people with as little experience as they had.

Like the Picciotto study reported above, Ploger and diSessa (1987a) also carried out an experiment concerned with the topic of probability. They report a case study of a 12-year-old student exploring a topic in probability by rolling simulated dice and recording the results within the Boxer computer system. They stress that the activity had two goals: that of learning probability and learning programming. In this experiment, the student was told that of three dice, two were loaded. He had to experiment to discover which dice were loaded and also how they were loaded. What was novel about the experiment is that the (simulated) throwing of the dice as well as the recording of the results were all carried out within Boxer. The authors analyse this experiment from the points of view of both educational content and those of programming.

The simplicity of what has been done here belies its importance. An object of investigation (die) has been implemented as a general computational object (program) in such a way that the correspondence between the two is obvious, including how to operate the object (roll the die = execute the program). (Ploger and diSessa, 1987, p. 4)

With respect to content, they emphasise several aspects. Firstly, the experiment began with a question which was both easy for the child to understand and which was also interesting. Secondly, it had a solution that was easy to comprehend, but was not easy to find. Thirdly, the student found the search for the answer challenging. In the course of this process, the student learned to use powerful computational tools. These aspects of learning were not separate features, they argue, but rather different benefits from a coherent, integrated, educational process.

A more recent study involving mathematics is reported by Williams and Rochelle (1993) who write of four 16 year-olds using Boxer to model the spread of a disease through a city. They say certain features of Boxer lent themselves to different stages of solving the problem (cf. Polya, 1945). At the point of formulation, Boxer's capacity to integrate text and program was particularly helpful. Students were able to deal with the concrete manifestations of variables in the form of named data boxes and the spatial block structure help them organise their thinking. Boxer's ability to permit line-by-line execution and its manner of building data structures gave students a simple means of solving for a particular case.

However, they say that the algorithm that had worked for one particular case began to go wrong when tried with another specific case. They write that because the model was an active one and was visually represented in a concrete manner, and because the process could be stepped through, the cause of the error was readily isolated. "The video shows the teacher and the two students using their hands and the mouse pointer to identify the source of error and to communicate with each other. They have objects to talk about." (p. 497).

Finally, they write that in presenting the report of the problem and its solution, certain features of Boxer are unique in that they enable a *written* description of the solution and the process to be juxtaposed on or integrated with the *active* expression of the model or the solution to the problem. Boxer, they say allowed an almost physical interaction between the person and the mathematical computational environment thus allowing students to engage with, provide models for and solutions to complex mathematical problems.

### **3.4: Summary**

This chapter has reviewed the literature relating to visualisation in mathematics, and the role of computers in that visualisation. It has also reviewed the visualisation literature in relation to programming in general, and to Boxer in particular. There have been two central themes throughout this review: visual thinking and visual representations.

Section 3.1 examined the role of visualisation in the practice of mathematics. It showed that although mathematicians frequently employed visual images, there were strong professional pressures (Davis, 1993) to omit such images in their proofs – perhaps the defining characteristic of mathematics. Similar pressures existed in the classroom where symbolic reasoning was generally accorded higher status and visual reasoning relegated to a supporting role if used at all.

However, section 3.1 also showed that with the advent of computer generated images, the status of visual reasoning was experiencing potentially far-reaching changes. Several authors (e.g., Turkle and Papert, 1990; Goldenberg, 1989) highlighted the fact that in working with computational objects, young students were gaining access to doing mathematics in ways similar to that of mathematicians. Other writers (e.g., Tall and Thomas, 1989) suggested that computer environments enabled students to adopt a holistic way of thinking and thus deal with concepts much earlier than might otherwise have been possible.

Section 3.2 moved the discussion to visual representations. The review pointed to the variety of visual representations used in mathematics, but noted major differences between them, and possibly in the nature of the reasoning which they sponsored. Such observations serve to reinforce Vergnaud's view (chapter 2) of the need to study concepts in contexts. This section also noted that the models and diagrams used to introduce concepts pre-supposed an understanding of certain spatial/geometric concepts. Consequently, misunderstandings about such concepts were often closely linked to misunderstandings about the models that were used to illustrate them. In particular, the number line representation was looked at in some detail as a Boxer version of it formed the basis of one of the number studies of this research (chapter 8).

The review of computer-mediated representations indicated that such misunderstandings were equally likely in computer environments. On the other hand, it was pointed out how sophisticated software could help students overcome conceptual obstacles, and

could enable them to engage with concepts at an earlier age than usual. The potential of using computerised function machines was discussed in the light of their potential for encapsulating actions into processes because the second number study in this research (chapter 9) involved students in constructing and interacting with one such device.

Section 3.3 reviewed the literature pertaining to the role of visualisation in programming and in Boxer. It began by pointing to the fact that previous efforts at making programming accessible did so at a cost of removing control from the user. Visual programming languages attempt to overcome the ‘complexity barrier’ by having the user directly manipulate screen objects. But such systems, as diSessa (1986a) points out, stress functional considerations at the cost of structural ones. For the Boxer group, programming is a long-term goal, and one which can best be accomplished by giving students access to structural as well as functional and patchwork models.

The second half of section 3.3 reviewed the available Boxer research. In doing so, it returned to the theoretical concerns of chapter 2: constructivism, concepts in context and how to build on students’ intuitions. It drew again on the twin notions of visual thinking and visual representation, but this time in the context of Boxer. The experimental results gave some indication of how learning to program in Boxer and learning mathematical knowledge could be integrated. In particular, they showed how, in interacting with visual representations, students might make connections between their intuitive notions and the more formal and abstract concepts of mathematics.

These two ideas: that of integrating programming with mathematical learning, and exploiting the medium to make connections between students’ intuitive notions of number and more abstract concepts were central to this study also. The long-term nature of this research implied that it would be necessary to develop students’ programming knowledge both as a precursor to the number studies, and in the course of those studies.

This chapter complements chapter 2 in providing a theoretical framework for the thesis as a whole. The following chapter provides a more detailed outline of the research and presents a methodological and pedagogical framework for the remainder of the thesis.

# **Chapter 4: Research Methodology**

## **4.0: Introduction**

The purpose of this chapter is to look in detail at the methodology of the present research. It begins by discussing the major research traditions in the field of educational research, their origins and the nature of the questions that they pose. My choice of methodology is then related to my research questions, to my theoretical disposition, and to the research tradition within which much of the work of a similar nature is located. In the subsequent discussion, I consider key characteristics of my chosen research methodology in relation to my role as participant observer, the nature of the research site, the students involved and the collection and analysis of data.

## **4.1: Issues for Research Methodology**

### **4.1.1: Theoretical Perspective**

It is a truism to say that research methods should be selected according to purpose. Nevertheless, there are distinct traditions or paradigms in educational research which differ in assumptions, goals and primary research questions. The roots of these differences each be located in different research traditions: experimental psychology on the one hand, and cultural anthropology on the other (Smith, 1988; Eisenhart, 1988).

Most educational research, according to Eisenhart (op cit.), has been conducted in the general tradition of experimental psychology – a tradition characterised by its attempts to emulate the physical sciences (positivism). It embraces a conception of scientific method, aims for deductive universal laws, and adopts a neutral observation language. Schools tend to be viewed as quasi-laboratories where the students' environment can be constrained to promote the development of certain skills assumed to be good for everyone.

Educational anthropologists or ethnographers, on the other hand, have been trained to assume that human behaviour and human learning are responsive to a context that is interpreted by its participants and that is dominated by social relationships. The school is seen as an institution that organises meanings and social relations in particular ways. They argue that to restrict investigation of social behaviour to experimental settings is to

discover only how people behave in such settings. The search for universal laws is rejected in favour of detailed descriptions.

From a positivist perspective however, the methodology favoured by ethnographers is seen as subjective and particular. Since the observer plays a significant role in the research, they argue that there is no way of knowing whether another observer in the same setting would not yield different results. Moreover, since resources and time limit observers to one group, there is no way of knowing whether this group is representative of others.

These two traditions also differ in the nature of the research questions that they each pose Eisenhart (*op cit.*). Research questions in the psychological tradition tend to be derivatives of the general question: ‘How can mathematics teaching and learning be improved?’ The main goals of the research are descriptive and prescriptive: to identify psychological, psycho-social, or instructional factors and processes that affect mathematics education and then design and implement treatments to achieve better results. In contrast, she writes, research questions posed by educational anthropologists interested in mathematics tend to take the general form: ‘Why is mathematics teaching and learning occurring in this way in this setting? The goals are descriptive and theoretical: to identify the socio-cultural processes that constitute mathematics education in a particular setting and to make sense of this configuration through the development, modification or adoption of theories of culture or social relations.

Dichotomising traditions of enquiry serves to polarise their differences, but it does so at the risk of obscuring the variety of methods that lie in between. The research strategy employed in ethnography is participant-observation, but then all research involves some participation. Observing two students programming with a prototype language on a commercial computer at the back of a classroom, as I intended to do, shared characteristics of an experimental approach. It was not ‘natural’ in the sense that the term tends to be used in the ethnographical literature. There were limitations in the extent to which the research intruded on the social relations of that classroom still less of the school. My research questions also suggested a cognitive bias. On the other hand, I felt that other factors disposed my research towards an ethnographical approach.

#### **4.1.2: Pre-conceptions**

The style of research that I was interested in was not the sort that sets out to prove or disprove a hypothesis. I had an area of interest which was to investigate how working in the computational medium of Boxer would affect students' developing sense of number. I also had a series of 'working questions' in order to provide a focus, but simultaneously I wanted to leave myself free to explore and report how Boxer was mediating the students' expression of number. Moreover, I was beginning my research from a Logo baseline and shared the constructivist paradigm within which much of that research was located. Aggregating these factors, I felt that the most appropriate research method would be one that tended towards an ethnographical approach.

But, there were also many unknowns. There was, for example, no guarantee that the Logo wisdom, accumulated over the previous decade, could be built on in a Boxer setting. Nor was it clear whether the technical problems of programming in Boxer and using a Sun work-station were within the capabilities of primary school students. There was also a question of access: how to best to deploy one computer in a typical primary classroom of 30 students. Finally, there were the mathematical unknowns. I wanted to adopt a holistic approach to my investigation of students' understanding of directed numbers and decimals – areas which were at, or beyond the limits of the normal primary school curriculum. It was not at all clear whether this was a viable objective. These then were the 'sensitising concepts' (Hammersley and Atkinson, 1983) at the beginning of the search for a research site.

#### **4.1.3: Research Setting**

My study was part of a broader Boxer project involving two other researchers so the criteria for selecting a school reflected the needs of this overall project. It included proximity to the University, a degree of computer literacy amongst the participating teachers, enthusiasm from these same teachers to undertake such a project and a willingness to accept research workers into their classrooms. There was also a requirement that the teachers concerned would participate in an induction programme prior to the beginning of the research.

The selected school is an inner-London primary school, little more than two miles from the University. The pupils attending the school come from a wide range of ethnic and social backgrounds. It is a two-form entry school, meaning there were two parallel classes in each year, with approximately 30 students in each class. The ethos of the

school is purposeful, disciplined and yet relaxed. The students, for example, address all members of staff including the Headteacher by their first names. While this might be regarded as an incidental idiosyncrasy of the school, in practice it meant that the role of participant-observer was that much easier to assume.

Both of the year 5 and year 6 class teachers were experienced practitioners who had some knowledge of Logo. In each of their respective rooms, there already was a single computer which was used for Logo, word-processing, drawing packages etc. At the point when the computers were placed in the school (January, 1991), the year six class had four terms experience of Logo and the year 5 had two terms experience. So, it would be reasonable to say that at the outset of the project, the teachers and students were familiar with Logo.

During the orientation week at the University, it was decided to place one of the two Sun computers, on which Boxer was mounted, in the year 5 room and the other in the year 6 room, and to place the laser printer – which could print out pictures of the screen for either computer – in the year 6 room. In English primary schools, it is usual to find students working in small groups, often on different activities linked by a common theme such as ‘Weather’ or ‘Space’. We agreed that the most promising way of introducing Boxer would be to integrate it with these activities and themes. Thus, while the classroom teachers carried on with the normal business of the class, the researchers would observe pairs of students programming in Boxer at the back or side of the room. In the absence of researchers, the teachers themselves would attempt to continue with the Boxer work.

#### **4.1.4: Field Relations:**

The expectations which the hosts have about the researcher’s identity and intentions can have serious implications for the amount and nature of the data collected. These may well lead to anxiety on the part of those under surveillance. As Hammersley and Atkinson (1983) point out, the challenge is to establish an affinity with the host while maintaining a critical distance.

I sought to allay these fears by playing down my role as university lecturer, stressing instead my (former) classroom teacher identity. Anxieties about mathematics were reduced by focusing on Boxer, and anxiety about Boxer were reduced by stressing that Boxer was new to all concerned. Occasionally because I was present, the teacher felt free to leave the room. On such occasions, students might engage in behaviour which

was clearly in breach of normal class rules. This posed a dilemma for me: whether to ignore the behaviour or to assume the mantle of teacher? Each course of action carried dangers for the relationships which I had built up with teachers and students. This was resolved whenever possible by commenting on the behaviour, but also deferring to the authority of the class teacher. In a sense, this was part of the wider tension between the observer role and the participant role, as summarised in the following diagram:

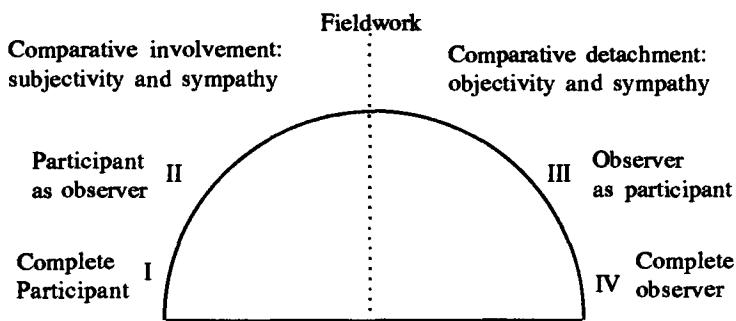


Figure 4.1: Theoretical social roles for fieldwork  
(Junker, 1960, from Hammersley and Atkinson, 1983, p. 93)

My role as participant-observer fluctuated during the course of the research. During the introductory phase of the longitudinal study (chapter 6), I was participating in that I was helping students overcome difficulties with software and hardware. In the later number studies (chapters 8 and 9), I was more inclined towards the observer role as I wanted to take a more detached view of how the students were utilising their Boxer programming knowledge, and how interacting in the Boxer was mediating their sense of number. However, it would be true to say that I was much more inclined to act as a complete observer with regards to the number investigations than with respect to programming. In the latter case, I still had to intervene occasionally when students reached an impasse.

## 4.2: Data Collection

In ethnography, the analysis of data is not a distinct stage of the research. Theory building and data collection are linked throughout the research. The intention of this section is to not only detail the collection of data, but to also indicate also how that process influenced the research and, of course, the nature of subsequent data collection.

#### 4.2.1: Research Timetable

For convenience, the account of data collection is divided into two distinctive stages. The first of these, the exploratory stage, describes the process leading to the progressive focusing of the research (reported in chapter 5). The second stage, the longitudinal study, denotes the period over which I observed four pairs of case study students. It is in turn sub-divided into distinct phases: a programming phase (reported in chapter 6) and a number phase (reported in chapters 8 and 9). These two phases were linked by a series of semi-structured interviews away from the computer with Boxer and non-Boxer students (reported in chapter 7).

Table 4.1: Timetable for the Research

The Exploratory Study Year 1	The Longitudinal Study		
	Year 2	Year 3	
	Programming Phase	Interview Phase	Number Phase
Jan. 1991 to July 1991.	Sept. 1991 to July 1992	Sept. 1992 to Dec. 1992	Jan. 1993 to July 1993
Class A (Y6, D)*			
Class B (Y5, M)			
	Class C (Y5, M)	Class C (Y6, D)	Class C (Y6, D)

\*The symbols in brackets denote the year group and the teacher.

It was common practice for the teachers to be assigned to a year. Thus, they might remain year 5 or year 6 teachers for a number of years. The students, on the other hand, changed teachers each year. In the original planning of the project, the intention was that the students would first experience Boxer with teacher M in the fifth year and then continue it with teacher D in the sixth year. As can be seen from Table 4.1, only class C (the one selected for the longitudinal study) achieved this.

Table 4.2: Timetable of Data Collection

Period	Activity	Data
Exploratory Stage Jan. 1991 – July. 1991	<ul style="list-style-type: none"> <li>• Exploring how students with Logo experience learned to program in Boxer.</li> <li>• Exploring how they learned number through its use.</li> <li>• Developing a classroom pedagogy for introducing Boxer.</li> </ul>	(a) Annotated print-outs (b) Written responses by teachers and students on Boxer.

<b>Longitudinal Stage Programming Phase</b> Sept. 1991– July 1992	<ul style="list-style-type: none"> <li>Investigating how students with no experience of Logo learned to program in Boxer.</li> <li>Investigating how different representations in Boxer affected students' conceptions of number.</li> <li>Investigating the issues involved in using the medium to introduce itself.</li> </ul>	<p>(a) Annotated print-outs of: First-Boxer, Second-Boxer; and Third-Boxer.  (b) Videos of case study students.</p>
<b>Semi-structured Interviews</b> Sept. 1992 – Dec. 1992	<ul style="list-style-type: none"> <li>Investigating students' interpretations and representations of number.</li> </ul>	<p>(a) Written responses by students on number.  (b) Audio recording of Interviews with Boxer and non-Boxer students.</p>
<b>Number Phase</b> Jan. 1993 – July 1993	<ul style="list-style-type: none"> <li>Investigating the issues involved in having students program their own microworlds.</li> <li>Developing a Boxer 'operational' approach to the investigation of number concepts.</li> </ul>	<p>(a) Annotated print-outs of: 'Number Line' and 'Function Machines'  (b) Videos of case study students and others.</p>

#### 4.2.2: The Exploratory Study

The exploratory study was necessarily broad in its scope, and as might be expected, it was 'loosely defined' in terms of goals (see Table 4.3). Some goals were set within the 'Turtles' microworld; others were chosen by the students themselves, or were in response to the current theme, while the remainder were suggested by the teachers or myself. An analysis of the 28 sessions observed during this period, reveals a breakdown of goals as follows.

**Table 4.3: Task Domains for Phase 1 of Exploratory Research**

Category	Number of sessions
Working mainly within the 'Turtles' microworld	8
Goals chosen by the students	9
Goals chosen in part or full by teachers/researcher	3
Theme-related tasks	8
Total	28

These categories are not discrete. Often, teachers or I would suggest extensions to tasks begun in the 'Turtles' microworld or started by the students themselves.

Data was recorded in the form of screen print-outs at approximately 20 minute intervals or after students had made a significant change to the screen display. In the latter case, this might be as a result of a student's program not performing as expected, or it might show a significant event in their programming, e.g., the use of modularity.

These print-outs were annotated to record what was happening. The screen print-outs therefore act as ‘markers’. They do not capture all that was happening – even on the screen. The annotations note what was happening on the screen as well as what the participants were doing and saying away from the screen. They also record my insights at that particular time. The following example gives some idea of the nature of these annotations:

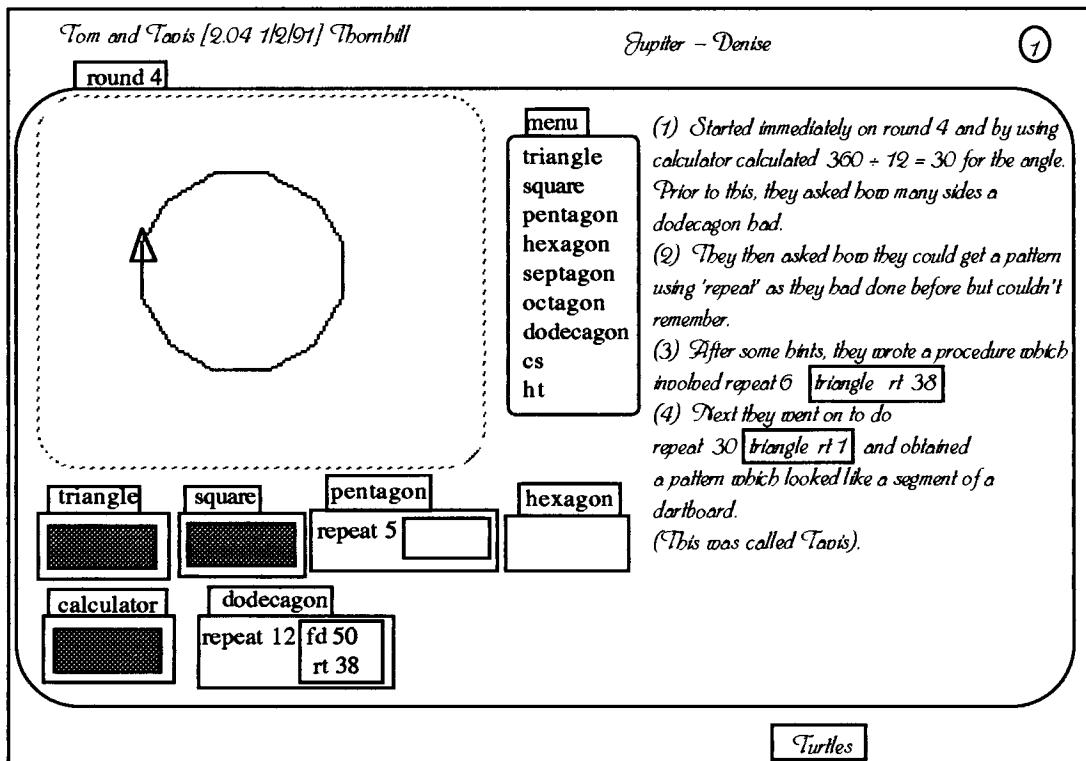


Figure 4.2: A typical print-out with annotations

The date, time and names of the students were noted on each page of the print-outs. The annotations were also numbered and times were noted against key events. Thus, it was possible to get some insight into how progress was made. In the longitudinal study, these print-outs were supplemented by systematic video recordings of the sessions.

#### 4.2.3: The Longitudinal Study

The longitudinal study consisted of three phases: a programming phase, an interview phase and a number studies phase. The research methods adopted in each phase were different, although all were characterised by case-study research.

#### 4.2.3.1: Programming Phase

The programming phase combined case-study research with exploiting the medium as a means of introducing itself (see chapter 5). I began by designing an introductory programming microworld ‘First-Boxer’. This was later followed by ‘Second-Boxer’ and ‘Third-Boxer’. At first, I observed eight pairs of students, four pairs from the year 5 class and four pairs from the year 6 class. My original intention was to work with and video all eight case study pairs, but this turned out to be over-ambitious in terms of the time needed. So, I decided – in consultation with the class-teachers – to work primarily with the four year 5 pairs. My intention now was to video all of the year 5 students working through the microworlds. As can be seen from Table 4.4, this was not quite achieved.

Table 4.4: Video Record for Programming Phase

Name	First-Boxer	Second-Boxer	Third-Boxer
Liam & Jason	✓	✓	✓
Nico & Joanna	✓	✓	✓ J with Faye J on own
Courtney & Laura	✓	✓ L with Meliha	✓
Jessie & Oliver	✓	✓	✗

#### 4.2.3.2: Semi-Structured Interviews

The exploratory stage had given some clues to how number activities might be structured in Boxer, but these were nevertheless still vague when the programming phase came to an end. I therefore decided to interview a group of non-Boxer students as well as the Boxer students with the intention of progressively focusing in on how the investigation of number could best be carried out in the medium.

Three distinct groups of students were interviewed: a pilot study group of eight students from the parallel year six class, a second group of eight students from the same class, and ten students from the Boxer class. This latter group included all of the case study students (seven at this stage), and three additional students who had some Boxer experience, a total of twenty six students. This study employed a series of semi-structured interviews. The students were first interviewed in pairs or individually (the format was modified in the course of the study). A second set of interviews was then

conducted in which the students in groups discussed certain of the representations arising from the earlier interviews.

With the pilot-study group, I began by attempting to engage all of the students in a discussion, in which members of the group critiqued each other's representations in a constructive way, in much the same way as the students in diSessa et al.'s (1991) study. However, this format was not entirely successful for three different reasons. Firstly, it was difficult to manage a group of eight students while allowing each child to express her/his view. Secondly, because the diagrams (at this stage) were labelled with their names, some students were unwilling to criticise drawings completed by their friends. Finally, these students – unlike those in diSessa's study – were not used to discussing each other's ideas. With the comparison and Boxer students, the follow-up discussions were conducted in groups of twos or fours (chapter 7). These interviews were audio recorded. A list of the students involved is given in Table 4.5.

**Table 4.5: Year 6 Students interviewed on Number**

Comparison Group	Boxer Case-study students	Boxer non-Case-study students
Ann-Maria Tahir Blake Hannah Ahmet Cerisse Rebecca Dillon	Nico Joanna Liam Jason Courtney Laura Jessie	Kathleen Rosie Sara

#### 4.2.3.3: The Number Studies

The exploratory study indicated the merits of involving students in the construction process, but it gave few clues as to how this might best be done. The data furnished by the semi-structured interviews supplied some of these clues. It pointed the way towards an 'operational' approach to number and moreover attested to the number line as being a potentially evocative representation. The number line study (chapter 8) was designed to put these ideas into practice. The function machine study (chapter 9) supplemented that study and helped to triangulate its findings.

In each of these studies, the students constructed their own microworld. In the first case, the number line goal was set by me. In the case of the function machine study, the goal was set by the students themselves. Each of these computational objects then became the means through which I observed students interactions with number.

Table 4.6 shows the students who were video-recorded on each of these:

**Table 4.6: Video Record for Number Phase**

Names	Number Lines	Function Machines
*Nico & *Joanna	✓	✓
*Liam & *Jason	✓	✓
*Laura & *Courtney	✓	✓
*Jessie & Kathleen	✓	
Rosie & Chloe		✓
Natalie & Sara		✓
Naomi		✓
Total number of students	8	11

\* (Original case study students.)

### 4.3: Case study students

The case study students were chosen by the class teacher to be representative of the mathematical ability range (one pair from each quartile) and to be balanced in terms of gender, but how she measured ‘ability’ was not known. Probing the teacher’s categories might have been seen as questioning her professional judgement. In any case, it was not clear what could be gained by doing so. It later transpired that she had applied another criterion of her own, in that she balanced the pairs in terms of their perceived social class.

**Table 4.7: Composition of the Case Study Pairs (September 1991)**

Name	Age, DOB	Boy/Girl	Maths Quartile
Liam	9, 27-11-81	Boy	3
Jason	9, 31- 7-82	Boy	2
Nico	9, 19-10-81	Boy	1
Joanna	9, 9- 9-81	Girl	1
Courtney	9, 28- 5-82	Girl	2
Laura	9, 23- 3-82	Girl	3
Jessie	9, 19- 5-82	Girl	4
Oliver	9, 10- 8-82	Boy	4

The teacher also selected the case study pairings. In some cases (Laura and Courtney, Liam and Jason), these were based on existing friendships. In other cases (Jessie and Oliver, Nico and Joanna), the teacher had a different agenda. In the former pair, the students were perceived as struggling with classroom work generally and working with Boxer was thought to be one way of ‘bringing them along’. In the latter case, Joanna

was paired with Nico, because the teacher thought that while each child was of comparable high ability, Joanna lacked confidence and she hoped the partnership would benefit her.

## 4.4: Data Analysis

### 4.4.1: The Interviews

All interviews are structured by *both* researcher and informant. The important distinction is between standardised and reflexive interviewing. In the latter case, questions are frequently generated by previous responses. In reacting to what the students did on the computer, I would ask questions such as: "Why did you do that?", "What do you think that means?", "What do you think might happen if ... ?" – in other words, unstructured or reflexive interviewing.

In recording the sessions, I was conscious of the need to capture what was said – the 'situated vocabularies' rather than translations of them. Thus, the significance of a student's utterances such as "Point fifty four." could be incorporated in the emerging theory. But, I also became alert to my own practice in referring to quantities such as 0.54 in several different ways: "Point five four.", "Nought point five four.", "Point oh five four." and even "Nought point oh four."

The success of such interviews depends not only on the relationship between the two parties but on assumptions made about the interviewer's knowledge. Informants may not 'spell out' what they mean because they may assume that the interviewer will draw on their background knowledge. For example, when Courtney and Laura were probed on the *meaning* of 0.2, Courtney replied: "It's a decimal." I responded: "Suppose I pretend I don't know what a decimal is." A little later, Laura exclaimed: "It's one tenth of two.", a description that conveyed its operational origins and was thus of importance for this research. These were captured on video as well as on annotated print-outs.

The use of video solved some problems but led to others. The original aim was to capture what was on the screen as well as what the students were doing away from it. But problems in matching a British video camera to an American computer meant that a clear picture of the screen could only be obtained by focusing the students out of the picture. The decision was therefore made to place primacy on what the students were doing, and to rely on the annotated print-outs for relating this activity to the screen display.

Using a video camera made it possible to more accurately locate the context in which an utterance was made. It demonstrated, in a way that would be difficult for words to capture, the physical interaction between the students and the various components of the Boxer environment on the screen. In chapters 8 and 9, for instance, the students' experiences in interacting with the Boxer number line and the function machine are analysed. But the words used therein cannot capture all of the finger-pointing between different parts of the graphical objects and the corresponding symbolic expressions in the program's doit box.

The use of the video recorder in the main body of the study enabled me to engage in reflexive interviewing in a way that would not have been possible had I relied solely on note taking. This explains why in chapters 8 and 9, the activities which the students pursued are seldom the same. A response which gave an opening into, say decimal quantities, could profitably be followed up. Likewise an incorrect response became not a dead answer but a 'strategic aperture' (Hoyles and Noss, 1992b) which could be exploited in the Boxer environment to yield more information and guide the next step in the interview. The presence of the video as a supplement to note-taking also meant that the nature of note-taking changed from being primarily a record of students' work to being more akin to an analytic memo. Such notes in turn led to preliminary hypotheses being refined as well as the generation of new hypotheses.

#### **4.4.2: Triangulation**

Technique triangulation can be achieved by comparing data produced by different techniques. In the computer-based interviews, the use of field notes together with the video record meant that there was an on-going form of technique triangulation. The number inferences made as a consequence of the semi-structured interviews were later tested through reflexive interviewing in the number line and function machine study. Moreover, the findings from function machine study acted as a form of triangulation with those from the number line study. There were differences as well as similarities, but these differences were also of importance to the emerging theory (chapters 8 and 9).

Another form of triangulation relates to respondent validation. Its value lies in the fact that the participants involved in the events may have access to additional knowledge of the context that is not available to the ethnographer. To this end, I invited both teachers to give their accounts of the Boxer work. The teachers themselves conducted a survey of the students' feelings about Boxer. Finally, at the end of the research I conducted a

semi-structured interview with the students about their feelings of working with Boxer. This was audio recorded.

Over the period of the research, I built up a relationship of trust with the teachers. I was thus able to compare the written accounts with what had been said or observed over that period. There was little in the different accounts to suggest any discrepancies in the data. Although not commented upon, either verbally or in writing, I did observe a marked preference, from students and teachers, for elaborate visual goals. This may relate to the primary school tendency for displaying work.

#### **4.4.3: The process of Analysis**

In the longitudinal study, my initial organisation of data was in terms of the students observed. This allowed a temporal recounting of what happened, but made it difficult to see what was common across groups. For the programming phase, the data was reorganised through categories derived from previous Logo research, e.g., ‘procedures’, ‘variables’, and so enabled me to compare students’ Boxer programming with Logo. At the same time, I imposed a second categorical layer, derived from the organisation of the microworlds, to highlight issues involved in using the medium to introduce itself.

Similarly, the organisation of the data from the number phase also underwent several changes. Initially, this too was in terms of the students observed. I then attempted to organise it in terms of school topic boundaries in mathematics. While this permitted a comparison with previous research, it moved too far away from what did happen, where students frequently mixed numbers and operations, and thus reinforced the reification of topic boundaries. In the current framework, the data is organised in terms of the operations used to introduce the numbers. It thus reflects the number line or function machine interactions, and goes some way towards showing how these Boxer objects mediated students’ expression of number.

Although no number of confirming instances can ever guarantee the validity of a theory, the chances of its acceptance can be increased if we specifically choose to study those cases where, because of the strength of rival explanatory factors, it seems least likely to be proved correct (Hammersley and Atkinson, 1983). My original choice of the traditionally non-visual area of number can be seen as exemplifying this notion in terms of the technical component of the research, while the decision to work with the full ability range can be seen as doing the same in terms of the students.

Each posed challenges. In the first case, it was some time before an obvious way of utilising the medium to visualise number emerged and when it did, it became apparent that it would necessitate considerable programming skills. In the second case, the early evidence from Jessie and Oliver seemed to suggest that the demands of Boxer were proving too difficult for students representing the bottom quartile of the ability range, and the teacher and I reluctantly agreed to drop them from the research. In the following academic year, when Jessie re-joined the research with her new partner Kathleen, some very unexpected differences were observed. It was apparent that Jessie, although from the bottom quartile like Kathleen, not only had a better understanding of programming, she also had a much more sophisticated insight into numbers than her partner (chapter 8).

The context in which the observations were made must also be made clear. There was no way of knowing, for example, whether the students interacted with the classroom teacher in a similar way when I was not present. Working with groups of one to three students on a computer gave me insights into students' reasoning that had never been possible in my own teaching career (with classes of 25 to 30 students). More precisely, I was able to observe how their interactions in the Boxer computer environment was shaping their emerging sense of number. None of these facts – the effect of the researcher, the small group sizes and the Boxer environment – detract from the status of the data or theory, but each serve as warnings against generalising the findings beyond such contexts.

#### **4.5: Summary**

The problem investigated in this thesis was to identify and examine how working in the computational environment of Boxer would mediate students' developing sense of number. The research concentrates on three inter-related aspects of the problem. Firstly, to study the programming activities of the students to gain insights into the kinds of learning involved. Secondly, to investigate their interpretations of number and to probe which representations were evocative for these students. Thirdly, to examine utilising the students' programming experiences as a basis for extending their developing sense of number. Clearly, an investigation of these areas could take place in several different ways. This chapter has detailed the important characteristics of the methods that I chose to use.

My joint decision with the classroom teacher to seek a method which ensured equitable access to all the students looks, at first sight, to be peripheral to the main research question. However, it proved to be crucial in several respects. Firstly, it showed how using the medium to introduce itself could overcome the obstacles posed by a shortage of computers in a normal classroom. Secondly, it moved the classroom teacher from being a user to being involved in the design process. Thirdly, giving the other children access to Boxer not only overcame the ethical question raised earlier, it proved important in the later function machine study when several of those children participated in the research.

The constructivist basis of this study has been alluded to throughout the previous chapters. The semi-structured interviews, which served as a bridge between the programming phase and the number studies, sought to build on and build in the knowledge which the students in this research had regarding number. Using a comparison group of students helped to accentuate differences between the two groups conceptions of number and thus point towards the research methods for next phase.

The two number studies of this phase of the research were characterised by two features. Firstly, each involved students not merely interacting in microworlds constructed by someone else, but in constructing the microworlds themselves. Secondly, the object of their construction was operational. Thus, the product of their construction became the agent through which number processes might be reified.

In chapter 5, the exploratory study is analysed. It gives an account of how students who had previous Logo experience adapted to Boxer. It also shows how my initial ideas for introducing number in the medium evolved.

## **Chapter 5: Exploring the Medium**

### **5.1: Introduction**

This chapter discusses the exploratory stage of the research, which lasted from January 1991 to July 1991. The intention was to use this period to generate more focused and coherent issues for the second longitudinal phase. Two groups of students were observed: a year 5 class (ages 9 – 10) and a year 6 class (ages 10 – 11). The class teachers were both experienced teachers of Logo, and all of the students had previous Logo experience. This stage was characterised by three aspects:

- distinctive strategies used by the students to learn the fundamentals of programming in Boxer;
- the creation of microworlds to explore how visual representations in Boxer mediated students' expression of number;
- the development of a pedagogical approach by the class-teachers in conjunction with the researcher.

### **5.2: Exploring Programming in Boxer**

During the course of this study, students were observed programming in Boxer in ways that differed to that reported in comparable Logo research (Hoyles and Sutherland, 1989). There were similarities too, which is hardly surprising since Boxer was designed as a successor to Logo (diSessa, 1986a). To take account of these differences and similarities, three categories of programming, derived from Hoyles and Sutherland's (1989) study with Logo will be utilised. These are: 'working at a syntactical level', 'making sense of', and 'goal-directed activity'.

#### **5.2.1: Working at a Syntactical Level**

Hoyles and Sutherland (*op cit.*) categorise this activity as consisting of the syntactically correct use of Logo primitives, procedures (or sequences of these) but with a focus on obtaining a screen output without any apparent reflection on how or why the output was achieved. In this Boxer study, the temptation to engage in such activity was exacerbated both by the facility for repeating a sequence of actions (pressing the linefeed key), and by the facility for getting an instant screen print-out on the classroom printer.

### Episode 1 – Focus on Product

This episode has been selected to show how Boxer's simplicity enabled Luisa and Sukey to obtain a complex visual pattern, and how these powerful visual effects became a goal in themselves.

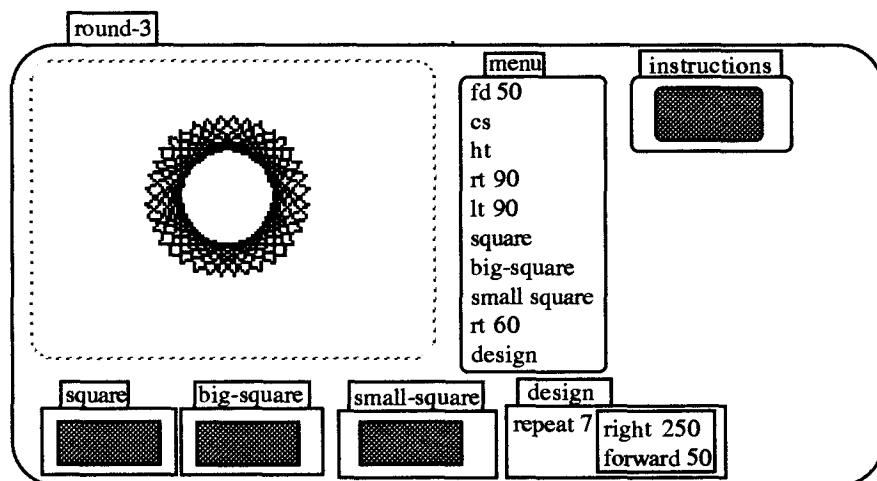


Figure 5.1: The ‘saw-tooth’ pattern

As part of the ‘turtles’ microworld (chapter 4), the students were invited, by an on-screen message, to complete a procedure called ‘design’ (Figure 5.1) which could then be used to create a screen pattern.

Two year 6 girls, Luisa and Sukey, typed ‘repeat 7 [right 250 fd 50]’<sup>1</sup> in the doit box called ‘design’. On running this, they obtained an ‘incomplete’ pattern, so they continued to execute it (by pressing the ‘linefeed’ key) until they made a ‘saw-tooth’ pattern (Figure 5.1). Later, they changed the input to ‘repeat’ from 7 to 40 and ran the program again. This was the first instance of the often observed practice of students combining ‘linefeed’ with ‘repeat’ to obtain ‘whizzy’ effects. The girls did not seem aware that the input to ‘right’ signified an angle, nor did they reflect on the relationship between this part of the program and the visual effect on the screen. At this point, Luisa and Sukey seemed only concerned to obtain striking visual patterns.

Against this, it could be argued that, although the students may not have fully comprehended the relationship between process and product, the product itself was highly motivating. Evidence to support this contention could be inferred from the way

---

<sup>1</sup> Note square brackets [ ] will be used in text to represent a doit box in Boxer.

they showed their print-outs to other students and by their requests to take them home to show to their parents. It was also noticeable that the desire solely for spectacular visual effects was a feature of their early encounters with Boxer and tended to fade as they gained more experience.

### 5.2.2: ‘Making Sense Of’

For Hoyles and Sutherland (op cit.), ‘Making Sense Of’ is exploratory activity in which students try out a new idea or procedure and reflect on what is happening. Sometimes such activity is completely non-goal directed, sometimes it takes place within goal-directed activity and sometimes a goal emerges from the activity.

#### Episode 2 – Attaining a goal by a Visual strategy

This episode has been selected to show how Boxer allowed two year 5 boys Finneas and Tariq to utilise a visual means of achieving a goal that might not have been attainable to them through analytical means.

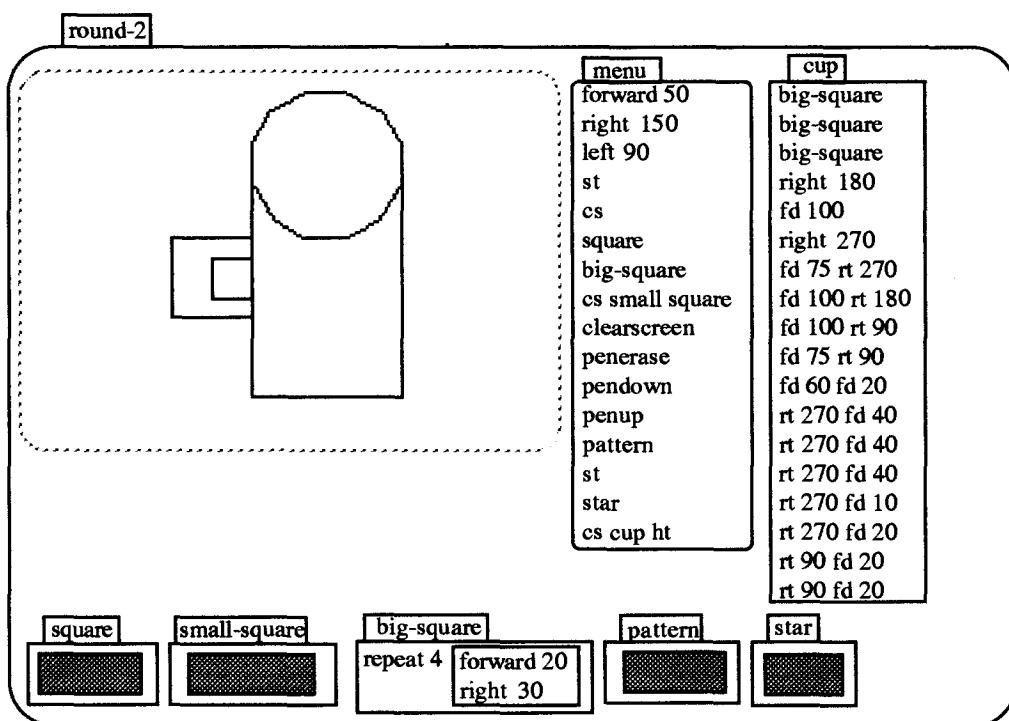


Figure 5.2: Finneas and Tariq’s cup doit box and drawing

Like, the previous episode, this also took place within the ‘Turtles’ microworld. An on-screen box entitled ‘big-square’ contained the line ‘repeat 4 [forward 20 right 90]’. The two boys changed this so that it read ‘repeat 4 [forward 20 right 30]’. They then

executed this line three times and – to their surprise – obtained a dodecagon. Noting the resemblance between this polygon and the top of a coffee cup, drawing a cup now became their goal.

They then began to type and execute commands in one of the boxes on the screen. At this point, their programming could be classified as ‘direct execution’. When they were sure that a command was doing what they wanted, they typed it inside a procedure called ‘cup’ (see Figure 5.2). In this way, they were able to tell where their program worked up to. They decided on dimensions for the various components of the cup by a ‘guess-and-test’ process. Mistakes or other unwanted effects were overcome by re-writing the commands for the most recent component and then running the line ‘cs cup ht’ from the menu.

It seems that for some parts of their program, they were operating at a syntactical level rather than reflecting on how or why the output was achieved. Some evidence for this could be seen in their use of ‘big-square’ three times for the top of the cup, in their use of ‘rt 270’ instead of ‘lt 90’, and in their later typing of ‘rt 270 fd 40’ four times even though they had used the ‘square’ procedure earlier on. On the other hand, they were pursuing a goal which had been generated by their earlier activities and one which they were unlikely to attain by analytical means. For example, they had no choice but to employ a ‘homing-in’ strategy to draw the bottom of the cup, since the mathematics of calculating the ‘diameter’ of a dodecagon with sides of 20 units and an external angle of  $30^\circ$  was simply not within their grasp.

#### Episode 3 – ‘Borrowing’ chunks of programs to attain a goal

This episode has been selected to show how a semi-structured Boxer environment gave two year 6 boys Tom and Tavis a ‘wedge-in’ to more sophisticated programming, and how the programming activity motivated them to engage in mathematical activity.

In ‘round-4’ of the ‘Turtles’ microworld, procedures for drawing a triangle and a square were given and students were invited to complete procedures for pentagons and so on (Figure 5.3).

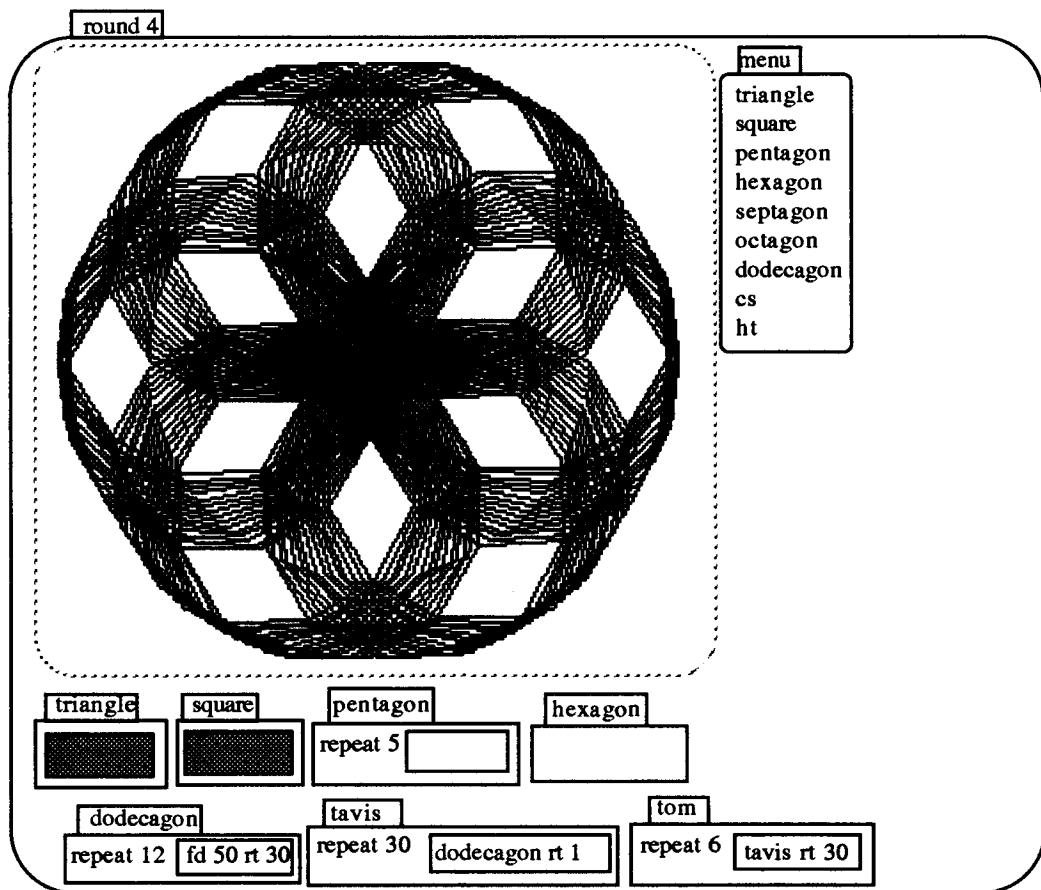


Figure 5.3: Tom and Tavis's dodecagon pattern

The boys began by asking how many sides a dodecagon had. On being told '12', they used a calculator to work out  $360 \div 12 = 30$ , opened the 'dodecagon' doit box, wrote the program 'repeat 12 [fd 50 rt 30]' and then ran it. Earlier, the boys had obtained a rotational pattern by running the procedure 'repeat 6 [triangle rt 60]' and they now tried to do the same with 'repeat 6 [triangle rt 38]'. Next they wrote 'repeat 30 [triangle rt 1]' and on running it, obtained a pattern which looked like a sector of a dart-board. This was called 'tavis'. Finally, they created a super-procedure 'tom' with which they rotated 'tavis' to get a full 'dart-board'.

They continued to experiment, first replacing the triangle in 'tavis' with a square and running 'tom', and then repeating this process in turn with a pentagon, hexagon, heptagon, octagon and finally a dodecagon (Figure 5.3). For each of these polygons (other than the square), they had to calculate the angle of rotation. Frequently, they would ask how many sides a particular polygon had before computing  $360 \div n$  on their

calculator. The connection between the number of sides  $n$  and the angle of turtle turn  $360 \div n$  appeared to be unproblematic.

Tom and Tavis's were motivated throughout this activity by their goal of obtaining spectacular visual effects. But this goal prompted them to use more structure in their programming than might otherwise have been the case. It also encouraged them to engage in the mathematical activity of working out the angles of polygons. For this study, it gave clues as to how similar semi-structured environments might be employed in the second phase of the research.

### **5.2.3: Goal-directed Activity**

In the third category of programming identified by Hoyles and Sutherland (op cit.) the activity is aimed at achieving a specific goal whether posed by the pupil or teacher.

#### Episode 4 – Teacher Intervention

This example has been chosen to show how a goal chosen by two girls Anoushka and Natasha was greatly enriched by appropriate intervention by the class teacher.

The girls set themselves the goal of drawing a house. They adapted procedures given in round-7 of the ‘Turtles’ microworld in order to do so. ‘Square’ was used for the body of the house, and ‘triangle’ re-named ‘roof’. They added ‘orientation’ commands to position and combine them. The girls then began to draw the door and the windows by typing and executing commands in the menu, using ‘penerase’ to correct errors.

The failure of ‘penerase’ to work for print-outs together with the effort required to completely re-draw their house made them receptive to the teacher’s suggestion of a creating modules for each component of the house and for navigating between them. She also set the girls the task of creating the house in one super-procedure. The students worked through their lunch-time to obtain the house shown in Figure 5.4.

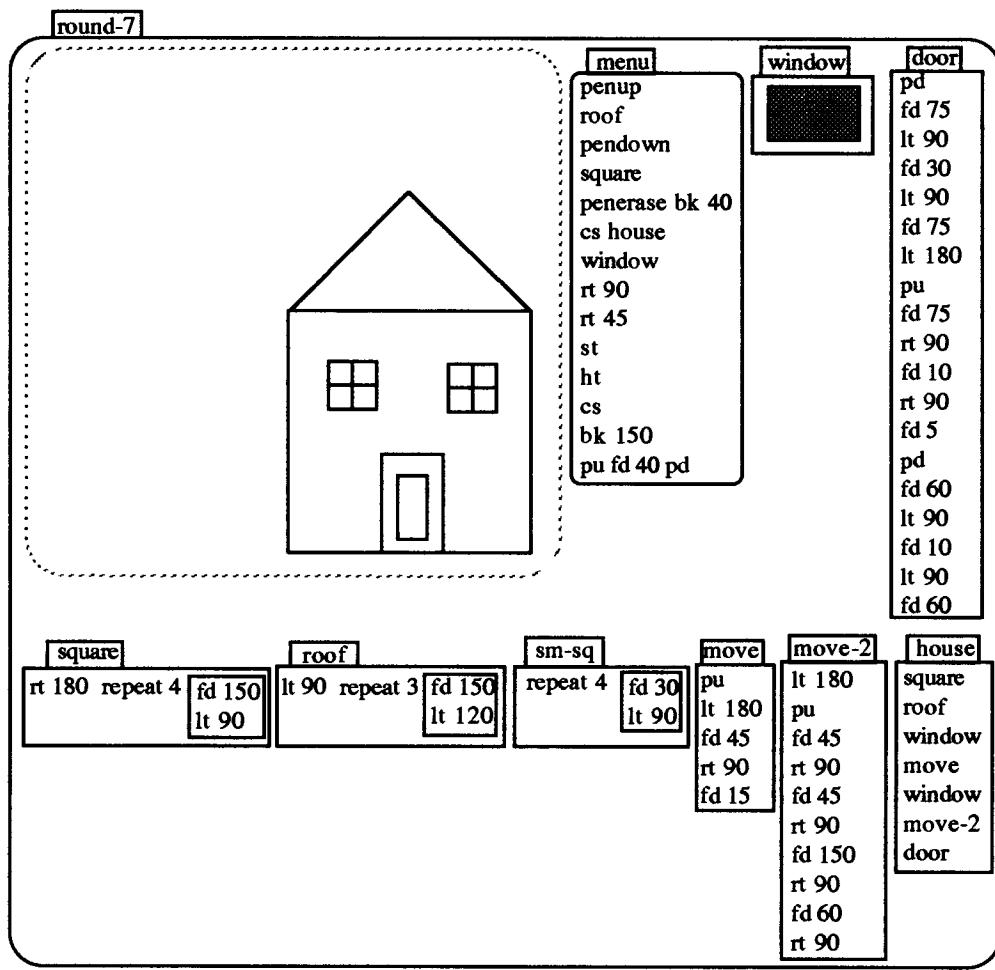


Figure 5.4: Anoushka and Natasha's house

The early part of Anoushka and Natasha's programming had similarities and differences to that seen frequently in Logo research. It was similar in that they executed commands directly. It was different in that they could easily adapt pre-written procedures. The latter part of their programming activities was quite different. The girls wrote separate modules for separate parts of their drawing 'roof', 'window' etc. and then used a super-procedure 'house' in which to combine them. The teacher's intervention was important in focusing attention on the merits of modularity, while the simplicity of creating such modules in Boxer meant the students were able to implement her suggestions.

Anoushka and Natasha goal of drawing a house drove their early activities. Their later actions were motivated by the goal of creating a procedure to draw a house in one move. The teacher's intervention was crucial in that it had moved the students' focus from the product to the process of their programming.

### 5.3: Exploring Number in Boxer

During the exploratory phase of the study, I investigated several different ways of exploiting the visual nature of Boxer for representing number. This section discusses one of these representations ‘line-numbers’ (Figure 5.5), in which parallel vectors were used to compare numbers. It was used only with class B (Table 4.1), i.e., the class who had Boxer experience, but were not involved in the longitudinal study.

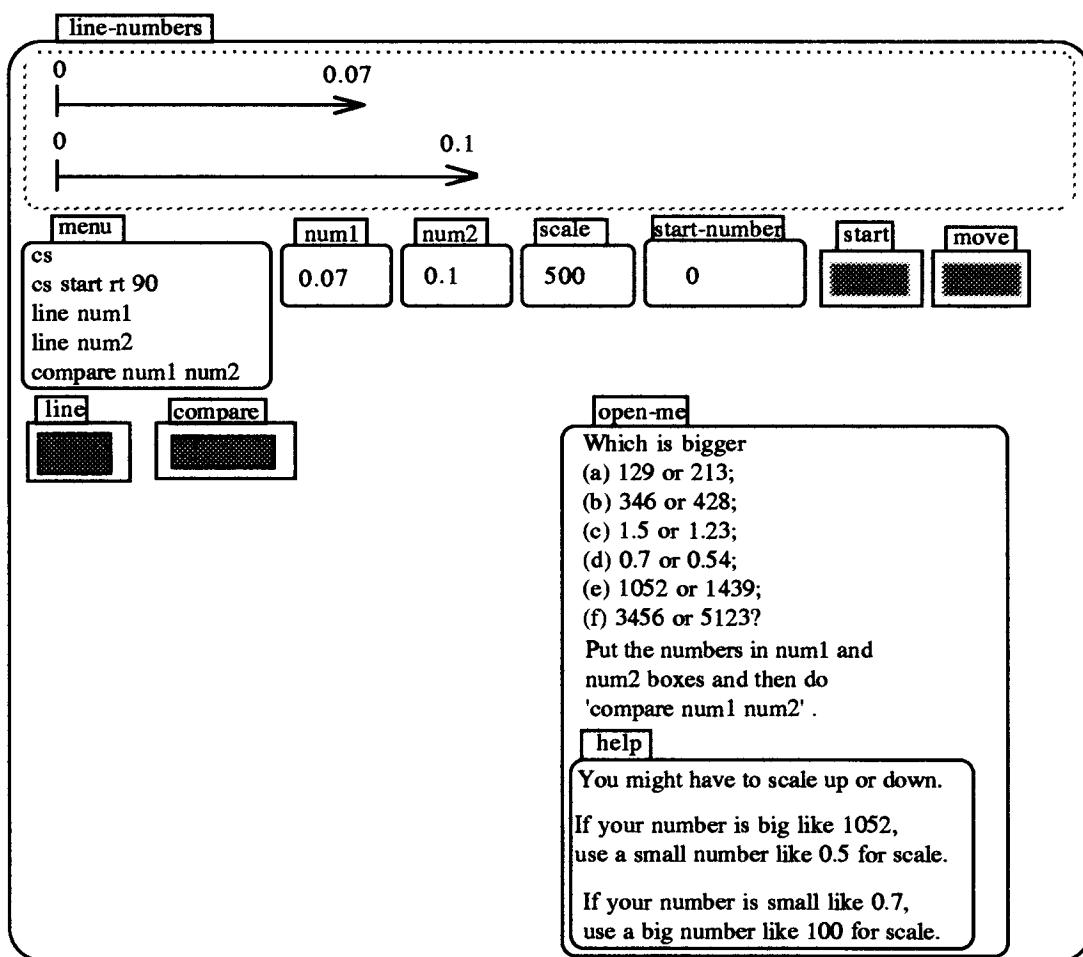


Figure 5.5: The ‘line-numbers’ representation

#### 5.3.1: Number Issues arising from the ‘line-numbers’ representation

I began by asking the students to answer each on-screen question. They were then invited to check their response by putting the numbers in the ‘num1’ and ‘num2’ boxes, and then run ‘compare num1 num2’ which drew vectors for each number. If the resultant drawing did not show up the differences clearly, the students were invited to either scale up or down. After the first session (with Leonie and Jermaine), the

microworld was modified by adding the ‘start-number’ box and adjusting the ‘line’ program. This enabled a better comparison of numbers, such as 1.12 and 1.2, which were very close (in value) to each other.

Of the screen questions, the problem which caused most consternation for the students was (c) comparing 1.5 and 1.23 (Table 5.1). But, although the other questions appeared unproblematic, caution needed to be exercised in interpreting correct answers as signifying understanding. Rather, answers needed to be analysed in conjunction with the students’ spoken responses. Sian, for example, answered (d) correctly, comparing 0.7 and 0.54, but reacted to the visual representation by saying: “How come that one is bigger and it’s only seven?”

**Table 5.1: Results from the on-screen Comparison Questions**

n = 6	(a) 129 or 213	(b) 346 or 428	(c) 1.5 or 1.23	(d) 0.7 or 0.54	(e) 1052 or 1039	(f) 3456 or 5123
	5	5	2	5	6	5

The students were now asked to set questions for each other, and to use the vectors to check them on screen. This was only partially successful, as the students’ lack of knowledge prevented them from choosing the most challenging questions, a fact that they themselves signified by asking me to set the questions. Their choices, responses, and actions are shown in tables 5.2, 5.3, and 5.4 along with extra questions set by me.

**Table 5.2: Leonie and Jermaine’s Number Challenges**

Questioner	Which is bigger	Response*	Scales
Jermaine:	(1) 0.07 or 0.1?	L: <b>0.07</b>	500
Leonie:	(2) 4089 or 2444?	J: 4089	0.5, 0.1
Jermaine:	(3) 1.12 or 1.2?	L: 1.2	100, 500
Leonie:	(4) 4.00 or 5.09?	J: 5.09	500, 0.1, 0.5, 100
Jermaine:	(5) 5.5 or 5.05?	L: 5.5	100
Leonie:	(6) 56.02 or 56.20?	J: 56.20	500, 0.1, 1.00, 5.00
Jermaine:	(7) 5.2 or 5.11?	L: 5.2	100
Leonie:	(8) 05.90 or 0.09?	J: 5.90	100
Jermaine:	(9) 111 or 111.1?	L: 111.1	0.1, 100, 20, 5
Interviewer:	(10) 0.0034 or 0.65?	L: 0.65 J: 0.65	100, 500, 1000
Interviewer:	(11) 0.054 or 0.21?	L: <b>0.054</b> J: <b>0.054</b>	100, 500
Interviewer:	(12) 0.01 or 0.1?	L: same J: 0.1	100, 500
Interviewer:	(13) 0.02 or 0.2?	L: 0.2 J: 0.2	100, 500

\*Incorrect answers are given in bold.

In the course of these interviews, I became more conscious of the importance of the links between the way students say decimal numbers and their understanding of them. Thus, 5.20 could be pronounced as “Five point two oh” or “Five point two nought” or as “Five point twenty”, but the latter articulation might indicate the established research finding of ‘Decimal Point Ignored’ (APU, 1985, p. 53). That research finding was replicated many times in this study, as can be seen in Leonie and Jermaine’s responses to question (11) of Table 5.2.

Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Which do you think is the bigger: point oh five four or point two one?	0.054 or 0.21
Jermaine:	Point oh fifty four.	Note expression.
Interviewer:	What about you Leonie, which do you think is the biggest?	
Leonie:	Point oh fifty four.	Again, note expression.
Interviewer:	Let’s have a look.	Runs program.
Both:	No.	They look at each other.
Jermaine:	It’s normally fifty four.	
Leonie:	It’s just the nought and fifty four.	

At the beginning of the interview with Ella and Steven, Steven said: “We’ve been doing fractions and stuff like this.” However, they too expressed decimals in ways that hinted at the DPI error.

Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	How about these: nought point one seven and nought point two?	Note the way the decimals were expressed.
Ella:	I think it is nought point two.	
Steven:	I think it is nought point seventeen.	Note the way Steven has changed the expression.

**Table 5.3: Ella and Steven’s Number Challenges**

Questioner	Which is bigger	Response	(Start No)* & Scales
Ella	(1) 453.5 or 454?	S: 454	(400), (450), 10
Interviewer:	(2) 0.17 or 0.2?	E: 0.2 S: 0.17	500, 1000, 5000
Interviewer:	(3) 0.314 or 0.67?	E: 0.67 S: 0.67	1000

Interviewer:	(4) 0.314 or 0.216?	E: 0.216 S: 0.314	1000
Steven	(5) 0.75 or 1.64?	E: 1.64	1000
Interviewer:	(6) 0.356 or 0.4?	E: 0.4 S: 0.356	1000
Interviewer:	(7) 0.3421 or 0.356?	E: 0.356 S: 0.3421	(0.3)1, 6, 12
Ella	(8) 99.5 or 99.1?	S: 99.1	(99) 5, 512
Steven	(9) 0.26 or 0.7?	E: 0.26	1000
Interviewer:	(10) 0.7 or 0.70?	E: 0.7 S: 0.70	(0.6) 1000, 5000
Interviewer:	(11) 0.5 or 0.50?	E: same S: same	1000, 5000, 50000
Interviewer:	(12) 0.3 or 0.30 or 0.300?	E: same S: same	5000
Interviewer:	(13) 0.03 or 0.7?	E: 0.7 S: 0.7	5000

\*The number in brackets indicates the start number if this is different from 0.

The APU finding ‘Largest is smallest’ (Foxman et al., 1985, p. 53) was put forward by Ella as a theory after running the program to compare 0.314 and 0.67 (question (3) in Table 5.3).

### Extract 3:

Speaker	Dialogue	Commentary
Ella:	I think the smaller it is the bigger it's worth.	
Interviewer:	Do you?	
Ella:	I don't know.	
Interviewer:	Well let's try another one then. You think the smaller it is the more it's worth? When you say the smaller . . .	
Ella:	The lower the number.	

After comparing the vectors for 0.314 and 0.216 (question (4) in Table 5.3) – chosen to confront her hypothesis – Ella said “It doesn’t depend on the size of the number.” by which she meant the number of digits in the number. Following question (9), I became aware that the students thought that numbers such as 0.5 and 0.50 were different. Questions (10), (11) and (12) were chosen to confront this belief.

Working interactively in the Boxer setting made it possible to immediately create visual counter-examples which in turn led to students generating new hypotheses. The pupils were not told that one quantity was greater than another, but faced with an on-screen representation which demonstrated that this was the case, they were motivated to reconsider their thinking about such numbers.

**Table 5.4: James and Sian's Number Challenges**

Questioner	Which is bigger	Response	(Start No). & Scales
James:	(1) 6751 or 5481?	S: 6751	1
Sian:	(2) 879 or 0.9?	879	1
James:	(3) 4371 or 4367?	S: 4367	100, (4000), (4300)10
Interviewer:	(4) .46 or .5?	S: .5	100, 500
Sian:	(5) .899 or .900	J: .899	(.850) 900, 5000, (.880) 10000
Jermaine: *	(6) 11.50 or 11.5?	S: 11.5	(10)10, (11.0)200
Interviewer:	(7) 0.213 or 0.45?	S: 0.45	500
Interviewer:	(8) 0.213 or 0.19?	S: <b>0.19</b> J: 0.213	500, 1000, 2000
Interviewer:	(9) 1.5 or 1.25?	S: 1.5 J: 1.5	2000, 200
Interviewer:	(10) 0.57 or 0.216?	S: 0.57 J: <b>0.216</b>	200
Interviewer:	(11) 0.475 or 0.4136?	S: 0.475 J: <b>same</b>	200
Interviewer:	(12) 0.5678 or 0.5681?	no clear answer	500

\*(Jermaine set question (6) for Sian and James.)

Sian, who despite having said correctly that 0.5 was bigger than 0.46 (question (4) in Table 5.4) reacted to the visual representation by saying “But how? It’s a lower number.” Later, having compared 0.213 and 0.45 (question (7) in Table 5.4) with the vectors, she put forward the hypothesis: “The lesser the number without the dot is it?”. The next question comparing 0.213 and 0.19 (question (8) in Table 5.4) confronted her hypothesis as her reaction shows: “But how come? Forty five was bigger.” After comparing 1.5 and 1.25 (question (9) in Table 5.4), she developed a new theory:

**Extract 4:**

Speaker	Dialogue	Commentary
Sian:	I think I know how to tell which one is bigger.	
Interviewer:	Do you?	
Sian:	If the front one is bigger the other must be ...	Refers to the digit immediately to the right of the decimal point.
Interviewer:	Aah. I see. What you’re saying is that if the number just after the decimal point is bigger ...	
Sian:	Yes. The first number.	
Interviewer:	Why don’t we test it on something?	

### 5.3.2: Design Issues arising from the ‘line-numbers’ representation

The ‘line-numbers’ study was useful in illuminating issues of design as well as of number. In interacting in this microworld, students needed to compare numbers such 0.07 and 0.1, or 3456 and 5413 on the same graphics screen. I chose not to automate the scaling. This meant that the students had to go into the ‘scale’ data box to change the scale factor. Frequently, this number was a decimal. So, in effect, the microworld provided a visual refutation of the ‘multiplication makes bigger’ notion, but whether the students were aware of this is a different matter. If numbers were close together, the students also needed to alter the ‘start number’.

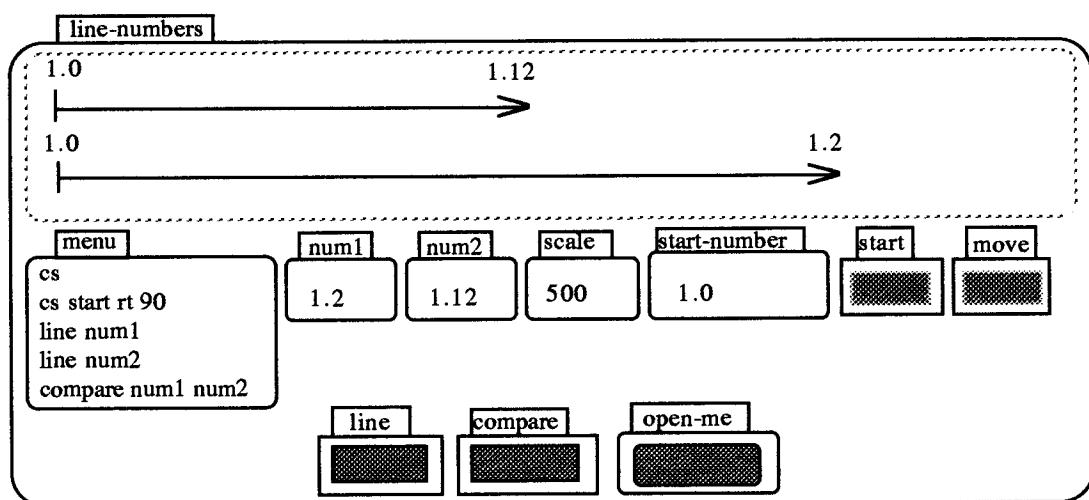


Figure 5.6: Controlling the ‘scale’ and ‘start-number’

The students became fluent in doing this, but again whether they fully understood what they were doing is debatable. By having to change both the scale and the start-number, it is possible that some students lost sight of the original task. There was frequent researcher intervention to help in choosing the scale and the start number. For some students, the choice of scale was at times haphazard, and a start-number too was occasionally chosen bigger than one or both of the numbers being compared. Steven, for example, chose a start number larger than either of the two values to be compared, and Ella said “You can’t do that. You’ll get two minuses.”

In not automating the drawing process, I had attempted to make accessible all the parts of the program that could be changed. However, whether the things that the students were manipulating were the things that mattered in relationship to the numbers they were comparing is open to question.

## **5.4: Developing a Pedagogical Approach**

It had been the original intention of teachers and researchers to integrate the Boxer work within the class themes, but in practice, this turned out to be difficult to do. The teachers were extremely busy in managing all the activities of a primary school classrooms of some 30 students, so there was a limited time that they could devote to the two students on the computer at any one time. In any case, the fact that there was only one computer per class imposed practical limits to this ambition.

Instead, the teachers adopted a ‘cascade model’ of distribution: certain students who appeared to have displayed enthusiasm and skill in learning to use Boxer were given unequal access with the intention that they could disseminate that skill amongst their peers. These students did become very proficient at programming in Boxer, recovering from ‘crashes’ and in using the operating system. They also proved to be very willing to help other students and were of great assistance to the classroom teacher. On the other hand, it was also true that at the end of this period, the number of classroom ‘experts’ was small relative to the size of the class. Moreover, there was a gender imbalance in this expertise – the boys were the ‘experts’. The class teacher was also aware of this imbalance, but as she pointed out (personal letter) in an evaluation of the first year, this imbalance reflected the make-up of the class, two-thirds of whom were boys. We began to look for another method of integrating Boxer work into the classroom activities. Events at the end of this period gave impetus to this search.

The year 6 class teacher, who by now was quite proficient in Boxer and who was also a key figure in suggesting programming goals and extending tasks, began a year’s maternity leave. This meant that the new year 6 class (who had begun Boxer in the fifth year) had a new teacher with no Boxer experience. To compound matters, it quickly became obvious that the new year 5 class (class C in Table 4.1) had very little experience with either Logo in particular, or computing generally.

To gain a deep insight into how students engaged with Boxer, it would be necessary to work with small groups of students over extended periods. On the other hand, to do so raised ethical questions of access by the rest of the class – an issue which was of concern to the year 5 classroom teacher. Boxer was new. It was exciting. The students could see what their peers were able to achieve with it. It did intrude on the

life of the class. The dilemma faced at the end of the exploratory period of the research was how to develop a pedagogical approach which reconciled these two needs.

## **5.5: Summary and Implications**

This chapter has considered issues which arose in the exploratory phase of the research concerned with programming, number and developing a pedagogical approach. The results replicate Logo research in the forms of programming observed, but differ in qualitative respects.

### **5.5.1: Programming Issues**

The results show that the visual format of Boxer changed the nature of students' programming in comparison to Logo. By having a single environment in which communication, editing and execution took place, it reduced the 'overheads' needed to achieve complex goals. It also gave these students access to programs written by more experienced programmers. On one side, this motivated them to use more structured programming techniques, and encouraged them to engage in mathematical activities. But, it brought to the fore the issue of the match between the programming and mathematical demands of pre-written programs. The Turtles microworld, for example, combined introductory programming with problems relating to finding the angles of polygons, and students often found the latter harder than the former.

In Boxer, as in Logo, students were often motivated by the desire to obtain powerful visual effects, but it seemed that simplicity of writing or adapting procedures in Boxer enabled students to attain complex goals with greater ease. The presence of the program and its consequence on the same screen together with line-by-line execution medium gave them the tools to exploit their perceptual thinking. The results also replicate Logo research in pointing to the importance of appropriate intervention, but raised the question of whether in Boxer this might be devolved from the teacher to the medium itself.

### **5.5.2: Number Issues**

The 'line-numbers' representation gave students a way of visually comparing numbers, especially decimal fractions. Interacting with this Boxer object motivated them to confront their intuitive notions of such quantities, and provoked them to generate alternative hypotheses. But, I felt that the representation was inadequate in limiting them to interpretation. My original – and simplistic – idea that working with Boxer

would somehow enhance students' understanding of number needed to be modified in the light of the data emanating from this study.

Working within my pre-constructed microworlds exposed their number misconceptions to them and to me, but it did nothing to address the root causes of those misconceptions. It seemed to me that they lacked meaning for the quantities that they were being asked to represent. I felt that there was a greater likelihood of generating such meaning if the students played a greater role in the construction of the representation itself.

#### **5.5.3: Pedagogical Approach**

The exploratory study threw up two apparently conflicting needs. On the one hand there was my need to observe students in depth and this implied working with small numbers. On the other hand, the class teacher and I felt that there were ethical grounds for making Boxer as accessible to the remainder of the students in the class as possible. In order to accommodate the different demands. I made two changes to the mode of working with students. Firstly, I made the decision to undertake case-study research. Secondly, I decided to exploit the medium as a means of introducing itself.

#### **5.5.4: Implications for the remainder of the thesis**

My main research aim implied that students needed to have a certain degree of programming expertise with Boxer. The prevailing circumstances at the beginning of the longitudinal study meant that the students who would participate in that study had no programming experience. There was therefore an obvious need to build up that expertise. The evidence from the programming activities together with the pedagogical considerations pointed to the potential of Boxer as a way of introducing itself. But, the evidence from the number activities led me to the conclusion that there was a need for greater involvement by the students in the construction process. The rest of this thesis can be seen in terms of my attempts to take account of – and indeed reconcile – these two conclusions from the exploratory phase of the research.

Chapter 6 analyses how I attempted to achieve the first objective through a process of iterative design, whilst chapters 8 and 9 recount two different studies aimed at accomplishing the latter. Chapter 7 discusses the interviews with the Boxer and non-Boxer students. As such, it represents a bridging chapter between the programming and number activities of the longitudinal stage of the research.

# **Chapter 6: Exploiting the Medium**

## **6.1: Introduction and Overview**

### **6.1.1: Introduction**

This chapter consists of an analysis of the programming, number and design issues which arose through using the medium of Boxer to introduce itself. It describes how a process of iterative design was used to construct introductory Boxer microworlds through which year 5 students (ages 9 to 10) learned programming ideas, and began to extend their concept of number. This first phase of the longitudinal research lasted from the September 1991 to July 1992.

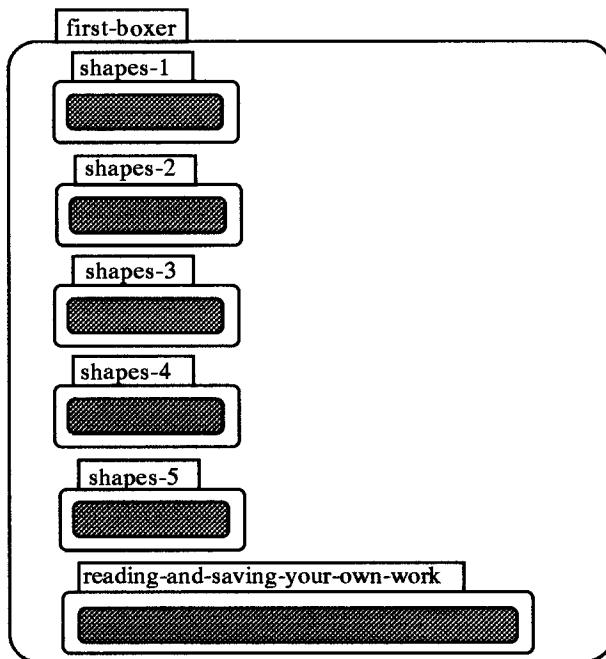
Altogether, eight case-study students were closely observed (chapter 4) using three microworlds known as 'First-Boxer', 'Second-Boxer' and 'Third-Boxer'. However, the microworlds were also being used simultaneously by the remainder of the year 5 class in so far as time, hardware and other classroom limitations allowed. The feedback from these two sets of users formed an integral component of the iterative design process.

This phase of the research had three interlocking aims:

- (i) to provide students with a foundation of programming knowledge on which to build the long-term number aim of the research;
- (ii) to continue the search for an evocative number representation;
- (iii) to document the design issues which arose through using the medium to introduce itself.

### **6.1.2: Overview**

Three distinct microworlds were employed in the research: First-Boxer, Second-Boxer and Third-Boxer. First-Boxer was designed to introduce fundamental programming ideas, while Second-Boxer was mostly devoted to the single idea of variables. Third-Boxer then exploited the notion of variable as a means of investigating number. The microworlds were divided into sections, each of which had a main programming or number focus. Figure 6.1 shows how First-Boxer appeared to users on opening, while Table 6.1 indicates the programming and number contents of the sections.



**Figure 6.1: First-Boxer**

**Table 6.1: Programming and Number Organisation of First- to Third-Boxer**

Main Focus of Section	First-Boxer Sections	Second-Boxer Sections*	Third-Boxer Sections
Sense of command.	Shapes-1		
Procedure as an entity.	Shapes-2		
Patterned procedures.	Shapes-3		
Sub- and super-procedures.	Shapes-4		
Modularity.	Shapes-5		
Variables		Variables-1 to -3	
Number in the context of area		Variables-4	
Directed numbers in the context of co-ordinates			Lines Writing
Directed Numbers and Decimals in the context of an on-screen counter.			Change-it, Build-it
Directed Numbers and Decimals in the context of a list processing command.			Lots-at-once

\* Second-Boxer also had a section called 'Set-up' whose main focus was the creation of boxes and screen management (Appendix 3).

The remainder of this chapter separates out the programming, number and design issues which arose in this phase of the research.

## 6.2: Programming

Hoyles and Noss (1987a) observe that a microworld cannot be defined solely in terms of its technical component, i.e., the software. Rather, they stress the need to also take into account the pedagogical, contextual and pupil components of such microworlds. In making a selection from the research data, my choice has been guided by three factors. Firstly, the selection attempts to illustrate how changes in the technical component of Boxer (compared to Logo) influenced the other components. Secondly, I have tried to include representative sections from each of the microworlds (The complete microworlds can be found in Appendices 2 - 4). Finally, the selection tries to show how the data from the students' interactions was also used to modify the microworld itself.

### 6.2.1: Developing a sense of command

Students first encountered commands in 'shapes-1'. Figure 6.2 shows 'shapes-1' partially open, whilst figure 6.3 shows it fully open.

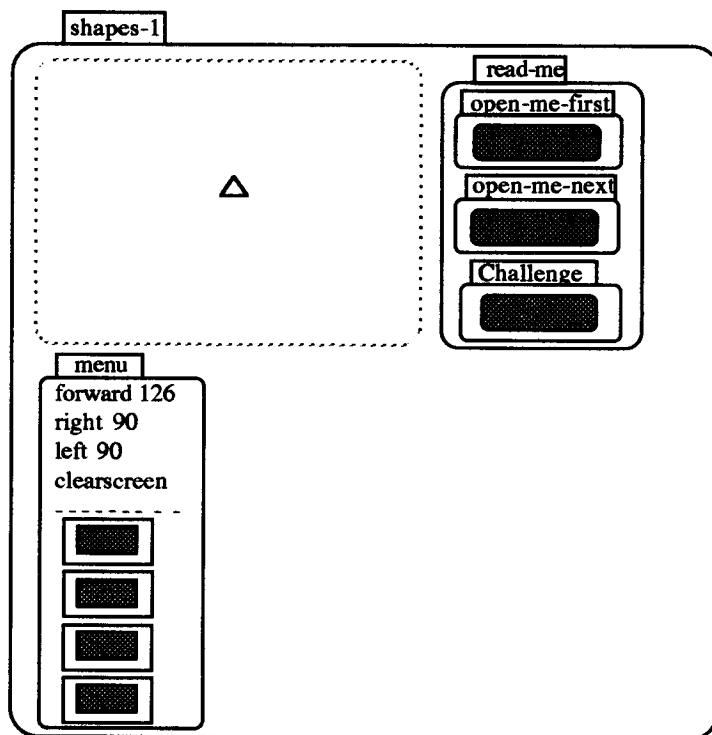


Figure 6.2: Shapes-1 partially opened

These are presented here to assist the reader. In reality, the facility for opening and closing boxes meant that students experienced many other intermediate views before

and between each of these. It needs to be stressed that figures 6.2 and 6.3 represent one version of ‘shapes-1’. Like all the other components of the microworlds, ‘shapes-1’, was modified as the research proceeded.

The first point that needs to be made here is the obvious one that all of the above details appeared on the screen. Boxer facilitated an on-screen form of presentation that would not have been possible in say, Logo. The scoping rule (chapter 3), for example, allowed the presence of more than one graphics box on the screen at once (e.g., the ‘challenge’ in Figure 6.3). Boxer’s hypertext capability for suppressing or revealing information through closing and opening boxes was another significant change from what was possible in Logo.

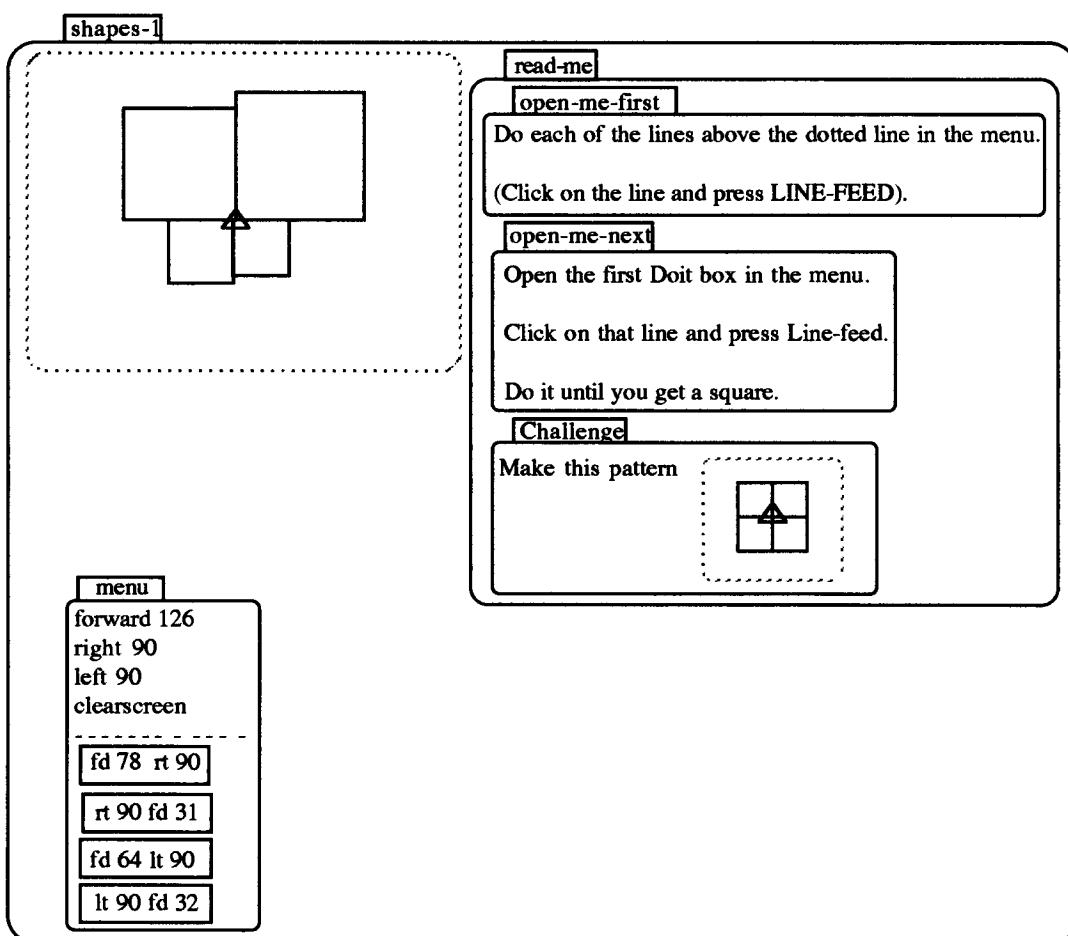


Figure 6.3: Shapes-1 fully opened showing the ‘window’ challenge

‘Shapes-1’ was intended to introduce students to the notion of a programming command, but to do so in a way that focused on the process and not merely the product. My initial interventions were designed to introduce the students to Boxer

commands by encouraging them to repeatedly execute the lines in the menu. Each of these repetitions led to their drawing of a square, but the squares were of different sizes and were each drawn in different areas of the screen (Figure 6.3). These features were designed to motivate students to reflect on the processes involved. My interventions were now aimed at encouraging prediction and reflection by drawing students' attention to the differences between each square and asking why such differences might have arisen. The following extracts from Laura and Courtney's first encounter with Boxer give some indication of what occurred in practice.

The girls began the session by creating a square through repeating the lines 'forward 126' and 'right 90' in the menu. Having, cleared the screen they drew a new square by repeating the top doit box in the menu '[fd 78 rt 90]' four times. Laura, now queried the screen product.

**Extract 1:**

Speaker	Dialogue	Commentary
Laura:	How come it's smaller?	Points into the screen graphics box at the square of side 78.
Interviewer:	What do you think?	
Laura:	Oh, because that one says seventy eight	She answered her own question by pointing at the 78 inside the program doit box (Figure 6.3).

Correct, as her inference was, the students' knowledge was still tenuous at this stage. The following exchanges came after Courtney had drawn a new square '[rt 90 fd 31]' by repeatedly executing the second doit box in the menu.

**Extract 2:**

Speaker	Dialogue	Commentary
Laura:	As we're coming down, they might be smaller.	Traces finger down the menu doit boxes, and then pinches fore-finger and thumb together on the edge of the small square to indicate an even smaller square.
Courtney	It'll get smaller.	
Laura:	Hey. It's getting bigger!	After Courtney had executed the third menu doit box '[fd 64 lt 90]'.

Laura:	Big, smaller, big, smaller.	Pointed to each of the doit boxes in turn.
Courtney:	No! small, big, small, big.	Leans towards Laura and points to each of the doit boxes in turn.

This was a typical incident in Boxer in that it highlights the way the students' dialogue was accompanied by pointing to the visual images on the screen. It also highlights the manner in which the students interacted with the programming code and the resulting drawing. The presence of both together on the screen meant that process as well as product was the focus of attention.

### 6.2.2: Procedure as an entity

'Shapes-2' was designed with the aim of developing the notion of procedure as an entity (cf. Papert et al., 1979). In Boxer, this can be achieved in several ways, perhaps the simplest being that of placing a series of commands associated with a composite action inside an un-named doit box.

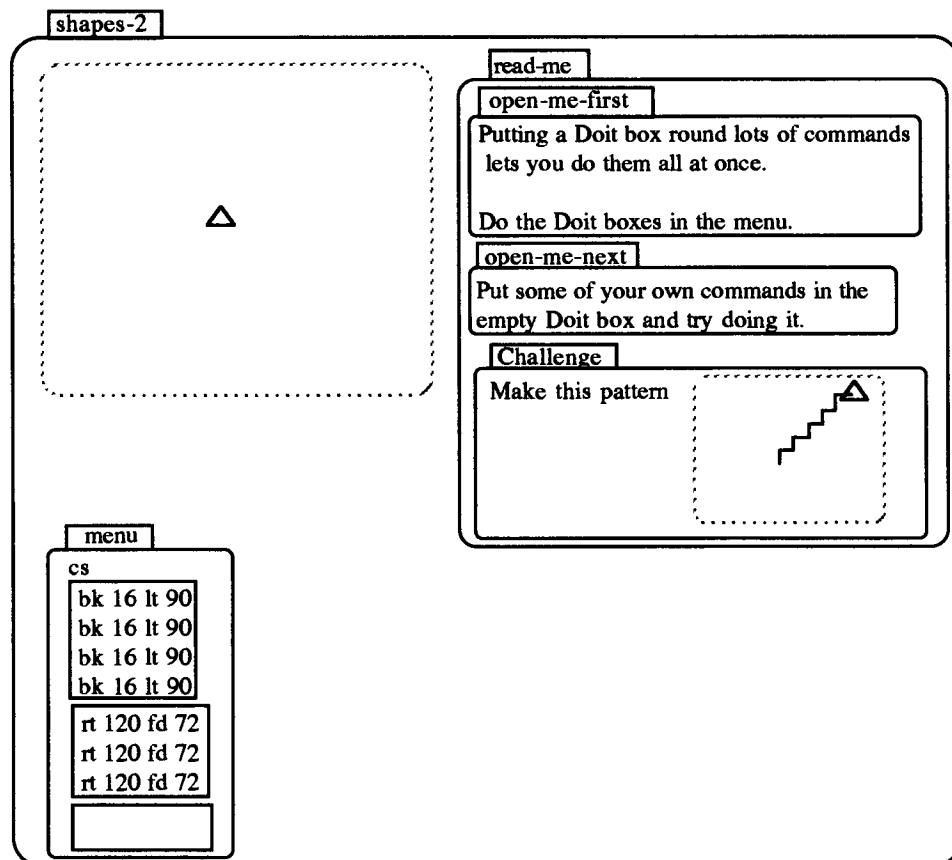


Figure 6.4: Shapes-2 fully opened showing the 'staircase' challenge

The un-named doit box can then be executed by placing the cursor outside of it and pressing the line-feed key. In Figure 6.4, for example, the first un-named doit box in the menu drew a square when executed while the second drew a triangle. The empty doit box was intended to give students an opportunity to produce a pattern by repeatedly running their own group of commands.

In designing shapes-2, my dilemma was how to introduce the notion of a procedure as an entity without focusing too much attention on the product. Demonstrating the power of the language to obtain interesting geometrical patterns from simple procedures needed to be balanced with reflection on the processes involved. The doit boxes in the menu gave students a means of obtaining the former, i.e., getting interesting patterns from relatively simple procedures. I saw this as being beneficial in terms of motivating students to undertake designs of their own. On the other hand, it quickly became apparent that students' attention was solely on the product. Initially, process reflection was achieved by way of my interventions, but as the microworlds developed, I attempted to design the challenges to take on this role. In particular, I created challenges which required the recognition of state-transparency. The following episode illustrates Laura and Courtney's first attempts to come to terms with this idea.

**Extract 1:**

Speaker	Dialogue	Commentary
Laura:	We got a square.	Points to screen output.
Courtney:	I don't know what we've done wrong.	
Interviewer:	Suppose I act out the turtle?	
Both:	[Courtney reads out commands from their doit box and tries to explain why the program won't work. Then Laura does the same.]	They disagree on why things are going wrong.
Interviewer:	Suppose I'm the turtle and you want me to draw out the staircase, tell me what to do.	Laura calls out the commands while I act out the turtle.
Laura:	forward ten right ninety forward ten left ninety forward ten right ninety forward ten left ninety forward ten right ninety forward ten left ninety	Laura calls out commands while Courtney types and executes them in the doit box.
Interviewer:	Are they starting to repeat anywhere? See which ones you can get rid of.	Refers to superfluous commands.
Courtney:	Yes!	The girls reduce the commands in the doit box to [fd 10 rt 90 fd 10 lt 90] which they run five times.

The girls were not alone in experiencing difficulties with state transparency. Other students also had difficulties with this idea. To help them overcome this impasse, three different strategies were adopted: (i) I or one of the students acted out the role of turtle with the other student calling out instructions; (ii) one student called out instructions and the other student drew the shape on paper; (iii) I encouraged students to use line-by-line execution to get one ‘step’ before going on to repeat it.

Jessie and Oliver, for instance, experienced considerable difficulties with the challenge. Jessie spoke of doing ‘half-a-box’, i.e., drawing the bottom and right sides of a square and repeating it. Their first attempt led to their drawing the top and right sides of a square and they now discussed drawing the stairs going down instead of up. After a number of attempts, during which they drew squares etc., I acted out the turtle, with Jessie giving the orders and Oliver writing them down. Some of these commands, e.g., ‘fd 90 rt 60’ revealed that Jessie had not appreciated the significance of the various inputs. By guiding me and drawing my path on paper, they were eventually able to draw the first step. They completed the staircase by repeated use of the linefeed key.

Shapes-2, like shapes-1 before it, was being used by 9 year old students who were still coming to terms with the physical manipulation of the keyboard, the Boxer conventions etc. For students with learning difficulties such as Jessie and Oliver, these factors were of some importance. Nevertheless, they were able to do the challenge - albeit in a form different to the other students – and to gain some insight into the structure of programming in Boxer.

This episode illustrates two significant features of Boxer. Firstly, it shows how interventions were being devolved to the medium in the form of on-screen challenges, and secondly it shows how Boxer provided an alternative means for students to achieve the same goal (repeatedly executing an unnamed doit box) thereby giving them a sense of achievement. In the event, all the students were able to do the challenge.

### **6.2.3: Structured Programming**

Shapes-3 introduced the ‘repeat’ command and initiated the use of sub-procedures and super-procedures in the form of un-named doit boxes inside other un-named doit boxes. Shapes-4 took this a stage further by naming the procedures. In Boxer, this is achieved simply by creating name tags at the top of the appropriate box. Shapes-5 now stressed the modular nature of these procedures.

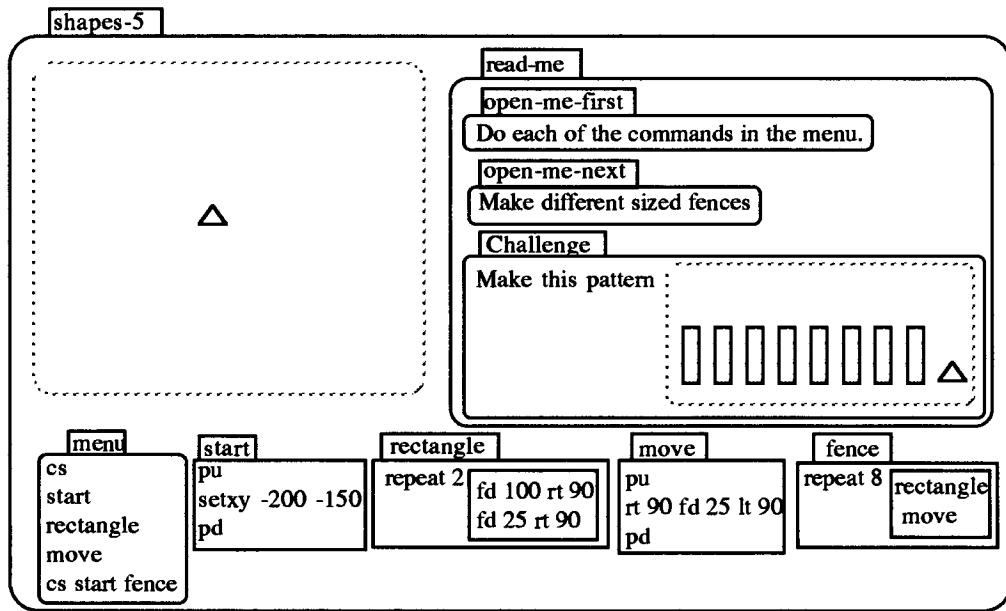


Figure 6.5: Shapes-5 fully opened showing the ‘fence’ challenge

In earlier episodes, students had expressed a desire to start their drawings elsewhere on the screen, i.e., other than at the centre. Shapes-5, was designed so that they could do just that, but it was constructed in such a way that they might see the advantages of modular programming. There was, for instance, a doit box called ‘start’, another called ‘move’ along with doit boxes named ‘rectangle’ and ‘fence’. The on-screen challenge, that of drawing a ‘fence with gaps’ (Figure 6.5), was also designed with the intention that students might approach it in a modular way. The following episode shows how Nico and Joanna reacted to the challenge.

The students’ approach to the earlier sections had been to ignore the screen instructions completely and launch immediately into the challenges. In shapes-5, they attempted to complete the ‘fence’ challenge all at once. At their first attempt, Nico and Joanna obtained a set of rectangles without gaps (Figure 6.6a). A later attempt led to the drawing of pairs of rectangles with gaps between each pair (Figure 6.6b).

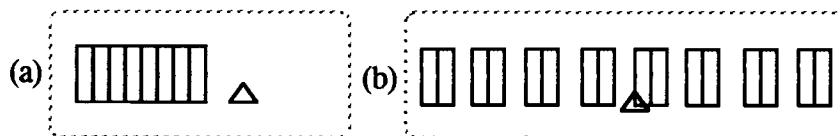


Figure 6.6: Nico and Joanna’s first attempts at the ‘fence’ challenge

In ‘shapes-4’, they had demonstrated that they knew how the ‘rectangle’ procedure worked and they appeared to understand how to use both the ‘start’ and the ‘move’

modules. However, at that point, they had not perceived the need to match the components of one module with the other. Likewise, in attempting the ‘fence’ challenge, they began by typing directly into the ‘rectangle’ program, and this led to their bugs. They were now receptive to my interventions which focused on modularity.

The use of the ‘start’ and ‘move’ modules in shapes-5 was part of what might be termed a ‘subliminal’ agenda: it was hoped that students would also adopt a modular way to organise their programs. However, over the course of the research, it was rare for a student to create a separate ‘move’ module. On the other hand, students did organise their Boxer programs in a modular way, the modularity being related to functionality. For example, in a house program, a distinct module would be created for the door, the windows and so on. It seemed that the boxes acted as natural receptacles in which to write distinct programs.

#### 6.2.4: Variables

The ‘Second-Boxer’ micro-world was intended to expand students’ programming power and to provide a wedge into their exploration of number concepts. This was achieved in two ways: firstly by giving them the means of creating Boxer environments for themselves, and secondly through the use of variables.

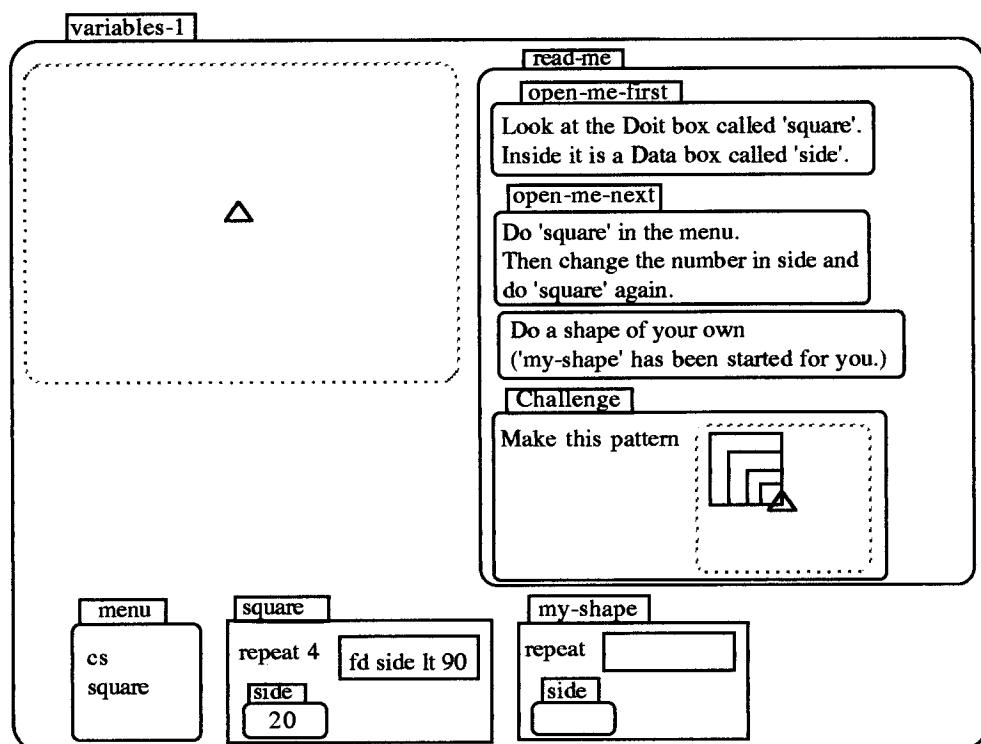


Figure 6.7: Variables-1

It consisted of five sections ‘set-up’ and ‘variables-1’, to ‘variables-4’. ‘Variables-1’ introduced the notion of variables in the form of internal data boxes. In the *doit* box ‘square’, the first line ‘repeat 4 [fd side rt 90]’ uses the variable ‘side’ which is the name of an internal data box (Figure 6.7). The instructions in the ‘read-me’ data box invited students to experiment with different values of side.

In creating a microworld devoted to the exploration of variables, I made a conscious design decision to avoid two of the common features of introductory Logo courses: firstly the emphasis on variables in the context of angles, and secondly the practice of introducing procedures, first without, and later with variables. In each case, I felt that the initial gains which these approaches allowed, were more than offset later by the gaps that needed to be bridged to more general notions. The design criteria were thus intended to facilitate novice’s introduction to programming in a way that would be contiguous with later more sophisticated demands.

I hypothesised that students would best appreciate the significance of variables if these were first represented by internal data boxes. By having to manipulate directly the value in the variable data box, I thought the students would come to have a better understanding of a variable as representing a range of values. This approach also had the advantage that it meshed well with the tendency in British mathematical textbooks to introduce variables by means of empty brackets or boxes (e.g., SMP, NMP).

All of the students were able to do the ‘nested-squares’ challenge by successively changing the number inside ‘side’. This in itself was significant since it gave them an intermediate form of programming power, i.e., between the Logo procedures of ‘square’ without and with variables. But the challenge itself raised questions. On the one hand, it prompted the students to ask if there was an easier way to obtain the ‘nested squares’ thus motivating the later use of general variables. On the other hand, at the time of the research, there did not seem to be an obvious link between the internal variable data box and the general variable which came with the ‘input’ line<sup>1</sup>.

The evidence from the interviews suggested that, just as students had earlier appreciated the arbitrary nature of naming procedures, so too did they appreciate that the names of variables were also arbitrary.

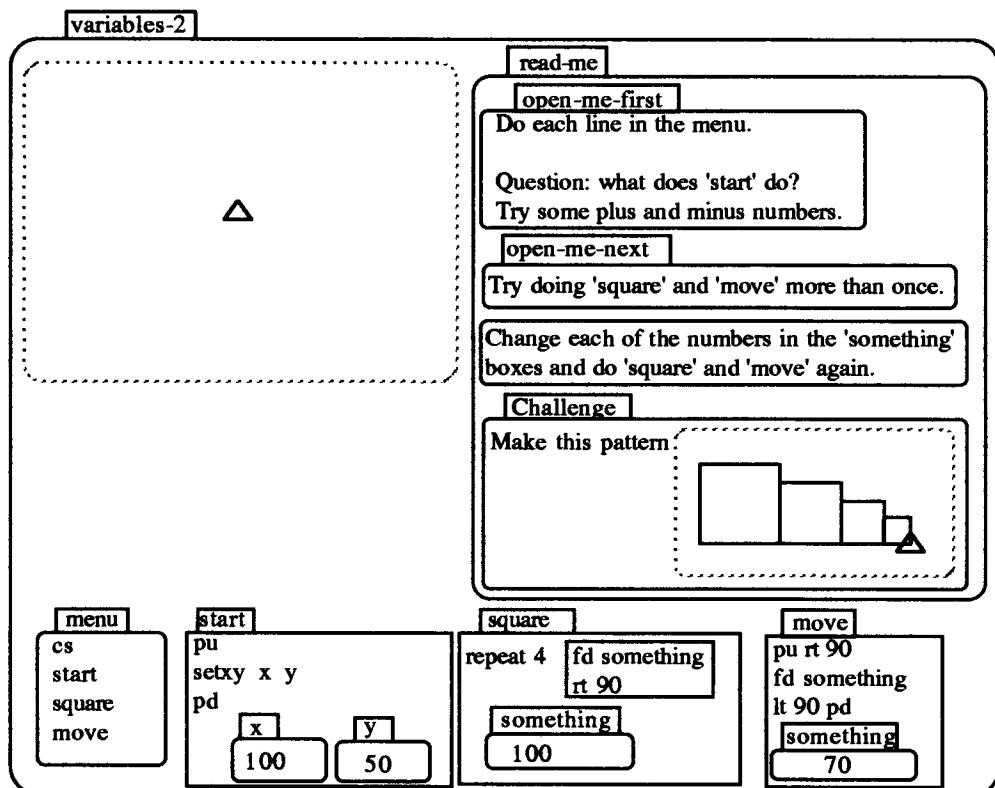
---

<sup>1</sup> 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.

**Extract 1:**

Speaker	Dialogue	Commentary
Interviewer:	Raises the possibility of changing the name of the variable 'side'.	
Joanna:	Joanna.	Changes 'side' to 'joanna' and runs 'square' from the menu.
Interviewer:	Reads out error message.	Message appears next to 'square' in the menu.
Nico:	Oh yes, because you have to change there as well.	Points to 'side' in the line 'repeat 4 [fd side rt 90]'.

'Variables-1', introduced the notion of variable in the form of an input for the side of a square. 'Variables-2' extended this usage by using variables in the contexts of co-ordinates ('x' and 'y') in a 'start' module and a distance ('something') in a 'move' module. The variable for the size of the square was now named 'something' also.



**Figure 6.8: Variables-2**

In an earlier version of 'variable-2', the numbers given in 'x' and 'y' in 'start' had negative values. But, this meant that the students could begin their drawing without reflecting on the processes involved. In a later version (Figure 6.8), the co-ordinates

given were such that the drawing would not begin on the bottom-left hand-side of the screen. The students therefore had to reflect on the meaning of these numbers. ‘Variables-2’ also planted the notion that the ‘square’ and ‘move’ modules needed to be co-ordinated. The names of the variables in each had the same name, but their initial values were different, again with the intention that students would have to reflect on the relationship between the two modules.

The students did not at first appreciate the need to change the value of the variable in ‘move’ to match that in ‘square’. Generally, they drew the first square and then used the ‘move’ module with its current value. The following extract from Jason and Liam illustrates this phenomenon.

Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	Try move and see if it does what you want it to. Does it?	
Both:	No.	
Interviewer:	So, how can you change it to get it to do what you want it to?	Both students point at the data box called ‘something’ in ‘move’.
Jason:	Change that so it’s the same as ‘square’.	Refers to variable values in ‘move’ and ‘square’.

Variables-3 extended the contexts in which variables were used. ‘start’ was unchanged, but in ‘move’, the variable name was changed to ‘amount’. The main change was the introduction of the ‘type’ command with the variable input ‘length’ (Figure 6.9). The ‘type’ command gave a means of integrating text and graphics, and the ‘Shrinking Lines’ challenge required students to exercise this facility. Few of them had difficulties in doing so. Jessie and Oliver, who hitherto had displayed little feeling for the size of numbers, had the following dialogue about the gap between the lines.

Extract 3:

Speaker	Dialogue	Commentary
Jessie:	Shall I put zero in here?	
Oliver:	A ninety, I know ninety.	Refers to the ‘amount’ data box in ‘move’.
Jessie:	It can’t be ninety. It has to be a smaller number like forty to get a smaller gap.	

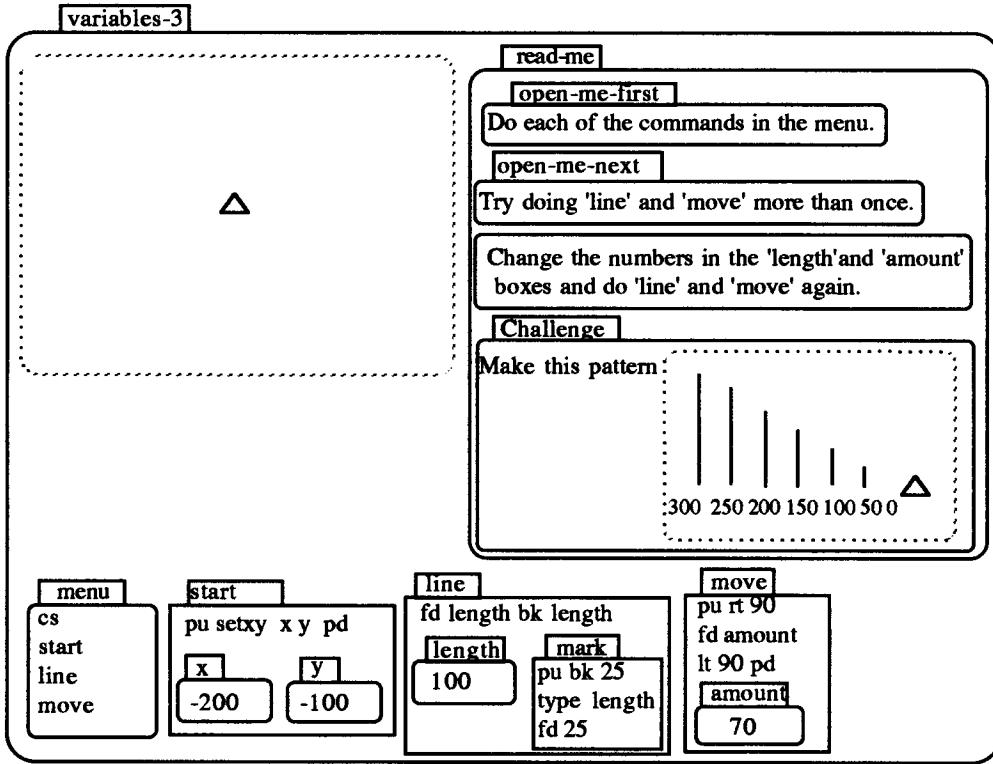


Figure 6.9: Variables-3

Variables-3, like its predecessors, introduced new programming commands, and the challenges were always related to these new commands. However, it is important to observe the cumulative effects of each successive section on students' programming. Each presented a model of modular programming, which progressively motivated students to attend to the links between modules. Each extended the range of contexts in which students used variables, and increasingly these contexts necessitated students to extend their notions of number. The following section details how concern with variables per se gave way to situations in which variables were subordinated to the investigation of number representations.

## 6.3: Number

### 6.3.1: Investigating Number in the context of Area

Variables-4 employed variables in the context of area. The technical component encapsulated a more explicit mathematical agenda than had previously been the case. The procedures 'rectangle', for example, incorporated commands to draw a rectangle, and calculate and print the area inside it (Figure 6.10).

Area seemed to be the kind of context that would lend itself to a visual representation in Boxer. I thought that by changing the values of the two sides and then seeing the resulting area typed in a rectangular box, students would begin to appreciate the connection between the numbers. My hope was that this might give me a context in which to introduce decimals. However, this was not successful – the major source of the difficulty stemming from the fact that the students' concept of area was not sufficiently robust.

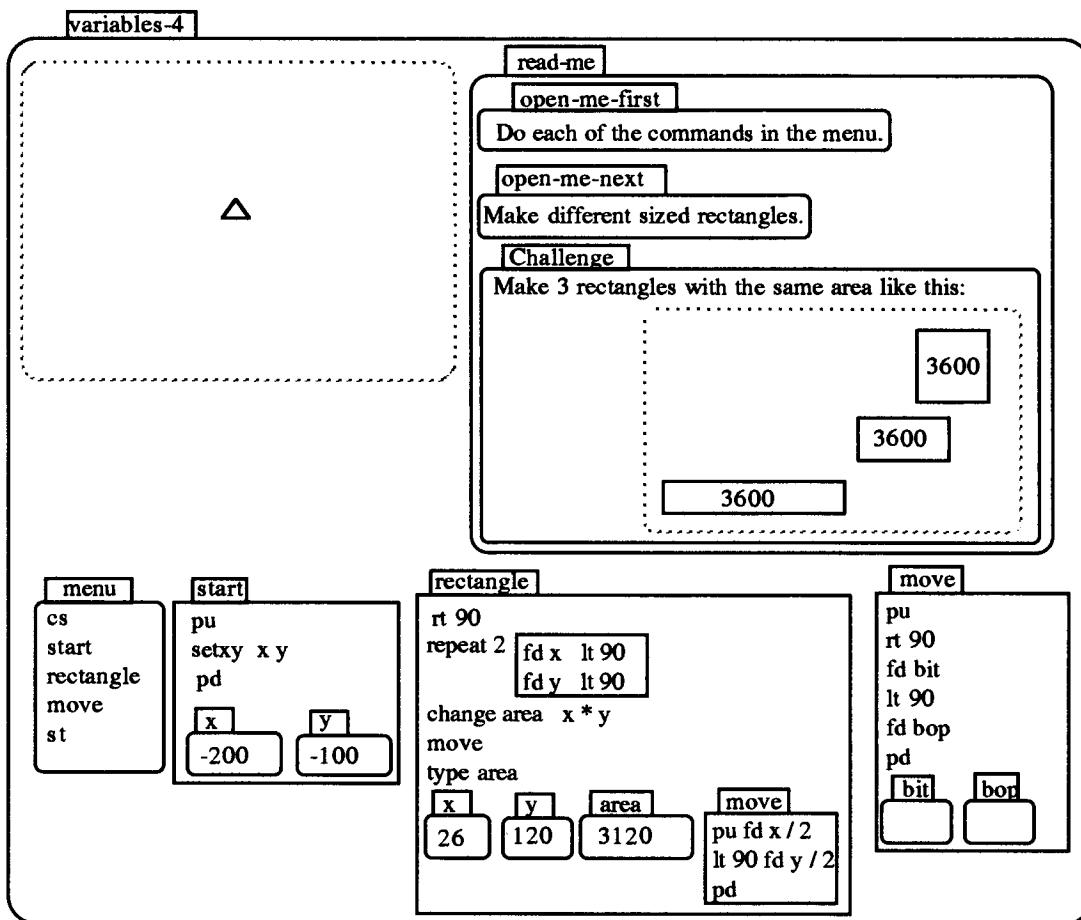


Figure 6.10: Variables-4

Oliver and Jessie at first professed ignorance of the notion of area (It had in fact been introduced during an earlier class lesson.).

Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	If I draw a rectangle 3 this way and 6 that way, would you be able to work the area out?	Drawing on paper.
Jessie:	36?	

Interviewer:	Do you know what area means?	
Jessie:	No.	
Interviewer:	That's what I was trying to find out. You haven't done area?	Both nod heads.
Interviewer:	Area means how many squares there are altogether.	
Jessie:	Oh yea, I know what area means.	
Interviewer:	Oh, you do? So how would you work out the area of this rectangle?	
Jessie:	Add 3 and 6?	

The following extract with Liam and Jason also shows initial difficulties with the relationship between area and sides. The boys were trying to find the sides of the square with area 3600.

**Extract 2:**

Speaker	Dialogue	Commentary
Interviewer:	When you say it should be even, Liam, what do you mean?	
Liam:	The same size. They should be the same numbers.	
Liam:	Let's try one thousand, eight hundred.	Appears to reason that the 'same numbers' can be found by halving.
Jason:	How about nine hundred times four?	Liam ignores Jason's suggestion.
Liam:	Try nine hundred by nine hundred .	Tries this and says it's a bit too big.
Both:	[They try 800, 600, 300, and 200.]	
Jason:	Sixty.	

Despite these difficulties, most of the students went on to do the challenge. They drew all three rectangles, mostly using a guess and test approach. In that sense, 'Variables-4' was successful. The students seemed to gain in terms of their understanding of area as a result of experimenting with the values of 'x' and 'y' and seeing the resultant rectangle on the screen. There was other incidental learning. Joanna, for example, queried whether a square was a rectangle leading to a discussion of definitions of each.

'Variables-4' was successful too in providing a meaningful context for using variables, that of relating the sides and area of rectangles. However, it did not succeed in terms of its original number aim: that of using area as a context in which to extend their notion of number to decimals. I felt that the students' knowledge of area was not a sufficiently

solid foundation on which to build such an approach. In ‘Third-Boxer’, I again used variables as a wedge into number but in different contexts.

### 6.3.2: Investigating Number in the context of Co-ordinates

The whole of ‘Third-Boxer’ was concerned with the investigation of number, with variables the vehicle for doing so. Two of its sections ‘Lines’ and ‘Writing’ used variables in the context of co-ordinates.

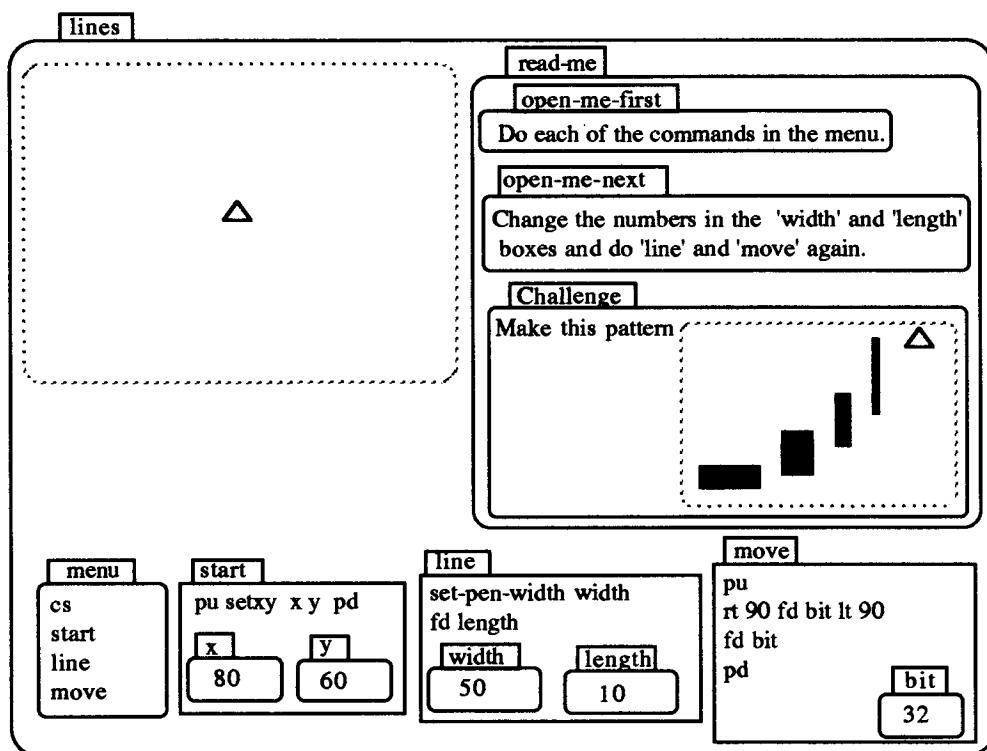


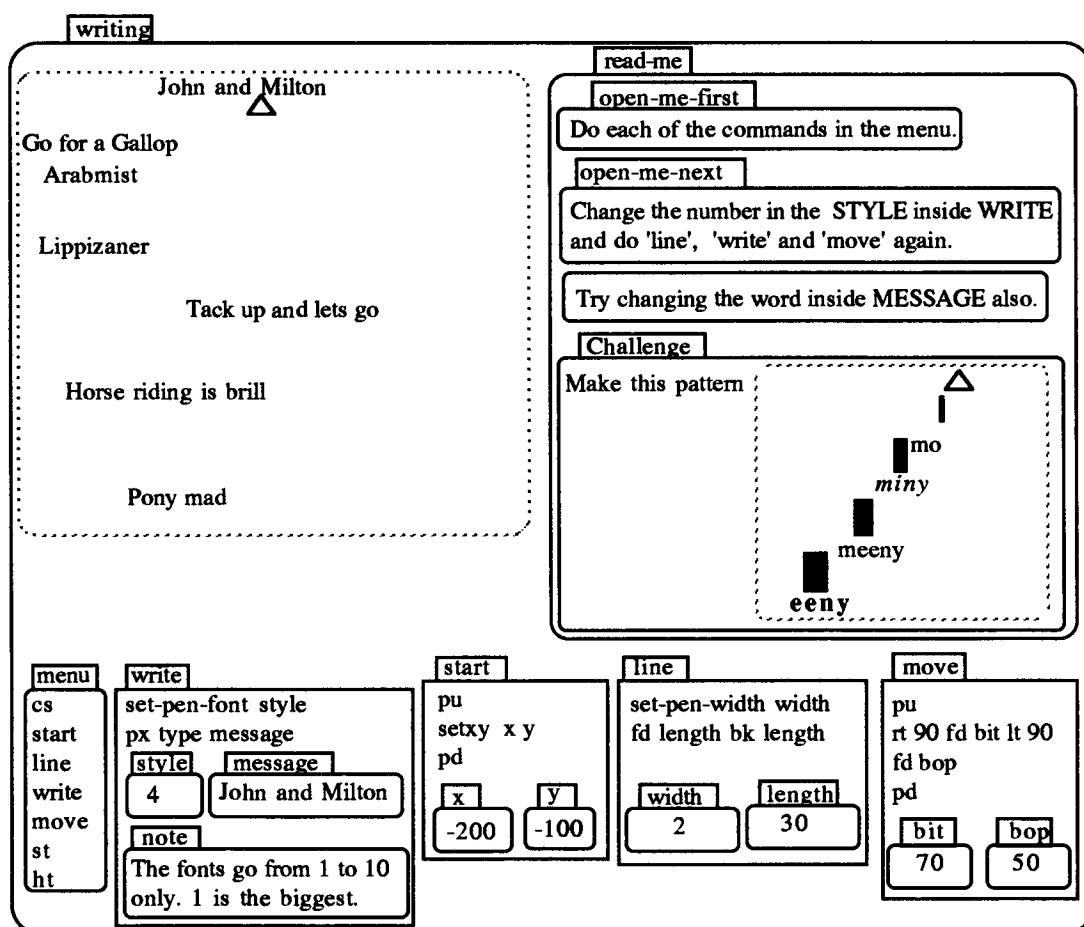
Figure 6.11: Lines

‘Lines’ was designed to familiarise students with the ‘set-pen-width’ command and in the process use variables in a different and wider context. In earlier sections, the ‘start’ doit box was a useful but not prominent part of the set-up. In this section, I wanted to focus attention on how it worked, and changed it so that the internal data boxes ‘x’ and ‘y’ had initial values of 80 for x and 60 for y. It was thus intended that start would *not* place the turtle where the students wanted it to, and hence provoke reflection on its operation.

The challenge forced the students to change the variable values in ‘start’ so as to begin in the bottom left-hand corner. It thus motivated the use of directed numbers. It also required students to go in and out of ‘line’ to change the values of the two data boxes

(‘width’ and ‘length’) there. Finally, the students needed to choose a suitable value for the variable in ‘move’. In order to draw the various lines, it was necessary both to use the ‘move’ module and to change the value of its variable ‘bop’. In the past, some students had avoided using the move module by recreating its commands in the menu, e.g., ‘pu rt 90 fd 30 lt 90 pd’, but it was noticeable now that they were making increased use of the modularity afforded by ‘move’.

The section named ‘writing’ extended the work begun in ‘lines’. The doit box ‘write’ contained a new command ‘set-pen-font’ so the students could experiment with Boxer’s fonts. At that time, these were somewhat limited. The ‘type’ command was now reintroduced but in the context of words rather than numbers. ‘Move’ was changed so as to have two variable data boxes, one for each direction (Figure 6.12).



**Figure 6.12: Writing**

By providing students with a context which combined text with graphics, I hoped to motivate them to use directed numbers to a greater extent than before. By and large,

this was successful, but the prototype version of Boxer that we were using caused difficulties. The ‘type’ command, for example, printed only the top line of data box text on the screen, and although different fonts appeared on the screen, the hard copy was always in the same small font.

Courtney and Laura, who were interested in horse-riding, changed the contents of ‘message’ so as to write: ‘riding is brill.’ across the centre of the screen. They then changed this to ‘Horse riding is brill.’ This led them to experiment with values of x and y in ‘start’. At one stage, the girls deliberately chose -290 as a value for y knowing that it would wrap round the screen to get where they wanted to be. I intervened at this point to show them how to achieve the same effect without wrapping. The girls now became engrossed in experimenting with negative numbers, the motivation throughout being that of getting their messages on different parts of the graphics box.

Extract 3:

Speaker	Dialogue	Commentary
Courtney:	The more you get higher, it goes down there.	This refers to the effect of large negative values of y.
Interviewer:	Where do you want it?	
Courtney:	Down.	
Laura:	Minus goes downwards. I'm understanding it now.	
Laura:	That's exactly where I want it.	

The girls’ choice of values for x and y became more and refined. At one point, they used (-1, 150) to position the message ‘John and Milton’ exactly where they wanted it. Boxer facilitated this experimentation through the simplicity of typing messages and of positioning those messages: in each case all that was required was to change the contents of the visible data boxes.

The prospect of typing their own messages on the graphics screen motivated other pairs also. In response to a request from Joanna and Faye, I drew a set of axes on paper to explain the screen co-ordinates. The girls then experimented with ‘start’, trying values such as (100, 75), (-200, 80), and (-200, -150). They looked at the challenge but decided not to do it, choosing instead to type their own messages.

The desire to get to ‘clear’ parts of the screen motivated the girls to ask if they could use ‘move’ to do so. This drew attention to the fact that my design employed two different conventions in ‘start’ and ‘move’, and prompted a discussion about the difference

between absolute and relative values of variables – not usually a source of interest to 9 year old students! For example, Joanna asked if using -150 in ‘move’ would move the turtle down to 150 (-150 on the y axis). After explanation of how ‘move’ worked (in comparison to ‘start’), she used it with values of (0, -150), (0, -150), and (-300, 0). When I asked if she could do this in one step, she correctly used (-300, -300). By the end of the session, the girls used ‘move’ for even small movements and the graphics box looked something like Figure 6.13.

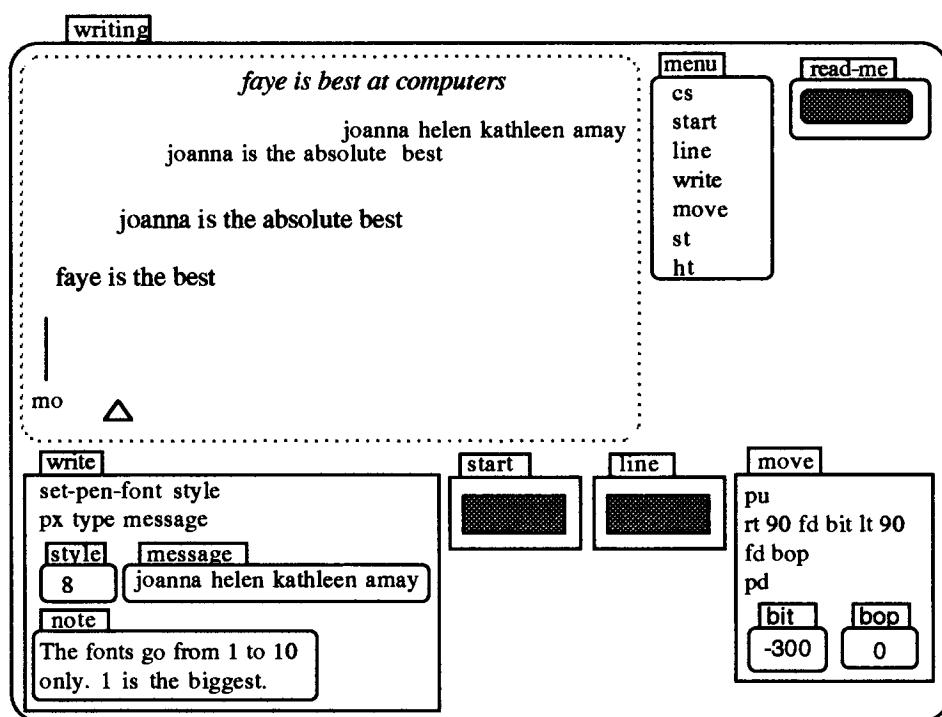


Figure 6.13: Joanna and Faye’s use of ‘move’ for writing

By now, the students were becoming sufficiently confident in working with the environment of Boxer for me to consider a more ambitious, but potentially more complex approach to number. I began to build an on-screen counter in Boxer. This formed the basis of the two sections of Third-Boxer called ‘change-it’ and ‘build-it’.

### 6.3.3: Investigating Number in the context of a screen Counter

The two sections of Third-Boxer called ‘change-it’ and ‘build-it’ differed from earlier sections of the microworlds. Firstly, neither contained a graphics screen – an apparently trivial point – but it meant that the programming activity had moved beyond the simple manipulation of a screen turtle to the control of number sequences. Secondly, each of these sections provided the means (the ‘change’ command) of

changing a variable under program control. Finally, these two sections presented a much more overt mathematical agenda than had previously been the case.

The pedagogical intentions embodied in the design of ‘change-it’ and ‘build-it’ were aimed at providing a meaningful context for the use of directed numbers and decimals. The on-screen challenges initiated this process by inviting the students to generate a sequence of positive whole numbers. The interviews then provided the setting in which further such sequences could be generated, and through which the notion of number could be extended.

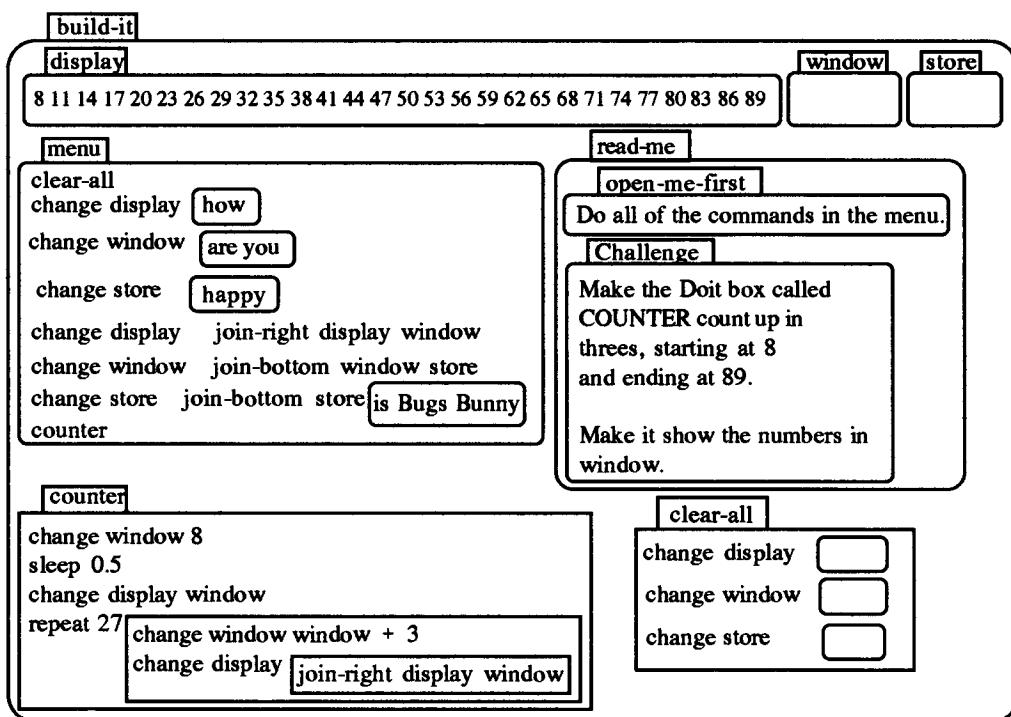


Figure 6.14: Build-it

The challenges of the earlier sections had motivated the students to ask if there was an easier way of changing variable values, so the introduction here of the ‘change’ command was seen to fulfil a perceived need. It was felt that the success of these two sections depended crucially on students understanding the functioning of every part of them. To this end, they each began with the students using the new commands (‘change’, ‘join-right’ and ‘join-bottom’) in the menu to see their effects before using them with the ‘counter’ doit box. Their initial usage was in terms of words rather than numbers as I felt this would give a clearer indication of the command’s action.

The ‘counter’ number-machine in ‘build-it’ generated a sequence of numbers which were displayed across the screen in the data box ‘display’ (Figure 6.14). I began by asking the students to predict what would happen when they executed each line. I then took time with every pair to explain what each line did in ‘counter’. For example, by placing a semi-colon in front of ‘sleep’, it was simple to show what happened when this was not run. Similarly, by varying the input to ‘sleep’, the effects could be seen instantly. The students demonstrated that they knew how to change the lines in ‘counter’ to set the start number, to vary the step-size, and to decide how many times they wanted to repeat. Interventions at this stage were thus intended to provoke reflection on the structure and function of the parts of the program and the logical sequence within it.

I then challenged the students to obtain particular outcomes such as starting at 8 and ending at 89 by repeatedly adding 3. For the students, this activity involved:

- making conjectures;
- carrying out the necessary calculations, using pen and paper or a calculator;
- changing the appropriate variables of the ‘counter’ program;
- running the program to test their conjectures ; and
- accounting for the result obtained.

In generating a sequence of numbers, there were thus three variables which could be changed: (i) the start number, (ii) the step-size and (iii) the number of repeats. Mathematically, using either the operation of addition or subtraction, this is equivalent to the arithmetic progression  $U_n = a + nd$  where  $U_n$  is the nth term,  $a$  is the first term,  $n$  the number of terms, and  $d$  the common difference.

In each of the following episodes, the initial intervention began with the symbolic code and invited prediction as to how changes to it would effect the desired mathematical outcome.

The first episode has been selected to show how the ‘counter’ gave Steven and James a way of conferring meaning on directed numbers. The boys were set the challenge of starting at 20, increasing in steps of 5, and ending at 120. The underlying formalisation of this activity is that of solving the equation  $20 + 5n = 120$ . For the boys, this task was a matter of deciding how many times they needed to repeat the process of adding 5. They changed the appropriate variables in the program to set the start number and the step-size. They used pen and paper calculations to work out that the number of repeats needed was 20 and ran the program.

In the course of their session, Steven asked: "Could you use minus?", and I now sought to exploit his curiosity by challenging the boys to reverse the earlier process, i.e., by starting at 120 and, using the same step size, ending up at 20. In formal terms, this is 'equivalent' to solving the equation  $120 - 5n = 20$ . The boys changed the step size from 5 to -5, and decided that they would again repeat 20 times. However, they forgot to change the start number inside 'counter', so that they began at 20 and subtracted 5 twenty times which resulted in the screen displaying -80. The visual feedback provided a scaffold for an extension to negative numbers which I now sought to exploit.

Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Why do you think it did that?	Points to the -80 in the 'display' data box.
James:	It started at 20. It just went away from 20. It took away 5s and just carried on.	Traces his finger across 'display' data box.

The following episode shows how the pedagogical situation could be structured so as to capitalise on this accidental foray into negative numbers. The boys were set the challenge of starting at 50 and ending at -50, again with steps of 5. The underlying formalisation this time is that of solving the equation  $50 - 5n = -50$ .

Extract 2:

Speaker	Dialogue	Commentary
Steven:	Repeat one hundred.	They run the program and see this is wrong.
Steven:	Repeat ten.	
Interviewer:	Steven says 'Repeat ten'. What do you think?	To James.
James:	Yes, because he has taken away fifty.	James has multiplied the number of repeats (10) by the step size (5) to get 50.
James:	I've got the answer now. It's twenty	After running the program with 'repeat 10 ...'
James:	I know. If ten makes it go to nought, then ten more will make it go to minus fifty.	

Throughout this exchange, the boys pointed alternatively at the program code inside 'counter' and the resulting display. Steven's initial suggestion of 'repeat 100' may

have stemmed from an intuitive grasp that the overall amount needed to move from 50 to -50 was a hundred (Note how subtraction is denoted as movement here). The screen display compelled him to attend to the inner relationship between the step size and the overall movement, from which he derived his second conjecture: ‘Repeat ten’. James, who until now was the silent partner, accounted for Steven’s explanation and then builds his own rationale upon it: “If ten makes it [Boxer] go to nought, then ten more will make it go to minus fifty.”

In another episode with Joanna and Faye, it can be seen how decimals also could be ascribed meaning in terms of step size. The girls were set the challenge of starting at 2, going up in steps of 0.25 and ending at 12 (the formalisation of which is  $2 + 0.25n = 12$ ). Joanna’s response is interesting in terms of the method she used.

**Extract 3:**

Speaker	Dialogue	Commentary
Joanna:	There’s four lots of 0.25 in one so if I did twelve lots of four?	She changed the start value to 2, the step size to 0.25 and the repeat number to 48, and ran the program, at which the screen display showed: 2.00 2.25 2.50 2.75 . . . 14.0
Interviewer:	You are right except for one thing . . .	
Joanna:	Oh yeah. It started at two. So it is ten.	Refers to ten lots of four.
Joanna:	[Changed the repeat number to 40, and ran the program ending at 12.]	

As with all of these episodes, the mathematical activity was an amalgam of thinking/writing/calculating off-screen, interspersed with changes to the program on screen and pointing both to the program code and the line of numbers which resulted from running that program. The way the students referred to the actions of Boxer as “It did ...” or “It started ...” etc. is noteworthy. The anthropomorphism suggests that Boxer was giving the students a dynamic model for the mathematical formalism.

#### **6.3.4: Investigating Number in the context of List Processing**

I had two motives in creating this section, each of which was related to the command ‘for-each-item’. In the first place, I hoped that this command would give me an powerful but more easily understandable alternative to recursion as a programming tool. Secondly, I hoped that it would be a powerful device for investigating number. The Boxer command ‘for-each-item’ provides a means of handling lists (data boxes). It

takes two inputs: a variable which ‘runs’ through the list and the list or box itself. In Figure 6.15, this variable is represented by the data box ‘n’ in ‘function’ and the list by the data box ‘nums’. With the function ‘ $n * 2$ ’ as below, the effect of running the ‘function’ doit box would be to change the contents of the ‘display’ data box to 2 4 6 8 10 12 14 16 18 20 as shown.

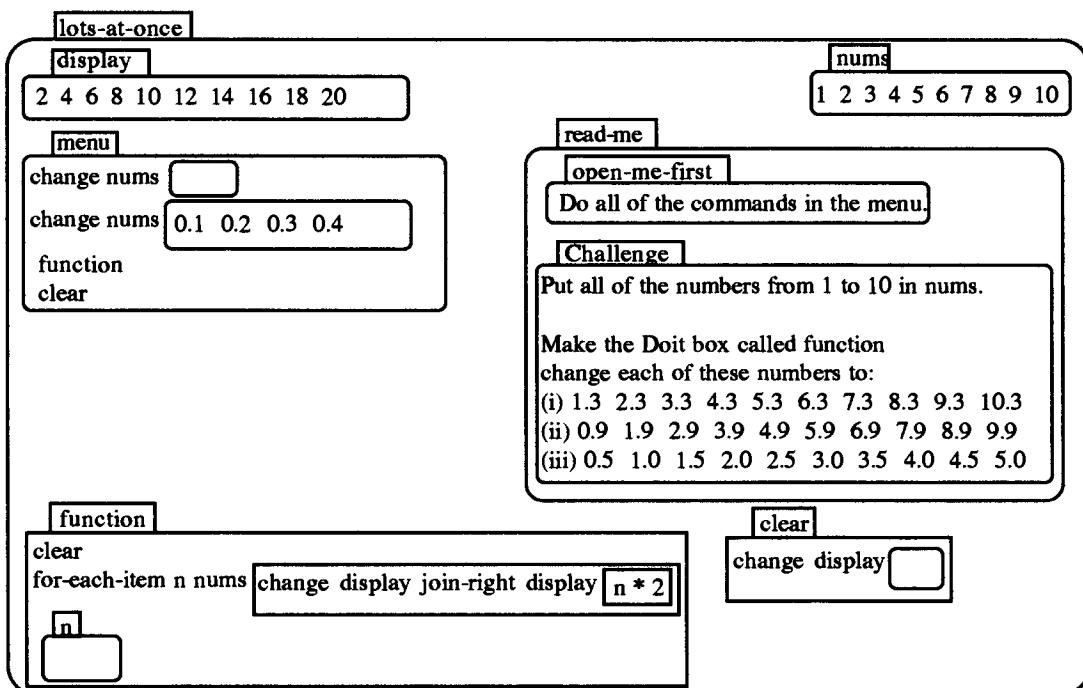


Figure 6.15: Lots-at-once

To begin with, I ran the ‘function’ doit box on the ‘nums’ data box containing 1 2 3 4 5 6 7 8 9 10, and explained its operation line-by-line. I then changed the doit box first from ‘ $* 2$ ’ to ‘ $* 3$ ’ and then to ‘ $+2$ ’ and ran it each time. When the students were confident about how to change the function, the challenge was set.

Liam and Jason, found the first part of the challenge hard, so I used drawings of flow diagrams (on paper) to help. Jason and Joanna, although in different groups, each suggested -0.5 for (iii) of the challenge (Figure 6.15). Having tried this value, they each realised their mistake and corrected it. Joanna said “I think it is take-away half of all of them”. This correct, but algebraically awkward formalisation (equivalent to  $n - (0.5 * n)$ ), was countered by my intervention suggesting that she tried dividing by 2 or multiplying by 0.5. Joanna adopted the latter of these, and ran the program again. The screen display exposed Joanna’s still fragile knowledge of decimal multiplication, as the following extract shows.

Extract 1:

Speaker	Dialogue	Commentary
Joanna:	It halves them. I was thinking it would take 0.5 from each of them.	After trying the program.
Interviewer:	Instead of 0.5 try 0.1.	
Joanna:	That will be one-tenth of it.	
Interviewer:	You are ahead of me.	But, Joanna again talks of taking away.
Interviewer:	Try it.	
Joanna:	I don't understand. I thought it would be take away one tenth not nine tenths.	Having seen the screen display.
Interviewer:	Instead of taking away a tenth, it gets one tenth of it.	
Joanna:	So this times by 0.1 gets a tenth.	

The ‘lots-at-once’ section was not successful either in terms of my programming aim or in terms of my number aim. In the former case, its structure proved harder to understand than I had anticipated. In the latter case, the mathematical challenge required students to be able to choose the correct function to map one set of numbers to the other, and they appeared to have considerable difficulty in doing this.

## 6.4: Design Issues

The previous two sections have given an account of how the medium of Boxer was exploited to introduce itself. It showed how Boxer environments were constructed to foster the learning of elementary programming ideas, and how these environments progressively integrated programming with number. As stated earlier, the pedagogical approach also entailed using these microworlds with the remainder of the class. The feedback from both sets of students led to several design changes as the microworlds were being used, a process referred to as ‘iterative design’. This section discusses these design issues.

### 6.4.1: Presentation

The transcripts of Oliver and Jessie’s early sessions with ‘First-Boxer’ reveals that much of the dialogue was concerned with the mechanics of using the computer, e.g., controlling the mouse on the optic pad, pressing the correct mouse button manipulating the cursor into the right position before pressing the linefeed key and so on. These mechanical skills caused ‘start-up’ problems for many students, but they seemed especially prominent for Oliver and Jessie.

The students, who were using the early version of ‘First-Boxer’, also seemed reluctant to read the instructions on the screen. It quickly became obvious that the source of their reluctance was twofold: reading and readability. Jessie read very slowly and Oliver - at first – did not seem to read at all. But, equally, the screen presentation was not as clear as I had thought. Together, these factors motivated a redesign of the presentational format of ‘First-Boxer’.

One of the sources of Oliver and Jessie’s difficulties was related to the technical jargon of programming. In the first version of this micro-world, I used expressions such as ‘Execute the commands’ which were meaningless to them. In a later version, the expression ‘Try doing each of the commands in the menu.’ proved to be little better since students 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 suggested 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 to change. Replacing the term ‘procedure’ by ‘doit box’ was more than just a simplification of language. The new terminology was supported both by its visual and operational characteristics. Doit boxes appeared as rectangular boxes on the screen, i.e., as physical containers for programs, and moreover these (unnamed) doit boxes could be executed in place. When sub-procedures were physically inside the super-procedure, it was possible to refer to them as ‘inside doit boxes’.

The visual nature of Boxer facilitated a form of communication that was not possible with earlier languages. 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 which clearly helped students to navigate from one to the other. Information overload was reduced by simply closing boxes no longer in use. But, communicating in this way raised a second design issue concerned with reading: namely whether students read what was written on the screen – clearly an important issue in a microworld designed to be used independently.

Perhaps not surprisingly, I found that, if students were presented with a lot of writing, they didn't read at all. On the other hand, I observed that students 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 students 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. Figure 6.16 shows the evolution of these ways of presenting screen information.

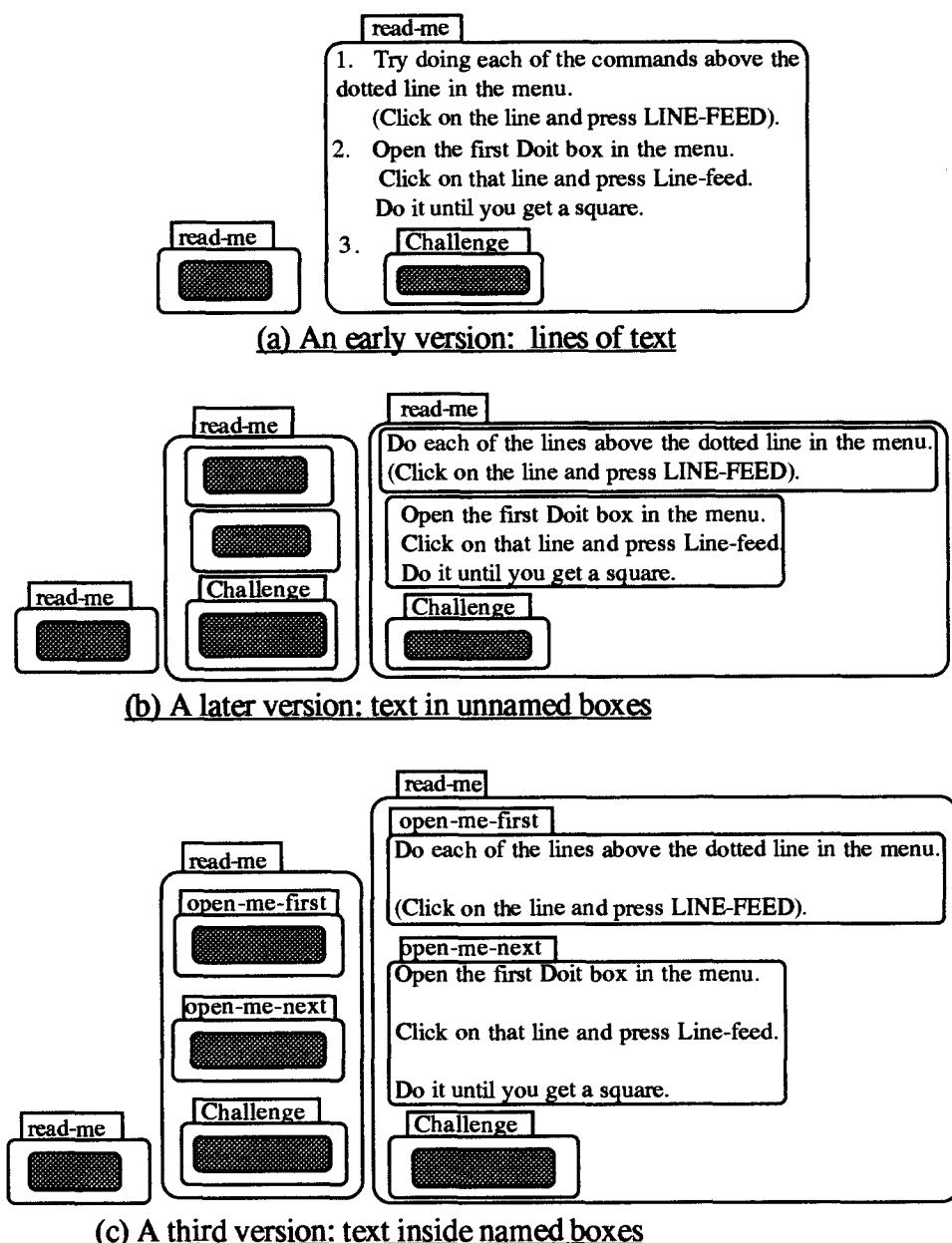


Figure 6.16: Changes to the presentation mode

Another presentational issue concerns the challenges in each section of the microworld. The notion of on-screen challenges supported by a graphic of the goal is one of the 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 students' difficulties or in order to better match the challenge to the key aim of that section of the micro-world. For example, in one early challenge (for shapes-1), I asked students to draw a hexagon, but this led to mathematical as well as programming difficulties and so it was removed. As stated earlier, I also increasingly devolved interventions to the challenges as a result of feedback, e.g., the need to consider state-transparency.

#### **6.4.2: Programming**

In my early attempts at writing programs in Boxer, I carried over practices from my experiences with Logo. Only later, as students' difficulties arose with matters such as the recognition of state-transparency, did I reflect that some of the programming problems might be alleviated by changes to the mode of presentation.

##### 6.4.2.1: Commands

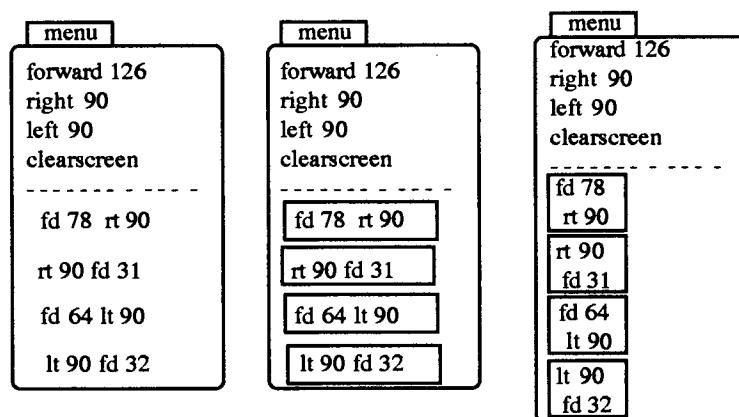


Figure 6.17: Changes to the presentation of commands

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. Firstly, placing text inside rectangular boxes emphasised the text as it would in traditional media, but it also drew students' attention to the fact that boxes were functional, i.e., they were programs that could be executed. Later still, each command

was placed on a separate line inside doit boxes. This last version meant that students 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 6.17 indicates the manner in which commands were presented evolved over the study.

#### 6.4.2.2: Procedures

Likewise, the way procedures were created evolved in the course of the study. In Logo, creating a procedure and naming a procedure are synonymous. This is not the case in Boxer, where procedures can be defined by un-named doit boxes. This facility proved to be of significance as the research progressed. Students frequently wrote and executed their programs line-by-line in an un-named doit box. This gave them a ‘correct-up-to-here’ facility. When the program did not perform as expected, they altered the offending line, placed the cursor outside the unnamed doit box and ran the whole program again.

The difficulties which the ‘Ls’ challenge in ‘shapes-4’ (Appendix 2) caused for several students led to changes in the way I approached the construction of procedures. Students, such as Joanna and Nico, for example, encountered problems because they were attempting to do too much at once. At their first attempt, they obtained a ‘squares-on-squares’ pattern rather than the ‘Ls’. With their next attempt, they obtained a set of ‘Ls’ without gaps between them. My interventions were now aimed at suggesting that, rather than trying to do all the ‘Ls’ at once, that they might first write a program to do a single L. Over the study as a whole, this evolved into a common programming practice.

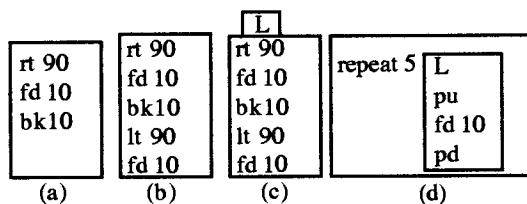


Figure 6.18: Combining ‘repeat’ with doit boxes

Figure 6.18 shows how this method might be adapted to the task of drawing five Ls. In (a), the commands can be stepped through line-by-line. In (b), it can be ascertained that the program to draw a single L is successful by running the whole doit box. Naming the box in (c) not only saves typing but reifies the existence of the procedure as an entity, whilst (d) shows a program to obtain the five Ls. The latter program can now be named also. This process seems rather natural in the visual environment of Boxer

and it is easy to overlook that it involves the creation and naming of procedures along with the use of modularity and sub-procedures. Moreover, debugging – usually seen as a post-programming repair activity – is built in to the construction of the program.

#### 6.4.2.3: Variables

The initial findings from the early parts of ‘First-Boxer’ replicated those of the research from Logo, e.g., students 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 so this represented an intermediate stage between the two Logo-like ways of using inputs. Because the students 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. Changing the contents of a data box made transparent the idea of changing the value of a variable, i.e., Boxer has visible variables. Some evidence for this can be deduced from Joanna’s description. Referring to the data boxes, she remarked: “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 students to ask if there was an ‘easier way to do it’. This led to a dilemma: the use of internal data boxes did lead to an appreciation of input, but their use imposed limitations. This was a design problem arising from the use of internal data boxes. For example, the nested square challenge could have been accomplished if the design had used either an external data box (global variable) or a general input variable. Figure 6.19 shows two ways of doing this.

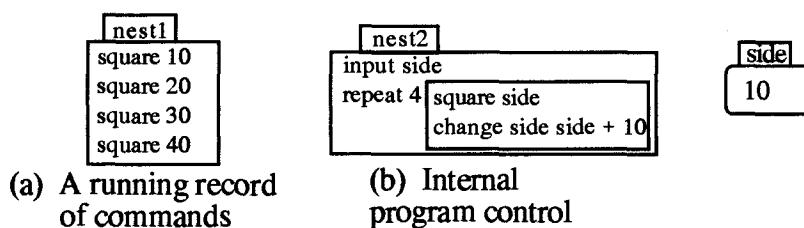


Figure 6.19: Two ways of doing the nested squares

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. In the event, students were introduced to external data boxes along with the command ‘change’ in the later sections ‘change-it’ and ‘build-it’ (section 6.3). Later still, they used general variables in the contexts of constructing number lines and function machines (chapters 8 and 9).

#### 6.4.2.4: Program Transparency

The section entitled ‘Build-it’ was in effect a design modification of ‘change-it’. In ‘change-it’, the numbers were displayed momentarily one-by-one in data box ‘display’. But, in ‘build-it’, all of the terms were printed across the screen. I made this modification because I thought that students might better appreciate the relationship between the terms of a sequence if they could see the pattern as it was being generated. Later, I added the line ‘sleep 0.5’ to emphasise this step-by-step generation of the sequence. Later still, the counter program itself was modified.

For my long-term number aim, I wanted students to construct their own microworlds. Therefore in presenting them with my own constructions, I felt that it was important that they should understand every part of it.

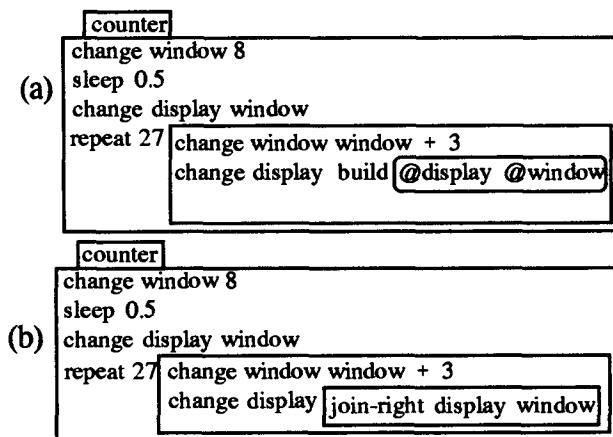


Figure 6.20: The two versions of ‘counter’

Originally, the ‘counter’ program used the ‘build’ command for concatenating terms across the screen (Figure 6.20a). Following trialling however, I felt that the programming code needed to achieve this aim was off-putting for students, so I made a design decision to create a new command ‘join-right’. From the students’ point of view, ‘join-right’ could be treated like any primitive command and it behaved in a

manner that could be understood intuitively. Moreover, the program used to create ‘join-right’ command was concealed in a ‘closet’ (chapter 3), and was thus available for inspection should any student be curious about its operation.

## 6.5: Summary and Implications

This chapter has examined the first phase of the longitudinal research in which the medium of Boxer was used to introduce itself. It has described how distinctive features of the technical component of Boxer (in comparison to Logo) influenced the way students learned to program, and reported how programming then became the vehicle for investigating number representations. It has also indicated how the students interactions with these microworlds became the data through which they were modified. The subsidiary role of this phase of the research in the study as a whole means that both programming and number findings were tentative at this point. The design findings, on the other hand, are unique to this phase and so are presented first.

### 6.5.1: Design Issues:

The idea of exploiting the medium to introduce itself evolved from the observations of the exploratory study. The findings raised important design issues which had not been anticipated at the outset of the research.

#### Presentation and Communication

The fact that the screen was being used as a means of communication as well as an environment in which to work raised several issues. On one level, the observation that changes to the presentational mode affects readability is common sense, and comparable to traditional media. But on another level, the fact that boxes can be closed and opened, hiding and revealing text, adds another dimension. An interesting insight into how one student perceived these characteristics can be gauged from Oliver’s remark: “It is sort of like a game book. You can make your own pages appear.”

The structure of the screen environment as a whole, and the on-screen challenges in particular, were modified as the research progressed as I attempted to devolve interventions to the medium. But the relationship between the microworld sections themselves remained relatively unchanged. An issue for future research is how the hypertext facilities of Boxer can be more fully exploited to offer alternative routes through a microworld or alternative challenges within it?

## Programming Constructs

The observation that presentational changes might affect program comprehension is again perhaps obvious. Yet, these are nevertheless non-trivial observations. Placing commands on separate lines which can be individually executed was a powerful means through which students connected the process and product of their programming. Similarly, presenting commands inside un-named doit boxes did more than merely emphasise those commands. Over the course of study, students adopted this as a way of working, i.e., they wrote and ran lines inside doit boxes, building up procedures as they went. This practice also meant that they debugged their programs as they wrote them rather than afterwards.

### **6.5.2: Programming Issues:**

Learning to program was secondary to the long-term goal of using this knowledge to construct learning environments in Boxer in which to investigate number. Nevertheless, at this point, some preliminary findings could be made.

#### Developing a sense of command

Developing a sense of command implies associating that command with a particular action, and relating its inputs (where applicable) to that action. The evidence suggests that students quickly acquired the sense of command, and this acquisition was influenced by the following two Boxer features:

- (i) the presence on-screen simultaneously of program and resultant drawing;
- (ii) the facility for line-by-line execution.

#### Procedure as an entity

Coming to appreciate a procedure as an entity implies associating it with a composite action. Noss (1985), for example, notes how using ‘procedures’ marks a significant step in students’ learning of programming. In Boxer, the students can *begin* their programming with procedures. The evidence suggests that the notion of a procedure was appreciated by students more quickly than might have been the case in Logo. The Boxer features which made this possible:

- (i) the facility for writing and executing commands directly inside doit boxes;
- (ii) the absence of the need to learn intermediate skills for entering and leaving the editor;
- (iii) the manner of naming procedures, i.e., the ‘name-tag’ boxes.

### Variables

The students in this phase of the research encountered variables in a wide range of contexts. They were also introduced to two of the three different manifestations of variables in Boxer: the internal data box and the external (global) data box. The evidence suggests that they came to appreciate the concept of a variable as representing a range of numbers. The Boxer features which contributed to this learning were:

- (i) the spatial appearance of data boxes as containers;
- (ii) the facility of being able to directly manipulate the contents of the data boxes;
- (iii) the manner of naming variables, i.e., the 'name-tag' boxes.

Hillel (1992) observes that the variable concept is introduced in Logo through generalised procedures, and hence the learning of the one is tied up with the learning of the other. The data of this study implies that in Boxer this need not be the case.

#### **6.5.3: Number Issues:**

In these microworlds, four different number contexts were investigated as part of my continuing efforts at finding an evocative Boxer representation. None of them were fully satisfactory. The area context was unsuitable because the students' notion of area was insufficiently developed. The co-ordinates context succeeded in motivating the use of directed numbers, but I could not see it doing the same for decimals. The counter representation supported visually the relationship between numbers by generating a pattern across the screen. However, I had some reservations about whether students would find it too difficult to program something similar. Finally, the list-processing command 'for-each-item' was not as accessible as I had hoped.

#### **6.5.4: Implications for the rest of the thesis**

The long-term nature of learning to program meant that the programming achievements of this phase of the research would have to be judged over the longitudinal study as a whole. On the other hand, choosing a representation in Boxer which would prove both visually evocative in terms of students' conceptions of number and be within their capacity to program was an immediate priority.

In order to assist my search for such a representation, I decided to investigate more closely the kinds of knowledge which these students already had of number, as well as the kind of number representations which they might spontaneously deploy. Chapter 7 gives an account of how I undertook this part of the research.

# **Chapter 7: Students' Interpretations and Representations of Numbers**

## **7.1: Introduction and Overview**

### **7.1.1: Introduction**

This chapter consists of an analysis of the semi-structured interviews that were conducted with the case-study students and a comparison group of non-Boxer students from a parallel year 6 class. The interviews took place during the Autumn term September 1992 to December 1992, following the programming phase of the longitudinal study (chapter 6), but prior to the number phase (chapters 8 and 9). The intention was to use this period to generate more focused and coherent issues for the number phase of the study.

A precedent for such an investigation can be found in diSessa et al. (1991), who report a study in which students 'invented' representations to depict motion. In discussing and critiquing their own and other students' representational forms, these students gained 'meta-representational' competence, i.e., they acquired an appreciation of the relationship between motion and the various ways in which it could be depicted. By undertaking a similar off-computer study with respect to number, it was hoped to establish some baseline data on the Boxer students' understanding of number, and to gain insight into the representations that they employed and found helpful.

The aims of this phase of the research were threefold:

- (i) to establish baseline data on the case-study students' knowledge of number in general, but of directed numbers and decimals in particular;
- (ii) to establish the kinds of representations that the students used for different numbers, and to solicit their views concerning the suitability of these representations for extended systems of numbers;
- (iii) to probe differences in the number knowledge and representations exhibited by the Boxer and non-Boxer students in the hope of gaining pointers for the next phase of the research.

Following a pilot study, modified interviews were conducted with the case-study students<sup>1</sup> and a comparison group of eight non-Boxer students from a parallel year 6 class.

### 7.1.2: Overview

Section 7.2 is the first of a two-part analysis of students' conceptual understanding of number. It is concerned with their interpretations of quantities obtained through extending the number system. It gives an account of findings which reveal students' thinking about the existence of such quantities, as well as the relationship between numbers. Questions about representations usually followed immediately after those on interpretations. Thus, if a question asked students to write down numbers between 0 and 1, the following question might ask them to create a representation to depict their numbers. However, for the purpose of this analysis, I have separated the interpretation from the representation items.

In section 7.3, the focus is on the documentation and analysis of students' visual images of numbers. For the purposes of analysis, it is helpful to draw again on Bruner's (1968) of enactive, iconic, and symbolic representation. Although, it is Bruner's notion of iconic representations that will be of most relevance to that discussion, nevertheless, the enactive roots of students' drawings can easily be discerned in their choices of icons, and at the other end, the merging of iconic forms with symbolic notation can be seen in the work of students like Joanna.

Von Glaserfeld (1987) observes that representations of numbers, do indeed, provide a complex illustration of what is iconic and what is not:

With regards to icons, Piaget's distinction between the "figurative" and the "operative" would seem to be of some importance. Number is not a perceptual but a conceptual construct; thus it is operative and not figurative. Yet, perceptual arrangements can be used to "represent" a number figuratively. Three scratches on a prehistoric figurine, for instance, can be interpreted as a record of three events. In that sense they may be said to be "iconic", but their iconicity is indirect. (1987, p. 223)

Section 7.3 documents the kind of representations which the students chose for natural numbers, rational numbers and directed numbers, and discusses the limitations inherent in certain of these.

---

<sup>1</sup> Interviews were also carried out with three other non-case study students from the Boxer group: Kathleen, Rosie and Sara, all of whom later participated in one or other of the number studies.

## 7.2: Students' Interpretations of Number

### 7.2.1: Natural Numbers

By way of a gentle introduction to the interviews, students were simply asked to choose any three numbers, and having done so, were then asked to select two of these numbers (Q1a and Q1b).

Q1a: Choose any three numbers and write them down.

Q1b: Now choose two of these numbers and write them down.

All of the students chose whole numbers, which would seem to suggest that, for them, the concept of number was equated with whole number. Moreover, most of the selected numbers were relatively small (all but one were within the range 1 - 100), suggesting further that the numbers chosen were those within their experience. There were no discernible differences between the Boxer and the comparison group of students.

Table 7.1: Students' choice of whole numbers

Non-Boxer		Boxer	
Ann-Maria	<b>10, 11, 15</b>	Nico	<b>3, 9, 8</b>
Tahir	<b>10, 15, 40</b>	Joanna	<b>3, 8, 15</b>
Blake	<b>3, 5, 9</b>	Jason	<b>1, 6, 11</b>
Hannah	<b>3, 20, 69</b>	Liam	<b>6, 19, 210</b>
Ahmet	<b>9, 15, 20</b>	Courtney	<b>5, 24, 18</b>
Cerisse	<b>3, 6, 9</b>	Laura	<b>1, 2, 3</b>
Rebecca	<b>10, 120, 3</b>	Jessie	<b>2, 4, 9</b>
Dillon	<b>1, 100, 5</b>	Kathleen	<b>9, 12, 55</b>
		Rosie	<b>12, 6, 22</b>
		Sara	<b>7, 3, 2</b>

The final two numbers chosen are in bold

### 7.2.2: Rational Numbers

Question 1, besides intending to help the students feel at ease, was also meant as a lead-in to question 3, which probed students' awareness of non-integer quantities. As part of the protocol, I continued to ask this question until the students gave two successive integers. I then asked the students to choose a number between these. In contrast to the responses to question 1, there were very marked differences between the Boxer and non-Boxer group's answers to question 3.

Q3 a, b, c. etc.: Can you write down a number that is bigger than [the smaller number] and smaller than [the bigger number]?

A frequent first response at this point was for the non-Boxer students to deny that there were any numbers between two whole numbers. Their later fractional answers, which followed researcher interventions, reinforced the contention that the term ‘number’ was seen - at least initially - as synonymous with whole number. It was very noticeable that all of these students gave fractional answers and most of these (Dillon’s final response was the exception) were obtained by employing a bisection strategy. It suggested that their notion of fractional quantities was tied to the physical operation of halving. Only one student (Dillon) was able to identify more than two numbers between the successive integers.

**Table 7.2: Students' successive choices of non-integers**

<b>Non-Boxer</b>		<b>Boxer</b>	
Ann-Maria	10½, 10¼.	Nico	3.5, 3.49, 3.489, 3.487.
Tahir	15½, 15⅓.	Joanna	8½, 8¼, 8⅓, 8⅙.
Blake	3½, 3⅓.	Jason	1½, 1⅓, 1⅔, 1⅕.
Hannah	3½, 3⅓.	Liam	19.5, 19.3, 19.1, 19.01.
Ahmet	9½.	Courtney	5.5, 5.1.
Cerisse	6¾, 6½.*	Laura	1½, 1¼, 1⅓.
Rebecca	4½, 4⅓.	Jessie	2½, 2⅓, 2⅔.
Dillon	5½, 5⅓, 5⅔.	Kathleen	9½.
		Rosie	6.5, 6.1, 6.05.
		Sara	2½, 2⅓.

\* Wrote 1½ instead of 6½

Some of the Boxer students also employed a bisection technique, but there were other indicators that this was not the sole strategy available to them. Joanna, for example when asked to produce a drawing to represent her numbers (see section 7.3), switched from fractions to decimals and chose a new set of numbers: 8.0, 8.25, 8.23, 8.10, and 8.05. The Boxer students also tended to choose more numbers. There was a marked difference between the two groups in their use of decimals. Five of the ten Boxer students gave a decimal expression as a number in between two whole numbers, in contrast to the non-Boxer students where no student gave a decimal answer to this question. In a later question (Q.9), I again probed the students' appreciation of decimals, and again, there were marked differences between the two groups.

**Q9a:** Do you know about decimals?

**Q9b:** Which is the bigger 0.5 or 0.13?

All of the comparison group claimed to know about decimals, but only two out of the eight chose 0.5 as bigger than 0.13 (Table 7.3). At the time of the interviews, the non-

Boxer students had begun to learn about decimals in the context of money, and this may account for the disparity between their two answers.

Eight of the ten Boxer chose the correct answer 0.5. But, again some caution needs to be exercised in interpreting their answers. Most of the students recognised 0.5 as a half, but several of them had difficulties in explaining why 0.13 was a smaller quantity. This may be because this is a difficult question to answer orally. I did not simply accept the Boxer students' answers as construing self-evident knowledge of decimals, but rather continued to probe the rationale for their choice.

**Table 7.3: Students' knowledge of Decimals**

Non-Boxer			Boxer		
	<b>9a</b>	<b>9b</b>		<b>9a</b>	<b>9b</b>
Ann-Maria	yes	0.13	Nico	yes	0.5
Tahir	yes	0.13	Joanna	yes	0.5
Blake	yes	0.13	Jason	yes	0.13
Hannah	yes	0.5	Liam	yes	0.5
Ahmet	yes	0.13	Courtney	yes	0.5
Cerisse	yes	0.13	Laura	yes	0.5
Rebecca	yes	0.13	Jessie	not sure	0.5
Dillon	yes	0.5	Kathleen	not sure	0.13
			Rosie	yes	0.5
			Sara	yes	0.5

Kathleen, who chose 0.13, said: "But, I think I'm wrong." Courtney, who chose the correct option, asked if it was a 'trick question', and could not explain why 0.5 was bigger than 0.13. Jessie chose 0.5 and knew it was a half, but she could not explain why it was bigger than 0.13. Laura justified her choice of 0.5 by saying: "The lower it is, actually the higher it is." The reasoning for this classic 'LS' (larger is smaller) error could be inferred from what she said a little while later, i.e., "That's a half and that's a thirteenth."

### 7.2.2: Directed Numbers

Question 5, like question 1 for decimals, was intended as a lead-in question for further investigation of students' conceptual understanding of directed numbers.

**Q5:** What is the smallest number that you know?

Often a first response by the non-Boxer students to this question was to ask "Is zero a number?". Assured that it was, many chose it as their smallest number. For some, there also seemed to exist a quantity '-0', which may have arisen because they had

heard of temperatures below zero, although this is speculation on my behalf. Six of the eight chose 0 or -0, with Rebecca's '0 –  $\frac{1}{4}$ ' being a variation on this.

A frequent first response by the Boxer students was to ask: if they could use 'minuses'. Five of the ten did so, choosing quantities ranging from Kathleen's -1 to Nico's minus sign followed by his depiction of a infinite number of nines. Of the remaining five students, three chose decimals and two chose 0.

**Table 7.4: Students' choice of smallest number**

Non-Boxer		Boxer	
Ann-Maria	-0	Nico	-99, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999 ....
Tahir	-0	Joanna	-1000 000 000 000 000 000 000 0
Blake	0	Jason	0.1
Hannah	-infinity	Liam	0.00001
Ahmet	0	Courtney	0
Cerisse	0	Laura	-infinity
Rebecca	0 – $\frac{1}{4}$	Jessie	0
Dillon	0	Kathleen	-1
		Rosie	-50
		Sara	0.1

The question arises as to why the responses of the two groups should have been so different? Neither of the groups had been exposed to negative numbers as part of the formal school curriculum. However, the Boxer students were coming into contact with both decimals and directed numbers in the course of their work in the computer environment. These findings suggested that these contacts were indeed affecting their expression of number. They appeared to have both a wider and a deeper insight of numbers than their non-Boxer counterparts. Two follow up questions (Q6 and Q8) reinforced this impression.

**Q6a:** Can you think of a cold temperature?

**Q6b:** Can you give me a colder temperature?

The responses to question 6 appeared to contradict the answers to question 5, in that many of the students who gave 0 as their smallest number responded by depicting numbers below zero. However, three points may be proffered as partial explanations. Firstly, there is the obvious one of temperature being a meaningful context: it is not claimed that they knew which number was smaller but rather which temperature was colder. Secondly, it could be that the term 'smallest' is ambiguous when negative numbers are involved, i.e., students may have been giving the smallest absolute value

$|x|$  of a number  $x$ . Thirdly, 0 may be the smallest number that students have experienced.

Several of the students, both Boxer and non-Boxer, used non-standard notation to depict the relative values of negative numbers in the context of temperature.

Table 7.5: Students' comparison of Temperatures

Non-Boxer			Boxer		
	6a	6b		6a	6b
Ann-Maria	20 below 0	30 below 0	Nico	0°c	-10°c
Tahir	1 c	1/2 c	Joanna	0°c	-1°c
Blake	C 3	-1 C o	Jason	-40° C	-60° C
Hannah	-0° C	-20° C	Liam	1 F	0.1 F
Ahmet	-0	Freezing	Courtney	1°c	0°c
Cerisse	-1	-0	Laura	4°c	-22°c
Rebecca	0 - 1/8 c	0 1/24 c	Jessie	Cold 1 or 2 o F°	Freezing o F°
Dillon	0 c -	100 c -	Kathleen	5°FH	below 1
			Rosie	-10	-30
			Sara	0.1	below freezing

Another question (Q8) was designed to elicit further information concerning students' conceptions of both decimals and directed numbers in terms of the real number continuum.

Q8: Suppose the temperature was minus one and it warmed up to one.  
Can you write down any numbers it would pass through?

Table 7.6: Students' interpretation of the Real Number continuum

Non-Boxer		Boxer	
Ann-Maria	2, 6, 8, 10	Nico	- .6 -.5 0 .3 .4 .5
Tahir	5, 7, 3, 8	Joanna	0.05, 0.10, 0.15, 0.25, 0.50, 0.75
Blake	-00 -1 00 1	Jason	-1 -0.1 -0.2 -0.3 -0.4 -0.5 -0.6 -0.7 -0.8 -0.9 0 0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9, 1
Hannah	-1 -0 1	Liam	00.1 00.2
Ahmet	-2	Courtney	-2 -4 -6 -8 -10 +
Cerisse	-7	Laura	-1/4, -1/8, -1/2, -1, +1/4, +1/2, +1/8, +1/16
Rebecca	0 1/2	Jessie	1/2
Dillon	+1 +50 +90 +Hot	Kathleen	1 1/2
		Rosie	-1.55 0.5 0.05
		Sara	0.1 +1 1/2 1/4 1/3 3/4 -1/2 -1/4 -3/4

The qualitative differences in the responses to this question by the non-Boxer and Boxer groups were quite stark (Table 7.6). Only three of the eight responses by the

non-Boxer group could be interpreted as being correct. The response of two of the ten Boxer students, Courtney and Kathleen were also incorrect, and Rosie's -1.55 was outside the specified range. Of the remaining seven students, six gave two or more fractional or decimal quantities. Jason, for instance, wrote down all the numbers between -1 and 1 obtained by using 0.1 increments.

On the other hand, it seems clear that there was some confusion with the relative sizes of negative numbers, particularly if these were not whole numbers. Joanna, who gave only positive decimal numbers in her answer, was asked whether she could write down any numbers between -1 and 0. The difficulties she had in doing so can be gauged from her comments: "I don't know. If you write a number like one point something that's making it smaller, because you go down in the negatives. But if you make it zero point something, that's above zero."

### **7.3: Students' Representations of Numbers**

Reacting to a conference on representation, Belanger (1987) warned of the dangers of neglecting students:

I have felt during our deliberations that students are strangely absent from our discussion; this is similar to the period of the 1960s when they were absent from the new math and the new science and there is a danger they will be missing from the new representations. One of the things we need to remember is that students construct representations. (1987, p. 105)

In the light of Belanger's warning, this section can be seen as constituting my efforts to ensure that students' contributions were included.

#### **7.3.1: Natural Numbers**

Having chosen two whole numbers for question 1, the students were asked if they could illustrate their choice with drawings.

**Q2:** Suppose you were trying to explain to a younger child which of your two numbers was the bigger, can you make a drawing that would help her or him to understand?

Their responses to this question indicated that the relationship between what was represented (the signified) and the mode of representation (the signifier) was more complex than had previously been supposed. It seemed that for some students, it was sufficient to depict the 'bigness' of one number relative to the other – they felt no need to accurately preserve the absolute size of each number. Secondly, there was some ambiguity as to what the representations were meant to convey. There was also some

doubt regarding whether the visual representation was intended to be a prop for the accompanying symbol or meant to stand independently of it.

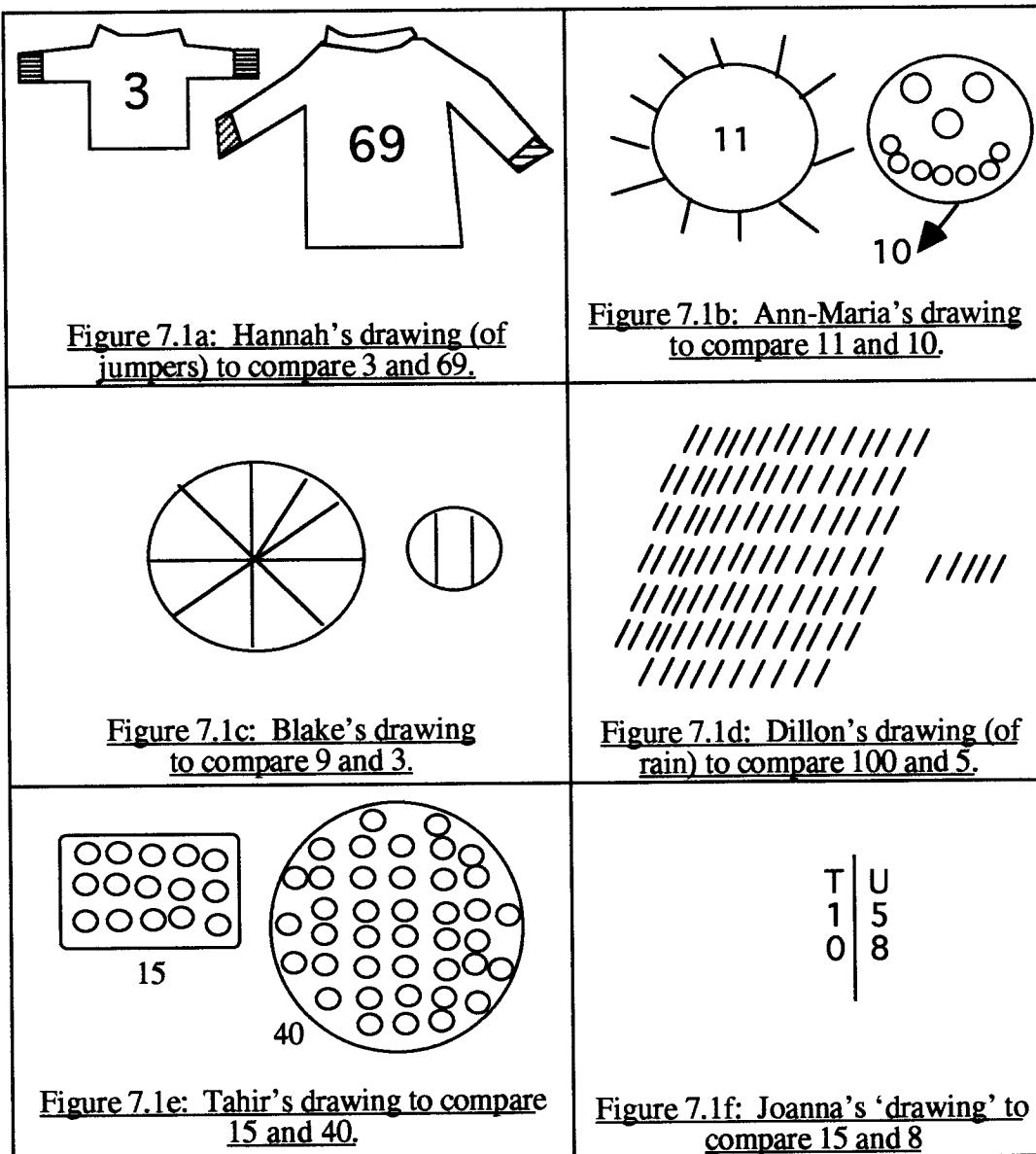


Figure 7.1: Students' representations to compare whole numbers

In the follow-up discussions, the students themselves critiqued the limitations in their own and other's representations. Hannah, who had drawn the picture of jumpers (Figure 7.1a)<sup>2</sup>, reacted to Maker's preference for it by pointing out what she thought was its weakness.

---

<sup>2</sup> The representations which make up this and subsequent figures are a sub-set of the representations given by all of the students. These can be found in Appendix 5.

Extract 1:

Speaker	Dialogue	Commentary
Maker:	I like this one. I like that because it shows size.	Points to the drawing of jumpers. [Figure 7.1a].
Hannah:	I don't think that's a very good idea.	
Interviewer:	You don't think your own idea is very good now. Why not?	
Hannah:	Because if it was a bigger number, this would be like gigantic.	Refers to the difficulties in keeping the drawings in proportion to the numbers.

Figure 7.1a was also criticised by another child because “It is not clear what the numbers stand for.” Other students criticised figure 7.1b because “One is for Summer and one is for Winter, but what’s that got to do with 10 and 11?” Figure 7.1c was thought defective because “The parts aren’t the same”, but there was disagreement between Rebecca and Cerisse about the merits of figure 7.1d (the raindrops).

Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	[to Rebecca] Why do you not like it so much?	
Rebecca:	It would be hard to count.	
Cerisse:	Yes, but even if it is hard to count, they’ll still see the difference because of the amount of raindrops.	Refers to depiction of relative sizes.

Liam, who thought the drawing of stones (Figure 7.1e) was good was countered by Joanna leading to a discussion on whether a diagram should be able to convey its meaning without the need for back-up writing.

Extract 3:

Speaker	Dialogue	Commentary
Liam:	That one. Fifteen is smaller than forty.	Points to Figure 7.1e.
Joanna:	Yea, but just looking at that, you couldn't tell the difference between thirty-nine and forty.	
Liam:	But you can write the numbers next to them.	
Joanna:	But, in that case, you can just write them.	

Joanna’s implicit use of an abacus mirrors a common teaching approach to place value. Before drawing/writing it, she asked “Do you want a ‘drawing-drawing’, or can I write it?” This hints at an intermediate interpretation of this representation, i.e., between that

of a drawing and the purely symbolic form of a number. Jessie, chose this diagram saying: "If they understand about tens and units, that one would be quite good". This was also the response of Rosie, who said: "If children already knew a lot about numbers, that one would be a good idea."

### 7.3.2: Rational Numbers

Just as question 2 was intended to evoke representations to illustrate the two whole numbers from question 1, question 4 was meant to fulfil the same role with respect to the rational numbers from question 3.

- Q4:** Suppose you were trying to explain to a younger child which of your two numbers was the bigger, can you make a drawing that would help her or him to understand?

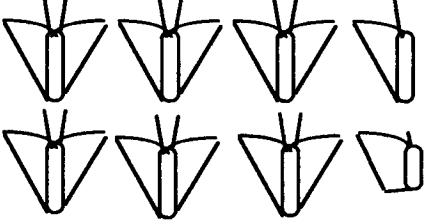
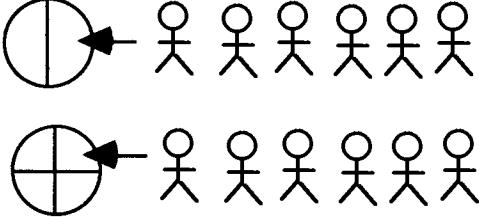
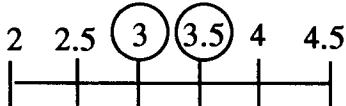
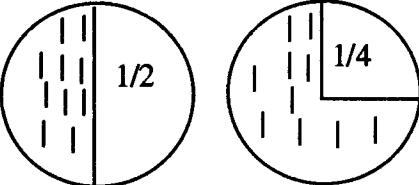
	
<p>8.0 8.25 8.23 8.10 8.05</p>	
<p>Figure 7.2c: Joanna's 'drawing' to compare 8 and 8.25</p>	<p>Figure 7.2d: Nico's drawing to compare 3 and 3.5</p>
<p>19 19.1 19.2 19.3 20</p>	 <p>Figure 7.2f: Ann-Maria's drawing to compare 10 1/2 with 10 1/4</p>

Figure 7.2: Students' representations to compare rational numbers

Often students chose representations for fractions which were similar to those used to compare whole numbers even when these are clearly inappropriate. An answer which included half a person or half a car seemed to be regarded as quite acceptable. Hannah was the exception to this. She later described her own diagram (Figure 7.2a) showing portions of butterflies as “Disgusting”. Students also mixed representations within the same drawing as in Cerisse’s case (Figure 7.2b), where stick people were used for whole numbers, but fractions were represented by parts of a circle. Another phenomenon observed was for students to represent the fractional parts only, ignoring the whole number part. In general, the students experienced some difficulties in conjuring up a representation for non-whole numbers (Dillon, for example, could not produce one). This may be due to the novelty of the idea, i.e., representing numbers rather than manipulating them.

The phenomena of using inappropriate representations, of mixing representations, and of depicting only the fractional parts were also observed amongst the Boxer students. But, also observed were tendencies towards representations which depicted all numbers and a general trend towards more symbolic notation. Joanna and Nico consistently chose conventional classroom representations, the implicit abacus model for Joanna (Figure 7.2c) and the number line for Nico (Figure 7.2d). Although he did not draw a number line, Liam’s depiction of numbers in a line (Figure 7.2e) was clearly equivalent, as can be gauged from the following extract.

**Extract 1:**

Speaker	Dialogue	Commentary
Liam:	You have nineteen and twenty there.	Writes 19 on paper, leaves a space and then writes 20.
Liam:	Now, in between these two numbers, there are lots of littler numbers. No, not littler. There are lots of other numbers. First, there is nineteen point one.	Writes this.
Liam:	... and then it goes up like this to ten.	Writes 19.2 and then 19.3.
Liam:	But then there are more littler numbers. No, more numbers. They, go up like this.	Counts up 19.4 19.5 etc.
Liam:	They go on to ten which is twenty.	Refers to next step in the sequence beyond 19.9.

This clearly means that  $19.9 + 0.1 = 20$ , but note how orally counting with decimals might go nineteen point eight (19.8), nineteen point nine (19.9), and then nineteen point ten (19.10).

In the follow-up interviews with both Boxer and non-Boxer students, criticisms of other representations tended to be linked to the accuracy of the representation of the fractional part. Rosie's comment on Figure 7.2a was typical: "Because if you were trying to show a small fraction, like a tenth, you wouldn't be able to see it."

### 7.3.3: Directed Numbers

As explained earlier, questions about directed numbers were asked in the context of temperature, and so the request for representations were similarly framed (Q7):

- Q7: Suppose you were trying to explain to a younger child about temperature, can you make a drawing that would help her or him to understand?

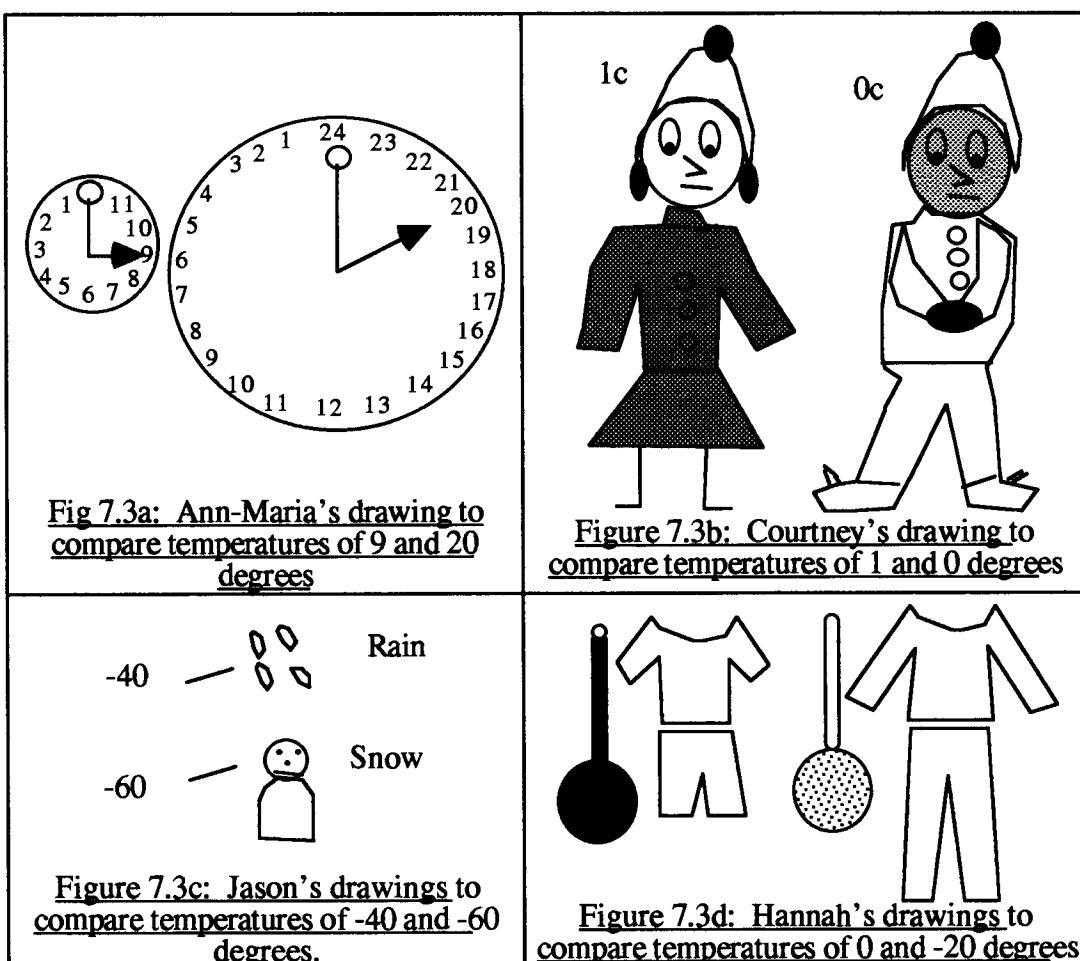


Figure 7.3: Students' representations to compare directed numbers

Not surprisingly, many students drew thermometers, but often these were graduated in a way that gave no clear indication of the difference between their two numbers, or not graduated at all. In this sense, they might best be considered as labels for temperature.

Hannah's drawings of summer and winter clothes (Figure 7.3d) alongside thermometers to denote the difference between her two temperatures of 0 and -20 degrees lends weight to this labelling conjecture. The use of light and warm clothing was also used by Courtney (Figure 7.3b), who added 'cold' and 'warm' shading to her images. Jason's juxtaposition of numbers next to images of rain and a snowman (Figure 7.3c) again draws attention to one of the deficits of this form of representation, namely the lack of clarity concerning the nature of the relationship between the number and its accompanying drawing.

## 7.4: Summary and Implications

This chapter has analysed the semi-structured interviews which were conducted with eighteen students in a non-computer setting. It has detailed how questions were posed which probed students' conceptions of number. Two indicators of these conceptions were used: interpretations and representations.

### 7.4.1: Students' Interpretations of Numbers

The findings show that there were few problems with whole numbers. However, this was not the case for non-integers. Pimm (1987) observes there is no such thing as a decimal number or a fractional number per se.  $2.5$  and  $2\frac{1}{2}$  are simply different representations for the same number. However, my questions designed to probe students' interpretations of such numbers suggested that the relationship between signifier and signified is not a simple one. Fractional notation, used predominantly by the comparison group, carried with it an action-based connotation of its derivation, i.e., 'a half of', 'a quarter of'. 'In-betweeness' could thus be signified by such expressions. The greater use of decimals by the Boxer group appeared to suggest a greater appreciation of a number continuum, but attaching meaning to decimal symbolism was problematic for both sets of students.

Investigating directed numbers in the context of temperature demonstrated the advantages of a meaningful context for all of the students. There were some conceptual problems in deciding whether zero was a number, and some unorthodox notation to depict temperatures below zero. The greater use of directed numbers by the Boxer group again pointed towards a greater appreciation of a number continuum, but there was some confusion with the relative sizes of negative numbers, particularly if these were not whole numbers.

#### **7.4.2: Students' Representations of Numbers**

It is apparent that there were a wide variety of representations employed by the students to denote number. Some of these (the thermometers) were used only for directed numbers. Others (the match-stick men) were used primarily for natural numbers, with some attempts to extend them to non-integers. It is not clear how these might be used for depicting directed numbers. Still others were idiosyncratic, as might be expected with such young students.

Vergnaud (1987b) points out that representations always involve keeping some features of the concept in focus whilst losing sight of others. Thus the way that symbols are used in the representation of the concept determines what is represented and what is not represented at particular moments. As Pimm (1995, p. 119) writes: "One function of representation can be to allow access to some event, although any choice of representation carries with it certain stressings and ignorings." Certainly, the students in this research often emphasised one feature at the expense of another. For example, in comparing mixed numbers, the fractional parts were stressed whilst the whole number parts were ignored. Ann-Maria's device of using a conventional 'parts-of-a-whole' diagram (Figure 7.2f) for fractions but tally marks for whole numbers can be viewed as one child's attempt to represent both.

Care must be taken that, in our critique of their constructions, we do not adopt a patronising stance. The limitations which students identified in their own models of numbers alert us to deficits in our own conventional representations. It is common, for example, to use parts of rectangles or circles to depict fractions, but it is questionable whether such a representation could be easily extended to include large numbers, and it is not at all apparent how directed numbers might be represented in this form. The number line representation may appear to overcome some of these difficulties, but it may be necessary to use a device such as 'zooming-in' or 'zooming-out' for small and large numbers respectively. This chapter shows how students' representations were particular to the numbers they were depicting, but this is not unlike the practice of mathematicians, and model builders in general.

#### **7.4.3: Implications for the Remainder of the Thesis**

The models used in elementary school textbooks to depict number vary according to both the type of number and the operation being modelled. Not unnaturally, there is an implicit assumption that if students are learning about number operations, then there

already exists a base level of understanding about the numbers themselves. The evidence of this chapter challenges that assumption.

It shows that, while students did not naturally use either directed numbers or decimals, contexts could be found which elicited the use of the former. In the latter case however, similar contexts very often elicited the use of fractions rather than decimal expressions, particularly from the non-Boxer students. Furthermore the students' own explanations seemed to point towards understanding gained as a result of operations rather than preceding them. A hypothesis was generated to the effect that the medium could foster 'an operational approach' to number.

Thus, the usual teaching sequence could profitably be reversed. Rather than beginning by introducing decimal quantities and directed numbers, to be followed by operations on those quantities, the operations would be the means by which the numbers themselves were first created. Subtraction would be the means by which directed numbers acquired meaning, while division would provide an introduction to decimal quantities. An epistemological foundation for such an approach can be located in the formal axiomatic extension of IN to Z and Z to Q. Peano's Axioms (Stewart and Tall, 1976) encapsulates precisely this technique to extend the Natural Numbers to the Integers, the Integers to the Rationals and the latter to the Real numbers.

The findings also show that students' representations of numbers tended to be linked to the particular numbers. In some cases, it was not clear whether diagrams were intended to stand alone and in others, it was uncertain what aspect of number was signified by the diagram. Where whole numbers and rational numbers were mixed, it was often the case that students also mixed representations: one aspect of the diagram to denote the whole number and another to denote the rational part. Representations for directed numbers were generally quite different than those for positive rational numbers. The desirability of having a single representation for all numbers was not generally appreciated.

The purpose of this phase of the research was to generate more focused and coherent issues for the number phase of the study. Strictly speaking, this could have been done using only the Boxer students. However, having a comparison group not only increased the knowledge base, but also accentuated differences between the groups which otherwise would not have been possible.

Two main differences were observed. In the first place, it seemed that the Boxer students were more inclined to interpret numbers in a manner that suggested a greater insight into the existence of a number continuum. Secondly, the Boxer students were more likely to employ representations which were amenable to extended systems of number. It was not possible to attribute these differences to any one episode they had experienced within the Boxer microworlds prior to these interviews, but it seemed most likely that the ‘counter’ representation (Chapter 6) might have influenced their notion of a number continuum.

However, this computational object did not appear to be within the capacity of the students to program. The challenge now was to find a representation which offered both the visual and dynamic support of such a device and yet could be constructed by the students on their own. This challenge was resolved by the creation of a new representation: the number line object. Chapter 8 gives an account of how the students constructed a dynamic number line object in Boxer, how interaction with this object fostered an operational approach to number, and thus enabled me to gain greater insight into how Boxer was mediating their developing sense of number.

# Chapter 8: The Number Line Study

Within computational environments which genuinely offer the opportunity for the expression of mathematical ideas, pupils can use ideas *before* they have fully discriminated the relationships involved, and moreover, use them in ways which inevitably push in the direction of formalisation. (Hoyles and Noss, 1992b, p. 32)

## 8.1: Introduction and Overview

### 8.1.1: Introduction

This chapter consists of an analysis of the programming, and number issues which arose as a result of students constructing and interacting with a Boxer number-line microworld. It describes how the case-study students began to utilise their programming knowledge to construct an ‘operational’ number line in Boxer, and how the insight gained through this construction influenced their subsequent interactions with it. It then analyses the ways that this ‘evocative computational object’ (Hoyles, 1993, cf. Hoyles and Noss) mediated their expression of number, particularly in respect of decimals.

This was the first of the two number studies which formed the final part of the longitudinal research, and which lasted from January 1993 to July 1993. A total of eight year 6 students (ages 10 – 11) were observed. Seven of these eight had participated in the earlier First to Third-Boxer microworlds, whilst one (Kathleen) had replaced Oliver as Jessie’s partner. With this one exception, all had approximately 20 hours programming experience in Boxer.

The exploratory work (chapter 6) implied that students should be involved in the construction of representations, rather than being confined solely to interpretation. The evidence from the interviews (chapter 7) indicated that, of the representations which students employed, the number-line was perhaps the most accessible and also the most flexible in terms of depicting rational numbers. The interviews also seemed to point to students acquiring meaning for numbers gained as a result of operations rather than preceding them. By having students program their own Boxer number line representation, which would form the basis of subsequent number investigations, my intention was to build on these earlier research findings.

The aims of this phase of the research were thus:

- to document the programming issues involved in students constructing a number line object in Boxer;
- to investigate how their interactions with this ‘operational’ object mediated their understandings of number.

### **8.1.2: Overview**

Section 8.2 consists of vignettes from two of the four case study pairs with the intention of showing how students began to apply and extend their programming knowledge in the course of constructing the Boxer number line.

Section 8.3 recounts how the students’ interactions with this Boxer object mediated their expression of number. It points to the importance of the construction process in structuring these interactions, and it attempts to link aspects of the students manipulation of the programming code with these expressions.

## **8.2: Constructing the Boxer Number Line**

Mathematicians seem commonly to possess an iconic number-line permanently available in their heads, and we might attempt to give children a similar icon. (Plunkett, 1981, p. 183)

As the longitudinal study progressed, the emphasis changed from programming to number. Programming became a means to an end rather than simply an end in itself. The following two episodes have been selected to show how, in the course of constructing the Boxer number line, the case study students began to apply and extend the programming knowledge gained in the earlier phase (chapter 6). But, it needs to be stressed that the programming activity was of equal importance in terms of the insight it gave to me into the ensuing interactions with the object of their construction: the number line.

### **8.2.1: Constructing the Number line 1**

The following episodes feature two girls Jessie and Kathleen during their first two hours (approximately) of programming.

Jessie and Kathleen were chosen by their teacher as being representative of the bottom quartile of this year six class. Jessie and Oliver formed one of the original four case-study pairs. This partnership had been dissolved towards the end of the previous year when it appeared that the work was becoming too difficult for them, and when it seemed that they were no longer willing to co-operate (chapter 4).

As the number-line study got under way, Jessie pleaded to again be allowed to participate in Boxer work, and even went to the extent of finding herself a new partner: Kathleen. As Kathleen, like Jessie, was also from the ‘bottom quartile’ of the year six class, this allowed the research to again represent the whole ability range. Nevertheless, it needs to be emphasised that Kathleen had very little Boxer experience in comparison to Jessie.

#### 8.2.1.1: Episode 1 – Separating Process and Product

This episode has been chosen to show how Kathleen and Jessie separated the process and product of procedures in the course of constructing their number line.

At the beginning of the session, I created an on-screen ‘challenge’ box which invited the girls to make a sideways ‘T’, the intention being that this might be repeated to make the number line (Figure 8.1).

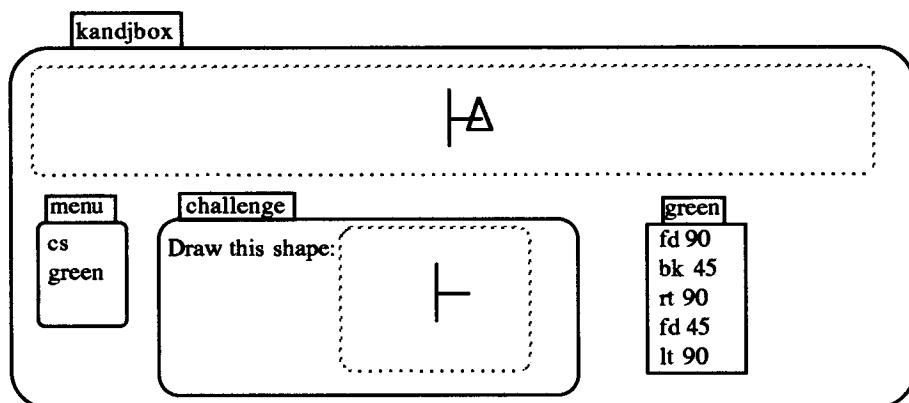


Figure 8.1: The ‘T’ challenge

The girls made a doit box which they named ‘green’ (the colour of Kathleen’s shirt), in which they typed the first command ‘fd 90’. I assumed this to be a case of confusing angle measure and linear measure. However, Jessie’s next command was ‘bk 45’ showing that I was mistaken. Kathleen said: “Now we have to get it this way.”, indicating right by moving her body in that direction. Jessie asked “What’s the angle? Shall I get a compass?” She meant a protractor and fetched one which had a moveable centre section. Holding this against the screen, she decided the angle was 90 degrees. The girls now executed the commands line-by-line from within the ‘green’ doit box until they obtained the sideways ‘T’ shape (Figure 8.1).

This episode shows one of the differences between interpretation and construction. Frequently, students are given the command and input as a pair, e.g., ‘rt 90’. Some of the difficulties they experience in separating process and product may stem from this practice. The fact that the Kathleen and Jessie needed to measure to determine that the angle needed was 90 degrees could be read as indicating their relative weakness mathematically, but it also shows that they had discriminated the inputs needed for ‘fd’ and ‘rt’.

### 8.2.1.2: Episode 2 – Building up procedures and Debugging

This episode has been chosen to show how the girls built up procedures and how they debugged their program.

The girls swapped positions and began to write a program to draw the number line. Jessie pointed to the doit box ‘green’ and said: “We can repeat that. We can do that again.” She created a new doit box which they called ‘mum’ and typed the line ‘repeat 5 green’ inside it. Upon running this, the girls discovered a bug (Figure 8.2). Jessie said: “Shall I do one?”

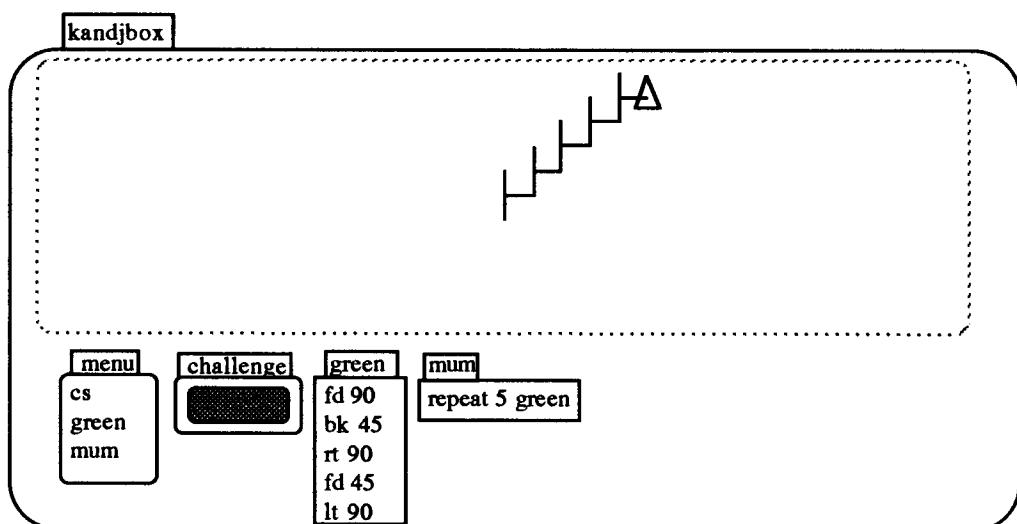


Figure 8.2: Jessie and Kathleen’s buggy procedure ‘green’

After Jessie had run ‘green’ repeatedly from the menu, Kathleen said: “I see what’s gone wrong”, but, before she had time to explain, Jessie said “We want to go back 90.” They re-wrote ‘green’ with ‘bk 90’ followed by ‘fd 45’ and so on, executed ‘mum’ and discovered they still had the same bug. Kathleen now said: “Do fd 45 and bk 90.” They ran ‘mum’ and were delighted to see that it worked. The session up to here had lasted approximately one hour.

This episode shows how the doit boxes enabled Kathleen and Jessie to *begin* their programming with procedures and to build up super-procedures from there. The bug in their sub-procedure ‘green’ was not apparent when it was completed. It drew the desired shape and the turtle faced in the right direction. It only became apparent when they ran the super-procedure ‘mum’. The fact that they had built up ‘mum’ seems to have enabled them to deconstruct it and hence helped them to debug it.

#### 8.2.1.3: Episode 3 – Relating Commands within and across Procedures

This episode has been chosen to show how the girls made finer discriminations as they related the effect of changing the input to one command to the need to change the input to another to compensate.

After a short break, I asked the girls how they would get the number line to start over on the left-hand-side of the screen. Much to my surprise, Jessie replied: “I know. I’ve done that before.” She then retrieved a print-out of work she had done almost six months previously. She copied out the line ‘pu setxy -300 -100 pd’ from inside a doit box called ‘start’. This did not get her quite where she wanted to be in the elongated graphics box.

The following transcripts capture the next thirty or so minutes in which the girls were placing the line in the desired starting position, adjusting the size of the graduation marks and the distance between these, and changing the input to ‘repeat’ to compensate for changes in the length of the line.

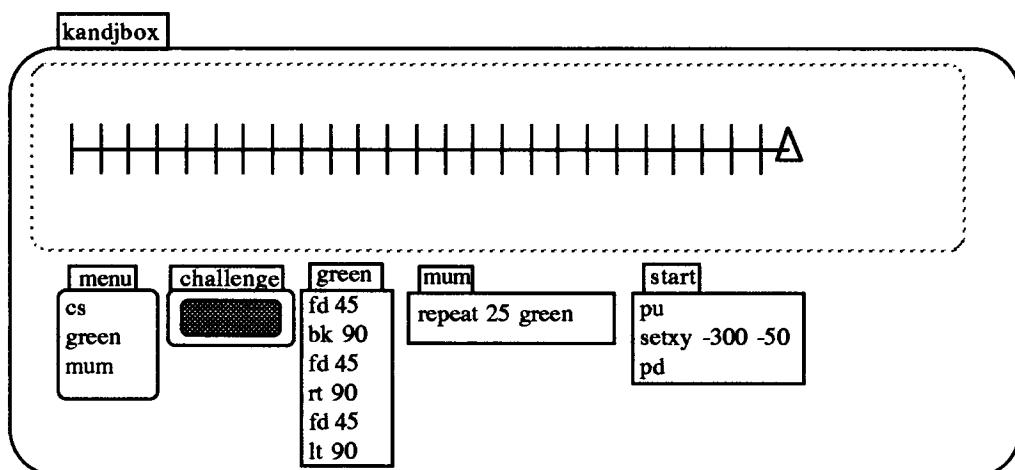


Figure 8.3: Stage 1 of Kathleen and Jessie’s Number Line

Extract 1:

Speaker	Dialogue	Commentary
Kathleen:	Shall we clear screen?	The girls have run ‘mum’ with an input of 15 to ‘repeat’.
Jessie:	If you don’t do clearscren, it will add 15 on the 15 already there and go way over.	Kathleen changes the input to ‘repeat’ from 15 to 18 and Jessie runs ‘mum’ again.
Kathleen:	Try 20.	
Jessie:	This is fun!	
Both:	Yes!	After the girls changed the input to ‘repeat’ to 22, and ran it again.

Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	Suppose you didn’t want these lines to be quite so long, would you know how to do this?	Points to vertical graduations.
Kathleen:	No.	
Jessie:	Change forward forty five and back ninety to forward forty and back eighty.	Jessie also changed the third fd 45 in ‘green’ and I questioned this.
Jessie:	We wants these bits (the horizontal components) to be the same as these bits (the vertical components).	Points to segments and graduations of number line in the graphics box.
Both:	The girls complete the changes to ‘green’, run ‘mum’ and observe that the no. line is not as long as before.	They change ‘repeat 22’ to ‘repeat 25’ to compensate.

Extract 3:

Speaker	Dialogue	Commentary
Both:	Smaller.	Kathleen changed 40 to 30 and 80 to 60.
Jessie:	Not right ninety.	She meant that ‘rt 90’ should not be changed.
Jessie:	We’ll need to change repeat.	She said this before the program was run. Subsequently, they changed ‘repeat 25’ to ‘repeat 32’.

The extracts in this third episode show how Jessie, in particular, had identified individual program lines with their effect on the screen, and moreover could predict how changes in one part of the program affected other parts.

### 8.2.2: Constructing the Number line 2

The following episodes feature Liam and Jason who were selected by the class teacher as representing the middle of the ability range. Unlike the episodes from Kathleen and Jessie, these are not continuous, but selected to illustrate different aspects of the programming involved in the number line construction.

#### 8.2.2.1: Episode 1 – Creating and Naming Procedures

This episode has been chosen to show how Liam and Jason created and named procedures in the course of constructing their number line.

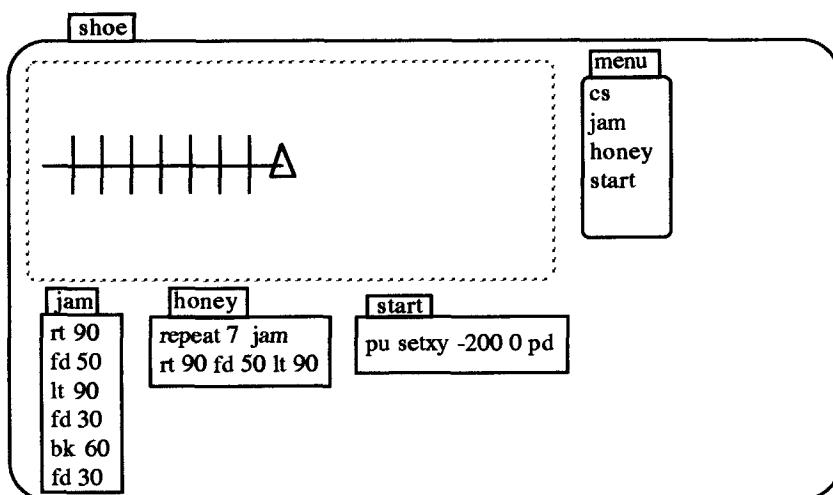


Figure 8.4: Stage 1 of Liam and Jason's Number Line

The boys began by creating a do-it box which they named ‘jam’. They executed the commands line-by-line to draw one segment of the number line, and also typed ‘jam’ in the menu. They then created a new do-it box named ‘honey’ in which they executed the line ‘repeat 7 jam’. ‘Honey’ too was added to the menu (Figure 8.4). The boys asked how they could start at the left-hand-side of the screen and I reminded them of how ‘setxy’ worked, but left them to experiment with its inputs. They created a ‘start’ do-it box in which they at first wrote ‘pu setxy 45 55’, but after a number of trials settled on ‘pu setxy -200 0 pd’.

The names which the students chose for the boxes themselves are worth remarking on. Early Logo writings (e.g., Papert, et al, 1979; Harvey, 1985) argued for the use of suggestive names for procedures and variables. Later writers (e.g., Hoyle and Sutherland, 1989) warned of the danger that students might think the names as having special significance. In this episode, the names chosen by Liam and Jason indicated

that they were aware of the arbitrary nature of names and had chosen names which were significance to them personally. It would appear that in Boxer the manner in which procedures are named (append a name box to a doit or data box) gives a directness to the naming process that seems to eliminate the danger of associating too much meaning to the name.

### 8.2.2.2: Episode 2 – Structured Programming

This episode has been selected to show how Liam and Jason's programming incorporated more structure than that reported for previous Logo research. It followed a period in which they had experimented with different inputs to 'repeat' to change the number of graduation marks on the line, and then adjusted the inputs to the commands in 'jam' and 'honey' to compensate for changes in the length of the line.

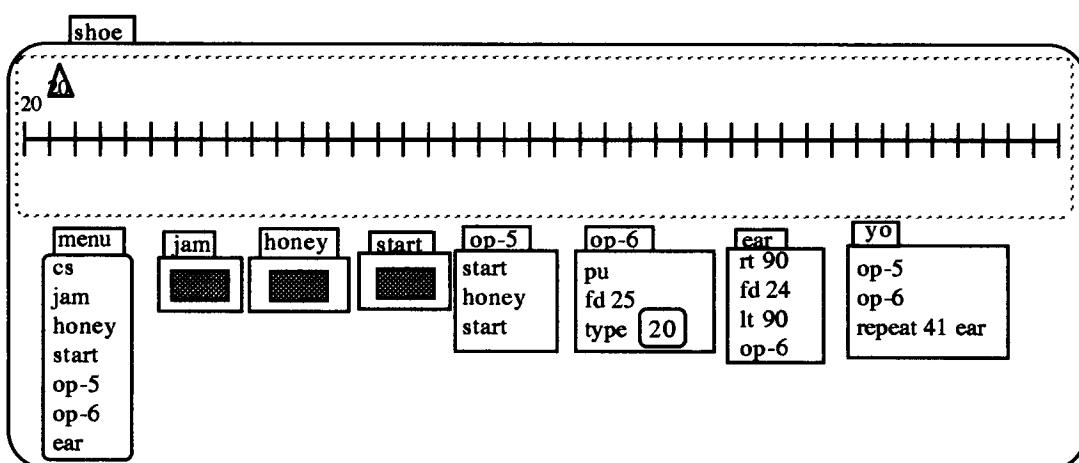


Figure 8.5: Liam and Jason's 'op-5', 'op-6' and 'ear' procedures

Liam suggested using 'start honey start' as a way of beginning in the appropriate position, drawing the number line and returning to the start position, and the boys created a new procedure 'op-5' to do this. In response to my challenge of typing a number above the left most graduation mark, the boys created a new procedure 'op-6', and when I challenged them to do this for every graduation mark, they first created 'ear' and then the super-procedure 'yo'. At this point they encountered a bug (Figure 8.5): the typed number continued to move up as well as along at each execution of the 'ear' procedure.

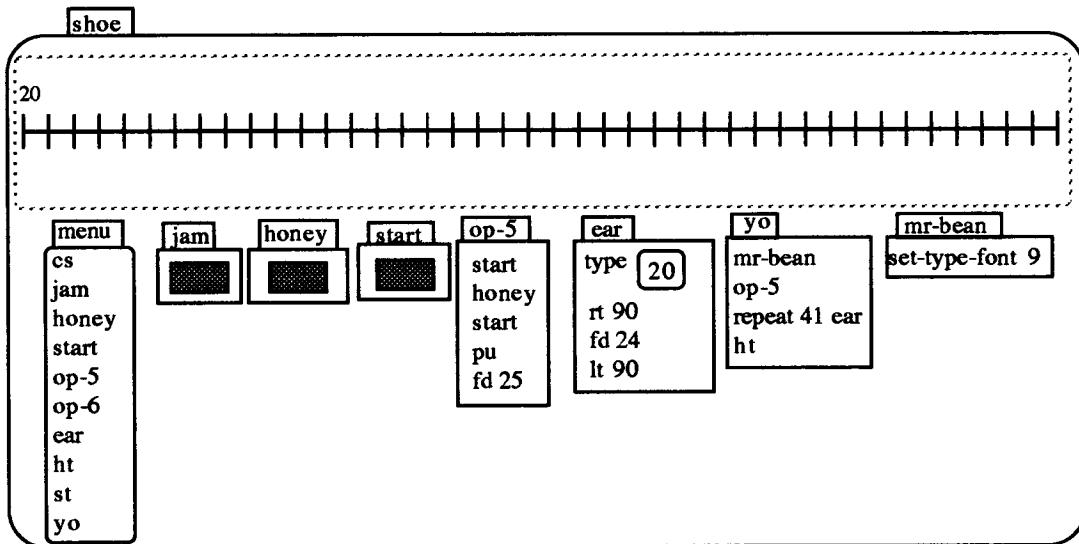


Figure 8.6: Stage 2 of Liam and Jason's Number Line

In response to my question: “What went wrong?” Jason immediately said “We need to take out the forward twenty five in op-6.” Liam now asked: “Can we get rid of some of these? Can we put ‘op-6’ in ‘op-5’?” He then instructed Jason to type ‘pu fd 25’ in ‘op-5’ after which then tested it, and saw that it worked. They moved the line ‘type 20’ from ‘op-6’ to ‘ear’, removed the line ‘op-6’ from ‘yo’ and then deleted the doit box ‘op-6’ completely. Finally, having complained that the fonts were “a bit boring”, they created a new procedure ‘mr-bean’ to change fonts and added the line ‘mr-bean’ to the super-procedure ‘yo’.

The degree of structure which Liam and Jason employed in their programs supports the contention that the long term programming aims embodied in the earlier phase of the research were coming to fruition. It seemed that with the doit boxes of Boxer, they found it as simple to write a procedure as to write a series of commands in direct drive. ‘set-type-font’, for example, could simply have been appended as a single line to ‘yo’, but the boys chose to create a special module ‘mr-bean’ for it. The evidence would also suggest that they were able to construct hierarchies of procedures in a flexible way.

The relationship between debugging and structured programming observed in this study was also different to that reported in comparable Logo research. In the number-line study, the students were inclined to create procedures line-by-line. In executing these lines, bugs in procedures usually became apparent. Students tended to follow a

similar practice in building up sub-procedures into super-procedures so that debugging was an on-going process.

The Brookline authors (Papert et al, 1979, p. 4.20) note that “Beginning students rarely separate the interface steps needed to go from one part of a design to another, as an independent sub-procedure, though it is a practice that can aid readability and debugging.” This was also the case in this study, but as this example with Liam and Jason shows, line-by-line execution allowed them to overcome the disadvantages of working in this way.

### **8.2.3: Linking Programming and Number**

The final part of the construction process was concerned with tabulating the number line with different numbers. This required the students’ program to be adapted to take a variable input and to change that variable under program control. This had been done in the programming phase using the ‘change’ command, and although I intervened to remind students how to do this, they appeared to have few conceptual difficulties with the programming itself.

The ensuing number investigations were centred on changing the appearance of the number line by manipulating the program’s parameters. From the students’ perspective, the focus was still on the Boxer number line, but now attention had been shifted from its construction to its operation.

## **8.3: Interacting with the Boxer Number Line**

The real problem which confronts mathematics teaching is not that of rigour, but the problem of the development of ‘meaning’, of the ‘existence’ of mathematical objects.  
(Thom, 1973, p. 202)

This section documents the way that the Boxer number line mediated students’ expression of number, and tries to identify facets of the representation and the interaction which influenced this expression. It identifies three different categories related to whether the students were operating within the natural numbers, extending the number line to include negative numbers, or dividing to obtain decimals.

### 8.3.1: Visualising Natural Numbers

The following episodes show how Jessie and Kathleen, and Courtney and Laura were able to exploit their Boxer number lines to express number relationships in a visual form.

#### 8.3.1.1: Episode 1– A new way of expressing familiar numbers

This episode has been selected to illustrate how Jessie and Kathleen utilised their Boxer number line to express large numbers and tables

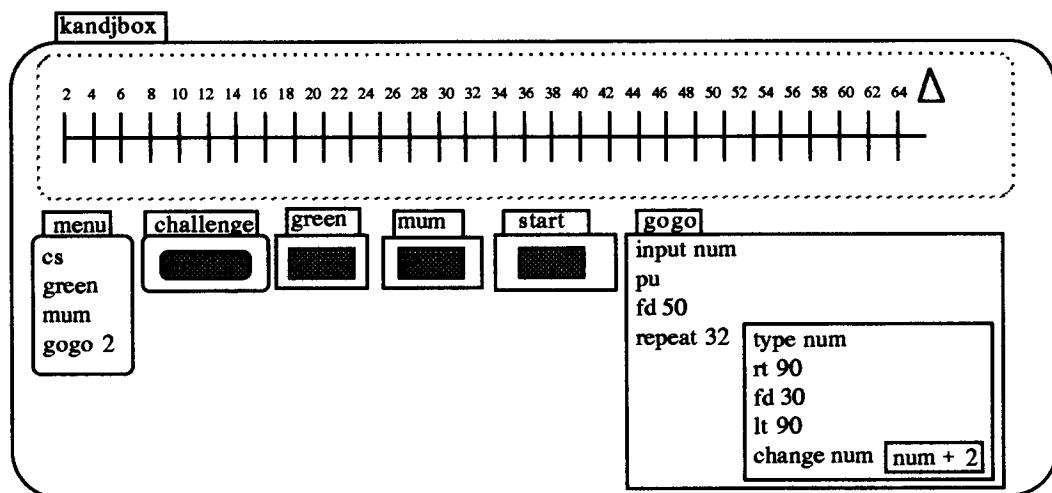


Figure 8.7: Expressing Patterns with Kathleen and Jessie's Number Line

#### Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Suppose now I wanted you to start at one and go up in twos?	Kathleen changes 'gogo 10' in the menu to 'gogo 1'.
Jessie:	Change that to two.	Points at 1 in the line 'change num [num + 1]' inside the doit box 'gogo'.
Jessie:	Do we want it on one?	Jessie questions the goal.
Kathleen	Yes.	
Jessie:	No 'cause we want it to go two, four, six, eight.	Kathleen changes 'gogo 1' in the menu to 'gogo 2'.

This extract shows that the girls were able to connect the components of the program with their effects on the screen sequence. This is not surprising, since they constructed it in the first place. Jessie's change of goal, from my starting value of 1 and increment of 2, to one which had also had a starting value of 2 is further evidence of their appreciation of the link between process and product. Rather than calibrating

the number-line 1, 3, 5, 7, etc., she chose to calibrate it 2, 4, 6, 8, etc., which she presumably found more aesthetically pleasing.

The next extract shows how they decided to use the number line to express what were for them large numbers, and presumably ones which they might have experienced difficulties in articulating otherwise.

**Extract 2:**

Speaker	Dialogue	Commentary
Interviewer:	Would you like to choose where to start and what to go up by?	
Jessie:	Start at a thousand and go up by hundreds.	Kathleen changes the program line inside 'gogo' to: 'change num [num + 100]' and the menu line to 'gogo 1000'.
Jessie:	That's brilliant!	After running it.

The number line now appeared across the screen graduated from 1000 to 4100 in steps of 100. Jessie's remark shows clearly that she appreciated the fact that the number line could be used to articulate large numbers.

Next, the girls decided to use the number line to express their 8 times table. Kathleen changed the program line inside 'gogo' to: 'change num [num + 8]' and the menu line to 'gogo 1'. Jessie disputed this latter decision by saying: "But, we have to change that to 0", meaning that her 8 times table should start at 0 rather than 1. On running 'gogo 0' from the menu, the girls got a new number line calibrated in steps of eight from 0 to 248.

This first episode shows how the girls were able to exploit the number line to visualise large numbers as well as their multiplication tables. While the emphasis had shifted from programming to number, it is clear that knowledge of the programming code – which they had acquired through constructing the number line – was the key to being able to express these numbers. A second example reinforces this claim.

#### 8.4.2.2: Episode 2 – Modelling ‘times’ tables with the number line

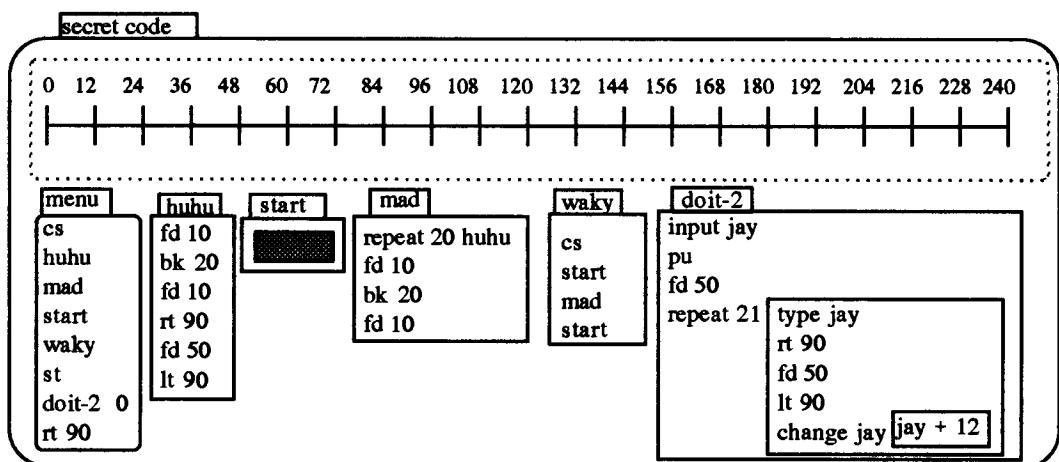
The following episode has been selected to show how Laura and Courtney's also used the number line program to model their ‘times’ tables, but also how the Boxer number line accentuated certain features of using the number line in this way.

**Extract 1:**

Speaker	Dialogue	Commentary
Laura:	Lets not start at nought. Lets start at one.	
Courtney:	Change jay to jay plus one.	Laura wrote the line 'change jay to [jay + 1]' and I intervened to point out that Boxer did not understand the word 'to'.
Interviewer:	Let's doit-2 one and see if it works.	Note how this sounds like "Do it to one".
Both:	Yeah. We've done it.	
Interviewer:	Try starting at a different number.	
Laura:	Fifty.	They run 'doit-2 50' and obtain a number line calibrated from 50 to 99 in steps of 1.
Laura:	It finishes at ninety nine.	

**Extract 2:**

Speaker	Dialogue	Commentary
Interviewer:	Suppose you didn't want to go up in ones. Suppose you wanted to go up in twos or fives.	
Laura:	You could change the 'jay plus one'.	Points to the line: 'change jay [jay + 1]' in 'doit-2'
Interviewer:	OK, try that.	
Courtney:	Here we go. doit-2 fifty.	The starting number is 50 and the increment is 5.
Interviewer:	What's it doing?	
Courtney:	It's going fifty, fifty-five, sixty, sixty-five, seventy, seventy-five, eighty etc. It's adding five to each number.	Note the expression: "It's going fifty..." They obtain a number line calibrated from 50 to 295 in steps of 5.



**Figure 8.8: Visualising the 12 time table with Courtney and Laura's Number Line**

**Extract 3:**

Speaker	Dialogue	Commentary
Interviewer:	Suppose you wanted it to do your three times table or your four times table ...	
Laura:	You'd go up like ... times five repeat.	'Times five' here is being related to '[jay + 5]' in the program line 'change jay [jay + 5]'
Interviewer:	What times table do you not like?	
Laura:	The twelve times table.	
Courtney:	Can we do twelve?	
Interviewer:	Sure. What about you Laura? Do you know how to do a twelve times table?	
Both:	They alter the program line to: 'change jay [jay + 12]' and run it.	Number line drawn in the centre of graphics box with typing unclear.

Responding to my challenges to adapt their number line to depict different sets of numbers involved more than simply changing the parameters to the 'doit-2- program. The students' typing program was independent of the line drawing program, so changes to the two needed to be co-ordinated. Changing from increments of 1 or 5 to increments of 12 led to numbers being typed too close together on the graphics screen. The girls compensated by reducing the number of graduations on the line. But, this meant the new line was squashed in the centre of the graphics screen. They then adjusted the width of each horizontal segment, and – for aesthetic reasons – the length of each vertical section.

Through construction and re-construction the students became aware that the number line had one more vertical than horizontal segment, a seemingly trivial point. But, it meant that in starting the number line at 0 to represent tables (as many students did), there was always one more number than one would expect in a times table. The number line was, in fact, providing a model of multiplication as repeated addition. It may, therefore have been modelling their intuitive notion of multiplication.

### 8.3.2: Visualising Directed Numbers

There is of course nothing new about using the number line to depict directed numbers. However, in many textbook approaches, it tends to be used as a means of modelling operations on the numbers (Küchemann, 1981). Students are introduced to operations on directed numbers in the form of  $-7 + 3$ , or  $-5 - -4$  etc. and the number line is then used to support these operations. The assumption underlying this

approach is that students already have a meaning for the numbers themselves. In the number line study, negative numbers were introduced simply as different starting positions, and students were then asked to predict how the number line would be graduated as a result of repeatedly adding a fixed positive whole number. Few of them had difficulties with this.

### 8.3.2.1: Seeing Directed numbers as a process

This episode has been chosen to show how Jessie and Kathleen expressed directed numbers through their Boxer number line.

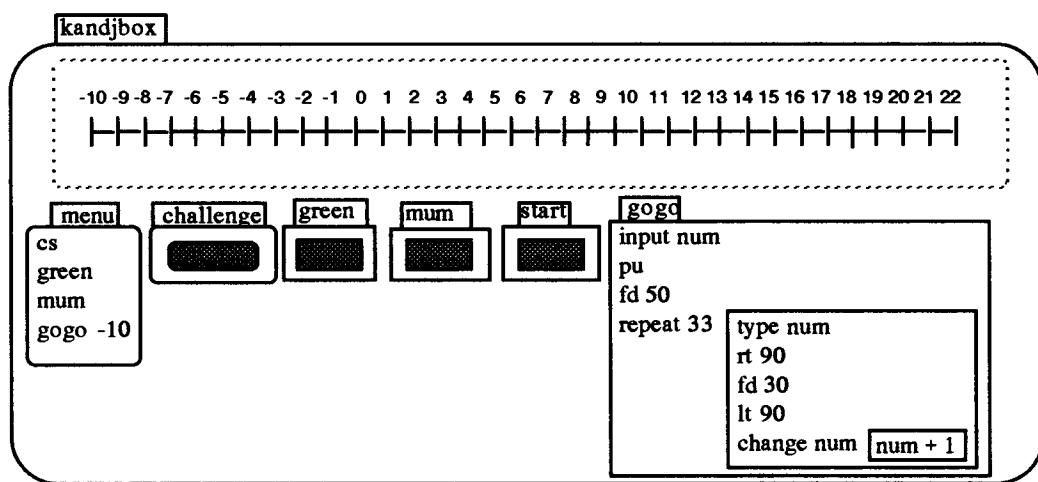


Figure 8.9: Visualising Directed Numbers with Kathleen and Jessie's Number Line

Extensions to negative numbers was often provoked by interventions which built on the students' successes with natural numbers. With the screen number line displaying the numbers 0 to 32 in steps of 1, I asked Jessie and Kathleen what would happen if they started at -10 instead.

#### Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	What do think will happen if we start at minus ten and go up in ones?	Figure 8.9.
Jessie:	Don't know.	
Interviewer:	No ideas? [no response] Okay, just put minus ten.	girls run 'mum' and 'gogo -10'
Interviewer:	Minus ten, minus nine, minus eight, minus seven, minus six ...	Reads from left to right on the number line.
Jessie:	There's a nought there as well.	Points to screen number line.
Interviewer:	There's a nought, one, two, and all the rest.	The number-line was calibrated in steps of 1 from -10 to 22 (Fig 8.9).

Jessie:	If we do minus half of thirty two. That's sixteen. If we do minus sixteen, then it will be sixteen each end.	Points to both ends of the number line in the graphics box.
Interviewer:	Try it.	

This last interjection by Jessie points to the way the Boxer number line was mediating her expression of directed numbers. Her remark “If we do minus half of thirty two.” suggested that the number line could be made symmetrical by beginning at -16. But, what is particularly striking is the manner in which she calculates this -16 from the previous end-point of 32. “Minus half of thirty two.” is in effect a composite function  $fg: x \rightarrow -(x/2)$  where  $g: x \rightarrow x/2$  and  $f: x \rightarrow -x$ , as her own solution demonstrates. It is hard to imagine a child spontaneously wishing to calculate ‘minus half of thirty two’ in a pen-and-paper medium, even with the aid of a number line. Here, the combination of the visual and the dynamic motivated such expressions.

She went on to change the line ‘gogo -10’ in the menu to ‘gogo -16’, ran the program, and obtained her symmetrically labelled number line from -16 to 16. That she was developing a sense of how these numbers behaved as a consequence of addition can be seen in the following examples.

#### Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	What do think would happen if you went up by plus two?	
Jessie:	Perhaps it would go sixteen, fourteen, twelve, ten, eight, and then up.	Refers to -16, -14, -12, -10, -8, etc.
Interviewer:	Try it.	

#### Extract 3:

Speaker	Dialogue	Commentary
Interviewer:	Pick a number.	
Both:	Minus two.	After discussion.
Interviewer:	What do think would happen if you start at minus two and go up by two each time?	
Kathleen:	Minus four, and then minus six.	
Jessie:	Nought, then two and then four.	
Interviewer:	Try it and see what happens.	
Interviewer:	So the next number was nought, not minus four. You were thinking of going the other way.	To Kathleen.

Although Kathleen was still encountering difficulties, the evidence suggests that Jessie was developing a sense of how directed numbers behaved as a consequence of interacting with the Boxer number line. The formalisations which she was able to deal with in the Boxer medium could be expressed algebraically as  $-16 + 2n$ , and  $-2 + 2n$  respectively.

### 8.3.3: Visualising Decimals

The following episodes show how students were able to express decimals through their Boxer number line. The operational approach which the number line embodied helped to link their existing knowledge of fractions with that of decimals.

#### 8.3.3.1: From Program code to Number expression

This episode has been selected to show how Laura and Courtney's number expressions were framed round their interaction with the number line, and how Laura's confusion of decimal and fractional notation was related to the program code.

##### Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Suppose you wanted to end at thirty?	The number line was calibrated from 0 to 60 in 20 steps of 3.
Laura:	Jay plus one.	Points to line: 'change jay [jay + 3]'.
Courtney:	I wanted to go up in twos.	Tries 'jay + 2' and ends at 40.
Laura:	Plus one.	Tries + 1 instead and ends at 20.
Laura:	And a half! You've got to have one and a half.	Yells out excitedly.
Interviewer:	How would you put one and a half in Boxer?	
Courtney:	One point five.	
Interviewer:	Try it.	

##### Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	Keeping the same number of repeats, what I'd like is for you to end up round about ten.	The number line was graduated in steps of 1.5 from 0 to 30.
Laura:	Half! Just point five.	
Interviewer:	Try it.	They do this and the number line appears as in figure 8.10.
Courtney:	Yes. We've got it right!	
Interviewer:	You were quick. How did you work it out?	
Laura:	Well if one is twenty, then half of that has to be ten.	Refers to earlier number line.

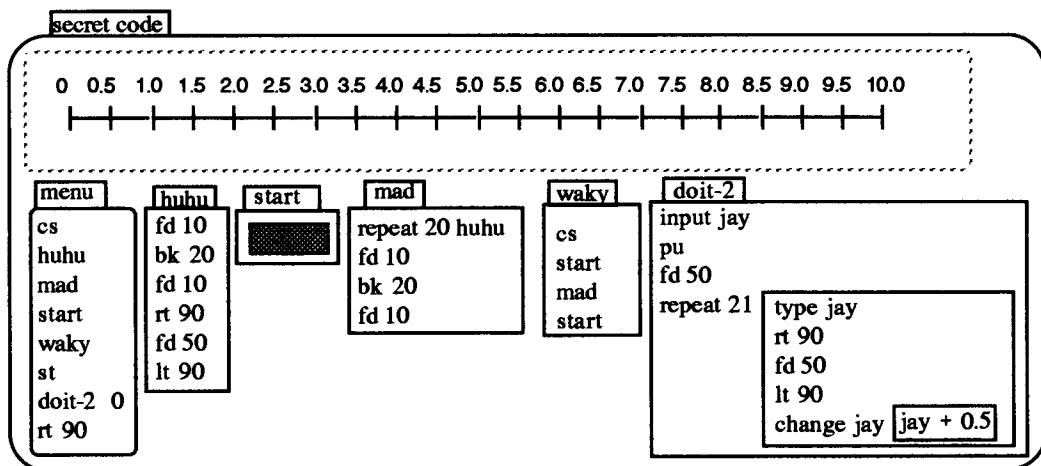


Figure 8.10: Visualising Decimals with Courtney and Laura's Number Line

Laura's expression "If one is twenty, then half of that's got to be ten." would appear to amount to reasoning that since  $20 \times 1 = 20$  then the solution of  $20x = 10$  must be 0.5. Its significance lay in the way it was derived from the earlier visual representation and the way it opened up the process for investigating other decimal expressions.

I now asked the girls if they could get the number line to end at 1. Courtney's first offering was: "Point one." which the girls inserted in the program code and which resulted in a number line ending at 2.0. Laura now said: "I think it's point twenty oneth." After running the program with 0.21, she read off the end value of the number line and remarked: "I can't see how it got to four point two."

#### Extract 3:

Speaker	Dialogue	Commentary
Interviewer:	I'd like you to start at nought and finish about one.	This would necessitate an increment of 0.05
Laura:	One eight. Try one twenty oneth.	
Courtney:	Point one.	Runs this and ends at 2.0.
Laura:	I think it's point twenty oneth.	
Laura:	Try point twenty one.	Runs this and ends at 4.2.
Laura:	I can't see how it got to four point two.	

At this point, I intervened to recall what had been tried so far. They could see that with an increment of 0.5, the number line ended at 10. With an increment of 0.1, the number line ended at 2. Therefore, there must exist a quantity which enabled it to stop at 1. 'Zero point zero' was a first attempt to articulate such a quantity.

**Extract 4:**

Laura:	Try zero point zero.	They try this and I point out that 0.0 is the same as 0.
Interviewer:	Recalls that 0.1 was a bit too big and 0 was too small.	
Laura:	Nought point five [waves hand about]. It's like half of ....	Laura trying to articulate the number that lies at the midpoint of 0 and 0.1.
Interviewer:	You try and say it.	
Laura:	Nought point nought five.	
Interviewer:	Laura is saying 'Nought point nought five'.	
Courtney:	I've got a feeling she's right.	Runs program.
Interviewer:	What made you choose that number, Laura?	
Laura:	Em ... [No answer.]	
Interviewer:	Reviews situation.	
Laura:	I chose one twenty oneth.	
Interviewer:	I didn't understand you. I wondered why you had chosen point two one.	

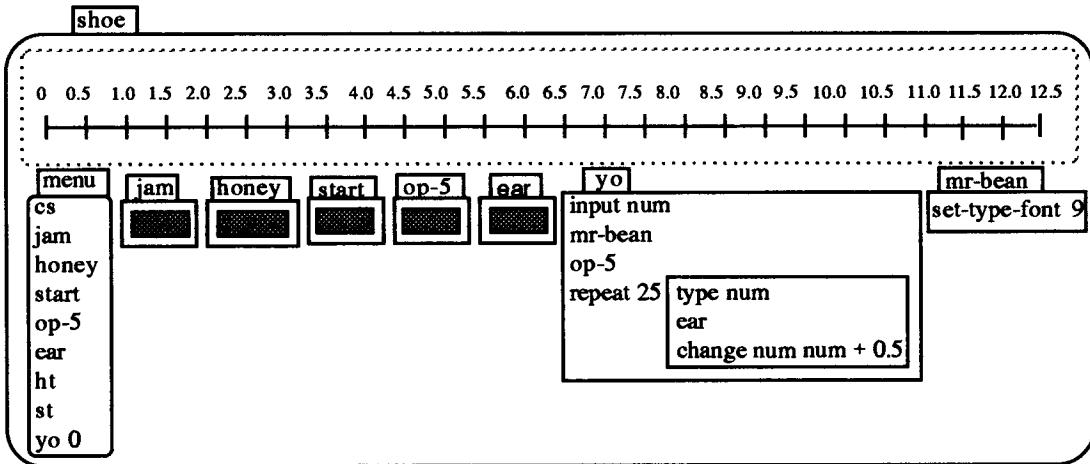
At the time of the earlier exchange, I thought that Laura's 'point twenty oneth' was simply a wild guess. However, later when she had discovered a way of articulating quantities between 0 and 0.1, she continued to express puzzlement as to why her earlier offering did not work. In the ensuing discussion, it emerged that she was confusing decimal and fractional notation, i.e., her 'point twenty oneth' was intended to express  $1/21$  in decimal form. Her choice of 'point twenty one' was directly related to the Boxer program. Since the program line responsible for graduating the number line read 'repeat 21' she concluded that the increment needed was the (multiplicative) inverse of 21, which she articulated as 'point twenty oneth'.

### 8.3.3.2: Inventing Notation for Decimals

The following episodes constitute further evidence for the compelling nature of this Boxer object in convincing the case study students that other numbers must exist. In each case, the students invented their own notation to express these numbers.

#### Episode 1: From fractional to decimal expressions

This episode has been chosen to show how Liam and Jason used a notation which they were familiar with (fractions) to express a quantity that they lacked familiarity with (a number between 0.1 and 0.2).



**Figure 8.11: Visualising the Number Continuum through Liam and Jason's Number Line**

At the beginning of this episode, the number line was graduated from 0 to 25 in steps of 1. The boys were challenged to change the increment so that the line ended at 'about 13'. The vagueness of the formulation 'about 13' was deliberate: my intention was that they employ a decimal not a *particular* decimal.

**Extract 1:**

Speaker	Dialogue	Commentary
Interviewer:	Suppose you started at one and you wanted to finish at about thirteen? What would you change?	The implication was that the number of graduations would remain the same.
Liam:	That. Point six or point four.	Points to the line 'change num num + 1' inside 'yo'.
Interviewer:	What about you Jason?	
Jason:	Point five.	The boys amend the line to 'change num num + 0.5'.

The number line was now graduated in steps of 0.5 up to 12.5 (Figure 8.11). Hart (1981, p. 216) writes that a half: "seems to be an honorary whole number" by which she implies students are familiar with it unlike other non-integer quantities. So, it might be thought that Liam and Jason's solution stems from this familiarity. However, the next extract belies this explanation.

**Extract 2:**

Speaker	Dialogue	Commentary
Interviewer:	Suppose you started at nought and you wanted to finish at about seven.	The assumption is that the input to 'repeat' will be unchanged.
Jason:	Point three.	The boys amend the program to 'change num num + 0.3'.

The boys obtained a new number line on the screen graduated from 0 to 7.5 in steps of 0.3. So far, the number line simply let them express what they already knew, but the activity of doing so was important in setting the scene for extending that knowledge.

**Extract 3:**

Speaker	Dialogue	Commentary
Interviewer:	Suppose you started at nought and you wanted to finish at about three point five.	
Liam:	Where's that?	Points to the on-screen number line.
Interviewer:	Roughly in the middle there.	Points to the centre of the on-screen number line.
Liam:	Try point one.	They run this.
Interviewer:	It looks like point one is a little on the small side.	After getting a number line graduated from 0 to 2.5 in steps of 0.1.
Jason:	Point one and a half.	
Interviewer:	How would you write down what Jason just said: point one and a half?	Hands Liam a piece of paper. Liam writes: $.1\frac{1}{2}$ .
Interviewer:	Would you mind changing that point one to point two?	Points to increment. The boys change this and run the program.
Interviewer:	Point one wasn't quite right and point two wasn't quite right.	Refers to new screen number line ending at 5.0
Jason:	In between point one and point two.	Refers to the value of the increment in the program line.
Liam:	Point one and a quarter.	
Interviewer:	Explains convention of point one and a half as point one five.	
Liam:	Can we try it?	They run it with 0.15 and then with 0.14 which gives the exact answer 3.5.

The activity was framed around the students' number line. It was therefore a relatively natural step from using it to depict natural numbers to representing decimals. The initial challenge simply involved bisection, and so the students' choice of 0.5 was not a difficult choice for them.

The later step was not so simple. It obliged them to seek a number between 0.1 and 0.2 – a quantity which they had difficulty articulating. Yet, they were convinced that such a quantity existed. The Boxer number line gave them visual proof that an increment of 0.1 in the program resulted in a number line that was graduated in steps

of 0.1 and ended at 2.5, and an increment of 0.2 in the program resulted in a number line that was graduated in steps of 0.2 and ended at 5.0, so there must be something in between. Liam's 'point one and a half' and later 'point one and a quarter' shows how the medium disposed him to these expressions.

At about this time, I explained that in order for Boxer to handle their expression, the number would need to be written as 0.15. The boys ran the program with an increment of 0.15 and watched as the screen number line was drawn and graduated from 0 up to 3.75. Without further intervention, they changed 0.15 to 0.14, ran the program and now got a number line that ended exactly at 3.5.

It is perhaps worth recalling the rather inexact formulation of the original question. The challenge was to finish at *about* three point five. Such an expression was necessary since challenges were often set in response to on-going activities and I did not have time to check the number of decimal places needed for an exact solution. So, the students were using the medium to get better (more accurate) results than I had anticipated.

#### Episode 2 – A historical notation for decimals:

This episode has been chosen to show how working with the Boxer number line also disposed Jessie and Kathleen to invent a notation to express decimals, although in their case, the notation did not owe its origin to fractions.

#### Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Suppose, you didn't want to go all the way to thirty two, like ending at sixteen?	The number line was calibrated from 0 to 32 in steps of 1.
Jessie:	Up in twos.	
Interviewer:	You did twos a minute ago and ended at sixty four. What about you Kathleen? Any ideas?	
Kathleen:	Minus two.	
Interviewer:	Minus two would take it the other way. Do you know any number smaller than one?	
Kathleen:	Minus one.	Correct answer to the question as stated!
Interviewer:	Minus one would take it the other way. Do you know any number between nought and one?	
Jessie:	Half.	

Interviewer:	Half. Right. Well, in Boxer, you have to write it as a decimal. So where it says num plus one, you put num plus nought point five.	Strictly speaking, this is not true. Boxer can handle fractional quantities. 'It' refers to the program line.
Kathleen:	[Does this]	
Interviewer:	Where did you end up?	
Jessie:	Sixteen.	

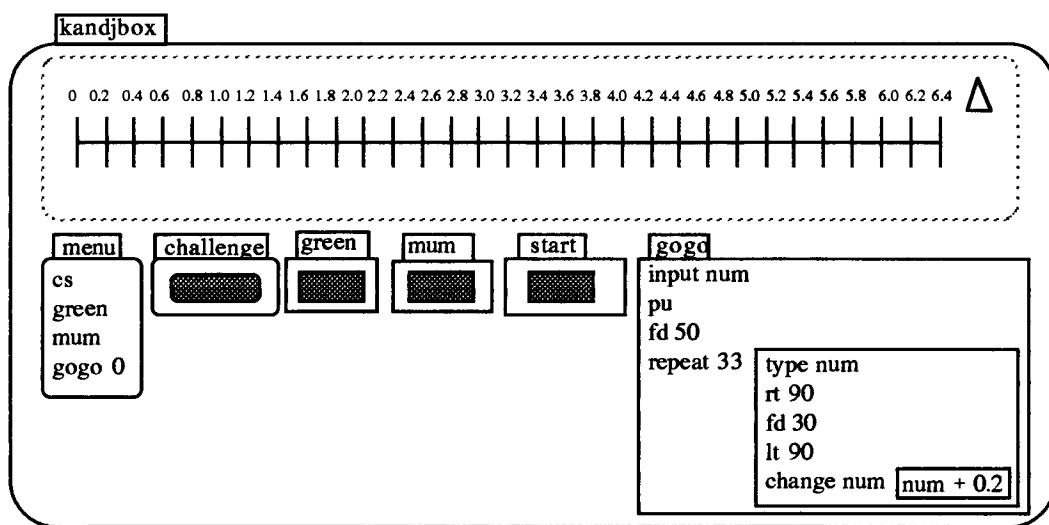


Figure 8.12: Visualising the Number Continuum through Kathleen and Jessie's Number Line

Kathleen and Jessie had drawn a number line which was calibrated from 0 to 16 in steps of 0.5. They were now asked if they could arrange for the number line to end at 8. Kathleen said: "I think its something under nought point five. Nought point two?" They changed the program code from '[num + 0.5]' to '[num + 0.2]', ran the program and obtained a line which ended at 6.4 (Figure 8.12). Kathleen now proffered: "Nought point three?" This was again inserted in the program and run, resulting in a new screen number line with an end point of 9.6. Jessie now interjected: "Nought point two point two." In written form, Jessie's answer is equivalent to 0.2.2.

#### Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	Suppose instead of finishing at sixteen I asked you to finish at about eight, any ideas.	The number line was graduated from 0 to 16 in steps of 0.5.
Both:	No response.	
Interviewer:	Do you know any decimals other than nought point five?	
Kathleen:	I think its something under nought point five. Nought point two?	

Interviewer:	Try it.	The girls alter the program code to 'change num [num + 0.2]'.
Kathleen:	Nought point three?	After running it with 0.2.
Interviewer:	Try it.	Again, the girls change the program code and run it.
Jessie:	Nought point two point two.	I showed her the conventional notation.
Interviewer:	Try it.	They run the program with 0.22.
Jessie:	Try point two three. It can't be anything if it isn't three.	
Kathleen:	Try nought point two four, Jessie. If it doesn't work, we can try something else.	Kathleen points to end of the line and reads the number
Jessie:	It's got to be five.	They run it.
Interviewer:	Have you ever come across nought point two five before?	
Both:	No.	

This non-standard notation, which Jessie had ‘invented’, actually has a precedent in the history of the development of the decimal system (Smith, 1925). But what is important about this incident is the way the medium pre-disposed the students to such an expression. Faced with a number line that ended in 6.4 for an input of 0.2, but giving an end-point of 9.6 for an input of 0.2, the students sought a number between 0.2 and 0.3. But, such numbers were not within their previous experience, with the consequence that they invented a notation to articulate their concept.

### 8.3.3.3: From Programming to Conceptual Precision

Working with the Boxer number line prompted me to set challenges for these students at a level of difficulty that I would hesitate to set for much older students using conventional media. Even when they did not resolve problems immediately, they persisted apparently convinced that the visual feedback would lead them eventually to the desired result. In persisting, they began to express decimals to greater and greater degrees of accuracy.

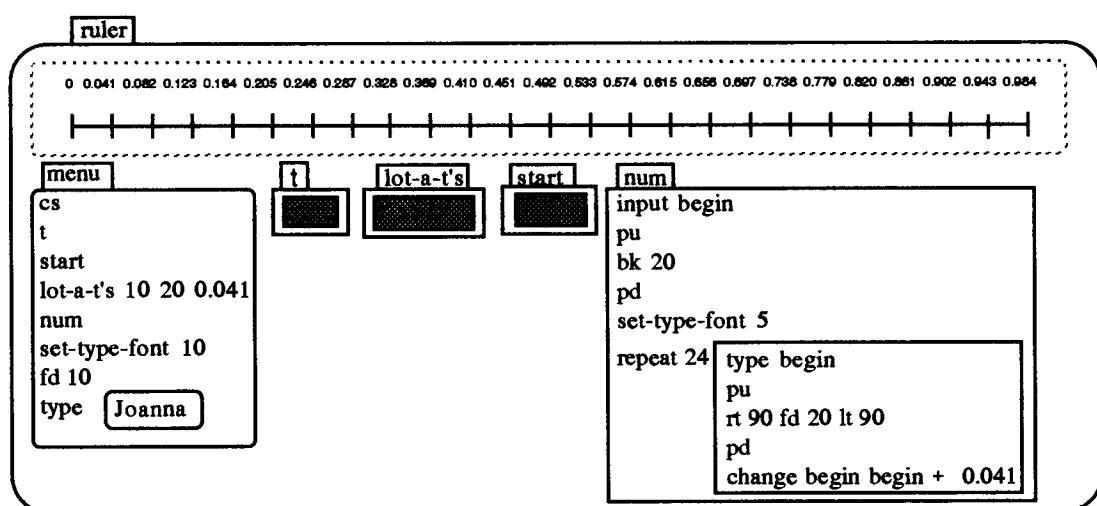
#### Episode 1: Accuracy of Expression

This episode has been selected to show how Joanna and Nico’s Boxer number line motivated them to persist in seeking a solution to a challenge (equivalent to solving the equation  $24x = 1$ ) to the extent of going to 4 decimal places. It suggests that the accuracy of the expression in this case serves as a measure of their understanding.

At the beginning of the following episode, the number-line was graduated in steps of 1 from 0 to 24, and the students were challenged to finish at twelve. Joanna instantly responded with: “Point five.”

**Extract 1:**

Speaker	Dialogue	Commentary
Interviewer:	I'd like you to finish at about twelve.	
Joanna:	Point five.	
Joanna:	Yes!	After running it.



**Figure 8.13: Expressing small decimal quantities with Joanna and Nico’s Number Line**

I now challenged the students to finish at about two point five. Joanna said: “Zero point one.”, to which Nico agreed. They ran the program and got a number line which ended at 0.24. While solving  $24x = 12$  and  $24x = 2.4$  within this computer setting might not seem too daunting, the next problem (equivalent to  $24x = 1$ ) was expected to be. The pair were asked to finish at about one.

**Extract 2:**

Speaker	Dialogue	Commentary
Interviewer:	Finish at about one.	Note the approximate nature of this and later questions.
Joanna:	Try point zero four. Maybe point nine six.	Before running program.
Joanna:	One!	The program had rounded up. Boxer’s printing-precision needed to be adjusted.
Joanna:	Ah. Zero point nine six.	After the adjusted program had been re-run.

Interviewer:	Could you adjust that to end up even closer to one.	Refers to the increment 0.04 in the line: 'change begin [begin + 0.04]'
Nico:	Changes increment from 0.04 to 0.041.	Figure 8.13
Both:	Point nine eight four.	Reads end number on new line. Both students pleased by answer.
Joanna:	Point oh four two.	The number line ends at 1.008. They then go on to try .0418, .0417, .0416, then .04168, and .04167.
Both:	One point oh oh oh oh eight.	At each stage, the students appear to be very pleased with the degree of accuracy that they are obtaining.

In this exchange, the assumption was that the only parameter that mattered was that for the increment. The start number was fixed at 0 and the number of repeats set at 24. Joanna's first choice of 0.04 was thus a very good first approximation to my initial challenge of ending at 1 (It resulted in the number line terminating at 0.96), while their second 0.041 was even better leading to 0.984. They now spontaneously ran the program with increments of .0418, .0417, .0416, then .04168. and finally .04167, suggesting that the precision afforded by manipulating the program's incremental parameter was also promoting conceptual precision.

#### Episode 2 – Relative sizes:

This episode has been selected to show how the visual feedback from their Boxer number line enabled Liam and Jason to eventually solve a challenge (equivalent to  $25x = 0.1$ ) by relating very small decimal quantities to other decimal numbers.

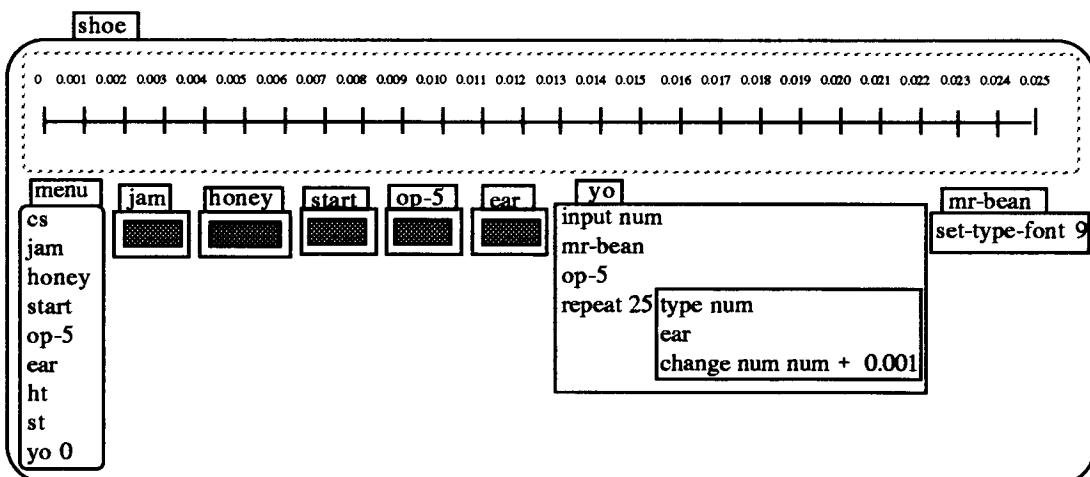


Figure 8.14: Expressing small decimal quantities with Liam and Jason's Number Line

**Extract 1:**

Speaker	Dialogue	Commentary
Interviewer:	I have a really hard question for you. I want you to end at nought point one over there.	The challenge is to begin at 0 and end at 0.1 with 25 increments.
Liam:	Try minus nought point one.	
Interviewer:	See what happens.	
Both:	They run the program.	
Interviewer:	It's going backwards.	
Liam:	Can we just delete that?	Refers to the minus sign.
Both:	They run the program with an increment of 0.1.	The number line is now graduated in steps of 0.1 and ends at 2.5.
Interviewer:	I want you to end on that.	Points to 0.1.
Jason:	Point minus one.	
Liam:	That would go down.	

This challenge caused difficulties for the boys and I intervened.

**Extract 2:**

Speaker	Dialogue	Commentary
Interviewer:	Your problem is to try to go up by very small amounts.	
Liam:	A hundredth.	
Interviewer:	Right. So how would you write a hundredth as a decimal?	
Liam:	Point oh one.	
Interviewer:	[To Jason] do you agree with that?	Nods his head.
Both:	They run the program and end up at 0.25.	
Interviewer:	Point two five is a bit too much. You have to make it smaller still.	
Jason:	Point oh ?	not clear
Liam:	You have to make it even smaller.	
Interviewer:	So how would you make it even smaller?	
Liam:	Point nought nought one.	
Interviewer:	Why don't you try that?	
Both:	They run the program and end up at 0.025.	Figure 8.14.
Liam:	Nought point nought two five. Change that to point nought nought four.	Reads end number and points to increment in program line.
Both:	They run the program and end up at 0.100	
Interviewer:	I know these numbers are new to you. I know they are hard. You gave me nought point nought nought one. Any idea what this number means?	
Liam:	A thousandth.	

These latter extracts show that the challenges were becoming difficult for the students. But these were difficult challenges. Yet, within the medium of this Boxer number line environment, they were able – albeit with some help – to resolve them. The manner in which the number line scaffolded their learning can be seen in Liam's deduction that since  $25 \times 0.001$  gave an end number of 0.025, what was needed was  $25 \times 0.004$  to obtain an end number of 0.1.

### Episode 3: Relating decimals to each other

This episode has been selected to show how in interacting with their Boxer number line, Kathleen and Jessie were able to impute meaning to very small decimal quantities by relating them to other decimal quantities.

#### Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Let's put nought point one there. Before you do it, can you tell me what will happen on the screen?	Points to the number line on the screen.
Jessie:	Nought point thirty two.	She had presumably interpreted $32 \times 0.1$ as 0.32.
Interviewer:	What about you Kathleen?	
Kathleen:	Same.	They run the program.
Interviewer:	In the end, where did it end up?	
Both:	Three point two.	

#### Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	Suppose I asked you to finish not at three point two up here but at one point six. Any idea how we would do that?	Points at end of the number-line drawn which is graduated in steps of 0.1 from 0 to 3.2.
Jessie:	Minus numbers?	This seems to be a relapse to a wild guess.
Interviewer:	No. Do you remember that we said minus numbers would take us the other way?	
Jessie:	I don't know any number smaller than nought point one.	This would appear to explain Jessie's earlier answer.
Interviewer:	Okay. What about you Kathleen?	
Kathleen:	Nought point nought?	A correct solution to the question as posed.
Interviewer:	Nought point nought, on its own, is the same as nought.	
Interviewer:	Suppose somebody said to you what is nought point one?	
Jessie:	A per cent? A hundredth?	

At this point, it would seem that the questions really were too difficult for the students. Jessie's "I don't know any number smaller than nought point one." and Kathleen's "Nought point nought." each express an impasse caused by not having a familiarity with numbers between 0 and 0.1. In the next extract, it can be seen how using such numbers provided the basis for their later discrimination of the behaviour of these quantities.

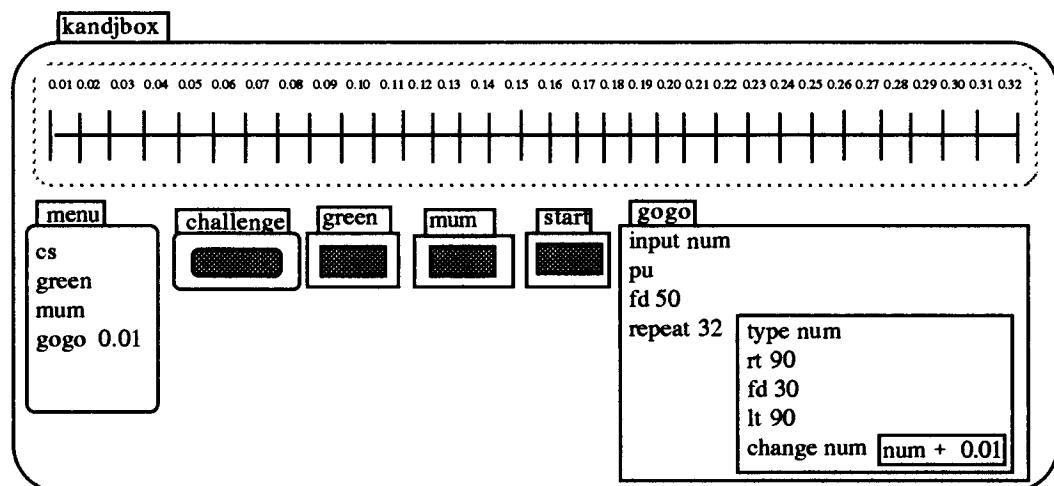


Figure 8.15: Expressing small decimal quantities with Kathleen and Jessie's Number Line

#### Extract 3:

Speaker	Dialogue	Commentary
Interviewer:	Change that to nought point nought one.	Refers to increment in the program.
Jessie:	That only comes to point three.	They re-do this and end at 0.32.
Jessie:	If you double that, you will double that	Asserts that doubling the increment will lead to twice the end no .
Interviewer:	Okay. Try it.	They do this and end at 0.64.
Jessie:	If you go up by nought point nought four, it will be even closer	Doubling strategy.
Interviewer:	Okay. Try it.	They do this and end at 1.28.
Interviewer:	Where did you finish?	
Jessie:	One point two eight.	
Interviewer:	One point two eight. You are getting very close to my original target of one point six.	
Jessie:	I think five would do it.	Means 0.05.
Interviewer:	Okay. Try it.	
Jessie:	One point six!	

While, I would not claim that Jessie and Kathleen had a robust understanding of the meaning of quantities such as 0.05, they were gaining a relative knowledge of certain decimal quantities, e.g., 0.04 is double 0.02, and so on. In operating within the medium, they were acquiring experiences which might serve as a basis for such understanding.

**Extract 4:**

Speaker	Dialogue	Commentary
Interviewer:	My challenge to you this time is to start at nought and finish close to one.	The number line was graduated from 0 to 1.6 in steps of 0.05.
Jessie:	Nought point six four	Reads from screen. She has used an increment of 0.02.
Interviewer:	That's quite good but it's not quite big enough.	
Interviewer:	What about you Kathleen? Any thoughts?	
Kathleen:	[no response]	Jessie changes the step size to 0.03.
Jessie:	Ninety six! [refers to 0.96]	Clearly recognises that this is close to one.
Interviewer:	Well done! I didn't expect you to get any closer than that.	

Jessie's unfamiliarity with decimals between 0 and 0.1 was not surprising. None of the comparison group in the semi-structured interviews had given a decimal answer at all. I doubt that the evidence would support the contention that she had a solid understanding of such quantities, but it does suggest, I believe, that within the medium of the Boxer number line, she was expressing these numbers in meaningful ways.

#### **8.4: Summary and Implications**

This chapter has analysed the Boxer number-line study. It has detailed how four pairs of case-study students first constructed and then interacted in their own number-line Boxer microworld. It shows how, in the course of constructing this number-line, they were able to apply and extend programming ideas developed in an earlier phase. It then indicates how the mathematical characteristics of this Boxer object mediated their expression of number.

### **8.4.1: Programming Findings**

#### Separating the process and product of a procedure

The evidence supports the contention that the students were able to separate the process and product of a procedure. Even students who were regarded as weak mathematically were able to discriminate the significance of the inputs to different commands. The programming activity, in which students were challenged to create number lines with varying degrees of precision, motivated them to attend to the interdependence of commands within and across procedures. The evidence suggests that they were able to do this.

#### Building up procedures and Debugging

All of the students in this study adopted a ‘bottom-up’ style of programming. In contrasting this way of working with the opposing ‘top-down’ mode of placing procedures in a hierarchy, the Brookline authors (Papert et al., 1979) observe that: “The latter ‘top-down’ approach usually allows better fitting of the hierarchy of the process to the logic of the product and hence makes procedures easier to plan, read, and debug.” (part II, p. 4.22). In this study, there did not appear to be such disadvantages in adopting a bottom-up style. It seems that in building up procedures, the students were also building up a picture of the process needed to debug them. The observations suggest that debugging was not a major obstacle to programming.

#### Creating and Naming Procedures

It would appear that in Boxer the manner in which procedures are named (append a name box to a doit or data box) gives a directness to the naming process that seems to eliminate the danger of associating too much meaning to the name.

#### Structured Programming

The degree to structure which students employed in their programs suggested that the earlier phase of the research influenced the way they programmed. It seems that with the doit boxes of Boxer, it is as simple to write a procedure as to write a series of commands in direct drive. The data points to students beginning programming with procedures. There is also evidence, which differs from comparable Logo research, that students were more able to create hierarchies of procedures.

### **8.4.2: Number Findings**

The Boxer number line object mediated students' expression of number in certain distinct ways which were directly related to its screen characteristics on the one hand and to nature of the numbers themselves on the other. Its screen manifestation linked the visual representation of the various number sequences with the program containing the formal Boxer description of that sequence. So, in manipulating the program's parameters, the students were making connections between the formal and the visual, and between conceptual and procedural notions of number. The form of this manipulation depended on the numbers. Natural number investigations were defined by setting a start number as a positive whole number and doing the same for the increment. The directed numbers investigations differed only in that the start number was no longer positive. For the decimal investigations, both the start number and the number of repeats were fixed, but my challenges motivated students to consider increments that were not whole numbers.

#### **Natural Numbers**

Initially, the students used the number line to depict sets of numbers which they could check. In this sense, the dynamic number line supported their existing knowledge and provided the scaffolding for extending it. In the room where the research took place, there was a number line poster on the wall graduated from 1 to 100. The students were quick to see the power of Boxer in producing number lines to obtain visual counterparts to their tables. Liam and Jason created a number-line which was in effect a 9 times table running from 0 to 360. Laura and Courtney tested the 5 times table beginning at 50 and then depicted the 12 times table. Jessie and Kathleen, the 'weakest' students, tried the 2 times, the 8 times and the 100 times tables beginning at 1000, describing the latter as 'brilliant'. Other students also used the number line to express multiplication tables. Their reactions to this practice were broadly similar. They appeared to think that in using the number line to depict their 'tables', they were somehow 'cheating', but this didn't diminish their pleasure in doing so.

#### **Directed numbers**

Manipulating the number line's parameters in order to display various natural number sequences, meant there was nothing new to learn in order to display other kinds of numbers. Directed numbers were introduced by simply changing the starting position of the number line, and asking the students to predict the consequences of repeatedly adding positive whole numbers to it. Few of the students found difficulties with this.

In generating a dynamic number pattern from left to right across the graphics screen, the number line object seems to have helped students to form a visual image of how the numbers themselves behaved under addition. Jessie's observation that the number line, which was calibrated from -10 to 22, could be made symmetrical by doing "minus half of thirty two." illustrates the influence of the visual on her thinking.

### Decimals

The Boxer number line appeared to mediate students' expression of decimals in three distinct ways.

(i) From program parameters to number expression: In the first case, it seemed that through manipulating its parameters, students were disposed to reason in particular ways. Decimals were conceptualised as small increments and their sizes related initially to whole numbers or to other (previously tested) decimal increments. In the case of Laura's confusion of decimals and fractional notation, the roots of her misconception could be attributed to her association of the number with the parameter for 'repeat'.

(ii) From program interaction to existence theorem: Secondly, rather than encountering difficulties with 'in-betweeness' as research has shown is frequently the case with conventional media (Hart, 1981, Markovits and Sowder, 1991), the Boxer number line seemed to pre-dispose the students to the belief that such numbers must exist. This manifested itself in several ways. It could be seen in remarks such as Jessie's: "I don't know any number smaller than nought point one." which suggested her difficulty was not with existence but rather with articulation. It could also be seen in the invention of notation by two of the four case study pairs to express these quantities. Liam and Jason's  $0.1\frac{1}{2}$  for a number between 0.1 and 0.2 was one example of this. Jessie and Kathleen's 0.1.1 was a second. Finally, it could be seen in the students' desire to attain greater degrees of accuracy than the problem required.

### (iii) From Programming to Conceptual Precision

Thirdly, the programming precision afforded by the Boxer number line appeared to be also promoting conceptual precision.

Some evidence for this could be seen in Joanna and Nico's successive trials of .0418, .0417, .0416, .04168 and .04167 in attempting to graduate the number line from 0 to 1 in 24 increments. The degree of accuracy at which these investigations took place

caused difficulties for several students. Nevertheless, the data shows that with the Boxer number line, students were able to attribute meaning to these very small quantities by virtue of relating them to each other. Jessie's recognition that 0.04 was double 0.02 was perhaps elementary, but it enabled this pair to 'close' in on a solution to their problem, which in formal terms, was equivalent to solving the equation  $32x = 1$ . Similarly, Liam's deduction was made from a number line graduated in 25 steps of 0.001 from 0 to 0.025. Recognising that four lots of 0.025 gave 0.1, he was then able to amend the increment from 0.001 to 0.004 to obtain an end point of 0.1.

In effect, these students were engaged in programming a 'zoom' effect, and in programming it, they appeared to be gaining access to the conceptual precision engendered by this process.

#### **8.4.3: Implications for the Remainder of the Thesis**

The number line study encapsulated the findings of the earlier stages of the research in three ways. Firstly, students constructed their own microworlds by programming in Boxer. Secondly the number line object itself proved to be the extensible and flexible representation that the semi-structured interviews had suggested. Finally, the operational nature of the pedagogical approach – also adopted as a result of these interviews – succeeded in provided a context in which students were able to confer meaning on both directed numbers and decimals.

The number-line object gave these students a means of moving back and forth between the notion of number as process and number as concept. Chapter 9 recounts how another such computational object, the function machine, did likewise.

# Chapter 9: The Function Machine Study

## 9.1: Introduction and Overview

### 9.1.1: Introduction

This chapter consists of an analysis of the programming and number issues which arose as a result of constructing and interacting with a function ‘number machine’ microworld in Boxer. It describes how in replicating a classroom project (making cardboard models of function machines) in Boxer, the students utilised and extended their programming knowledge. It shows how the insight gained through this construction influenced their subsequent interactions with it. It then analyses the ways this ‘operational’ Boxer object mediated their expression of number.

This phase of the research, which lasted from May 1993 to July 1993, was the final part of the case-study research. A total of eleven year 6 students (ages 10 – 11) were observed, six of whom were from the original eight case-study students (chapter 4). The remaining five acquired their Boxer knowledge through using the First to Third-Boxer microworlds independently (chapter 6).

At the beginning of the Summer term, the sixth year class had spent approximately two weeks constructing cardboard models of function machines. Figure 9.1 shows a typical example. The classroom teacher used this model to structure a follow-up activity in which the students had to ‘guess’ each other’s function.

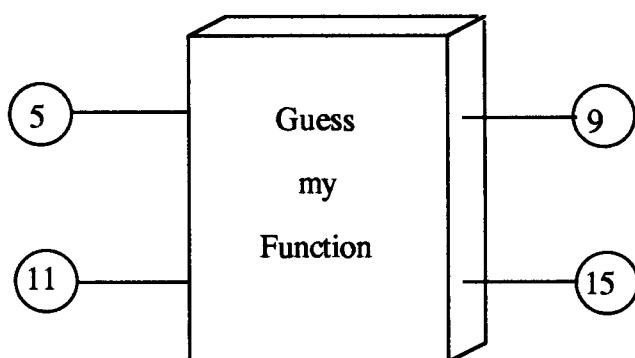


Figure 9.1: A typical cardboard model of a Function Machine

Some of the students then asked if they could ‘do’ their function machines in Boxer. It transpired that what they wanted to do was to use Boxer to draw a likeness of their

cardboard models on the screen. However, it quickly became apparent that, with a little more programming, it would be possible to make these models operational.

The interview findings (chapter 7) indicated that students were more likely to attribute meaning to number if representations could be found that conveyed an action-based derivation for those numbers. The number line study (chapter 8) was my first attempt to implement those findings. The students' request to build their function machines in Boxer now provided another opportunity to put those findings into practice.

The aims of this phase of the research were thus:

- to investigate the programming implications of students constructing their own function machine microworld in Boxer;
- to investigate how their interactions with this 'operational' object mediated their expression of number.

In following analysis, a distinction is made between the programming and number aspects of the activity. However, for the students, the construction of *their* function machine could not be separated from the subsequent activity.

### **9.1.2: Overview**

Section 9.2 consists of an analysis of the programming issues which arose through students' construction of their own function machine microworld in Boxer. It shows how students in re-creating their classroom number machines in Boxer programmed in ways that were significantly different to that reported in existing Logo research.

Section 9.3 recounts how the students' interactions with this Boxer object mediated their expression of number. It points to the importance of the construction process in structuring these interactions, and it attempts to link aspects of the students' manipulation of the programming code with these expressions.

## **9.2: Constructing the Boxer Function Machine**

### **9.2.1: Students' Programming in Boxer**

In chapter 5, three categories of programming, 'working at a syntactical level', 'making sense of', and 'goal-directed activity', derived from Hoyles and Sutherland's (1989) study with Logo, were utilised to analyse students' programming in Boxer. This chapter is only concerned with the last of these activities, and it will again draw

on that work to make finer categorical distinctions. Hoyle and Sutherland (*op cit.*) classified pupil goals along the following dimensions:

Loosely defined	Well defined
Real world	Abstract

They concluded that only when students were working towards a well-defined abstract goal do they begin to perceive a need to break down their program into sub-procedures or modules at the beginning of a project in a top-down way. They also found that students only took up the idea of sub-procedure when it was a useful problem solving tool for them. Subsidiary goals of this phase of the research was to assess the extent to which the new setting (Boxer instead of Logo) might change the situation regarding modularity, sub-procedures, debugging and the use of variables.

#### 9.2.1.1: Modularity

Research has shown (Hoyle and Sutherland, *op cit.*) that novice programmers frequently favour programming in ‘direct drive’. Experts point to the disadvantages of programming in such a way, particularly when things go wrong. As Papert (1993, p. 52) writes: “A strong argument for modularising programs is that they facilitate debugging.”

In most implementations of Logo, ‘direct drive’ as a way of working is in contrast to procedural programming. The advantage of the former is that by directly executing a command, the result can be seen on screen. The disadvantage is that there is no simple way of collecting together these commands to form a procedure. To define a procedure often necessitates moving into editor mode, writing the procedure, exiting the editor, and then executing it as a whole. There are thus discontinuities in two senses between these ways of writing programs: line execution and program execution cannot easily be combined, and the place of definition and that of execution are different. As Noss (1985) and Hoyle and Sutherland (1989) show, students frequently experience confusion concerning whether they were in defining or execution mode.

#### Episode 1 – Direct Drive, Direct Execution and Modularity:

This episode has been selected to illustrate how direct execution affected Laura and Courtney’s programming in the function machine study. The girls goal was to draw their ‘face’ number machine (Figure 9.2) in Boxer.

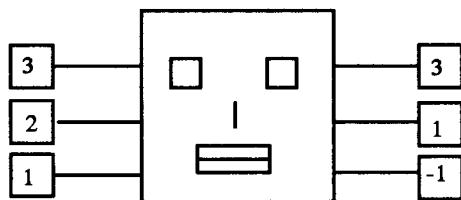


Figure 9.2: Laura and Courtney's cardboard face function model for  $2x - 1$

In beginning their drawing, the girls set themselves the goal of drawing the outline of the face in the centre of the graphics screen. They did not – as might have been expected – define a ‘square’ procedure. Instead, they created a module with the arbitrary name ‘cryu’ (Figure 9.3) into which they typed commands, executing them line-by-line.

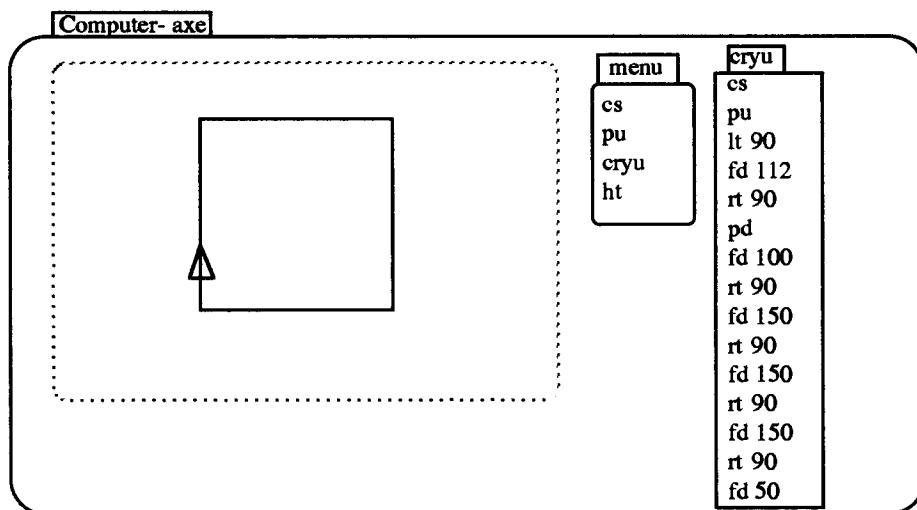


Figure 9.3: Laura and Courtney's ‘cryu’ procedure

The motivation for programming in this way seems to have been related to their desire to ‘centralise’ the drawing. They combined the navigation commands ‘pu lt 90 fd 112 rt 90 pd’ with the commands for drawing a square, while the square itself began and ended one-third of the way up the left-hand side.

As the drawing progressed, the girls alternated tasks in that one would create modules to draw the eyes, the other the nose and so on. ‘layrh’ (Figure 9.4), for example, is a module to draw the eyes but it also contains two ‘move’ modules: the first beginning quarter way up the left-hand side of the large square and the second moving from one eye to the next. Similarly ‘mat’ combines navigation commands (from the right eye) with the commands to draw the nose. Finally ‘lopter’ combines navigation

commands (from the nose) with the commands to draw the rectangular mouth. The module ‘ridd’ was added later to draw the central line of the mouth.

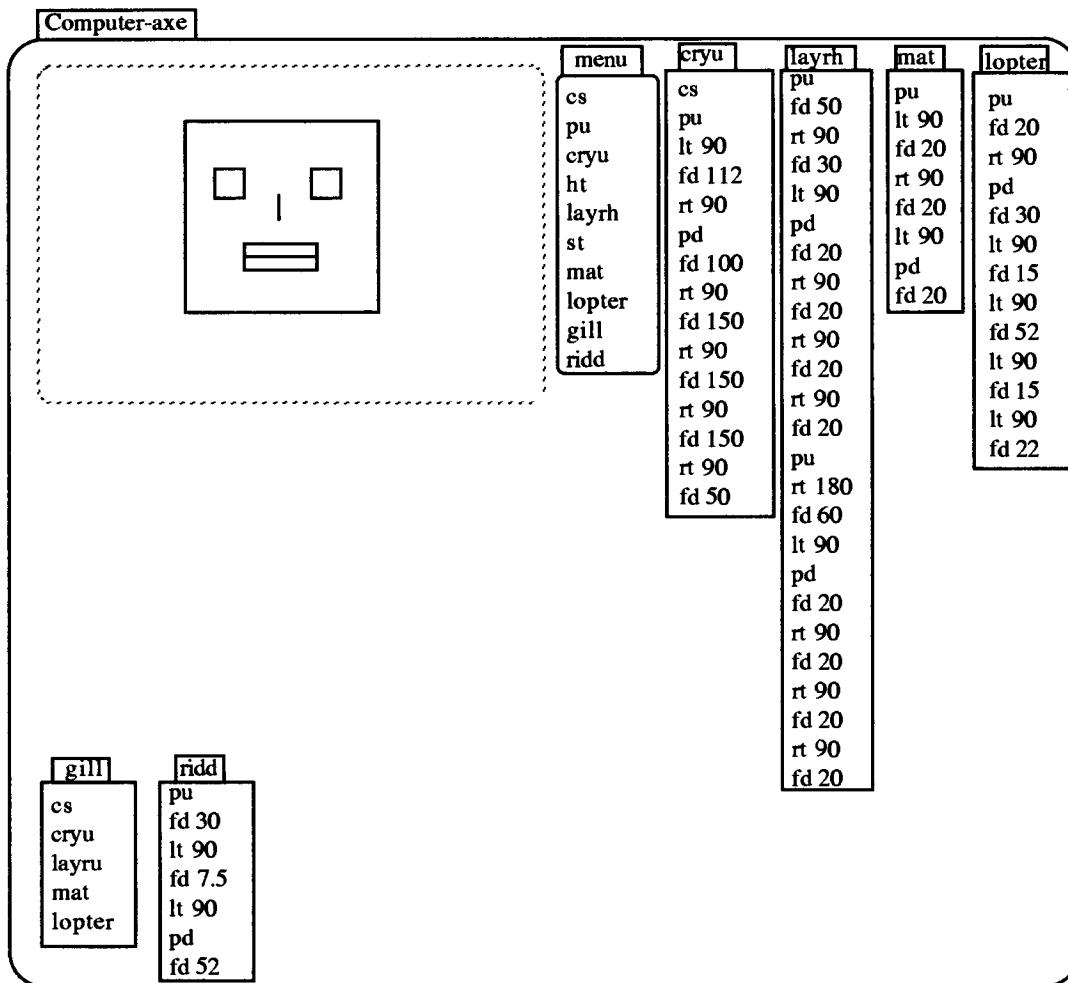


Figure 9.4: Laura and Courtney’s ‘semi-modular’ programming

Laura and Courtney’s programming illustrates the propensity to continue using collections of commands when they might have used modules: the use of direct commands in ‘cryu’ when they could have used a ‘square’ module and a ‘move’ module. Similarly, ‘layrh’ is a long list of commands which could have been made more elegant by including two ‘square’ procedures for the eyes. Against this, could be set the degree to which they did employ modules, e.g., ‘mat’ for the nose, ‘lopter’ and ‘ridd’ for the mouth. These were modular in that different programs were written for different parts of their drawing. However, none of these programs are strictly independent, since each also contains navigation commands.

It would seem that in Boxer, direct execution as a programming technique represents a half-way stage between direct drive and modular programming. It offers the instant feedback of direct drive, but it also permits executed commands to be combined into procedures. As Lorna and Courtney's programming shows, these may be semi- rather than fully independent modules.

#### Episode 2 – Direct execution, Debugging and Modularity:

The second episode illustrates how direct execution affected the relationship between modularity and debugging in Liam and Jason's construction of their function machine. Their goal was to draw a number machine resembling Figure 9.5.

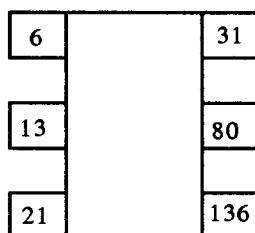


Figure 9.5: Liam and Jason's cardboard function model for  $7x - 11$

They created a module 'make' to position and draw the main body (rectangle) of their number machine. 'Make' was also typed in the menu.

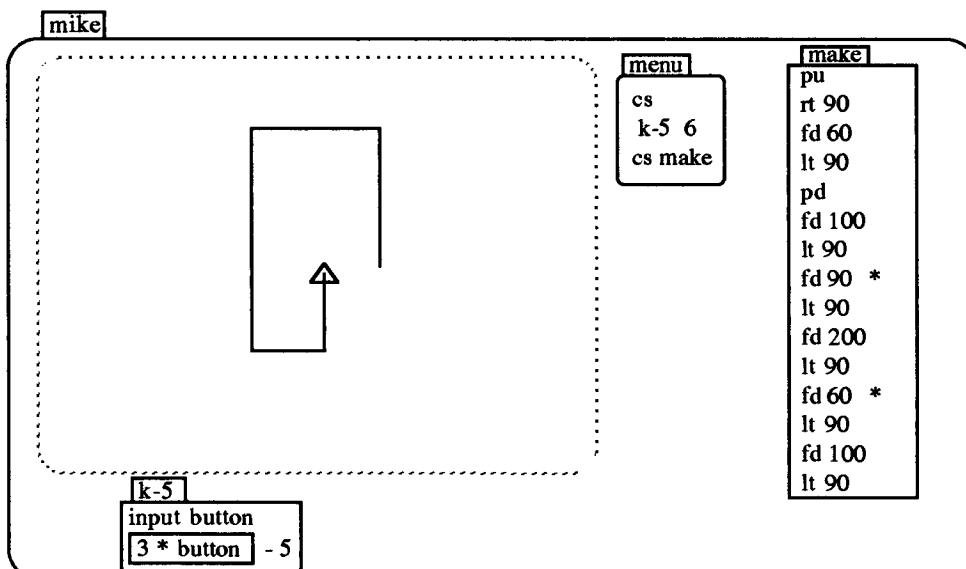


Figure 9.6: Liam and Jason's 'make' procedure and the first bug

From then on, they adopted the procedure of first typing new lines in the 'make' doit box, and then executing the line 'cs make' from the menu. However, because they

changed their minds about the desired size of the central rectangle several times in the course of drawing, they obtained a shape similar to that in Figure 9.6.

The boys went inside the ‘make’ module and began to execute the commands one-by-one. This enabled them to locate the bug in their program (marked with asterisks in figure 9.6) fairly quickly. They replaced each line with ‘fd 120’, and successfully drew the intended rectangle.

Liam and Jason then created a new module ‘live’ to draw the three right-hand output boxes. At this point, the turtle was at the middle of the right-hand side of the large rectangle facing inwards, so they included the commands ‘rt 270 fd 100’ in ‘live’ in order to begin at the bottom right-hand corner. Having done so, they executed the line ‘cs make live’ from the menu, and encountered their second bug (Figure 9.7).

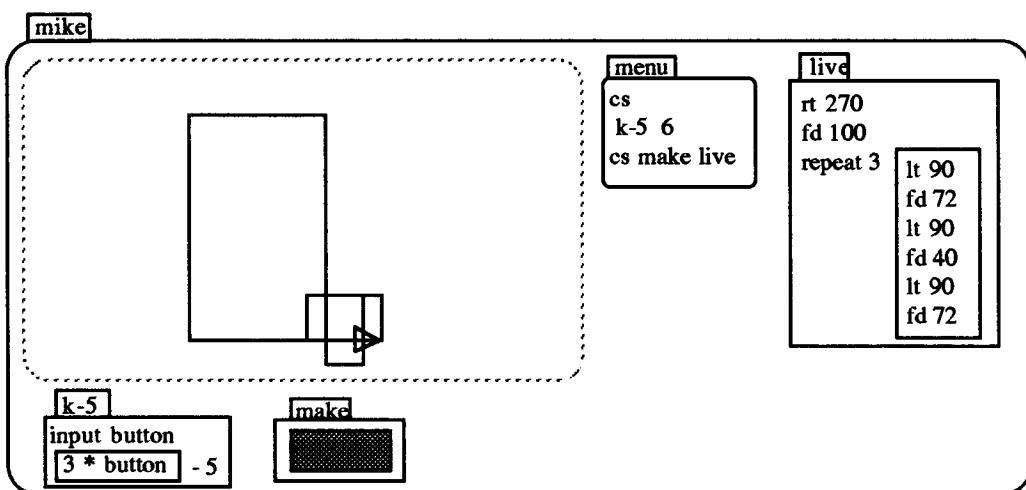


Figure 9.7: Liam and Jason’s ‘live’ procedure and the second bug

Liam intervened to say: “Do ‘make’ and then each one and see where we go wrong.” In doing this, i.e., in making each output box separately, the boys decided to move the top ‘lt 90’ out of ‘repeat’ and place it above it. They then added the extra lines ‘rt 90’, ‘fd 40’, ‘rt 90’. In effect, the repeated procedure draws three sides of the output box, and moves up ready to draw the next box. However, because they had repeated it three times, the boys got three boxes and three moves instead of two of the latter. They resolved this by changing ‘repeat 3’ to ‘repeat 2’ and then supplementing this with a final move (Figure 9.8).

A third major bug emerged as a result of the boys attempting to re-use their ‘live’ module to draw the left-hand boxes. The boys made a copy<sup>1</sup> of their ‘live’ module and re-named it ‘life’. In order to get ‘life’ to begin at the equivalent position at which ‘live’ started, i.e., at the centre of the left-hand side of the rectangle facing inwards, they went back inside ‘live’ and added the lines ‘fd 120’, ‘lt 90’, ‘fd 100’, ‘lt 90’. But, they failed to adjust ‘life’ for the left-hand side of the rectangle and so got another bug (Figure 9.8). Liam asked: “Shall we do it line-by-line?” As they did this, Liam pointed at one of the ‘lt 90’ lines in the ‘life’ doit box and said: “I know what it is. That has to be right. We didn’t change it.” They now went into ‘life’ and interchanged the commands ‘lt’ and ‘rt’, ending up with their desired shape.

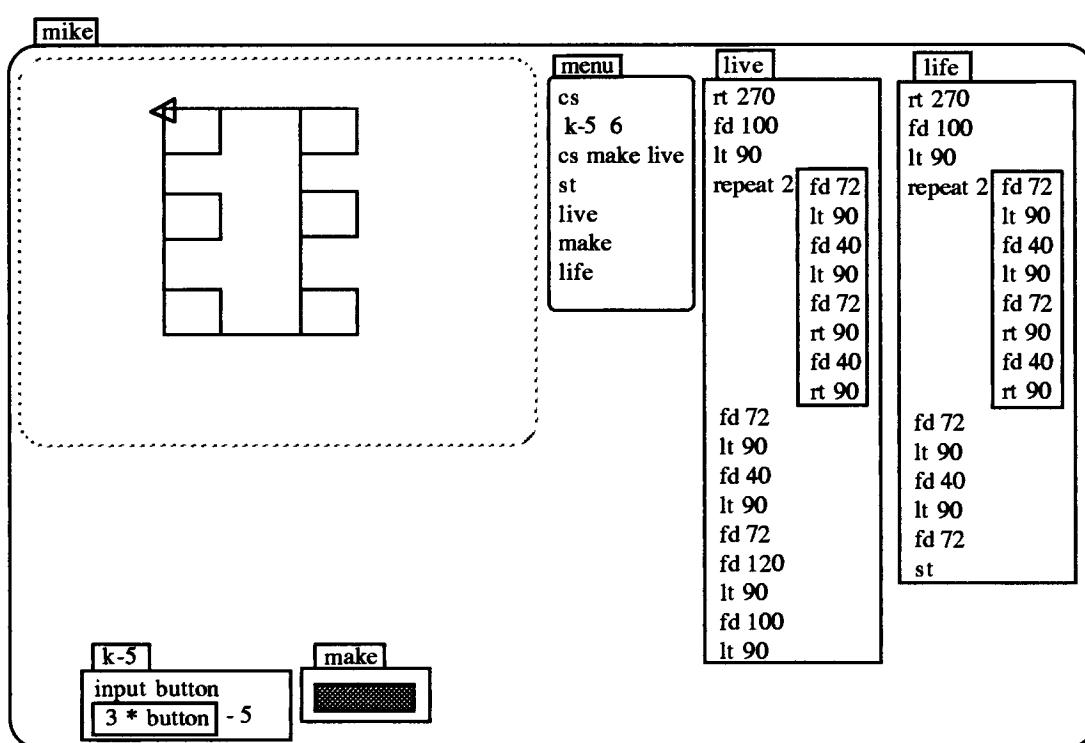


Figure 9.8: Liam and Jason’s third bug

It could be argued that the bugs which Liam and Jason encountered were self-inflicted, i.e., they were a consequence of their programming not being strictly modular. Yet, their attempt to re-use the module ‘live’, showed that they appreciated at least some of the merits of modularity. The difficulties stemmed from their inclusion of ‘extra’ navigation commands. The crucial point is that Boxer’s single mode and line-by-line execution made it more natural for them to locate and

<sup>1</sup>This can be done by ‘dragging’ over the box in Boxer.

remediate the bugs in the course of programming rather than afterwards and hence helped overcome some of the disadvantages of a semi-modular form of programming.

### Episode 3 – Modularity and Perceived Complexity of Goal:

This episode illustrates how the complexity of the goal seemed to influence the degree to which students employed modularity. It features Naomi – a girl with relatively little Boxer experience – who nevertheless made extensive use of modularity. Naomi's goal was to reproduce her cardboard function spider in Boxer (Figure 9.9).

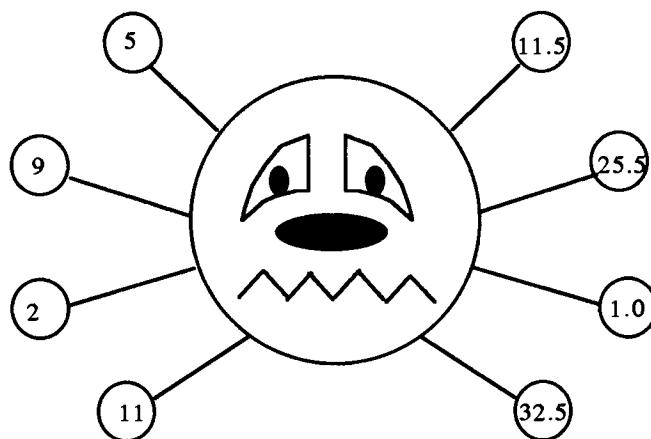


Figure 9.9: Naomi's cardboard spider function model for 3.5x - 6

Naomi created a module called 'spider', obtaining the 'body' of the spider by using 'stamp-hollow-circle 100', a Boxer primitive. She now went on to draw all 8 legs encountering few problems in doing so. A combination of the repetitiveness of typing the same commands over and over, together with the space needed to accommodate her typing, made her susceptible to my suggestion that she group the commands for the leg in a doit box which she could then repeat. Not unnaturally, she called this second module 'leg'.

By the end of the session, she had obtained a working function machine consisting of a large central circle and eight orbital circles (Figure 9.10). In one sense, she had achieved as much as the other students – albeit with help. But Naomi said she wanted to draw the spider's eyes, nose etc. Each of the eyes required the drawing of arcs etc. for which there were no ready-made primitives. I thought that she would encounter both programming and mathematics difficulties in pursuing this goal. However, Naomi was insistent on drawing a spider machine in Boxer that exactly resembled her

original cardboard model. Modular programming now became the problem-solving tool through which she could achieve her goal.

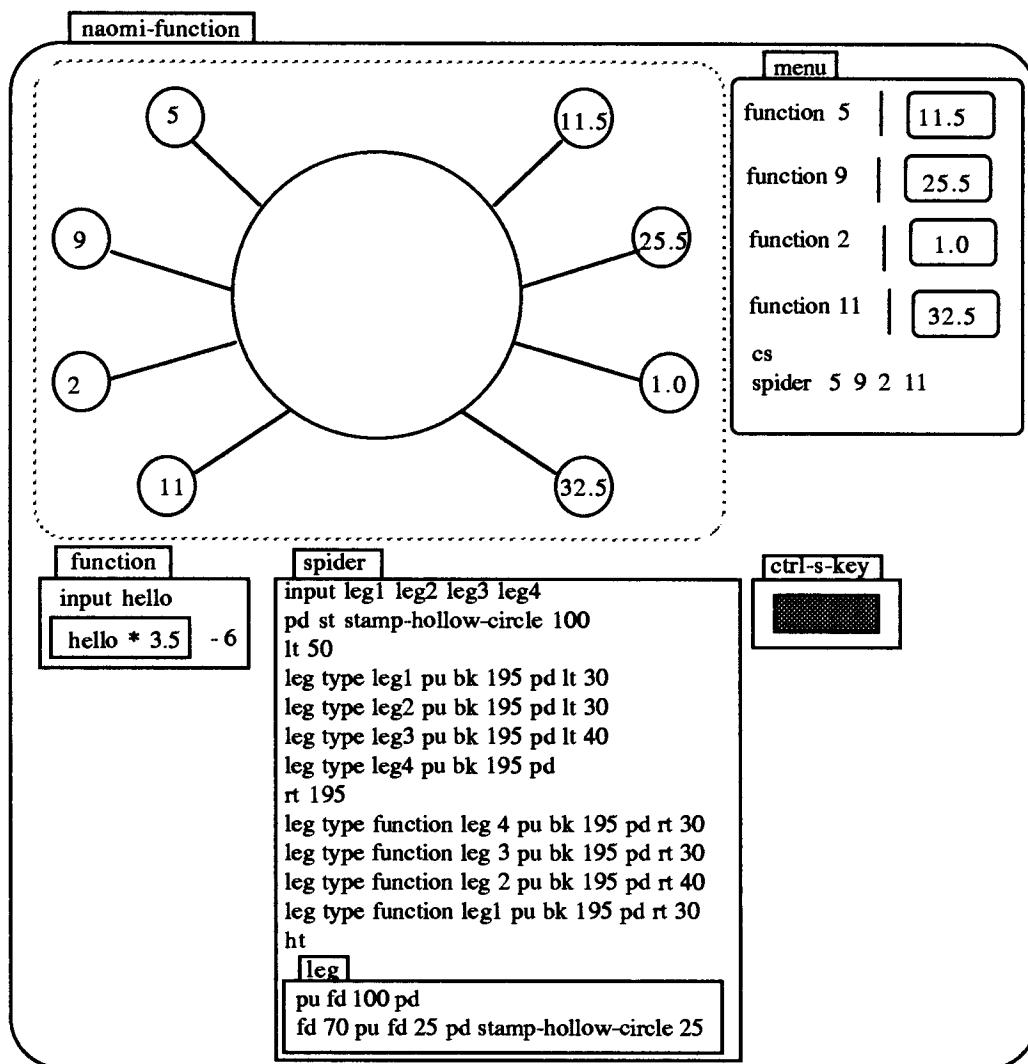
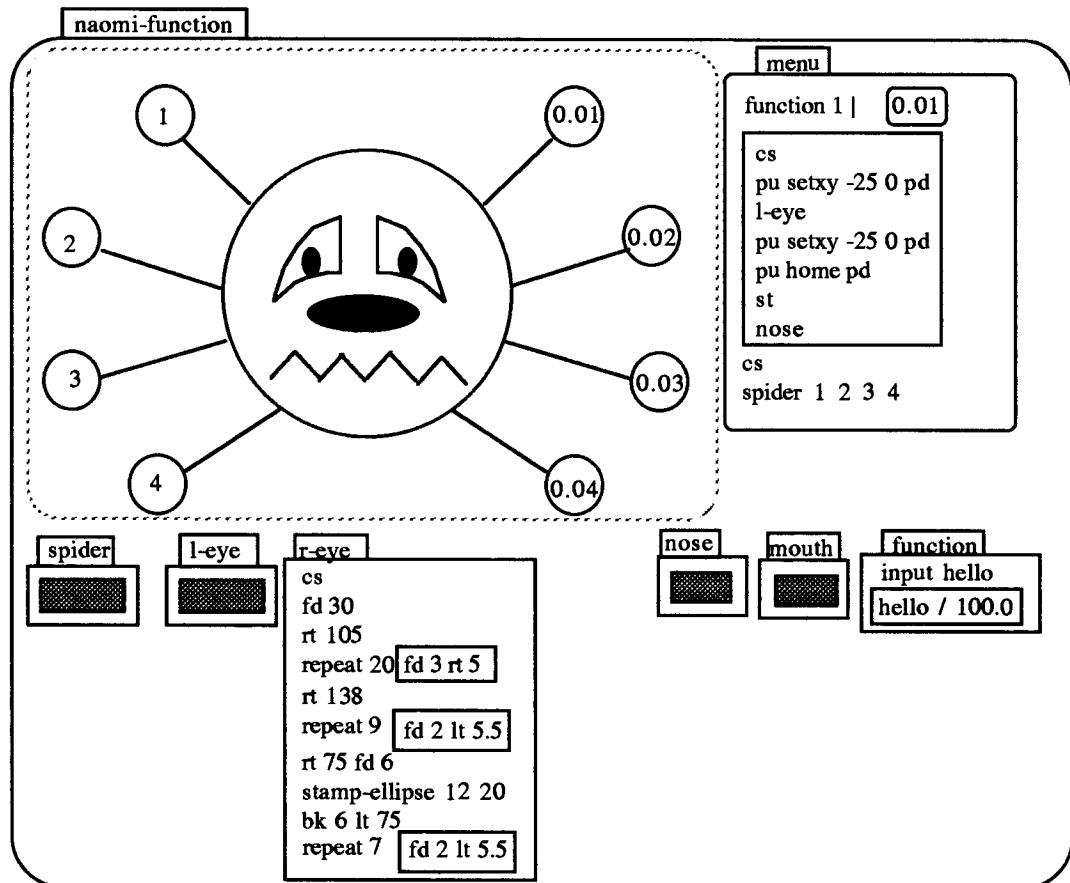


Figure 9.10: Naomi's Spider Function Machine

She asked: "Is left its left or our left?" before creating a new module which she named 'l-eye'. She experimented with angles and sizes, executing her 'l-eye' program line-by-line. If the drawing was not quite right, only the last line needed to be re-written, after which the whole doit box could be re-run. Once Naomi was satisfied that her drawing of an eye in Boxer resembled that on her cardboard model, she creating a new module 'r-eye' and then copied across all the lines from 'l-eye', swapping 'lt' for 'rt' where appropriate.

The next problem was to place the eyes in an appropriate position inside her spider. For this task, she ran each command from the menu (Figure 9.11) and, when it matched her intentions, moved it inside the super-procedure ‘spider’. Neither the nose nor the mouth caused great problems, but it was noteworthy that she created separate modules for each of these components also.



**Figure 9.11: Modularity and Complexity in Naomi’s Spider Machine**

Naomi’s function machine like that of Laura and Courtney was based on a drawing of a face. It was therefore more complex than Liam and Jason’s which simply had input and output boxes along with a central body. Like Laura and Courtney, she employed a considerable number of modules in her programming, and this modularity seems to be linked to her perceived complexity of the goal. However, the procedures themselves were more modular than those of Laura and Courtney in that they did not contain navigation commands. This increased modularity stemmed from her practice of using modules as a way of testing each component without repeatedly drawing the main body of the spider. So, ultimately both quantity of modular procedures together

with the degree of modularity could be attributed to her perceived complexity of the goal.

#### 9.2.1.2: Sub- and Super-procedures

The extent to which students made use of sub- and super-procedures also seemed to differ according to how they perceived their goals. If the goal was seen as relatively complex, then there was a corresponding increase in structured programming and vice versa. Some students viewed super-procedures as a ‘correct-up-to-here’ facility. They thus employed them throughout their construction. Others preferred to run modules one by one from the menu, only combining them into a super-procedure at the end. In their case, naming the super-procedure was in effect announcing the completion of their task.

In Boxer, sub-procedures can be physically inside the super-procedure. This means that the sub-procedure can only be accessed from within the super-procedure, i.e., it is local to it. It is doubtful if any of these young students appreciated the benefits of having local procedures, but some did place sub-procedures inside the super-procedures. The following examples illustrate each of these various practices.

#### Episode 1 – Super-procedures and Perceived complexity of goal:

The form of programming used by Joanna and Nico was at the opposite extreme to Naomi. Joanne’s cardboard model for her function looked something like Figure 9.12, but in drawing it in Boxer, the students decided to dispense with the top ‘dome’ part of the model and its inside ‘bulb’.

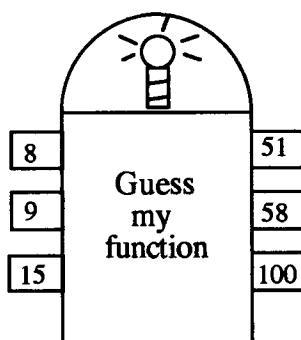


Figure 9.12: Nico and Joanna’s cardboard function model for  $7x - 5$

They began by creating a procedure ‘rectan’ to position and draw the main body of the number machine. Unlike every other set of students, they did not create sub-

procedures to draw the input and output arms. Rather, they included the commands to do this inside ‘rect’, using line-by-line execution to check that they were obtaining their objective. At first, they used a single command to a line but as regularities began to become apparent, blocks of commands were grouped (Figure 9.13). They used the ‘control-K’ and ‘control-Y’ keys to copy and paste these groups of commands, i.e., a labour-saving device rather than elegance was the motivation.

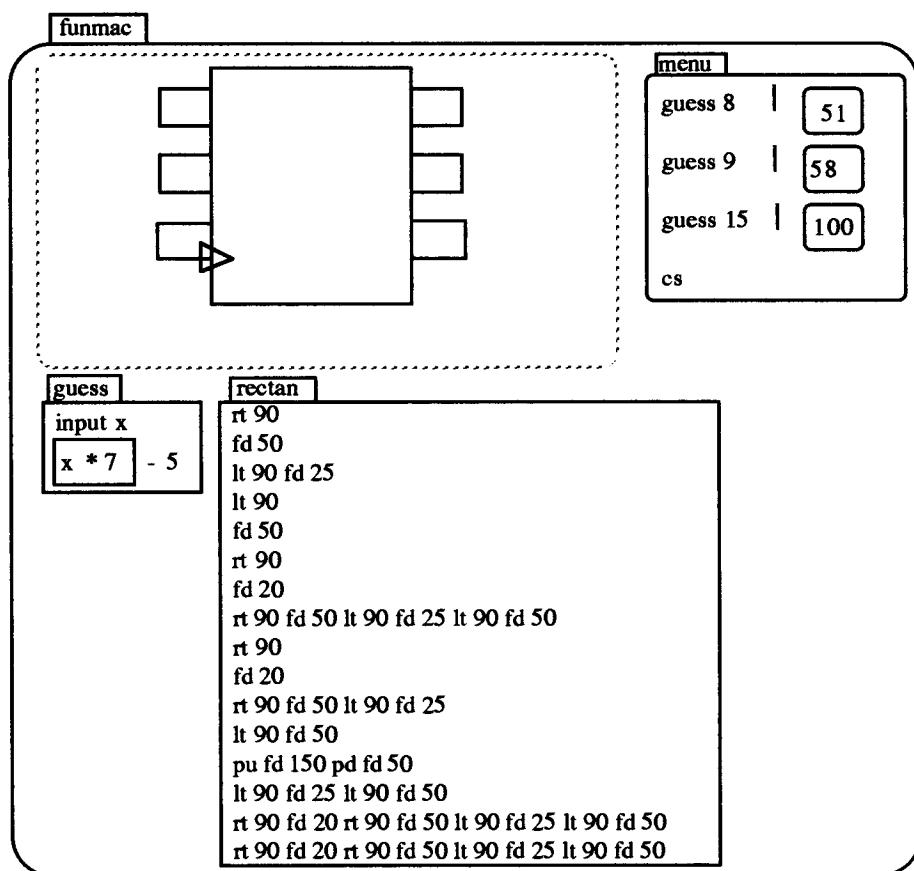


Figure 9.13: Joanna and Nico’s Function Machine under construction

Although the drawing was completed with little difficulty, it was noteworthy that they used very little structured programming in the process. These two students were the two most able of the case-study students, so it seems reasonable to ask why the programming techniques employed in drawing their function machine seem the least sophisticated?

One answer may be found by reflecting on the way they dispensed with the ‘non-essential’ parts of the cardboard model. They abstracted what seemed to matter. A speculative answer for their programming technique would be that they also

dispensed with sub-procedures etc. because they saw no need for them, the goal being perceived as simple to achieve. Boxer's facilities for line-by-line execution (and copying) enabled Joanna and Nico to overcome any immediate problems of programming in such a direct way.

### Episode 2 – Using un-named doit boxes as Super-procedures

This example has been selected to show how Rosie and Chloe adopted a practice (possible in Boxer, but not Logo) of using unnamed doit boxes for super-procedures whilst the program was under construction. They used this to run and join sub-procedures, naming the super-procedure when the task was complete. Their goal was to reproduce their cardboard function model in Boxer (Figure 9.14).

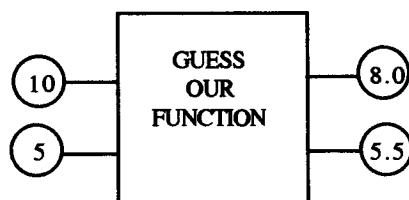


Figure 9.14: Rosie and Chloe's cardboard function model for  $x/2 + 3$

The girls began by creating an un-named doit box, in which they placed the commands to draw a square 'repeat 4 [fd 180 rt 90]'. This led to the drawing being to the right rather than about the centre of their graphics screen. They expressed the desire to begin their drawing to the bottom left of the central position, and I helped them create a 'start' module to do so.

Next, they made a new module, which they called 'in', to draw one of the left-hand arms. They executed commands line-by-line from within 'in' and, following some experimentation until it was seen to work, appended it to the commands in the un-named super-procedure (Figure 9.15). 'fd 60 lt 90'. Then, having executed a 'rt 90' command, they used 'in' again to draw the top left-hand arm.

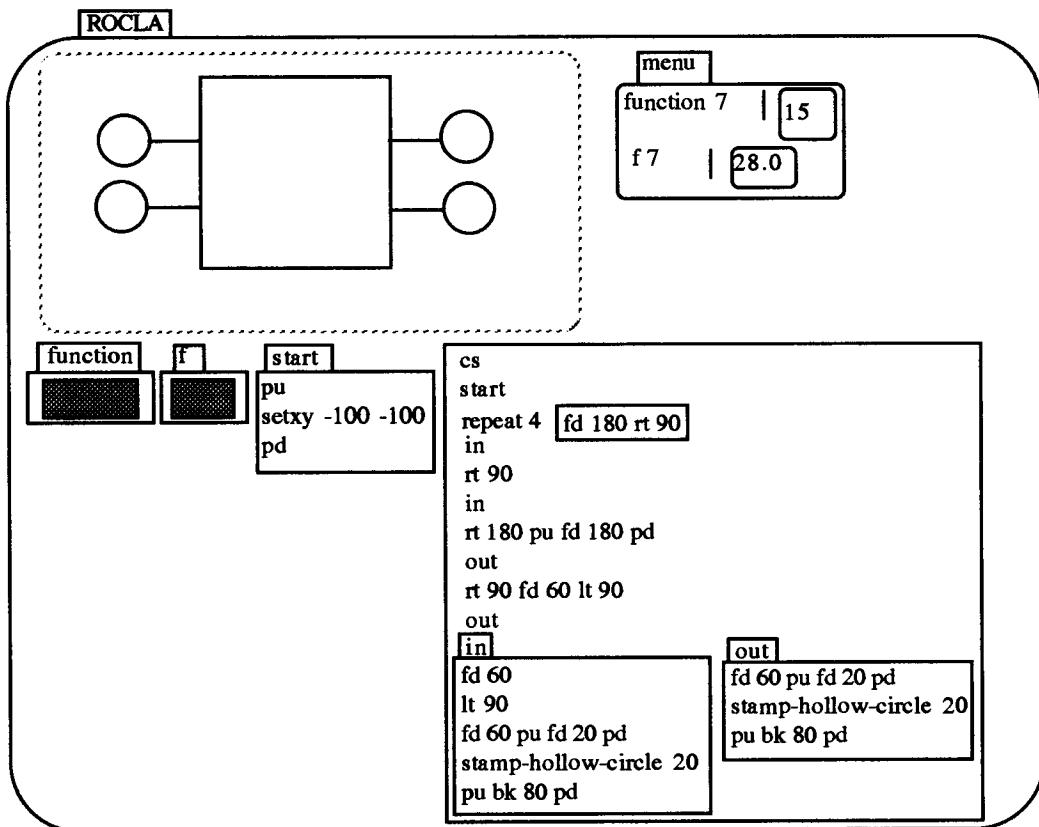


Figure 9.15: Rosie and Chloe's named sub-procedures 'in' and 'out' inside an un-named super-procedure

With the left-hand arms of their number machine drawn, the girls set out about doing the same for the right-hand side. But first, they executed a line ('rt 180 pu fd 180 pd') in the super-procedure to move the turtle across to the right-hand side of the square. They created a new module 'out' to draw one of the right-hand arms, which they also repeated. Interestingly, 'out' in contrast to 'in' did not include navigation commands. Instead, these ('rt 90 fd 60 lt 90') are part of the super procedure (un-named doit box) in which they were executing their program line-by-line. Later, when variables were used, they created new sub-procedures 'in1', 'in2', 'out1' and 'out2'. When their super-procedure drew the whole of the number machine, they named it 'machine'.

The use of sub-procedures for sub-components of the drawing was typical of the research overall, but I observed no consistent pattern as to whether these were placed inside the super-procedure or not. The manner in which this pair used the un-named super-procedure can be compared to the earlier use of a sub-procedure by Naomi. In each case, they were seen as a means to an end, i.e., as problem-solving tools.

### Episode 3 – Named Super-procedures:

This episode has been chosen to show how Sara and Natalie adopted a different strategy: that of naming their super-procedure from the start. Their goal was to draw their number machine ‘Nintendo Gameboy’ in Boxer (Figure 9.15).

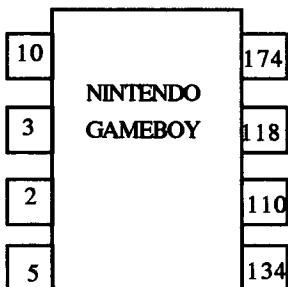


Figure 9.16: Sara and Natalie’s cardboard Nintendo Gameboy function model for  $8x + 94$

The girls began by creating a super-procedure, which they named ‘gameboy’, to draw their number machine. Next, they created an internal doit box called ‘rect’ which they then stepped through to obtain the main body of their machine (Figure 9.17).

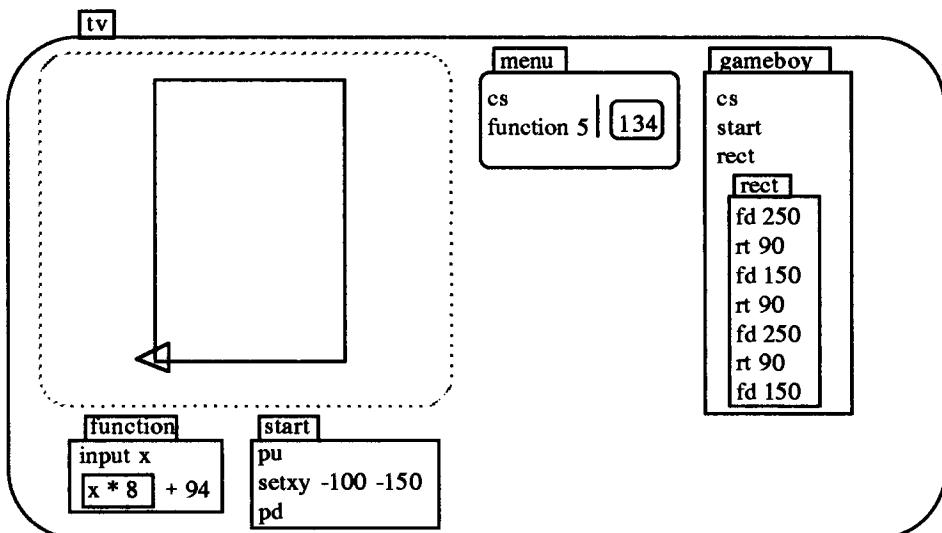


Figure 9.17: Sara and Natalie’s named sub-procedure ‘rect’ inside a named super-procedure ‘gameboy’

Their next aim was to draw the four left-hand-side ‘input boxes’. They created a module called ‘square’ inside ‘gameboy’ to draw the first (bottom left-hand) square, appended the command ‘square’ to ‘gameboy’ along with the navigation commands ‘rt 90 fd 60 lt 90’ and they repeated this until they had all four squares (Figure 9.18).

At this stage, they asked how they could make more screen space and I suggested that they put all their work inside a new doit box. They named this box 'malcom-x' (sic).

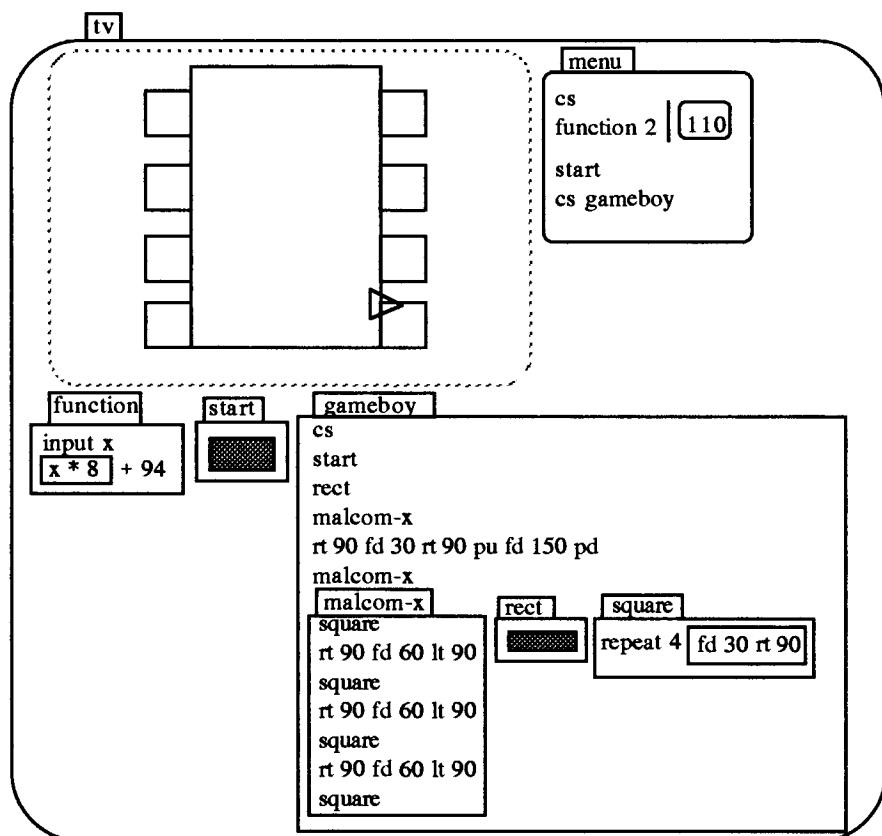


Figure 9.18: Sara and Natalie's developing use of structured programming

The girls then moved the turtle across to the bottom right-hand corner, and tried to use 'malcom-x' again, but it drew the four squares down and off the screen. Natalie saw that she could use still use 'malcom-x' if she started at the top right-hand-side instead, and so she went back into 'gameboy', amended the navigation commands, after which they ran the 'gameboy' again. Later still, when there was a need to take account of the input and output variable values, they created a new sub-procedure 'malcom-x2' for these right-hand output boxes.

Sara and Natalie's approach to super-procedures was different to Rosie and Chloe's in that they named the super-procedure from the start and then executed it from the menu. It was also different in that they employed a super-procedure 'malcom-x' inside another super-procedure 'gameboy'. It was similar in that they saw the merits in re-using sub-procedures elsewhere, e.g., their use of 'malcom-x' twice. It seems that they too saw super-procedures as problem solving tools.

## **9.2.2: Linking Programming and Number**

The final part of the construction process was concerned with making the graphic of the function machine operational. This entailed using variables and functions and it was through these that the programming activities were linked to the number activities.

### **9.2.2.1: Variables**

The students' goal in constructing their function machines in Boxer was to draw a likeness of their cardboard models, including typing their numbers in the input and output boxes. Few of the students were able to make the move from specific inputs to using variables without help. I adopted two different approaches to intervention. The first approach was to wait until they had successfully drawn the function machine using the 'type' command for input and output numbers before intervening to suggest variables. The second was to intervene at an earlier stage to suggest that they use 'type function 1', 'type function 2' etc. for the output numbers. My later intervention was to ask if they could now make their function machines work for any number.

The first approach entailed adding an input line to the super-procedure, and replacing the specific inputs to the 'type' command with variables. For the output boxes, the students also needed to use their function on their variable, i.e., they needed to write something like 'type function x'. The following examples illustrate some of the students' responses.

Sara and Natalie adapted 'malcom-x' to include the line 'input num1 num2 num3 num4'. They changed the program so that 'type num1' 'type num2' 'type num3' and 'type num4' replaced their specific inputs of 'type 10' 'type 3' 'type 2' and 'type 5' for the four input numbers in the left-hand-boxes. They then repeated this on the right-hand side also, but got the same numbers again and in the wrong order. Naomi chose leg1 leg2 leg3 and leg4 to depict her four input variables, but she thought that she could also get the output numbers by writing 'type leg5', 'type leg6' etc.

With Laura and Courtney, the output 'typing' lines were written (at my instigation) as 'type function 1', 'type function 2' and 'type function 3' from the start. When I asked them to change their function machine to handle other numbers. it became clear that the microworld as it stood was limited and all of the lines concerned with typing would have to be changed every time the numbers were changed. This provided the

motivation for the use of variable inputs which were now included in ‘toer’, ‘toer,2’ and ‘gill’. I took a similar approach with Liam and Jason.

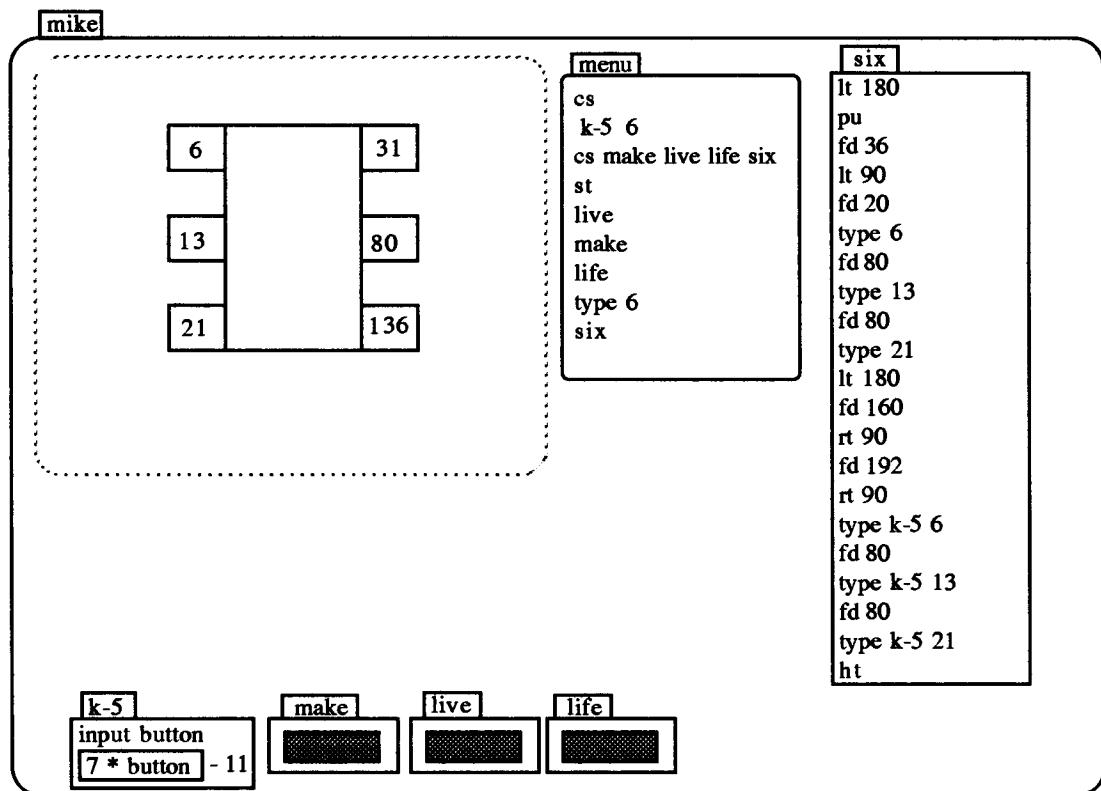


Figure 9.19: Changing the Domain and Range for Liam and Jason’s Function Machine

Once the boys had achieved their goal of typing the numbers in all of the output boxes, I intervened to ask about varying the input numbers.

Extract :

Name	Dialogue	Commentary
Interviewer:	You know that you have just done this for six, thirteen and twenty-one. If you wanted to use other numbers, would you know how to do that?	Points to 6, 13 and 21 in the input boxes.
Liam:	Yeah, you just type ...	
Jason:	... yes, you change all these [points to the numbers in the doit box ‘six’]	Note the reference to the program lines.
Liam:	Say, You wanted this to be twelve [refers to the ‘13’]. You delete [He does this].	Liam acting out the role of teacher.
Jason:	But, you have to change the bottom one as well. [points to the line ‘k-5 12’]	This refers to the need to match the output with input, i.e., in this case ‘k-5 12’ and 12.

Another pair, Joanna and Nico, had no difficulties in moving from specific numbers to variables. Nico said “Just, change the inputs.” They changed the commands ‘type 8’, ‘type 9’ and ‘type 15’ to ‘type num1’, ‘type num2’ and ‘type num3’ respectively. Nico wrote the line ‘type guess num3’ etc. to correspond with the line ‘type num3’ showing that he understood the need to apply the function ‘guess’ for the output numbers.

In this section, I have given an account of how students adapted their function machines to handle variables. The idea that their Boxer function machine could be made into a functioning machine was often a revelation for the students and it set the scene for the ensuing number investigations. Few of the students seemed to have any difficulties with the *concept* of variables. The difficulties that arose were primarily concerned with adapting the program for the output boxes to include their function. My approach (with Laura and Courtney, and Liam and Jason) of including the function with their specific values from the start proved the most successful in motivating the use of variables.

Questions relating to the concept of function arose at two main points in the study: as the students expressed their function rule in Boxer, and after they had made the graphic of the number machine ‘operational’. In the first case, the students needed to formalise their language “Times two and add one.” etc. in order to express it in Boxer. In the second case, they became aware that their number machine (together with its rule) operated on a larger domain of numbers than those they had used on their cardboard models.

#### 9.2.2.2: Formalising functional expressions

##### Episode 1 – From Intuitive to Formal expressions for Functions:

This first episode with Rosie and Chloe has been selected to illustrate how the Boxer formalism structured their expression of functions.

Rosie and Chloe asked to ‘do’ their function machines in Boxer. My response was to inquire what their machines did. The girls fetched their cardboard models, and I helped them make the doit box ‘function’ (Figure 9.20) for Rosie’s ‘Times two and add one’ function, which she had used with her cardboard model. They expressed this in Boxer as ‘[number \* 2] + 1’.

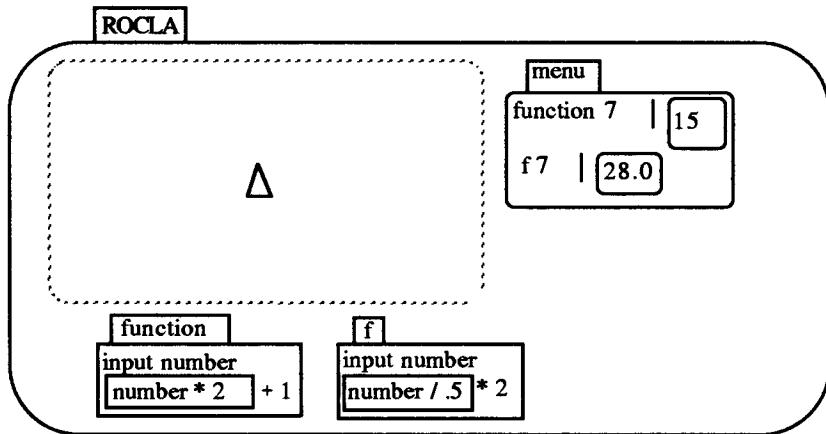


Figure 9.20: Articulating informal algebraic expressions through Rosie and Chloe's function doit boxes

The students decided to create a doit box for a new function 'f', which they spoke of as "Divide by point five and times by two." and which they expressed in Boxer as '[number / .5] \* 2'. This seemed an unusual way to articulate a function which is algebraically equivalent to '4x', so I asked the girls what would happen if the function had an input of 7. Using a calculator, they keyed in the steps  $7 \div .5 \times 2$  and obtained an output of 28. They then checked it by executing 'f 7' from the menu.

Within the medium of Boxer, the formalism '[number / .5] \* 2' directly articulated the function which the students wished to express. Of course, the need for unambiguous programming imposes a discipline which is not unique to Boxer. Tall (1983), for example, writes of similar difficulties in relating the formalism of algebra in BASIC to conventional pen-and paper algebra. However in Boxer, the students were not merely adopting a textual convention such as '(number / .5) \* 2', they were writing directly into executable boxes. The doit box provided a 'functional' model for their expression in the same way that the '=' symbol on a calculator does for numerical operations (diSessa, 1986b).

Shuard and Rothery (1984) have observed how the translation of mathematical expressions from verbal to written form is not one-to-one. Thus, the 'Divide by point five and times by two' function in its verbal form clearly denotes priority: what is said first is done first. However, the written 'form  $x \div .5 \times 2$ ' could be ambiguous. The Boxer expression '[number / .5] \* 2' contains no ambiguity: what is in the box is executed first. In this sense, doit boxes are 'operational' brackets. Even if the students had not inserted 'number / .5' in a doit box, there would still have been a significant difference between the Boxer expression 'number / .5 \* 2' and the written

algebraic expression ‘ $x + .5 \times 2$ ’ In the latter, students would face what Tall and Thomas (1991, p. 126) have termed the ‘parsing obstacle’, i, e., the need to unravel the sequence in which the algebra should be processed. In the former, they could learn how the computer prioritised such expressions by experimenting with key-presses.

### Episode 2 – Detaching Signifier and Signified in functional expressions:

This second episode illustrates how Liam and Jason in expressing functions in Boxer, could on the one hand attach meaning to functional and variable names and on the other hand detach the arbitrary nature of the names from the underlying concepts.

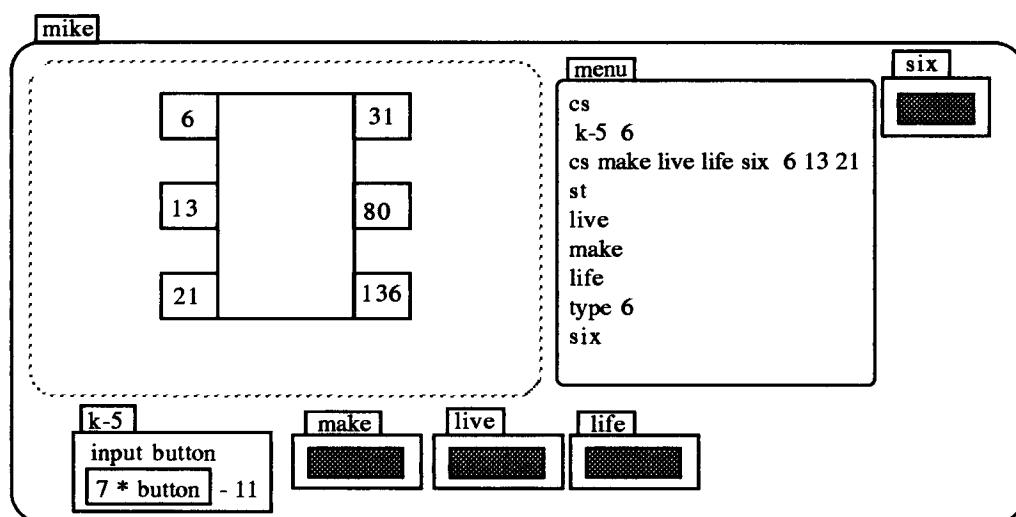


Figure 9.21: Expressing functions through informal algebra with Liam and Jason’s k-5 doit box

Liam and Jason created a doit box called ‘k-5’ in which to put Jason’s ‘Times three. Take away five’ function. They expressed this function in Boxer as ‘[3 \* button] - 5’. ‘k-5’ was then typed in the menu and tested it with an input of 6 (Figure 9.21). Whilst this expression mirrored the pen-and-paper convention, neither their function name ‘k-5’ nor their variable name ‘button’ did. The fact that they chose to use such names demonstrated that they were aware of their arbitrary nature, but it also shows, I believe, how within the medium of Boxer, the *concepts* of function and variable were disassociated from their algebraic symbolism.

In a pen-and-paper context, deciding which features of a function are essential and which are arbitrary may not be simple. Consider the following examples of functions which students encounter at later stages of mathematics.

$$(i) f:x \rightarrow 2x + 1, (ii) g:x \rightarrow 2x + 1, (iii) f:t \rightarrow 2t + 1, (iv) f:x \rightarrow \frac{x}{2} + 1.$$

The first three all represent the same function, but the formalism is different in each case. (i) and (ii) differ only in the name (letter) denoting the function. (i) and (iii) differ in the letter denoting the variable. (ii) and (iii) differ in both of these. Finally, (iv) uses the same names (letters) for the function and the variable as (i), but the function itself is different.

Within the context of their function machine, students were able to visualise the concept of a function not simply as a letter  $f$  or  $g$  along with a rule, but rather as the name of a doit box which performed the function. Similarly, a variable was more than an ‘x’ on the page. It stood for what went into the function machine – a use stressed by the preceding word ‘input’. Thus, Liam and Jason felt confident in choosing their own names: ‘button’ ‘number’ etc. The evidence suggested that the Boxer function machine was giving students a means – ‘a generic organiser’ (Tall and Thomas, op cit., p. 132) of accessing a form of algebraic thinking, and as the following section relates, this was crucial to their subsequent visualisation of number.

### **9.3: Interacting with the Boxer Function Machine**

This section documents the way that the Boxer function machine mediated students’ expression of number, and tries to identify facets of the representation and the interaction which influenced this expression. It separates interactions into two categories according to whether the students were dealing with directed numbers, or with decimals, and in terms of the operations involved. It indicates how the content, methodology, and the students’ construction of meaning were all structured by their interactions with these computational objects.

#### **9.3.1: Directed Numbers**

##### 9.3.1.1: An Operational approach to Directed Numbers

The semi-structured interviews (chapter 7) which hinted that these students found the idea of directed numbers unproblematic was confirmed in this part of the research. But, it also demonstrated that procedural misconceptions could co-exist with conceptual understanding. The episodes selected here illustrate how the medium enabled students to correct these misconceptions and to generate their own meanings.

### Episode 1 – Conceptual and Procedural Understanding

This episode illustrates how the function machine enabled Naomi to overcome procedural misconceptions concerning directed numbers.

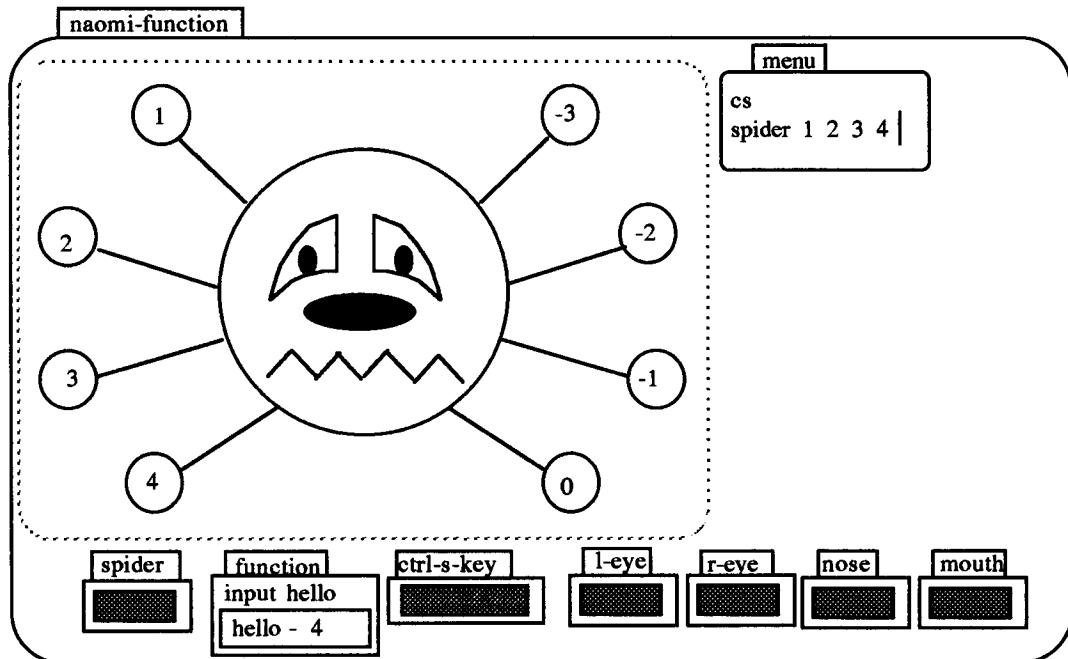


Figure 9.22: Approaching Directed numbers through subtraction with Naomi's Function Machine

#### Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Suppose you changed your function machine into a take way four machine.	The inputs were 1, 2, 3 and 4.
Naomi:	Take way four?	She changes the function's expression to 'hello - 4'
Interviewer:	Yes.	Naomi runs 'spider 1 2 3 4' from the menu
Interviewer:	Any idea why you got those numbers?	Points to -3, -2, -1, 0 down the spider's output legs.
Naomi:	Because one take-away four is minus three. If you've got one and you take away one, you get nothing, and if you take away more it goes into minus numbers. After one, you get minus one, minus two, minus three, minus four, and so on.	

The next extract shows how Naomi gave a different formulation for the subtraction of directed numbers, as well as showing that my articulation left something to be

desired! In this case, I challenged her to find a function to map the input to the output numbers.

**Extract 2:**

Speaker	Dialogue	Commentary
Interviewer:	Suppose instead of that, I wanted to end up with minus ten, minus eleven, minus twelve, minus thirteen.	Reading down, the inputs are 1, 2, 3 and 4.
Naomi:	You won't have minus ten, minus eleven, minus twelve, minus thirteen like that though, because that's a smaller number [points at 1], so that will take up more minuses.	Naomi points <u>down</u> the screen and corrects me!

Naomi initially changed the function to ‘hello - 15’, and obtained -14, -13, -12 and -11 on her spider graphic. She observed that the output numbers differed in each case by one from the target numbers, and so she changed the function to ‘hello - 16’. Re-running ‘spider 1 2 3 4’, resulted in her spider displaying -15, -14, -13 and -12. Realising that she had adjusted the function in the wrong direction, she re-entered the function doit box, and finally changed the function to ‘hello - 14’.

This episode was typical in that it revealed that Naomi did not appear to have any difficulties with the *idea* of negative numbers, and moreover had a good conception of their relative sizes. Nevertheless, she had procedural misconceptions. The function machine supported both her conceptual and procedural understanding. In the former case, the operation inside the function machine was the means through which negative numbers came into being. Her description: “ ... that will take up more minuses.” seems to have stemmed from the way the function machine carried out successive subtractions such as  $1 - 4$ ,  $2 - 4$ ,  $3 - 4$ ,  $4 - 4$ , and displaying the results down the screen. In the latter case, it enabled her to overcome her procedural misapprehension by allowing her to directly change the function’s expression in the doit box and observe the consequences. It thus gave her the means to self-correct these initial misconceptions.

**Episode 2 – Directed Numbers as a Process:**

This episode has been selected both because it offers evidence that the operational nature of the function machine fostered what might be termed an ‘operational’ form of reasoning, and also because it demonstrates how Boxer enabled Joanna and Nico to formalise Nico’s expression “Take away two lots of what you’ve got.”

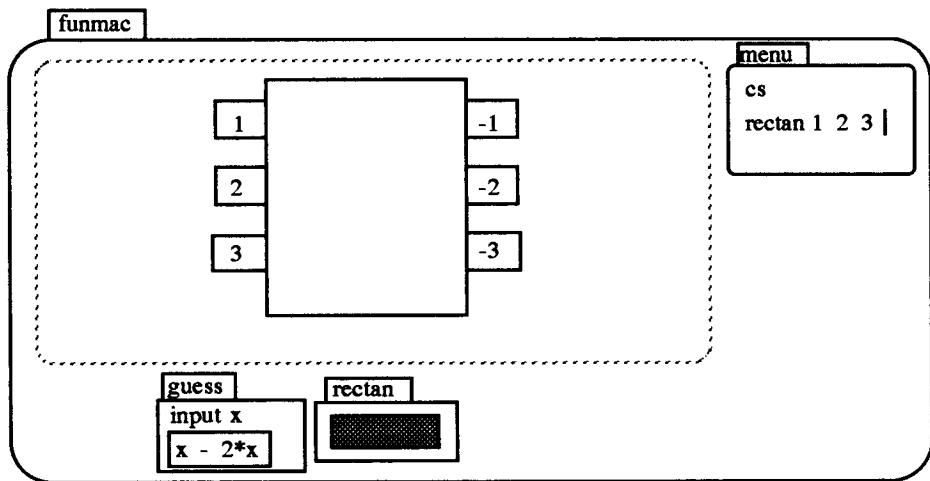


Figure 9.23: Approaching Directed Numbers through functions with Joanna and Nico's Function Machine

In probing the students' conceptions of directed numbers, I began with a somewhat clumsy formulation of the intended question. The intention was for the students to find a function to map 1, 2 and 3 to -3, -2, and -1 respectively, i.e.,  $x \rightarrow x - 4$ , but by expressing the negative numbers in descending order, it was interpreted to mean finding a function to map 1, 2 and 3 to -1, -2, and -3 in that order, i.e.,  $x \rightarrow -x$ . On the other hand, this ambiguity opened up windows into the students' conceptions of numbers generally, e.g., Joanna's "You can't go under nought by dividing it."

**Extract 1:**

Speaker	Dialogue	Commentary
Interviewer:	Start at one, two and three here and end up with minus one minus two and minus three.	Points to input boxes.
Interviewer:	You'll have to change your function machine.	Refers to the function doit box.
Nico:	Divide by ten.	
Interviewer:	What do you think Joanna?	
Joanna:	You can't go under nought by dividing it.	
Interviewer:	I agree.	
Joanna:	Taking away?	
Nico:	Take away the number you've got. How would you do this in Boxer?	This refers to $1 \rightarrow -1$ , $2 \rightarrow -2$ , $3 \rightarrow -3$
Nico:	Take away two lots of what you've got.	
Interviewer:	Oh, I see what you are saying. It's probably because I said minus one minus two and minus three.	Changes expression in 'function' to ' $x - 2*x$ ' and they run it.
Interviewer:	Well done! That's not what I expected.	
Nico:	What were you expecting?	Explains.

My request for a function that mapped 1 2 and 3 to -1 -2 and -3 was ambiguous in its articulation. The function I expected the students to obtain was  $x \rightarrow x - 4$ , but Nico's: "Take away two lots of what you've got." was probably a better response to the question as formulated. In a pen-and-paper context, Nico's expression would translate as the function  $x \rightarrow x - 2x$ , and might seem an unusual way to write the function  $x \rightarrow -x$ . In this Boxer context, writing ' $x - 2*x$ ' inside the 'function' doit box directly captured the students' informal expression without any loss of clarity.

Joanna's response "You can't go under nought by dividing it." amounted to saying that for any positive numbers  $n$  and  $m$ ,  $\frac{n}{m} > 0$ . Whether this insight was a situated abstraction or more general is not clear. What is clear is that it is a non-trivial insight, and one which seemed to be engendered within the operational context of this microworld. These students were *reasoning* about directed numbers and not simply carrying out algorithms with them. Their number machine formed the core of this way of thinking since it was its operation that they were discussing.

#### 9.3.1.2: Relating Addition and Subtraction through Inverse Functions

Although I have separated out subtraction and addition for the purpose of analysis, in the Boxer situation, one usually followed the other. Each of the following episodes came immediately after the students had succeeded in using subtraction to map positive whole numbers to negative numbers. I then reversed the formulation by asking them to find a function that would map negative whole numbers to positive whole numbers.

##### Episode 1 – Constructing Meaning for adding Directed Numbers:

This episode has been selected to show how two girls Sara and Natalie, who had little knowledge of directed numbers, could nevertheless construct meaning for such quantities in the context of their Boxer function machine.

Sara and Natalie were asked to predict what the function machine ' $x + 5$ ' would do to inputs of -4, -3, -2, -1. Neither was able to answer. However, when the program was run and the numbers 1, 2, 3, and 4 were displayed, Natalie drew her finger from left to right across the graphic screen from the numbers in the input boxes to the numbers in the output boxes (Figure 9.24) and said: "You could do four add one is five, three add two is five, two add three is five, one add four is five." This observation amounts to a

situated abstraction in that she has abstracted the relationship that if  $-x + 5 = a$ , then  $x + a = 5$ , but it is situated in that it is tied to the visual display on the graphics box.

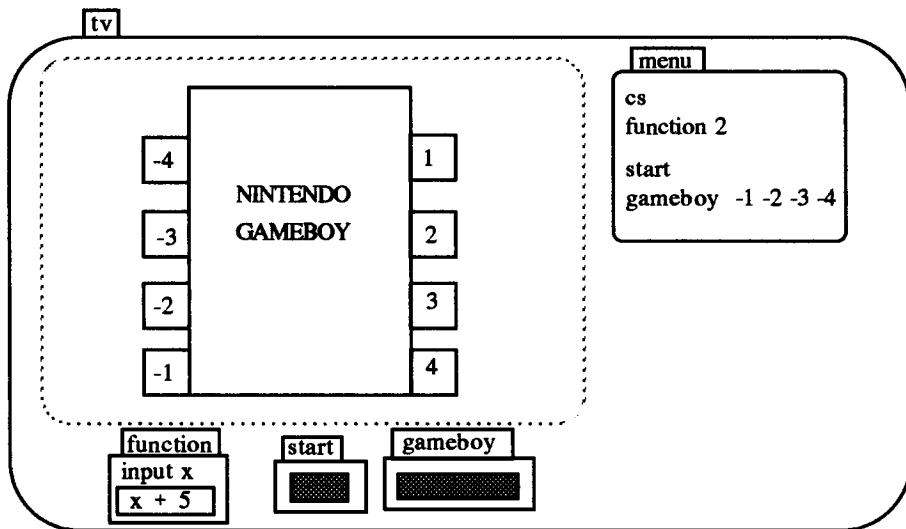


Figure 9.24: Approaching Directed Numbers through inverse functions with Sara and Natalie's Function Machine

It is not only the proffered solution that is situated, but the whole exchange. My question, for example, in a pen-and-paper setting could be expressed simply as '5 - 4', '5 - 3' etc. This would undoubtedly summon up the subtraction algorithm, whereas '-4 + 5', '-3 + 5' etc. would possibly cause a search for some other number 'facts'. Likewise, the students' discovery of a pattern between input and output numbers was structured by the situation. It would, of course, be possible to find such patterns with a conventional 'arrow' diagram, but then directed numbers are seldom introduced in this way. A little later, I asked the girls to predict how the functions  $x \rightarrow x + 10$  and  $x \rightarrow x + 20$  would behave with the same inputs of -4, -3, -2, -1.

#### Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Let's change that to ten so we have a plus ten there. Now before we do it, what do you think?	Sara changes the function to ' $x + 10$ '.
Sara:	Six, seven, eight, nine.	Both point to the output boxes on the graphic.
Interviewer:	You are going the other way round. Try it.	To Sara.
Sara:	[To Natalie] You were going down and I was going up.	Refers to order of dealing with -4, -3, -2, -1.

In interacting with the visual trace of their number machine's action, the girls were able to formulate a rule about how addition affected negative numbers. But, in the course of the dialogue, something else happened. We became aware of our predilection for listing these quantities in the opposite way to their positive counterparts. It seemed natural, for example, to say minus one, minus two, minus three etc. Yet in doing so, we were counting down rather than up. This can be seen in the menu command 'gameboy -1 -2 -3 -4' above.

### Episode 2 – Connecting directed numbers through Functions:

This episode has been chosen to show how the number machine setting pre-disposed Liam and Jason to search for a *function* as a way of building connections between directed numbers.

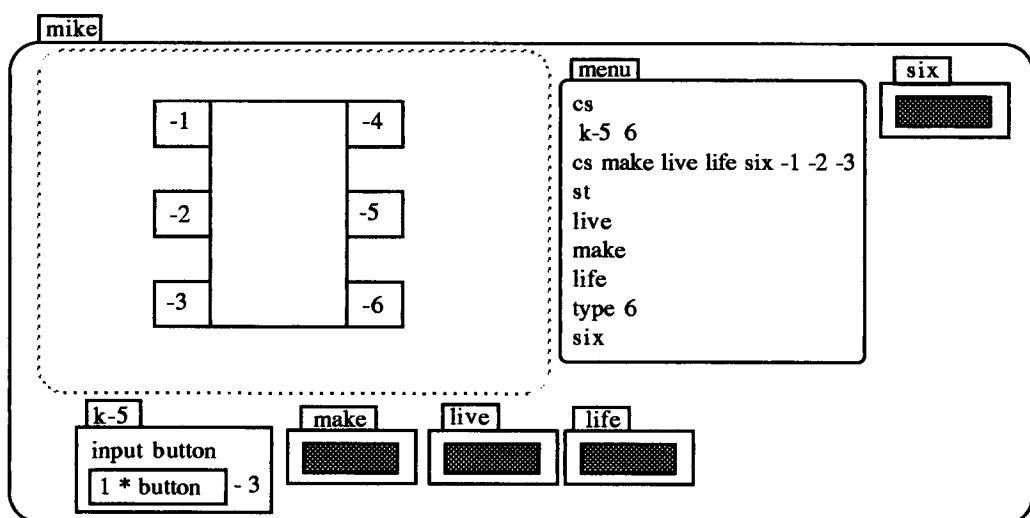


Figure 9.25: Approaching Directed Numbers through functions with Liam and Jason's Function Machine

#### Extract 1:

Name	Dialogue	Commentary
Interviewer:	Suppose this time you've got minus one, minus two and minus three going in. And coming out is two, one, zero.	The function was: -1 --> 2 -2 --> 1 -3 --> 0
Liam:	Take away one.	
Jason:	Take away three. Get rid of ...	Points inside 'k-5' doit box.
Both:	They run '[1 * button] - 3' and get -4, -5, -6	
Jason:	How do you do nothing?	
Liam:	Nought.	They run '[0 * button] - 3' and get -3, -3, -3

Liam:	Minus one take away three . . .	
Jason:	. . . minus four.	
Jason:	It's still not right.	
Interviewer:	Perhaps you could say what you think is wrong.	
Liam:	Minus one take away three equals minus four. It's an add!	Liam realises that $ x $ was getting bigger.
Interviewer:	Okay . . .	
Liam:	How do I do add though?	
Jason:	It's that one.	Points to + key.
Jason:	Why do you want add?	They run 'button + 3' and get 2, 1, 0
Jason:	Oh yes, now I see. It's add.	Gestures towards screen.

The boys' first attempt led them to create the function '[1 \* button] - 3' and instead of getting 2, 1 and 0 from their input numbers of -1, -2 and -3, they obtained -4, -5, -6. This didn't deter them, and they immediately tried the function '[0 \* button] -3', and got outputs of -3, -3, -3. Unfortunately, they used -3 instead of - 3 and so when they tried other coefficients of the variable in the doit box, they continued to obtain the screen trace of -3, -3, -3. Nevertheless, this provoked Liam to consider afresh the effect of subtracting 3 from -1, and hence prompted him to exclaim: "It's an add." Jason was running a little behind Liam at this point, but he too seemed to have seen why the function was 'an add'.

Some evidence for this could be gleaned from my follow-up question, in which I asked the boys to predict the effect of having an 'add two' machine (again with inputs of -1, -2 and -3). The students answered immediately and together "One, zero and minus one". Thus, while their initial attempts at finding a function to map -1, -2, and -3 to 2, 1, and 0 were incorrect, through their interactions with the function, they began to see *how* their solution differed from what they expected and this visual feedback ultimately gave them sufficient clues to create the desired function.

In pen-and-paper settings, the equivalent algebraic expressions to Liam and Jason's initial attempts might appear as  $1x - 3$  and  $0x - 3$ . It is perhaps more likely that they would appear as  $x - 3$ , and  $-3$  respectively. But, these symbolic shortcuts cut out a stage of expression that many students need to go through, i.e., the recognition for themselves that  $1x = x$  and  $0x = 0$ . Boxer's formalisations for functions constitutes an operational algebra in that the students could see the effect of the expression by running it, and it was this which seemed to have given the students the extra prop for supporting their thinking.

### 9.3.2: Decimals

#### 9.3.2.1: An operational approach to Decimals

The findings from the semi-structured interviews (Chapter 7) suggested that, while students did not naturally use either directed numbers or decimals, contexts could be found which provoked the use of the former. In the latter case however, similar contexts very often elicited the use of fractions rather than decimal expressions. It appeared from the evidence obtained that, for these young students, fractions construed an action-based (operational) conception of number, i.e., a quarter of a whole, or a tenth of a whole, whereas symbols such as 0.25 or 0.1 conveyed no such meaning. Two episodes have been chosen to illustrate how in interacting with their Boxer function machine, students began to construct such meaning.

#### Episode 1 – Building bridges between informal and formal expressions for Decimals

This episode has been selected to show how Liam and Jason were able to attribute meaning to decimals by directly relating them to the expressions inside their ‘k-5’ function doit box.

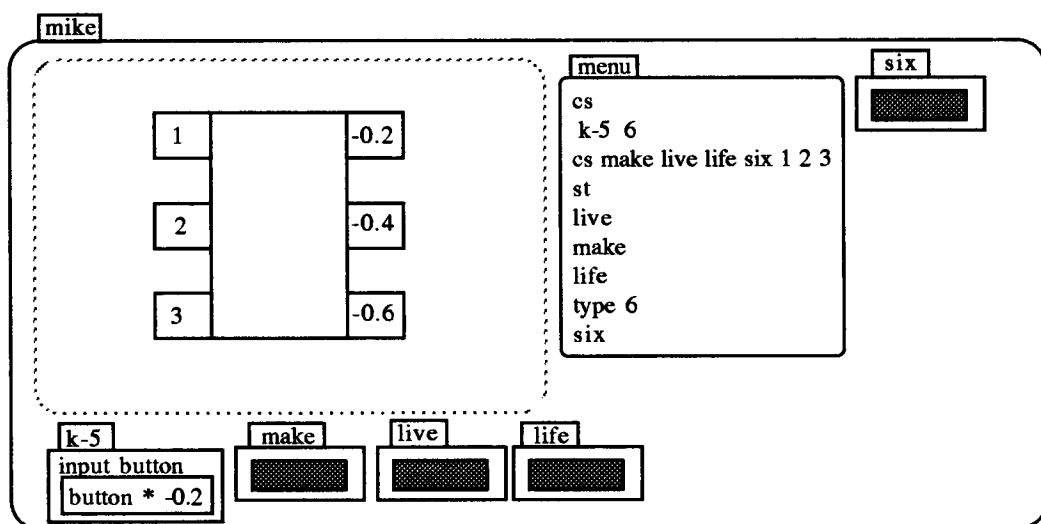


Figure 9.26: Approaching Decimals through functions with Liam and Jason's Function Machine

The boys were challenged to find a function that would map 1, 2, 3 to 0.1, 0.2, 0.3 in that order. Jason said: “Take away nought point two.” They entered the k-5 doit box to alter the function to ‘button - 0.2’. Instead, they wrote ‘button \* -0.2’. They forgot

to delete the multiplication symbol '\*' (left over from a previous function) and they wrote '-0.2' for '- 0.2'. So, on running it, they multiplied by -0.2.

**Extract 1:**

Name	Dialogue	Commentary
Interviewer:	This time, [points to each of the outputs in turn] I'd like nought point one, nought point two, nought point three coming out. How do you think you are going to do that?	The challenge was to find a function to map: 1 ---> 0.1 2 ---> 0.2 3 ---> 0.3
Jason:	Take way nought point two. [They run this]	I point out that they have multiplied by -0.2.
Liam:	Times nought point one. See if I have done it right.	
Jason:	It's getting there.	
Interviewer:	You are doing two things ...	Both Jason and I point into the doit box at the function.
Jason:	Get rid of the minus.	
Liam:	I've done it.	

There are several noteworthy features of this exchange. Jason's "Take away nought point two." suggests that, at the beginning of this episode, the boys did not have a well developed knowledge of a functional relationship between the whole numbers 1, 2, 3 and the decimals 0.1, 0.2, 0.3. The error itself and its consequences were both rooted in the function's expression in the 'k-5' doit box. The expression in the doit box was 'button \* -0.2' rather than 'button – 0.2' as the boys had intended. The screen trace of -0.2, -0.4 and -0.6 enabled the boys to exploit their mistake first in changing the function's expression to 'button \* -0.1' and then to 'button \* 0.1'. They did not seem to find it surprising that multiplication by 0.1 resulted in a smaller number.

Of course, there was a degree of serendipity in their choice of multiplication by 0.1, and I intervened at this point to probe the meaning which they attached to 0.1.

**Extract 2:**

Name	Dialogue	Commentary
Interviewer:	You've used the function to change one to nought point one, two to nought point two, three to nought point three. But what have you actually done? Why does it work?	
Liam:	'cause times point one is point one.	
Interviewer:	That's circular. It doesn't tell me much.	

Interviewer:	Why should point one of two give you point two?	Points at function machine.
Liam:	It's point one.	
Interviewer:	Okay. A better question then. What is nought point one?	
Liam:	It's less than one.	
Interviewer:	Well that's true. If you had a bar of chocolate and I asked you to split it up so that everyone got point one of it, what would you be doing?	
Liam:	Taking a hundredth of it. A thousandth? A thousandth of a millionth?	
Interviewer:	What do you think, Jason?	Doesn't answer.
Liam:	A tenth of one.	

This was a fairly typical extract in that the initial utterances were often quite wild indicating that the meaning which the students attached to symbols such as 0.1, 0.01 etc. were not yet stable. But the final phrasing of Liam's answer "A tenth of one." for 0.1 was also typical. It seems to be derived from the fact that they could see the Boxer expression 'button / 10.0' inside the 'k-5' doit box in front of them.

Following this exchange, I (perhaps unwisely) drew attention to the equivalence of multiplication by 0.1 and division by 10. The boys were then successively challenged to find functions which (with the same inputs of 1, 2 and 3) led to 0.01, 0.02, 0.03 and then 0.001, 0.002, and 0.003 respectively. I then probed the meaning they attached to the decimal fractions.

### Extract 3:

Name	Dialogue	Commentary
Liam:	That's one-hundredth, two-hundredths, three-hundredths.	Points to 0.01, 0.02, 0.03 in the output boxes
Interviewer:	Why are you saying one-hundredth, two-hundredths, three-hundredths?	
Liam:	In one hundred, there are three digits, and there's three digits there.	
Interviewer:	Okay, that's one explanation, but suppose I said what's the connection between this and your function machine?	[points to 0.01 in the output box]
Liam:	It's divided by a thousand. I mean one hundred.	

Liam's ways of linking 0.01 with 100 were not idiosyncratic to him. A number of other students discovered this same pattern or 'backward rule', i.e., division of 1 by 10 led to 0.1 (ten backwards); division of 1 by 100 led to 0.01 (a hundred backwards)

and so on. Boxer was mediating the students' expressions in that they were relating the Boxer formalisation 'button / 10.0' in the function doit box to the display 0.1 on the graphic number machine. This was not as straightforward as it seems since we were expressing functions as 'button / 10.0' rather than 'button / 10', and it thus required recognition of the equality of 10 and 10.0.

On the other hand, Liam's description of 0.01 as "Its [one] divided by a hundred." was also common. This articulation was also rooted in the medium. Rather than using the conventional description of 0.01 as 'one hundredth', it was expressed as "One divided by a hundred." Like many of the other students, Liam appeared to be making direct links between the decimal '0.01' and the function's expression 'button / 100.0' which generated it. In this sense, the function machine gave him a way of conferring an action-based meaning onto decimals comparable to the meanings they already attached to certain fractions (chapter 7).

#### Episode 2 – Constructing meaning for Decimals:

This episode has been selected to show how Naomi's initial attempts to link whole numbers and decimals were somewhat confused and how, in interacting with her Boxer function machine, she began to construct meaning for decimal quantities.

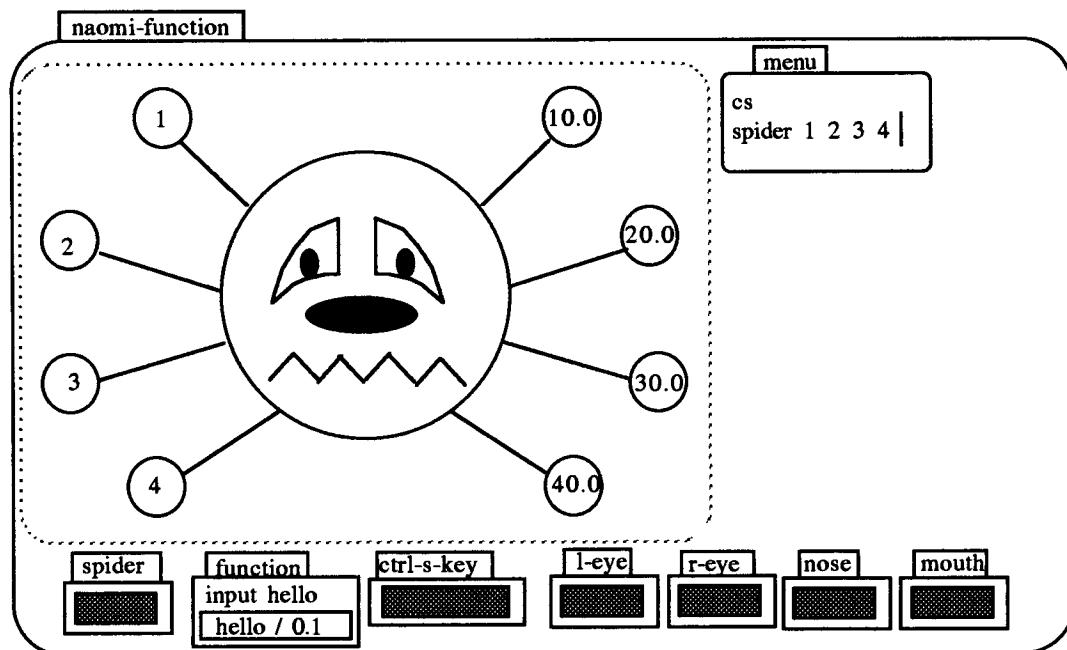


Figure 9.27: Approaching Decimals through division with Naomi's Function Machine

I began by setting Naomi the task of finding a function, that would map 1, 2, 3, and 4 to 0.1, 0.2, 0.3, and 0.4 respectively.

Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Suppose over here that I wanted nought point one, nought point two, nought point three and nought point four, could you find a function that might do that?	Points to the output legs of the spider.
Naomi:	Nought point one, nought point two, nought point three ... [starts to change function]	She changes the 'function' doit box to 'hello / 0.10' and gets out 10.0, 20.0, 30.0 and 40.0.
Interviewer:	What do you think happened?	
Naomi:	I put ... No wait a minute. Let me try again.	She now changes the 'function' doit box to 'hello / 0.1'.
Naomi:	How did it manage to do that again?	Refers to the same output numbers
Naomi:	Ah I know. Point oh one.	She changes the 'function' doit box to 'hello / 0.01'
Naomi:	Oh gosh. It's getting worse.	
Naomi:	Wait a minute! I know. It's probably going to get even worse.	Changes 'function' to 'hello / 10' and then to 'hello / 10.0'.

In this episode, Naomi's efforts to find a function to map 1, 2, 3, 4 to 0.1, 0.2, 0.3, 0.4 led her to divide by 0.10, then 0.1, then 0.01 and finally 10. Her first two attempts yielded outputs of 10.0, 20.0, 30.0 and 40.0, and reveal her puzzlement at obtaining the same answers for 0.10 and 0.1. Naomi clearly thought that these were different quantities at this point. Her next attempt using 0.01 produced a 'worse' answer, but nevertheless seemed to provide her with a clues to her final – and correct – strategy of dividing by 10.

The next extract show how Naomi was asked not merely to find a function which would map 1, 2, 3, 4 to 0.01, 0.02, 0.03, 0.04 but to justify it as if explaining to a younger child.

Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	Could you change your function so that what you get out is nought point nought one, nought point nought two, nought point nought three and nought point nought four?	The input numbers were 1, 2, 3, 4.

Naomi:	Maybe.	She changed the function to 'hello / 100.0' and ran 'spider 1 2 3 4'.
Interviewer:	Do you understand what has happened?	
Naomi:	Yes, because a tenth is one tenth of a hundredth [stops here as she realises her account is garbled].	[She may be trying to say that one tenth of a tenth is a hundredth].
Interviewer:	Suppose you were trying to explain to someone what your machine does, what would you say?	
Naomi:	It's dividing by one hundred.	
Interviewer:	Suppose someone – maybe in a younger class – came along to you and said "What do those numbers mean?"	Points to 0.01, 0.02, 0.03, 0.04 in the spider's output legs.
Naomi:	These are whole numbers and these are zero point one and [stops] point zero one [hesitates]. Part of the number is whole [stops]. There's like a tiny little bit at the end of the number. It's hardly a number at all.	Points to left hand legs.
Naomi:	I know. One hundredth.	Points to output leg of the spider with 0.01 in it.

Naomi's initial articulation : "There's like a tiny little bit at the end of the number. It's hardly a number at all." typifies the struggle which young students have in conceptualising decimal quantities. Her later identification of 0.01, 0.02, 0.03, and 0.04 with one hundredth, two hundredths, three hundredths etc. seemed to have been directly related to her interactions with the function machine. Moreover, it marked a key instance in her developing sense of decimal numbers. From this point on, she seemed to be able to generalise her discovery.

#### Extract 3:

Speaker	Dialogue	Commentary
Interviewer:	Suppose I said I wanted one thousandth, two thousandths, three thousandths, and four thousandths, but as decimals, could you do that?	Points in turn to 0.01, 0.02, 0.03, 0.04 in the output legs of the spider.
Naomi:	You'd just have to add another nought onto there.	Points to the line 'hello/100.0' in the function doit box and changes it to 'hello/1000.0'
Naomi:	You get more and more noughts after there. If you wanted to get a millionth, you'd divide it by a million.	Points to output numbers 0.001, 0.002, 0.003, 0.004.

These extracts illustrate how Naomi set about creating functions to map 1, 2, 3, and 4 to decimals such as 0.01, 0.02, 0.03, and 0.04. To do so, she needed to go inside the program ‘function’ doit box. Initially, her choices led her in the opposite direction. However, from the trace on the machine’s graphic, she was able to create functions that not only gave her the desired mapping, but also gave her an operational means of generalising the relationship between whole numbers and decimals. The Boxer formalisation seems to have structured Naomi’s expression of decimals in that she articulated them in terms of operation in the doit box.

### 9.3.2.2: Multiplication as the Inverse of Division

Although I have separated out division and multiplication for the purpose of analysis, in the Boxer situation, one usually followed the other. Each of the following episodes came immediately after the students had succeeded in using division to map whole numbers to decimals. I then reversed the formulation by asking them to find a function that would map decimal to whole numbers.

#### Episode 1 – Linking multiplication and division through reverse patterns

This episode illustrates how Sara and Natalie related the inverse nature of multiplication and division through a reverse pattern in the way the Boxer graphic displayed decimal expressions.

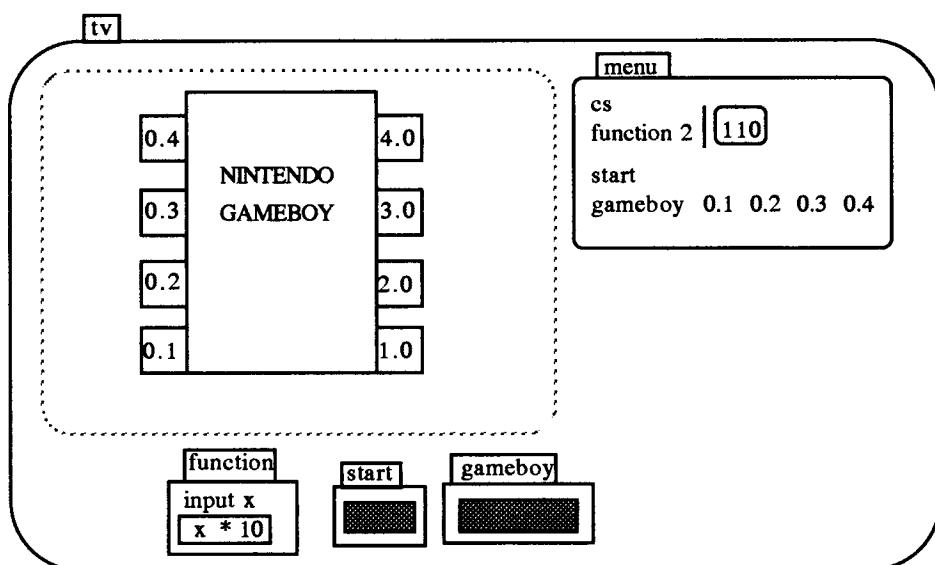


Figure 9.28: Relating Decimals through inverse functions with Sara and Natalie’s Function Machine

Following an episode in which I asked the girls about the relationship between whole numbers and decimals, and which involved the use of division, I now asked the girls about functions involving multiplication.

Extract 1:

Speaker	Dialogue	Commentary
Interviewer:	Let's change those numbers to nought point one, nought point two, nought point three, nought point four. Change the function to times ten.	Sara changes the inputs to 0.1, 0.2, 0.3, 0.4, and the function to ' $x * 10$ '.
Natalie:	Nought point forty, nought point thirty. [Anticipates question]	Points to output boxes on screen.
Interviewer:	Try it and see	
Sara:	Backwards.	Refers to pattern: 0.1 --> 1.0, 0.2--> 2.0, 0.3 --> 3.0, 0.4 --> 4.0
Interviewer:	Backwards. Why do think that is?	
Sara:	You changed it round there.	Points to the menu box, where the inputs of 1, 2, 3, 4 have been changed to 0.1, 0.2, 0.3, 0.4.

My motivation, in reversing the operations along with the input/output numbers, was to investigate whether the function machine would pre-dispose Sara and Natalie to seeing the connection between division and multiplication. The girl's identification of a 'backwards' pattern of the form  $0.1 \rightarrow 1.0, 0.2 \rightarrow 2.0, 0.3 \rightarrow 3.0, 0.4 \rightarrow 4.0$  was something I was not even aware of (Figure 9.28). But, it was probably the most obvious visual connection between the numbers. My focus was on the functions as objects. Their focus was on the functions as processes. The next extract reinforces this contention.

Extract 2:

Speaker	Dialogue	Commentary
Interviewer:	Any idea what will happen?	Changes the inputs to 0.01, 0.02, 0.03, 0.04 and the function to $x \rightarrow x * 100$ .
Natalie:	That will be there. That will be there. That will be there.	Points to inputs and output boxes, and indicates that the numbers will be reversed.
Interviewer:	Okay. You are saying that they're all going to be reversed. Why did you say that Natalie?	They run it and see it works.
Natalie:	Because when we did it before, that's what happened.	Refers to the earlier operations of division and multiplication by 10.

Interviewer:	Suppose someone said to you 'You times point oh one by a hundred and you get out one. Why is that?	
Natalie:	Because it's underneath. It's a hundred underneath one.	

The students again identified a ‘backward’ pattern in multiplying by 100, i.e., they saw the links 0.01 --> 1.00, 0.02--> 2.00, 0.03 --> 3.00, 0.04 --> 4.00. In fact, this pattern arose because of the way the prototype version of Boxer handled decimals at that time. As pointed out in chapter 2, visualisation can be misleading as well as an aid to learning, and in this case, the visual feedback seemed to be pre-disposing them towards a syntactical rule. On the other hand, Natalie’s explanation: “Because it’s underneath. It’s a hundred underneath one.” to account for the effect of multiplying 0.01 by 100 suggested an operational rationale. Her expression indicated that she was relating 0.01 back to the Boxer formalisation ‘ $x / 100.0$ ’ inside the ‘function’ doit box.

### Episode 2 – Inverse functions as a way of conferring meaning

This episode illustrates how in the Boxer function machine context, Liam and Jason were able to attribute meaning to decimals by relating them to the inverse nature of multiplication and division.

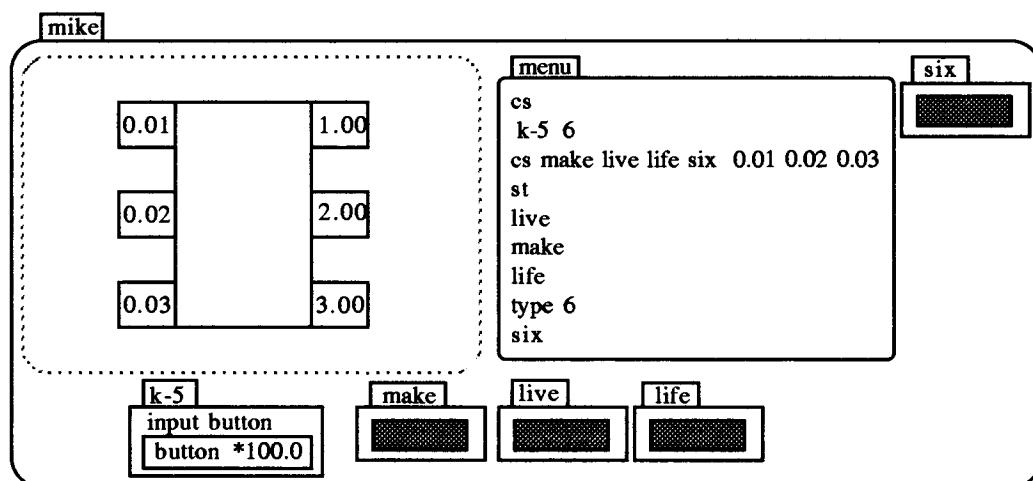


Figure 9.29: Relating Decimals through inverse functions with Liam and Jason’s Function Machine

In using their function machine to express division, Liam and Jason like several other students discovered the so-called ‘backward’ rule. One way of probing whether they were simply acquiring a syntactical rule was by reversing the question formulation. I

now tried this. Liam and Jason were challenged to construct a function which, with inputs of 0.01, 0.02, and 0.03, gave outputs of 1, 2 and 3.

Extract 1:

Name	Dialogue	Commentary
Interviewer:	Suppose instead of having nought point nought one coming out, we had nought point nought one going in ...	Points from output to input boxes on number machine
Jason:	... and, you had 1.00 coming out.	Also points to number machine.
Interviewer:	That's what I was going to ask you. I was going to ask you to try and get one, two and three coming out.	
Jason:	Change that.	Points inside 'k-5' to the function: 'button / 100.0'
Both:	Change inputs to 0.01, 0.02 and 0.03, change the function inside 'k-5' from 'button / 100.0' to the function 'button * 100.0' and run it.	Here, the boys are 'running ahead' of me!

There are different ways that this episode could be interpreted. It may have been that, by pointing at the numbers 0.01, 0.02 and 0.03 in the output boxes and then saying I wanted them in the input boxes, I had predisposed the boys to reversing the operation. On the other hand, I didn't expect them to immediately know that multiplying by a hundred was the inverse of dividing by a hundred. It is, of course, possible to find a function to map 0.01, 0.02 and 0.03 onto 1, 2 and 3 without knowing this. However, the evidence does seem to point to the Boxer function doit box 'k-5' as being critical in making this link.

Extract 2:

Name	Dialogue	Commentary
Interviewer:	Suppose Denise came back and said "Right, Liam and Jason. Why is a hundred times nought point oh one equal to one?"	Denise was the year 6 teacher.
Liam:	Because nought point oh one is a hundredth of one. So if you times nought point oh one by a hundred, you just get one.	Inverse relationship.
Interviewer:	What about the next one?	
Liam:	If you divide two by a hundred, you'll get nought point oh two.	Note the rephrasing.

The phrasing in Liam's first utterance suggests that the Boxer function machine together with the Boxer function 'k-5' have given him a means of ascribing meaning

to decimals such as 0.01. A ‘hundredth of one’ contains an action-based derivation unlike say ‘one hundredth’. His later expressions “So if you times it by a hundred, you just get one.” and “If you divide two by a hundred, you’ll get nought point oh two.” suggest that his situated knowledge is underpinned by a functional connection in each direction, i.e., from whole numbers to decimals and from decimals to whole numbers.

In interacting with the Boxer function machine, the students had used the function doit box as a way of mapping whole numbers to decimals and vice versa. It would appear that in doing so, they had begun to recognise the inverse relationship between these two operations, and this gave them the explanatory power for describing the number concepts themselves.

## **9.4: Conclusions and Implications for Research**

This chapter has analysed the Boxer function machine study. It has shown how a project that had previously been undertaken with conventional classroom resources, was greatly enriched by its re-construction in the medium of Boxer. It has demonstrated how students deployed programming ideas introduced in an earlier phase of the research and developed them in ways that differed from what is known on the basis of existing Logo-based research. It has also shown how the Boxer function machine representation mediated students' expressions of number in ways that were significantly different from pen-and-paper representations.

### **9.4.1: Programming Findings**

In one way, the data supports previous Logo research, but in other ways, it is radically different. It supports Hoyles and Sutherland's (1989) finding that students were more likely to employ structured programming when engaged in well-defined abstract goals. Likewise, it supports their contention that the use of such programming techniques was directly related to their usefulness as problem solving tools.

On the other hand, the results of the function machine study extends and challenges those findings in the context of Boxer. It suggests that the uptake of structured programming by the Boxer students stemmed not only from the abstract nature of the goal, but also from the perceived complexity of the goal. It provides evidence that the Boxer students were far more likely to employ sub-procedures and modular programming than their Logo counterparts, and it suggests the more sophisticated

programming employed by these students emanated from the structure of Boxer itself. Modular programming and super-procedures appeared to be used more frequently in Boxer than Logo because the design of Boxer (editor and workspace combined; line execution and box execution; boxes as containers for programs; naming of boxes etc.) made them more effective problem solving tools.

### Modularity

In Logo, modularity is frequently contrasted to 'direct drive'. Defining a procedure implies waiting until it is finished in order to test it as a whole, the advantage being that if it does not work, there is no need to write all of it again. On the other hand, it usually means entering an editor mode and thus, to see the result of executing each line as it is written, would necessitate an inordinate (and confusing) degree of movement between modes. Not surprisingly, many students go for instant feedback by staying in the workspace and running their program line-by-line. The problem with this form of programming, of course, is that if something should go wrong, the programmer must begin again, i.e., there is no simple way of collecting together these commands to form a procedure.

In Boxer, the distinction between editor mode and workspace is removed. The system is always in 'editor' mode. Programs can be executed line-by-line and then combined to form a procedure. In this research, these Boxer facilities directly influenced the way the students programmed. The students frequently adopted line-by-line execution as a way of programming, but, because these lines were inside doit boxes, they were being combined into procedures as they were being written. So, the students were getting the best of both worlds: the instant feedback of direct programming together with procedures that collected all that had been done so far.

Of course, there is more to modularity than its opposition to direct drive. One of these is the notion that if modules are independent, they can be used elsewhere. Another is the advantage of this form of programming as a means of locating bugs. Virtually all of the students created modules in the course of their function machine programming. The Boxer doit box as a visual container for programs together with Boxer's manner of naming programs (a name-tag attached to the box) led to students identifying programs with different parts of the function machine drawings (e.g., Naomi's 'l-eye', 'r-eye', 'nose' etc., Liam and Jason's 'make', 'live', 'life' etc.). But, these were not necessarily completely independent.

The students' use of line-by-line and box execution allowed them to overcome bugs as the program was being written, rather than afterwards. But, in some ways, this mitigated against their adoption of a strictly modular way of programming, since they were able to overcome the disadvantages of including 'navigation' commands within modules (e.g., Courtney and Laura) by running the box again and altering the navigation commands. It would probably be truer to say that the students used a semi-modular way of programming.

### Sub- and super-procedures

One of the advantages of using sub- and super-procedures is that it permits the construction of complex programs out of simpler ones. This seemed to have been appreciated by the students in this study, but the extent to which they made use of these techniques again depended on their perceived complexity of the goals. Joanna and Nico, perhaps the most accomplished of these novice programmers, drew the whole of their function machine with a single procedure 'rect'. It seemed that for them the task did not warrant the need for structured programming.

Most students however, viewed super-procedures as a 'correct-up-to-here' facility, and employed them throughout their construction. Certain distinct approaches were observed. Super-procedures, either in the form of un-named doit or named doit boxes, were built up by running modules along with 'joining' commands from within the box. Bugs were overcome by correcting the last line run and then executing the whole super-procedure in situ, in the case of the un-named doit box (Figure 9.30a); or from the menu, in the case of the named doit box. Rosie and Chloe, Sara and Natalie, and Naomi programmed like this. Some students (Sara and Natalie for example) placed their sub-procedure doit boxes inside the super-procedure doit boxes (Figure 9.30b), but most students chose not to. The disadvantage of programming like this was that it used up a lot of space down the screen. The advantage was that the programs were 'local' to that super-procedure. It seemed that the students appreciated the former but not the latter.

Other students used the menu as a kind of half-way house in that they built up super-procedures by running each module from there. Courtney and Laura, and Liam and Jason worked like this. In the case of the two girls, the modules were combined in a named super-procedure only at the very end. This was done as a labour saving device as they tired of executing the modules on-by-one from the menu (Figure 9.30c). The two boys however combined the modules by successively joining them on a single

line in the menu (Figure 9.30d). At first, they had ‘cs make’, then once ‘live’ was tested, ‘cs make live’ and so on. Thus, it can be seen that Boxer permitted a flexible approach to building super-procedures.

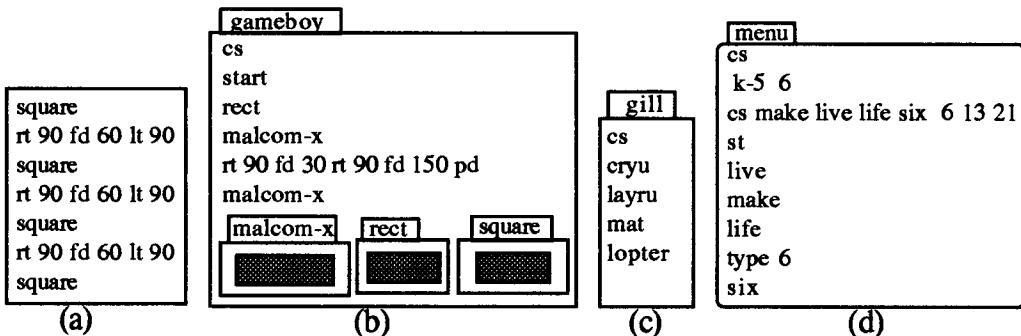


Figure 9.30: Ways of constructing super-procedures in Boxer

### Variables and Functions

Variables played a significant role in the function machine study, both in terms of making the graphic functional and in the subsequent activities. Previous research (Noss, 1985; Sutherland, 1988; Hillel, 1992) had shown how the concept of variable can derive meaning from its use with Logo. However, even within Logo settings, there are degrees of abstraction. At one level, variables may ‘stand for’ the side of a square, or the angle of a turn. These have physical referents. The variables in the function machine study were more abstract entities in that they stood for inputs to functions, and as such they were much closer to the notion of variable as used in algebra generally.

With the cardboard models, the students did not express their functions in algebraic form, but rather in ‘rhetorical’ fashion (Boyer, 1968). For example, Sara’s function was originally expressed as a “times 8 add 94” machine. In re-creating their function machines in Boxer, they needed to re-express their informal expressions into the formal language of Boxer. This resulted in expressions which appeared non-standard in comparison to pen-and-paper algebra. Sara now expressed her function as ‘[number \* 8] + 94<sup>2</sup> while Nico’s “Take away two lots of what you’ve got.” was expressed as ‘x – 2 \* x’ and Jason’s “Times one and take away three” was re-stated as ‘[1 \* x] - 3’. The medium enabled them to articulate their informal expressions.

---

<sup>2</sup> It is interesting to relate that this ‘syncopated algebra’ (Boyer, op cit.) also mirrors an intermediate stage in the historical evolution of algebra, between the rhetorical and the symbolic, in which words and symbols were combined.

Moreover, these Boxer expressions constituted an *operational algebra*: they could run them to check that they worked.

Unlike earlier Logo studies (Leron and Zazkis, 1986; Sutherland, 1988), functions and variables were not the subject of the research per se. Nevertheless, it can be claimed that through using functions in their study of numbers, students were picking up many of the essential properties of the function concept. Within Boxer, they were able to detach signified from signifier. Different names for variables were used ('button', 'number' etc.) and different names for the function itself ('function', 'k-5' etc.). The equivalence of functions was recognised and there was some linking of functions with their inverses. The range of numbers which the function operated on was also extended to include decimals and directed numbers.

#### **9.4.2: Number Findings**

The interplay between graphic and program was intimately connected. Both were visual, and both were close to each other in a physical as well as a metaphoric sense. At times, we found ourselves using the description 'function machine' to refer to the graphic, yet at other times, using it to refer to the program. The content, pedagogy, and the students' construction of meaning were all structured by this interplay.

The content was not 'directed numbers' nor 'decimals' much less sub-categories of these such as 'adding directed numbers' or 'multiplying decimals' as is often the case. Rather, the subject of enquiry was the function machine itself and how it behaved over a range of numbers. I did not teach nor ask students to subtract or add directed numbers nor did I ask them to multiply or divide decimals. Instead, I invited them to find functions that mapped one set of numbers to another, or I asked them to predict the effects of applying certain functions to numbers. Moreover, they were asked to account for that behaviour.

The students' ability to work proficiently within the medium with directed numbers and decimals was in marked contrast to research findings with traditional media (chapter 2). The findings reported in this chapter were selected to illustrate how the function machine supported their construction of meaning. In doing so, I have chosen a worse-case scenario, i.e., the examples show what happened when things went wrong, when meaning was not clear etc. As stated above, I adopted a holistic approach to number, but for clarity of analysis, the findings are presented under the headings of directed numbers and decimals.

### Natural Numbers

The data shows that the students initially checked the operation of the Boxer function with reference to the numbers on their cardboard models, which in most cases were natural numbers. My early interventions were aimed at probing whether they were aware that the Boxer model would operate over a range of numbers and not just the three or so numbers etched on their cardboard model. It was clear that in some cases there was uncertainty about this.

### Directed Numbers

Surprisingly, the students experienced few difficulties with the *idea* of negative numbers, i.e., they seemed to consider negative numbers a natural consequence of subtracting a larger number from a smaller one. This, if it is a general finding, has interesting consequences for the teaching of subtraction since in the common algorithmic approach, young students are frequently admonished “You can’t take away a larger number from a smaller one.”

However, subtracting and adding negative numbers produced mixed results, with initial responses which were often in the wrong direction. This was perhaps exacerbated by the tendency (by researcher and students) to say the negative numbers in descending order, i.e., -1, -2, -3 etc. Whereas, in the case of subtracting a larger number from a smaller one, students derived support from the formalisation in the function doit box ‘number – 4’ etc., in the case of subtracting and adding negative numbers, they appeared to be generating syntactical rules from the trace on the graphics of the machine. These rules were correct (equivalent to  $-a - b = -(a + b)$  and  $-a + b = b - a$ ), but they were situated in the medium.

Perhaps, of equal significance, was the way that the function machine structured the discourse around concepts and not simply algorithms. The focus was on the operation of the function machine rather than the numbers per se. “You can’t go under nought by dividing it.” was one instance of this. Another involved the successive trials of the functions ‘ $1x - 3$ ’ and ‘ $0x - 3$ ’ in order to map  $-1, -2, -3$  to  $2, 1, 0$  respectively.

### Decimals

In general, the students did not find the notion of decimals as a consequence of division as readily comprehensible as they had directed numbers as a result of subtraction. This could be deduced from their first responses to challenges inviting them to change their function machine so that it mapped whole numbers to decimals.

Frequently, this evoked incorrect choices of operation or operators. However, it was also clear that these were first responses. Within a relatively short time of interacting with the function machine, the students could construct such mappings, the consequences of their actions giving them the means of closing in on the functions.

The Boxer function machine appeared to mediate students expression of decimals in several distinct ways.

(i) In the first case, students gave operational descriptions of the numbers. Examples of this were 0.1 being referred to as a ‘tenth of one’ and 0.01 as ‘a hundredth of one’. These descriptions linked the decimals to the operations within the function machine which brought them into being, and contrast with those used in conventional settings such as ‘one tenth’ and ‘one hundredth’. It suggests that the Boxer function machine activities were indeed enabling students to confer an action-based meaning to these decimals, i.e., of linking operational and conceptual notions. This supported the hypothesis generated from the data of the semi-structured interviews (chapter 7).

(ii) Secondly, for division, students generated a syntactical ‘backward’ rule which related patterns in the decimal expression to those in the operator in the doit box. They observed that when they divided numbers such as 1 by 10, 1 by 100 and so on, the resulting decimals 0.1 and 0.01 looked like the operators 10 and 100 written backwards. This was not as obvious as it appears because I was expressing 10 as 10.0, 100 as 100.0 etc.<sup>3</sup>.

(iii) Thirdly, students directly linked decimals like 0.01 to the operation in the doit box. Examples of this were Natalie’s: “Because it’s underneath. It’s a hundred underneath one.” and Naomi’s “If you wanted to get a millionth, you’d divide it by a million.”

(iv) For multiplication, students also generated a syntactical ‘backward’ rule but in this case it related patterns in input and output boxes on the graphic. Multiplication by 100 mapped 0.01 --> 1.00, 0.02 --> 2.00, and 0.03 --> 3.00. Again, this was a by-product of the way the software forced a specified decimal precision.

---

<sup>3</sup> In Boxer, numbers can be expressed in the form of fractions or decimals. One way of ensuring the latter is to express the operator in decimal form, e.g., 10.0 instead of 10, and this is what I was doing.

(v) The close interplay between decimal expression on the graphics box and the operation in the function doit box seems to have disposed some students to recognise inverse relationship between division and multiplication. Liam's "Because nought point oh one is a hundredth of one. So if you times nought point oh one by a hundred, you just get one." is an example of this.

#### **9.4.3: Implications for the Thesis as a whole**

Hoyles (1993) has observed that software environments differ both in the extent to which mathematical ideas are embedded within them, and in how far software interactions are self-evidently structured by mathematics discourse. The drawing of a function machine could have been done in Logo. Boxer, like Logo, is a functional programming language, that is the underlying model is the mathematical idea of a function. It is therefore possible to represent functions in the programming language in a way which resembles their algebraic formalisation. However, Logo and Boxer differ in how they *represent* functions in their programming code.

The differences lie in Boxer's facility for permitting direct visual manipulation (Hutchins, Hollan and Norman, 1986). In the function machine study, both *articulatory directness* (the closeness of relationship between entities in the domain and their embodiment on the screen) and *semantic directness* (the closeness of relationship between a user's intentions and their expression in the interface language) were evident. This could be seen in the visual manner with which the students and I interacted with the programming code and related this to the drawing, and in the degree that this was explicitly related to the number concepts being investigated. More cogently, it was evident in the operational manner by which the students expressed those concepts.

# Chapter 10: Overall Conclusions and Implications of Research

This thesis has been undertaken to investigate how working in the computational environment of Boxer mediated students' developing sense of number. Following an exploratory study, the major part of this investigation was carried out by means of a longitudinal study, which consisted of a programming, an interview, and a number phase. The longitudinal study was concerned with three issues: the learning of programming, the learning of number, and developing a pedagogical design for integrating the first two issues.

This chapter will synthesise the findings from all three phases of the research by relating them to these issues, consider their limitations, and draw out conclusions of the study. The subsequent section will then weigh the implications for future research.

## 10.1: Programming

The students programmed throughout the longitudinal study, but the emphasis changed as the research progressed. In the programming phase, the focus was on learning to program with number issues being secondary. In the number phase, the position was reversed. Programming was now a means to an end: that of learning about number. The following sections synthesise the programming findings from both phases. As stated in chapter 3, at the time of this study, there was not yet a developed research literature in Boxer, so the Logo literature will often serve as a baseline for these findings.

### 10.1.1: The General Programming Environment

#### 10.1.1.1: The technical language of programming

This study found that the technical language of programming was open to change in Boxer. The spatial metaphor at the heart of Boxer's structure provided a hook for a language to describe that structure as well as its execution. In the former case, 'doit box' could be substituted for procedure, 'inside doit box' for sub-procedure, 'data box' for variable and so on. In the latter case, 'Do the lines' could replace 'Execute the commands' and 'Do the box' could replace 'Run the procedure'. This language had referents on the screen and hence was better able to convey its meaning to students. It

indicates that in Boxer, there is less of a jargon barrier for students to overcome in order to learn to program.

#### 10.1.1.2: Single mode of operation

In Logo, procedural programming is frequently contrasted to ‘direct drive’. This dichotomy of observed behaviour reflects the technical dichotomy of what is possible. Procedural programming generally means changing to editor mode, defining or altering procedural definitions, and then exiting the editor and moving back to the workspace. Not surprisingly, novice programmers get confused about which mode they are in and many choose to execute commands in direct mode despite the disadvantages of doing so (Hoyle and Sutherland, 1989). Boxer effectively removes the dichotomy of modes (and the confusion) by having a single mode of operation.

#### 10.1.1.3: Separating the process and product of commands

The evidence suggests that certain constructs of Boxer help students to discriminate the process from the product of commands. These include having a single mode of operation and having line-by-line execution. Students were quickly able to identify components of programs with actions on the screen and to appreciate the significance of the inputs to different commands. In the programming phase of the longitudinal study, several changes to the presentational mode were found to assist this process. These included presenting a single command to a line, combining this with writing commands inside un-named doit boxes, and creating on-screen challenges which motivated students to attend to state-transparency.

#### 10.1.1.4: State transparency

In Boxer, the practice of grouping commands one-to-a-line within a doit box, can help students’ appreciation of state transparency. In Logo, it is common practice to combine commands for linear and angular motion on the same line, e.g., ‘`fd 50 rt 90`’. Within procedures, there is no good reason for separating them. However, the ubiquity of this practice may conceal one reason why novices encounter problems with state transparency. In the course of this study, the design evolved from grouping commands on a line as above, to grouping commands on a line within a doit box, and finally separating out commands, one to a line, within a doit box.

#### 10.1.1.5: Modelling Structured Programming

The visual environment of Boxer to can be exploited to provide explicit models of structured programming. It is not obvious in Logo how structured programming can be

modelled on the screen. The editor lay-out need not reveal clues to the inter-relationship between different procedures. This fact may influence the practice often adopted in introductory Logo courses of creating a learning hierarchy which places structured programming somewhere down the line. The design employed in this study exploited the visual environment of Boxer to deploy explicit models of structured programming right from the start. The programming gains exhibited by the students indicates that this had beneficial effects.

### **10.1.2: Procedures**

#### 10.1.2.1: Procedure as an entity

In Boxer, the evidence suggests that there is no discontinuity between direct drive and procedural programming. It is thus possible to begin programming with procedures. The point at which students are introduced to procedures in Logo, as Noss (1985) points out, depends very much on the pedagogical design, but both he (*op cit.*) and diSessa (1986d) observe a ‘learning plateau’ between these two forms of programming. The evidence from this study suggests that this did not occur, with students quickly appreciating the notion of a procedure as an entity. The physical form of the doit box as a rectangular container for the program clearly helped in this respect.

#### 10.1.2.2: Naming and procedures

In Boxer, the separation between naming and creating procedures means that students are more likely to adopt procedural programming. In Logo, naming a procedure is synonymous with creating a procedure. In Boxer these can be distinct processes, with procedures in the form of un-named doit boxes forming an intermediate stage between direct execution of single commands and named procedures. The manner of naming procedures, i.e., the ‘name-tab’ boxes, is also quite different in Boxer. These changes – which the research design exploited – resulted in different findings to the Logo research. Firstly, students did not attach undue significance to procedural names. Secondly, procedures (in the form of un-named doit boxes) were created as the norm rather than the exception.

#### 10.1.2.3: Procedures and debugging

In Boxer, debugging can be an on-going rather than a post-hoc activity. All of the students in this study adopted a ‘bottom-up’ style of programming. This did not appear to disadvantage them. As the study progressed, many students adopted a practice of writing and executing lines inside un-named doit boxes. In doing so, they were in

effect in building up procedures – a practice which meant that debugging became an ongoing process rather than a post hoc activity. The observations suggest that debugging was not a major obstacle to programming.

#### 10.1.2.4: Sub- and super-procedures

In Boxer, structured programming was adopted as a practice because it was seen as an effective means of problem solving. One of the advantages of using sub- and super-procedures is that it permits the construction of complex programs out of simpler ones. This seemed to have been appreciated by the students in this study, but the extent to which they made use of these techniques depended on their perceived complexity of the goals: a task perceived as complex appeared to evoke a greater use of super-procedures. Thus structured programming was adopted as a practice because it was seen as an effective means of problem solving, a finding which extends that reported in Logo research by relating it to changes in the technical component of Boxer.

#### 10.1.2.5: Procedures and modularity

In this study, the Boxer facilities of line-by-line execution and executable boxes predisposed students to a semi-modular style of programming. The students frequently adopted line-by-line execution as a way of programming, but, because these lines were inside doit boxes, they were being combined into procedures as they were being written. From one perspective, these changes can be seen to facilitate modular programming techniques, but from another, they can be seen to remove some of the disadvantages of working in a direct way. Virtually all of the students created modules in the course of their function machine programming. But, these were not necessarily completely independent. Students were able to overcome the disadvantages of including navigation commands in modules by means of direct execution.

#### 10.1.2.6: Procedures and problem solving

In Boxer, students employed more structured programming because the structure of the language made their usefulness as problem-solving tools more transparent. The findings support Hoyles and Sutherland's (1989) finding that students were more likely to employ structured programming when engaged in well-defined abstract goals. But, it suggests that the uptake of structured programming by the Boxer students stemmed not only from the abstract nature of the goal, but also from the perceived complexity of the goal. Thus, if students perceived a goal as being made up of several parts, it was more likely that they would create separate procedures for those parts.

Likewise, it supports their contention that the use of such programming techniques relates to their usefulness as problem solving tools. However, the evidence points to the Boxer students employing more structured programming than their Logo counterparts, and it suggests that the use of more sophisticated programming techniques emanated from the structure of Boxer itself. Students used procedural and modular techniques because the visual structure of the language (boxes as containers for programs, name tags for boxes etc.) made their usefulness as problem-solving tools more transparent.

### **10.1.3: Variables**

#### 10.1.3.1: Declaring variables

In Boxer, syntactical changes to the way variables are depicted helps the students' understanding of variables. The manner of creating variables in Boxer is quite different to that in Logo. Instead of syntactical markers such as the colon (:) or apostrophe ('), variables are declared either by means of the 'input' command or by the use of named data boxes. The evidence suggests that the physical appearance of variable data boxes, the manner of naming the (name-tab boxes), the means of changing variable values (change the contents of the data box), and the research pedagogy which employed variables in a wide range of contexts, all influenced the students' understanding of variables. They appreciated the concept of a variable as representing a range of numbers. They also appeared to understand that different variable names could depict the same quantity, and that the same variable name could signify different quantities.

#### 10.1.3.2: Introducing variables

The constructs of Boxer are such that it makes pedagogical sense to introduce variables through the 'input' line from the first time that procedures are used. In Logo, it is common practice to introduce commands first with specific values and later with variables. In 'First-Boxer', I replicated this practice. However, by the time of 'Second-Boxer', the possibility of always using declared variables occurred to me. Local variables in the form of internal data boxes with names such as 'side', 'angle' made this an alternative viable practice. At the time of this research, there did not seem to be a clear (visible) link between variables as represented by internal data boxes and the more general variable introduced by means of the 'input' command. In current versions of Boxer, this link is made more explicit. Pressing 'doit' on the line 'input x' replaces the letter 'x' with an internal data box named 'x'. Values can then be placed in this box and run internally thus reinforcing the feeling of direct manipulation. In view

of the earlier finding, it would now make pedagogical sense to introduce variables through the ‘input’ line from the first time that procedures are used.

#### **10.1.3.2: Variables and functions**

The concrete nature of variables and functions help students understand these concepts, but also make it feasible to deploy these algebraic devices as a way of shedding light on other areas. Previous research (Noss, 1985; Sutherland, 1988) had shown how the concept of variable can derive meaning from its use with Logo. This study extends that finding by showing not only how variables can derive meaning from their use with Boxer, but also how numbers can derive meaning through the use of variables. The transparent nature of variables in Boxer enabled students, to use Mason’s (1988) expression, to see through the particular to the general.

In the function machine study especially, variables were used in the context of functions and together, function and variable, played a significant part in enabling students to confer meaning on directed numbers and decimals. At the same time, students were building up their notion of the function concept itself. This process was greatly assisted by the concrete manner in which functions are expressed in Boxer.

#### **10.1.4: Discussion of Programming Issues**

This thesis was concerned with how students’ developing sense of number might be mediated through working in the computational environment of Boxer. Learning to program was thus a means for preparing the ground for the learning of number. In that sense, it is located in existing research (Noss, 1985; Hoyles and Noss, 1993b) which has concentrated on the creation of Logo mathematical environments. This study indicates that Boxer is an even more effective means for creating mathematical learning environments in the classroom.

It owes its effectiveness to the structure of the medium, particularly its visual structure. This research shows that over the two year period of the longitudinal study, students were able to move from a position of knowing no programming to one in which they were able to construct their own microworlds. This is all the more impressive if the limitations are borne in mind. The fact that, as a part-time researcher, I could visit the school on only one day a week combined with the fact that there was only one computer in the class meant that each pair of students had a gap of between two and four weeks between one Boxer session and the next. Despite these conditions, the important conclusion is that students learned to program with a relatively high degree of structure.

Assessing the impact of IT in schools, Hammond (1994) observes how the authors of two major research projects, the Plait report (Gardner et al., 1992) and the ImpacT report (Watson, 1993), both attribute the failure of teachers and pupils to fully exploit software to: "... their lack of knowledge of the programs or the principles behind them." (Hammond, op cit., p. 256 and p. 257). Seen from this perspective, a research design in which students program their own software to pursue their own goals ensures both knowledge of, and principles behind, that design. An important conclusion of this study is that Boxer makes this possible for 10 to 11 year old students.

Noss (1985, p. 33) observes that: "It is the language which determines the nature of the interaction between learner (programmer), machine and the ideas being programmed." and he argues that the essence of the case for learning to program rests on the process rather than the product. Gray and Tall (1994), in discussing the use of software to learn mathematics, write: "... we have evidence that the use of the computer to carry out the process, thus enabling the learner to concentrate on the product, significantly improves the learning experience." (p. 137). In the past, these were necessarily discrete domains for the novice programmer.

The results of the present research suggest that this need no longer be the case. My number studies indicate that Boxer makes it possible for young students to program their own interactive software, and in the process gain the kind of mathematical insight of the product that was previously the prerogative of the professional programmer. As Dubinsky (1991) writes:

... it seems that if a student implements a process on a computer, using software that does not introduce programming distractions (such as complex syntax, constructs that do not relate to mathematical ideas etc.), then the student will, as a result of the work with computers, tend to interiorize the process. If that same process, once implemented, can be treated on the computer as an object on which operations can be performed, then the student is likely to encapsulate the process. (1991, p. 123).

My claims are twofold: firstly, that the design of the two number studies corresponds to Dubinsky's description and secondly that, in these microworlds, programming was the operational algebra which enabled them to make connections between conceptual and procedural notions of number.

## 10.2: Number

In the number phase of the research, the students constructed two computational objects: the Boxer number line and the Boxer function machine and each mediated their sense of number in ways that related to the programming code which they had built into these objects. In each case, the construction process seems to have been critical in terms of the students' ability to identify and manipulate the program's parameters. Familiarity with *their* number line or *their* function machine ensured that attention (rather than being diverted on matters of syntax or such like) was focused through the computational object on to the mathematics which its construction embodied. The following sections summarise the number findings which emerged from the students' interactions.

### 10.2.1: The Boxer Number Line Object

#### 10.2.1.1: Visualising natural numbers

Their construction of a number line in Boxer enabled students to gain new insight into familiar sets of numbers. The data shows that the students initially utilised the Boxer number line to express familiar sets of numbers such as their multiplication tables. Expressing numbers symbolically in Boxer mediated their reading of the visual. They – and I – became aware that there were subtle differences between the number line representation of their tables and the usual non visual depiction of them. Thus a seven times table, beginning at 0 and ending at 84 had 13 rather than 12 numbers printed on the screen. It appeared to me that the number line was modelling their intuitive notion of multiplication as repeated addition. It appeared to them that, in exploiting the number line to represent their tables, they were somehow cheating, a view which perhaps again highlights the uncertain status of visual reasoning in mathematics.

#### 10.2.1.2: Visualising directed numbers

The Boxer number line was used to explore only one aspect of directed numbers, namely the effect of beginning with a negative number and then repeatedly adding positive whole numbers to it. Running the program generated the numbers term-by-term along the number line, so providing a visual model of the relationship between the successive numbers. Few of the children found difficulties with this. Directed numbers was not reified into a separate topic area. Rather, the activity was contiguous with their earlier work with natural number sequences.

### 10.2.1.3: Visualising decimals

The Boxer number line appeared to mediate students' expression of decimals in three distinct ways.

In the first case, it seemed that through manipulating its parameters, students were disposed to reason in particular ways. Decimals were conceptualised as small increments and their sizes related initially to whole numbers or to other (previously tested) decimal increments. Their construction of the number line meant that they were modelling the multiplication of decimals by repeated addition, and this gave them a means of overcoming the 'multiplication makes bigger' misconception.

Secondly, their construction of the Boxer number line seemed to pre-dispose the students to the existence of a number continuum. Evidence for this could be seen in students' remarks which suggested their difficulties was not with existence but rather with articulation. It could also be inferred from the invention of a notation by two of the four case study pairs to express these quantities. Finally, it could be seen in the students' desire to attain greater degrees of accuracy than the problem required.

Thirdly, the programming precision afforded by the Boxer number line appeared to be also promoting conceptual precision. It was no more difficult to program the number line in steps of say 0.01 rather than steps of 0.1 for example, whereas conceptually, these are some distance apart. Evidence for this could be seen in the way that students constructed new solutions on the basis of previous ones. There was, in fact, a dialectic between data and conjecture, which motivated students to adopt sophisticated modes of thinking. Students were able to program a number line to do what they wanted it to do. This gave them power over decimals so they could directly manipulate and visualise numbers with three and more decimal places with an ease that would not have been possible in conventional media. It then became an intellectual challenge for students to seek greater degrees of accuracy than the task required.

### 10.2.1.4: Limitations of the number line object

The number line object represented my first attempt to engage students in the construction of their own microworld. It therefore needed to be simple to program. As the number investigations got under way, it became clear that whilst the relative simplicity of the program was a strength in terms of its transparency, it imposed other limitations. For example, in generating numbers across the screen, the numbers were

typed from left to right. This meant that if the program line which effected the incrementation was of the form ‘change num [num – 1]’, then the numbers on the screen line would be in the opposite direction to the usual convention, i.e., negative numbers would be off to the right.

One response to this might have been to say it did not matter. In an earlier experiment with an on-screen counter (chapter 6), children were not perplexed by sequences of numbers which descended from left to right. Another response might have been to rewrite the program so that it took account of both the starting number and the operation being used. In the event, I chose not to investigate the subtraction of directed numbers with the Boxer number line, as I felt that representing directed numbers in a way that conflicted with their usual classroom representation might be potentially confusing. I rejected the latter option as I felt that the programming required to carry it out (almost certainly involving conditional statements) was not within the students’ capacity at that time. I also rejected the option of writing such a program myself, as I felt this would trade programming power for programming transparency.

The direct consequences of these programming decisions was that the number line investigations with directed numbers were concerned only with their behaviour under addition. Since few of the children found this behaviour problematic, I tended to explore those elements of number that did cause conceptual difficulties, namely decimals. The term ‘decimals’ has come to reify certain distinct if related topics (chapter 2). Many of these, e.g., multiplication of two decimals, multiplying by powers of ten etc., were not touched upon. As stated earlier, my focus was on students’ developing sense of number, i.e., on their *conceptions* of decimals. My investigations were thus deliberately limited to exploring how the Boxer object of the number line mediated those conceptions.

### **10.2.2: The Boxer Function Machine Object**

#### 10.2.2.1: Visualising natural numbers

In programming a function machine in Boxer, students were extending an earlier classroom project: that of constructing cardboard models of functions. These cardboard models displayed input and output numbers, which in most cases were natural numbers. The data shows that in every case, the students began by checking the correspondence between the numbers on the Boxer function machine with those on their earlier models. It was clear that in some cases, students’ conceptions of functions

were tied to the numbers on their models. Their early interactions with their computational function machines were instrumental in making them aware that their function machine could operate over a wider domain.

#### 10.2.2.2: Visualising directed numbers

Few of the Boxer students experienced difficulties with the existence of negative numbers. In this case, they derived support from the formalisation in the function doit box, and the trace down the function machine graphic. Negative numbers were seen as a natural consequence of subtracting a larger positive whole number from a smaller one.

However, several students encountered procedural difficulties with both subtraction and addition of negative quantities, i.e., when the input numbers to the function machines were negative. Initial predictions were often in the wrong direction. This was perhaps exacerbated by our tendency (researcher and students) to express negative numbers in descending order, i.e., -1, -2, -3 etc. Frequently, the students appeared to be generating syntactical rules from the trace on the graphics of the function machine. Sara's observation of the effect of the function ' $x - 10$ ' on inputs of -4, -3, -2 and -1 "Because they're both minuses, you like add them." was typical. These rules were often correct, the above example being equivalent to  $-a - b = -(a + b)$ , but it was also apparent that they were situated in the medium. The Boxer function machine gave students a way of overcoming procedural difficulties with both subtraction and addition of negative quantities.

#### 10.2.2.3: Visualising decimals

Whereas the idea of subtraction as a way of introducing directed numbers was generally borne out, the parallel idea that division might be a way of introducing decimals did not seem to be as intuitive. Initial responses to challenges often evoked incorrect choices of operations or numbers. However, it was also clear that in the course of interacting with the function machine, the students began to construct meaning for decimals in terms of their functional relationship with whole numbers.

The Boxer function machine appeared to mediate students' expression of decimals in several distinct ways.

- (i) In the first case, students gave operational descriptions of the numbers. Examples of this were 0.2 being referred to as a 'tenth of two' and 0.01 as 'a hundredth of one'. These descriptions linked the decimals to the operations within the function machine

which brought them into being, and contrast with those used in conventional settings such as ‘one tenth’ and ‘one hundredth’. It suggested that the Boxer function machine activities were enabling students to confer an action-based meaning to these quantities.

- (ii) Secondly, students directly linked decimals like 0.01 to the operation in the doit box, e.g., “It’s a hundred underneath one.” It would appear that whereas the rationale in (i) above might be called a ‘post-operational’ description, this might be termed a ‘pre-operational’ explanation. In both cases, meaning was derived from the operations.
- (iii) Thirdly, students generated a syntactical ‘backward’ rule which related patterns in the decimal expression to those in the operator in the doit box. For division, they observed that when they divided numbers such as 1 by 10, 1 by 100 and so on, the resulting decimals 0.1 and 0.01 looked like the operators 10 and 100 written backwards. For multiplication, students also generated a syntactical ‘backward’ rule but in this case it related patterns in input and output boxes on the graphic. Multiplication by 100 mapped 0.01 --> 1.00, 0.02 --> 2.00, and 0.03 --> 3.00. This was a by-product of the way the software forced a specified decimal precision.
- (iv) Finally, there was evidence that the recognition by some students of the inverse relationship between division and multiplication was related to the close interplay between the decimal expression on the graphics box and the operation in the function doit box.

#### 10.2.2.4: Limitations of the function machine object

Perhaps, the first limitation that needs to be acknowledged concerns the extent of my intervention. In particular, I intervened to help most of the students with the use of variables in the latter part of the construction process. My interventions tended to be inversely proportional to the students' knowledge. For the non-case study students in particular, I intervened when it appeared they had reached an impasse. I had greater knowledge of, and confidence in the case-study students, and so was inclined to intervene less.

There were limitations in the number investigations also, perhaps the most obvious being in the range of numbers and functions explored. I did not, for example, ask students to multiply by decimals like 0.29 and I did not even contemplate division by decimals. I was primarily interested in how operational links could be established between whole numbers and other quantities and this interest influenced my priorities.

No attempt was made to link the students' work within the Boxer context to pen-and-paper contexts. One reason for this was that much of what we were doing within Boxer was beyond the normal primary school curriculum. Secondly, I felt that there was little to be gained and much to be lost by engaging in symbolic manipulations on paper. My agenda was to explore how Boxer mediated students' expressions of number – not to see if it improved their understanding in pen-and-paper situations.

### **10.2.3: Discussion of Number Issues**

In summarising the CMF research, Johnson (1989) observes that the students practical activities (in algebra) were "... not analogous with the formalization ..." or, (in the case of algorithms), "... there is really little or no match between the two procedures." (p 222). A significant feature of the two number studies in this research was the fact that there was no divide between the practical and the formal. The students' competence in programming meant that they were able to construct Boxer computational models of the formal. Interacting with these models did not take them away from the formal, so there was no problem of transfer between different modes. Rather, Boxer provided the students with a direct visual means for manipulating formal mathematical expressions thus giving them access to a way of mathematical thinking which is usually the preserve of more advanced students (Tall and Thomas, 1989).

Contrasting street mathematics with school mathematics, Nunes et. al. (1993) observe that in the written forms of representations typically used in schools, the meaning of the situation is purposely not represented, and that as situational aspects are lost, syntactic rules become the focus. The lesson for school mathematics, they write, is about ways of "... preserving meaning during the mathematising of situations" (p. 147). The findings of the exploratory and interview studies indicated that the situation is not simply about the preservation of meaning, since for many students, certain numbers do not have meaning for them in the first place.

The two pedagogical principles which guided the number studies, i.e., the holistic and operational approaches, were intended to ensure that meaning was conferred on numbers in this study. In the first place, students were able to derive meaning for unfamiliar quantities by relating them to familiar numbers. In the second place, number symbols derived meaning from the operations which brought them into existence. The computational objects were the catalyst for making connections between the number operations and number concepts.

## **10.3: Pedagogy**

Each of the three phases of the longitudinal study had its own distinctive pedagogical design, and all of them contributed to the constructionist nature of the research. The following sections summarise the findings relating to these, discuss their limitations, and draw out conclusions.

### **10.3.1: Developing a Pedagogy for Introductory Programming**

In the programming phase, the pedagogical approach exploited the medium to introduce itself. Using Boxer in this way raised issues which are not comparable with Logo research. Three distinct, though not necessarily discrete, issues were identified: communication and presentation; the iterative design process, and devolving intervention to the medium.

#### 10.3.1.1: Communication and Presentation

Using the medium to introduce itself depends on students not only being able to read the text on the screen, but also willing to read it. In this phase, three different modes of presentation were tried: lines of text; lines of text in closed boxes; and lines of text in closed boxes labelled with names such as ‘read-me-first’. The latter method was found to be the most likely to lead to text being read. Some of the students compared Boxer to a game-book, so the motivation for reading may have been related to their enjoyment of such activities. A tentative conclusion is that exploiting Boxer’s hypertext capabilities is likely to lead to a greater engagement by the students.

#### 10.3.1.2: The iterative design process

The pedagogical approach adopted in the programming phase owed a great deal to the prevailing circumstances. It was developed with the intention of reconciling two needs: that of observing small groups of students in depth, and that of giving access to Boxer to the remainder of the class. The process of iterative design was the means by which these two needs were met. These microworlds were also being used independently by the other students in the year 5 class when I was not present, and the classroom teacher frequently made suggestions as to how they might be clarified for the students. In working with the case study students, the need for excessive intervention was taken as a cue for the need to redesign the technical component. Using Boxer to introduce itself

meant that we were devolving interventions to the medium, and this raised further issues.

#### 10.3.1.3: Devolving intervention to the medium

Firstly, there is the question of whether, in enacting such design changes, we were making the microworlds more prescriptive? Secondly, it raises the question of whether the organisation of these microworlds into sections pre-disposed the students to a linear way of working. Each section of the 'First-Boxer' microworld had a similar structure: closed boxes which contained instructions or suggestions, a menu data box containing commands for the key idea in that section, a series of doit boxes containing some of the programs for that section and finally a challenge. Each of these were individually modified as the research progressed, but the relation between the sections themselves remained relatively unchanged. A question for future research is whether the hypertext facilities of Boxer could be more fully exploited to offer alternative routes through a microworld or alternative challenges within it?

Perhaps the most significant question that needs to be faced is whether the introductory microworlds simply constituted computer tutorials? The answer to this question depends very much on what constitutes a 'tutorial'. I would claim that these Boxer microworlds re-defined the notion of 'tutorial' in two main ways. Firstly, there was 'space' within the design for the students to adopt their own means of meeting the challenges, or indeed pursue their own goals. Secondly, there was 'space' for the teacher and myself to respond to the students' experiences in redesigning the microworlds themselves. A greater exploitation of Boxer's hypertext facility is likely to give even more flexibility.

As stated above, the research design employed in this phase was a response to the circumstances in the school, principally the fact that there was only one computer in a class of 30 students. While, this may suggest one mechanism by which this all too familiar problem may be overcome in school settings, I believe that the iterative design process has something to offer even in circumstances where resources are plentiful. Teachers have always gone beyond the textbook or scheme to design worksheets or other materials for their students. However, until recently, the parallel activity in a computer environment was not possible. The research design employed in this study indicates one way in which the reconstructible medium of Boxer can open up the possibility of teachers writing software tuned to the needs of their students.

### **10.3.2: Creating a Constructionist Pedagogy for Number**

The programming phase of the longitudinal study provided students with the means of constructing their own microworlds. The interview phase sought to determine the best way of doing this by building on students' existing conceptions of numbers. Finally, the number phase implemented a constructionist design based on these earlier phases.

#### **10.3.2.1: Building on students' knowledge**

The semi-structured interviews were aimed at generating more focused and coherent issues for the number phase of the study. These interviews were concerned with two facets of number knowledge: students' interpretations and representations of number.

In the first case, the interviews provided valuable data concerning students' conceptions of integers and non integers, and gave some insight into their understanding of the number continuum. They also revealed how students were able to assign an action-based meaning to certain fractions. In the second case, eliciting students' representations not only illuminated their earlier offerings, but also gave valuable insight into their notions of the role of representations generally. Consulting students as part of the research design helped in attaining a constructivist perspective, and in the process contributed to the epistemological integrity of the research.

#### **10.3.2.2: Constructing Computational objects**

The number phase of the research embraced a constructionist (Papert, 1993b) approach to number. In it, the students constructed public entities with which they engaged in multiple and complex relationships and through which they 'concretised' number concepts. The two number studies had three features in common. Firstly, in each of them, the students constructed their own microworld. Secondly, the computational objects at the core of each microworld were made up of representations which were extensible, i.e., they could be used to represent natural numbers, directed numbers and decimals. Thirdly, they were operational. These characteristics afforded me a way of adopting a pedagogical approach that was at once holistic and action-based.

Hoyles and Noss (1992a) observe that a key question in evaluating Logo activities is to ask: "What do the children think they are doing?" (p. 451). In each of the number studies, the discourse was predicated around the operation of the computational object rather than on number algorithms per se. This was of significance in two ways. I was asking, and they were answering, questions about *their* number line or *their* function

machine. Having constructed these, they had a vested interest in knowing how they worked. Secondly, my questions were of a form that directly related to the structure of the number object. Thus questions on the number line might be phrased: “Can you make your number line [in n steps] start at minus fifty and end at 30 ?” or for the function machine: “Can you change your function machine so that 1 2 and 3 go in and 0.01 0.02 and 0.03 come out?” Thus, these computational devices directly linked action-based notions of number with the idea of numbers as objects, i.e., they fostered ‘proceptual thinking’ (Gray and Tall, 1994).

The above paragraphs abstract commonalities of the number line and the function machine objects. But, as the previous number section shows, there were significant differences between these two objects also. In the first case, the number line object mediated students’ developing sense of number in ways that might be described as geometrical. The evidence, which included the invention of notation and the application of an unusual degree of precision, suggests that the number line scaffolded their understanding into the number continuum to an extraordinary extent. In the second case, the function machine mediated their sense of number in ways that could be described as being broadly algebraic. In this case, the evidence included operational descriptions of numbers, the linking of numbers through functions and the linking of numbers through functions and their inverses.

## 10.4: Implications for Research

### 10.4.1.1: Linking Microworlds

The Logo literature is replete with examples of concept acquisition *within* microworlds. However, less attention has been given to the linking of microworlds with each other. As diSessa (1986a) argues in relationship to Boxer:

Multiple microworlds, each strongly tuned to its own particulars, may be too disparate to allow any economy of learning the interactive medium. (diSessa, 1986a, p. 129)

In a similar vein, the Vygotskian or Brunerian notion of ‘scaffolding’ has been used to show how carefully constructed microworlds can support concept acquisition. But, as far as I am aware, there is nothing in the literature to describe how learning might be supported across microworlds.

Two issues arise: the first concerns programming in the mathematics curriculum; the second concerns the role of programming in general. The school mathematics’ curriculum takes place over eleven or so years. Few would suggest that it should be

compressed into say, two years. Yet, studies aimed at evaluating programming have seldom lasted even this long. It would be foolish to claim that Logo, or other languages, have not had greater success in school simply because they were not used over a long enough period. Clearly, there were many other factors involved. However, it is equally clear that there is a need to get away from what has been termed ‘the technological fix’ (Weizenbaum, 1984), i.e., to see the computer – or programming – as a simplistic solution to complex problems. We need to examine afresh the role of programming vis-a-vis the mathematics curriculum and learning in general.

As Noss (1985) points out, the history of programming in school is inextricably intertwined with the mathematics curriculum, and the justification for teaching the former has often been made in terms of their common underlying structure, e.g., in constructing algorithms. Not surprisingly, programming has become or is viewed as an esoteric activity. If, on the other hand, programming is seen as a new literacy, as an expressive medium, then the justification for including it alongside mathematics – and indeed alongside other subjects – as a long-term goal becomes reasonable. One of the implications of this study is that such long-term goals are worthwhile.

#### 10.4.1.2: Boxer Computational Objects

The notion of a computational object is now a familiar one in the literature. Turkle and Papert (1990) use the term ‘transitional objects’ to refer the role played by screen entities in bridging the gap between the concrete and the abstract. Hoyles and Noss (cf. Hoyles, 1993) have coined the term ‘evocative computational objects’ in order to emphasise the resonance between these objects and the knowledge domain. I wish to draw a further distinction between a computational object, such as a polygon drawn in Logo, and the nature of the computational objects as constructed in this research.

The essence of the latter is that they are interactional. Unlike Logo, Boxer objects can themselves become objects for (further) manipulation. They thus make it possible for mathematical activities to continue beyond the initial programming phase. In this study, the students’ construction of the interactional objects of the number line and the function machine gave them a powerful means of visualising number, and enabled them to confer operational meaning on those entities. A question for future research is how other such Boxer objects can be exploited for the investigation of similar or different mathematical domains.

## References

- Abelson, H., Van Couvering, D. and Ploger, D. (1987). The Boxes of Boxer: An Introduction. *Boxer Technical Report No. T1*. University of California, Berkeley.
- Ackermann, E. (1993). From Decontextualised to Situated Knowledge: Revisiting Piaget's Water-Level Experiment. In Harel, I. and Papert, S. (1993). *Constructionism*. New Jersey: Ablex.
- Adams, S. (1988). Children's Knowledge of Dinosaurs: Towards a Framework for Generative Knowledge. *Boxer Experimental Report No. E3*. University of California, Berkeley.
- Adams, S. (1989). Developing Databases and Knowledge Spaces with Boxer: An Illustration based on Dinosaur Knowledge. *Boxer General Report No. G4*. University of California, Berkeley.
- Adams, S. and diSessa, A. A. (1990). Learning by Cheating: Children's Inventive use of a Boxer Motion Microworld. *Journal of Mathematical Behaviour*, 10/1, 79 – 89.
- Alson, P. (1989). Path and Graphs of Functions. *Focus on Learning Problems in Mathematics*, 11, 1, 99 – 106.
- Arcavi, A. and Bruckheimer, M. (1981). How shall we teach the multiplication of negative numbers? *Mathematics in School*, 10, 5, 31 – 33.
- Balacheff, N. (1990). Towards a *Problématique* for Research on Mathematics Education. *Journal for Research in Mathematics Education*, 21, 4, 258 – 272.
- Barwise, J. and Etchemendy, J. (1991). Visual Information and Valid Reasoning. In Zimmermann, W. and Cunningham, S. (Eds.), *Visualization in Teaching and Learning Mathematics*. Providence, RI: MAA Notes Series, Vol. 19, 9 – 24.
- Beattys, C. and Maher, C. (1989). Exploring Children's Perceptions of Multiplication through Pictorial Representations. *Proceedings of the Thirteenth International Conference for the Psychology of Mathematics Education*. France, 109 – 116.
- Behr, M., Lesh, R., Post, T. and Silver, A. (1983). Rational-Number Concepts. In Lesh, R. and Landau, M. (Eds.), *Acquisition of Mathematics Concepts and Processes*. Academic Press, 91 – 126.
- Belanger, M. (1987). Excerpts from the conference. In Janvier, C. (Ed.), *Problems of Representation in the Teaching and Learning of Mathematics*. NJ: Lawrence Erlbaum Associates, 99 – 108.
- Bell, A. W., Costello, J. and Küchemann, D. (1983). *Research on Learning and Teaching: Part A*. NFER.
- Bell, A., Swan, M. and Taylor, G. (1981). Choice of Operations in verbal Problems with Decimal Numbers. *Educational Studies in Mathematics*, 12, 399 – 420.

- Ben-Chaim, D., Lappan, G. and Houang, R. T. (1985). Visualizing rectangular Solids made of small Cubes. *Proceedings of the Ninth International Conference for the Psychology of Mathematics Education*. Holland, 159 – 164.
- Ben-Chaim, D., Lappan, G. and Houang, R. T. (1989). The Role of Visualization in the Middle School Mathematics Curriculum. *Focus on Learning Problems in Mathematics*, 11, 1, 49 – 60.
- Bergerons, J. and Herscovics, N. et al. (1990). Psychological Aspects of Learning Early Arithmetic. In *Mathematics and Cognition: A Research Synthesis by the International Group for the Psychology of Mathematics Education*. Cambridge: Cambridge University Press, 31 – 52.
- Bishop, A. (1977). Is a Picture worth a Thousand Words? *Mathematics Teaching*, 81, 32 – 35.
- Bishop, A. (1979). Visualising and Mathematics in a Pre-Technological Culture. *Educational Studies in Mathematics*, 5, 2, 135 – 146.
- Bishop, A. (1988). Review of Research on Visualization in Mathematics Education. *Proceedings of the Twelfth International Conference for the Psychology of Mathematics Education*. Hungary, 170 – 176.
- Bishop, A. (1989). Review of Research on Visualization in Mathematics Education. *Focus on Learning Problems in Mathematics*, 11, 1, 7 – 16.
- Boyer, C. B. (1968). *A History of Mathematics*. London: Wiley
- Bright, G., Behr, M., Post, T. and Wachsmuth, I. (1988). Identifying fractions on Number Lines. *Journal for Research in Mathematics Education*, 19, 3, 215 – 232.
- Brooks, R. (1980). Studying Programmer Behavior Experimentally: The Problems of Proper Methodology. *Communications of the A.C.M.*, 23, 4, April, 207 – 213.
- Brown, M. (1981). Place Value and Decimals. In Hart, K. (Ed.), *Children's Understanding of Mathematics 11–16*. London: Murray, 48 – 65.
- Brown, M. and Küchemann, D. (1981). “Is it an ‘add’, Miss?” Part 3. *Mathematics in School*, 10, 1, 26 – 28.
- Bruner, J. (1968). *Towards a Theory of Instruction*. New York: Norton.
- Bruner, J. (1980). The Course of Cognitive Growth. In Anglin, J. (Ed.), *Beyond the Information Given*. London: George Allen and Unwin, 325 – 351.
- Bruner, J. (1985). Vygotsky: A Historical and Conceptual Perspective. In Wersch J. (Ed.), *Culture, Communication, and Cognition: Vygotskian Perspectives*. Cambridge: Cambridge University Press, 21 – 34.
- Campbell, K., Collis, K and Watson, J. (1995). Visual Processing during Mathematical Problem Solving. *Educational Studies in Mathematics*, 28, 177 – 194.
- Carpenter, T., Corbitt, M., Kepner, H., Lindquist, M. and Reys, R. (1981). Decimals: Results and Implications from National Assessment. *Arithmetic Teacher*, 28, 8, 34 – 37.

- Carr, K. and Katterns, B. (1984). Does the Number Line help? *Mathematics in School*, 13, 4, 30 – 34.
- Clement, J. (1989). The Concept of Variation and Misconceptions in Cartesian Graphing. *Focus on Learning Problems in Mathematics*, 11, 2, 77 – 87.
- Clement, J. (1991). Constructivism in the Classroom. *Journal for Research in Mathematics Education*, Vol. 22, No. 5, 422 – 428.
- Clements, M. A. and Del Campo, G. (1989). Linking Verbal Knowledge, Visual Images, and Episodes for Mathematical Learning. *Focus on Learning Problems in Mathematics*, 11, 1, 25 – 33.
- Cobb, P. (1986). Making Mathematics: Children's Learning and the Constructivist Tradition. *Harvard Educational Review*, 56, 3, 301 – 306.
- Confrey, J. (1985). Towards a Framework for Constructivist Instruction. *Proceedings of the Ninth International Conference for the Psychology of Mathematics Education*. Holland, 477 – 483.
- Confrey, J. (1994). A Theory of Intellectual Development Part 1. *For the Learning of Mathematics*, 14, 3, 2 – 8.
- Confrey, J. (1995). A Theory of Intellectual Development Part 2. *For the Learning of Mathematics*, 15, 1, 38 – 48.
- Cook, N. and Kersh, M. (1980). Improving Teachers' Ability to Visualize Mathematics. *Proceedings of the Fourth International Conference for the Psychology of Mathematics Education*. Berkeley, 377 – 383.
- Cunningham, S. (1991). The Visualization Environment for Mathematics. In Zimmermann, W. and Cunningham, S. (Eds.), *Visualization in Teaching and Learning Mathematics*. Providence, RI: MAA Notes Series, Vol. 19, 67 – 76.
- Cuoco, A. (In press). Computational Media to Support the Learning and use of Functions. In diSessa, A. A., Hoyles, C., Noss, R. and Edwards, L. (Eds.), *Computers for Exploratory Learning*. Berlin Springer-Verlag, x1 – x29.
- Davis, P. (1993). Visual Theorems. *Educational Studies in Mathematics*, 24, 333 – 344.
- Davis, R. (1984). *Learning Mathematics: The Cognitive Science Approach to Mathematics Education*. London: Routledge.
- Davis, R. (1986). Conceptual and Procedural Knowledge in Mathematics: A Summary Analysis. In Hiebert, J. (Ed.), *Conceptual and Procedural Knowledge: The case for Mathematics*. Hillsdale, NJ: Lawrence Erlbaum, 265 – 300.
- Davis, R. (1992). Understanding "Understanding". *Journal of Mathematical Behaviour*, 11, 225 – 241.
- Dickson, L. et al, (1988). *Children Learning Mathematics: A Teacher's Guide to Recent Research*. London: Cassell.

- diSessa, A. A. (1983). Phenomenology and the Evolution of Intuition. In Gentner, D. and Stevens, A. (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum, 15 – 33.
- diSessa, A. A. (1985a). Learning about Knowing. In Klein, E. (Ed.), *Children and Computers. New Directions for Child Development* (28), San Francisco: Josey-Bass Inc., 97 – 124.
- diSessa, A. A. (1985b). A Principled Design for an Integrated Computational Environment. *Human-Computer Interaction*, 1(1), 1 – 47.
- diSessa, A. A. (1986a). Notes on the future of Programming: Breaking the Utility Barrier. In D. A. Norman and S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum, 125 – 152.
- diSessa, A. A. (1986b). Models of Computation. In D. A. Norman and S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum, 201 – 218.
- diSessa, A. A. (1986c). Principles of Design for an Integrated Computational Environment for Education. In Sendov, B. and Stanchev, I. (Eds.), *Children in an Information Age: Tomorrow's Problems Today*. Oxford: Persimmon, 97 – 109.
- diSessa, A. A. (1986d). From Logo to Boxer, A New Computational Environment. *Australian Educational Computing*, July, 8 – 15.
- diSessa, A. A. (1987a). The Third Revolution in Computers and Education. *Journal of Research in Science Teaching*. 24, 4, 343 – 367.
- diSessa, A. A. (1987b). Reference and Data Construction in Boxer. In Gorny, P. and Tauber, M. J. (Eds.), *Visualization in Programming*, Fifth Interdisciplinary Workshop in Informatics and Psychology, Lecture notes in Computer Science, 282, Berlin Springer-Verlag, 151 – 163.
- diSessa, A. A. (1987c). Artificial Worlds and Real Experience. In Lawler, R. and Yazdani, M. (Eds.). *Artificial Intelligence and Education. Volume 1: Learning Environments and Tutoring Systems*. New Jersey Ablex, 55 – 78.
- diSessa, A. A. (1988a). What will it mean to be educated in 2020? In Nickerson, R. and Zodhiates, P. (Eds.), *Technology and Education: Looking towards 2020*. Hillsdale, NJ: Lawrence Erlbaum Associates, 43 – 66.
- diSessa, A. A. (1988b). Knowledge in Pieces. In Forman, G. and Pufall, P. (Eds.), *Constructivism in the Computer Age*, Hillsdale, NJ: Lawrence Erlbaum Associates, 49 – 70.
- diSessa, A. A. (1989). Computational Media as a Foundation for New Learning Cultures. *Boxer General Report No. G5*. University of California, Berkeley.
- diSessa, A. A. (1990a). Local Sciences: Viewing the Design of Human-Computer Systems as Cognitive Science. *Boxer General Report No. G6*. University of California, Berkeley.

diSessa, A. A. (1990b). Social Niches For future Software. In Gardner, M., Greeno, J., Reif, F., Schoenfeld, A., diSessa, A. A. and Stage, E. (Eds.), *Towards a Scientific Practice Of Science Education*, Hillsdale, NJ: Lawrence Erlbaum, 301 – 322.

diSessa, A. A. (1993). Designing Newton's Laws: Patterns of social and Representational Feedback in a Learning Task. *Draft Report*, Boxer Group. University of California, Berkeley.

diSessa, A. A. (1995 – in press). The Many Faces of a Computational Medium: Teaching the Mathematics of Motion. In Hoyles C., Noss R. and Edwards L. *Computers for Exploratory Learning*. Berlin Springer–Verlag, 337 – 359.

diSessa, A. A. and Abelson, H. (1986). Boxer: A Reconstructible Computational Medium. *Communications of the A.C.M.*, 29 (9), 859 – 868.

diSessa, A. A., Abelson, H. and Ploger, D. (1991). An Overview of Boxer. *Journal of Mathematical Behaviour*, 10, 2, 3 –15.

diSessa, A. A., Hammer, D., Sherin, B., Kolpakowski, T. (1991). Inventing Graphing: Meta-representational Expertise in Children. *Journal of Mathematical Behaviour*, 10, 2, 117 – 160.

Donaldson, M. (1987). *Children's Minds*. London: Fontana Press.

Dörfler, W. (1991). Meaning: Image Schemata and Protocols. *Proceedings of the Fifteenth International Conference for the Psychology of Mathematics Education*. Assisi, 17 – 32.

Dreyfus, T. (1990). On Difficulties with Diagrams: Theoretical Issues. *Proceedings of the Fourteenth International Conference for the Psychology of Mathematics Education*. Mexico, 27 – 34.

Dreyfus, T. (1991). On the Status of Visual Reasoning in Mathematics and Mathematics Education. *Proceedings of the Fifteenth International Conference for the Psychology of Mathematics Education*. Assisi, 33 – 48.

Dreyfus, T. and Halevi, T. (1990). QuadFun – A Case of Pupil Computer Interaction. *Journal of Computers in Mathematics and Science Teaching*, 10 (2), 43 – 48.

Driscoll, M. (1984). What Research Says. *Arithmetic Teacher*, 31, 6, 34 – 35.

Dubinsky, E. (1991). Reflective abstraction in Advanced Mathematical Thinking. In Tall, D. (Ed.), *Advanced Mathematical Thinking*. Dordrecht, The Netherlands: Kluwer, 95 – 123.

duBoulay, B. (1986). Some Difficulties in Learning to Program. *Journal of Educational Computing Research*, 2 (1), 57 – 73.

Duffin, J. and Simpson, A. (1993). Natural, Conflicting and Alien. *Journal of Mathematical Behaviour*, 12, 313 – 328.

Dunkels, A. (1988). Visualization of Whole Numbers with Applications to Divisibility Rules. *Invited paper to ICME 6*, Budapest, 1 – 12.

- Eastwood, M. (1983). More Models for Directed Numbers. *Mathematics in School*, 12, 3, 34 – 35.
- Eisenberg, T. and Dreyfus, T. (1986). On Visual versus Analytical Thinking in Mathematics. *Proceedings of the Tenth International Conference for the Psychology of Mathematics Education*. London, 153 – 158.
- Eisenberg, T. and Dreyfus, T. (1989). Spatial Visualization in the Mathematics Curriculum. *Focus on Learning Problems in Mathematics*, 11, 1, 1 – 6.
- Eisenberg, T. and Dreyfus, T. (1991). On the Reluctance to Visualize in Mathematics. In Zimmermann, W. and Cunningham, S. (Eds.), *Visualization in Teaching and Learning Mathematics*. Providence, RI: MAA Notes Series, Vol. 19, 25 – 38.
- Eisenhart, M. (1988). The Ethnographic Research Tradition and Mathematics Education Research. *Journal for Research in Mathematics Education*, 19, 2, 99 – 114.
- Ellerton, N. and Clemens, M. (1990). Cognitive Dissonance versus Success as the Basis for Meaningful Mathematical Learning. *Proceedings of the Fourteenth International Conference for the Psychology of Mathematics Education*. Mexico, 177 – 184.
- Eves, H. (1976). *An Introduction to the History of Mathematics*. London: Holt, Rinehart and Winston.
- Fischbein, E. (1977). Image and Concept in Learning Mathematics. *Educational Studies in Mathematics*, 8 (2), 153 – 165.
- Fischbein, E. (1978). Intuition in Mathematical Education. *Proceedings of the Second International Conference for the Psychology of Mathematics Education*. Osnabruck, 148 – 176.
- Fischbein, E., Deri, M., Nello, M. and Marino, M. (1985). The role of implicit models in solving verbal problems in Multiplication and Division. *Journal for Research in Mathematics Education*, 16, 3 – 17.
- Forman, E. and Cazden, C. (1985). Exploring Vygotskian Perspectives in Education: the cognitive value of peer interaction. In Wersch J. (Ed.), *Culture, Communication, and Cognition: Vygotskian Perspectives*. Cambridge: Cambridge University Press, 323 – 347.
- Foxman, D., Ruddock, G., Joffe, L., Mason, K., Mitchell, P. and Sexton, P. (1985). *A Review of Monitoring in Mathematics 1978 – 1982*. London: H.M.S.O.
- Gadanidis, G. (1994). Deconstructing Constructivism. *The Mathematics Teacher*. 87, 2, 91 – 97.
- Gelman, R. and Gallistel, C. (1978). *The Child's Understanding of Number*. Cambridge, Mass: Harvard University Press.
- Ginsberg, H. (1977). *Children's Arithmetic: The Learning Process*. New York: Van Nostrand.

- Goldenberg, E. P. (1987). Believing is Seeing. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 197 – 203.
- Goldenberg, E. P. (1988). Mathematics, Metaphors, and Human Factors: Mathematical, Technical, and Pedagogical Challenges in the Educational Use of Graphical Representation of Functions. *Journal of Mathematical Behaviour*, 7, 135 – 173.
- Goldenberg, E. P. (1989). Seeing Beauty in Mathematics: Using Fractal Geometry to Build a Spirit of Mathematical Inquiry. *Journal of Mathematical Behaviour*, 8, 169 – 204.
- Goldenberg, E. P. (1991). A Mathematical Conversation with Fourth Graders. *Arithmetic Teacher*, 38, 8, 38 – 43.
- Graeber, A. and Baker, K. (1991). Curriculum Materials and Misconceptions concerning Multiplication and Division. *Focus on Learning Problems in Mathematics*, 13, 3, 25 – 37.
- Graeber, A. and Tirosh, D. (1990). Insights Fourth and Fifth Graders bring to Multiplication and Division with Decimals. *Educational Studies in Mathematics*, 21, 6, 565 – 588.
- Gray, E. and Tall, D. (1994). Duality, Ambiguity, and Flexibility: A “Proceptual” View of Simple Arithmetic. *Journal for Research in Mathematics Education*, 24, 2, 116 – 140.
- Greeno, J. (1983). Conventional Entities. In Gentner, D. and Stevens, A. (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum, 227 – 252.
- Greeno, J. (1991). Number Sense as Situated Knowing in a Conceptual Domain. *Journal for Research in Mathematics Education*, 22, 3, 170 – 218.
- Greer, B. (1987). Didactical Obstacles in the Learning of Multiplication and Division. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 402 – 407.
- Greer, B. (1987). Non-conservation of Multiplication and Division involving Decimals. *Journal for Research in Mathematics Education*, 18, 37 – 45.
- Greer, B. and Mangan, C. (1986). Choice of Operations: From 10-Year Olds to Student Teachers. *Proceedings of the Tenth International Conference for the Psychology of Mathematics Education*. London, 25 – 30.
- Grossman, A. (1983). Decimal Notation: An important Research Finding. *Arithmetic Teacher*, 30, 9, 32 – 33.
- Grossnickle, F. and Perry, L. (1985). Division with Common Fraction and Decimal Divisors. *School Science and Mathematics*. 85 (7), 556 – 566.
- Hadamard, J. (1954). *The Psychology of Invention in the Mathematical Field*. New York, NY: Dover Publications.

- Hammersley, M. and Atkinson, P. (1983). *Ethnography Principles in Practice*. London: Tavistock.
- Hammond, M. (1994). Measuring the impact of IT on Learning. *Journal of Computer Assisted Learning*, 10, 251 - 260.
- Harel, G. (1989). Learning and Teaching Linear Algebra: Difficulties and an Alternative Approach to Visualizing Concepts and Processes. *Focus on Learning Problems in Mathematics*, 11, 2, 139 – 148.
- Harel, I. (1990). Children as Software Designers: A Constructionist Approach. *The Journal of Mathematical Behavior*, 9, 1 (special issue).
- Harel, I. (1991). *Children Designers*. New Jersey: Ablex
- Harel, I. and Papert, S. (1993). *Constructionism*. New Jersey: Ablex.
- Hart, K. (Ed.), (1981). *Children's Understanding of Mathematics 11–16*. London: Murray.
- Hart, K. (1987). Practical work and Formalisation: Too Great a Gap. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 408 – 415.
- Hart, K. (1989). Place Value: Subtraction. In Johnson, D. C. (Ed.). *Children's Mathematical Frameworks 8–13*. Windsor: NFER-Nelson, 8 – 45.
- Hart, K. (1989). Fractions: Equivalence and Addition. In Johnson, D. C. (Ed.). *Children's Mathematical Frameworks 8–13*. Windsor: NFER-Nelson, 46 – 75.
- Harvey, B. (1985). *Computer Science Logo Style, Volume 1*. London: The MIT Press.
- Hasemann, K. (1981). On Difficulties with Fractions. *Educational Studies in Mathematics*, 12, 71 – 87.
- Hefendehl-Hebeker, L. (1991). Negative Numbers: Obstacles in their Evolution from Intuitive to Intellectual Constructs. *For the Learning of Mathematics*, 11, 1, 26 – 32.
- Hershkowitz, R. (1989). Visualization in Geometry – Two Sides of One Coin. *Focus on Learning Problems in Mathematics*, 11, 1, 61 – 76.
- Hiebert, J. and Lefevre, P. (1986). Conceptual and Procedural Knowledge in Mathematics: An Introductory Analysis. In Hiebert, J. (Ed.), *Conceptual and Procedural Knowledge: The case for Mathematics*. Hillsdale, NJ: Lawrence Erlbaum, 1 – 24.
- Hiebert, J. and Wearne, D. (1986). Procedures over Concepts: The Acquisition of Decimal Number Knowledge. In Hiebert, J. (Ed.), *Conceptual and Procedural Knowledge: The case for Mathematics*. Hillsdale, NJ: Lawrence Erlbaum, 199 – 223.
- Hiebert, J. and Wearne, D. (1992). Links between Teaching and Learning Place Value with Understanding in First Grade. *Journal for Research in Mathematics Education*, 23, 2, 98 – 122.

- Higginson, W. (1980). Upstaging Piaget: The Child as Dialectical Constructivist. *Proceedings of the Fourth International Conference for the Psychology of Mathematics Education*. Berkeley, 231 – 236.
- Hillel, J. (1992). The Notion of Variable in the context of turtle graphics. In Hoyles, C. and Noss, R. (Eds.), *Learning Mathematics and Logo*. London: MIT Press, 11 – 36.
- Holmes, B. (1987). Children's Inferences with Print and Pictures. *Journal of Educational Psychology*, 79 (1), 14 – 18.
- Hoyles, C. (1987). Tools for Learning – Insights for the Mathematics Educator from a Logo Programming Environment. *For the Learning of Mathematics*, 7, 2, 32 – 37.
- Hoyles, C. (1993). Microworlds/Schoolworlds: The transformation of an innovation. In Keitel, C. and Ruthven, K. (Eds.), *Learning from Computers: Mathematics Education and Technology*. NATO ASI, Series F: Computer and Systems Sciences, Vol. 121, Springer-Verlag, 1 – 17.
- Hoyles, C. and Noss, R. (1987a). 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 Noss, R. (1987b). Seeing What Matters: Developing an Understanding of the Concept of Parallelogram through a Logo Microworld. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 231 – 236.
- Hoyles, C. and Noss, R. (1987c). Expanding Children's Mathematics with the Computer. In Johnson, D. and Lovis, F. (Eds.), *Informatics and the Teaching of Mathematics*. North Holland: Elsevier Science Publishers B.V., 125 – 135.
- Hoyles, C. and Noss, R. (1992a). Looking Back and Looking Forward. In Hoyles, C. and Noss, R. (Eds.), *Learning Mathematics and Logo*, London: MIT Press, 431 – 468.
- Hoyles, C. and Noss, R. (1992b). A Pedagogy for Microworlds. *Educational Studies in Mathematics*, 23, 31 – 57.
- Hoyles, C. and Noss, R. (1992c). Introducing Boxer. *Micromath*, 8, 3, 32 – 35.
- Hoyles, C. and Noss, R. (1993a). Microworlds/Schoolworlds: The transformation of an innovation. In Keitel, C. and Ruthven, K. (Eds.), *Learning from Computers: Mathematics Education and Technology*, NATO ASI, Series F: Computer and Systems Sciences, Vol. 121, Springer-Verlag, 1 – 17.
- Hoyles, C. and Noss, R. (1993b). *Deconstructing Microworlds*. In Freemason, D. L. (Ed.), *Advanced Educational Technologies in the Teaching of Mathematics and Science*, Berlin Springer-Verlag, 415 – 438.
- Hoyles, C. and Noss, R. (1993c). Out of the Cul-De-Sac. *Proceedings of the Fifteenth International Conference for the Psychology of Mathematics Education*. Assisi, 83 – 90.

- Hoyles, C. and Sutherland, R. (1989). *Logo Mathematics in the Classroom*. London: Routledge.
- Hutchins, E., Hollan, J. and Norman, D. (1986). Direct Manipulation Interfaces. In D. A. Norman and S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human–Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum, 86 – 124.
- Janvier, C. and Garançon, M. (1989). Graphical Understanding of Simple Feedback Systems. *Focus on Learning Problems in Mathematics*, 11, 2, 127 – 137.
- Jaworski, B. (1990). Scaffolding. *Proceedings of the Fourteenth International Conference for the Psychology of Mathematics Education*. Mexico 91 – 98.
- Johnson, D. C. (1989) (Ed.). *Children's Mathematical Frameworks 8 –13*. Windsor: NFER-Nelson.
- Kaput, J. K. (1986). Information Technology and Mathematics: Opening New Representational Windows. *Journal of Mathematical Behaviour*, 5, 187 – 207.
- Kaput, J. K. (1989). Supporting Concrete Visual Thinking in Multiplicative Reasoning: Difficulties and Opportunities. *Focus on Learning Problems in Mathematics*, 11, 1, 35 – 47.
- Kerslake, D. (1979). Visual Mathematics. *Mathematics in School*, 8, 2, 34 – 35.
- Kerslake, D. (1981). Graphs. In Hart, K. (Ed.), *Children's Understanding of Mathematics 11–16*. London: Murray, 120 – 136.
- Kerslake, D. (1986). *Fractions: Children's Strategies and Errors*. Windsor: NFER-Nelson.
- Kieren, T. (1988). Personal Knowledge of Rational Numbers: Its Intuitive and Formal Development. In Hiebert, J. and Behr, M. (Eds.), *Number Concepts and Operations in the Middle Grades*. NCTM, 162 – 181.
- Kilpatrick, J. (1985). Reflection and Recursion. *Educational Studies in Mathematics*, 16, 1 – 26.
- Kilpatrick, J. (1987). What Constructivism Might Be in Mathematics Education. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 4 – 27.
- Klein, P. (1990). Remembering how to read Decimals. *Arithmetic Teacher*, 37, 9, 31.
- Konvissier, M. (1989). Graphical Composition of Functions. *Focus on Learning Problems in Mathematics*, 11, 2, 107 – 115.
- Kruteskii, V. (1976). *The Psychology of Mathematical Abilities in Schoolchildren*. London: The University of Chicago Press.
- Küchemann, D. (1981). Cognitive Demands of Secondary School Mathematics Items. *Educational Studies in Mathematics*, 12, 301 – 316.

- Laborde, C. (1993). Do the pupils learn and what do they learn in a computer based environment? The case of Cabri-Geometre. In Jaworski, B. (Ed.), *Proceedings of the Sixth International Conference on Technology in Mathematics Teaching (TMT93)*, Birmingham, England. University of Birmingham, 39 – 52.
- Larkin, J. and Simon, H. (1987). Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science*, 11, 65 – 100.
- Leddy, T. (1977). Mis-Directed Numbers. *Mathematics Teaching*, 78, 26 – 29.
- Leonard, M. (1990). A Cognitive Benchmark: Learning Programming in Boxer. Paper presented to the *American Educational Research Association* annual meeting.
- Leonard, M. (1991). Learning the Structure of Recursive Programming in Boxer. *Journal of Mathematical Behaviour*, 10, 17 – 53.
- Leron, U. and Zazkis, R. (1986). Functions and Variables. *Proceedings of the Second International Conference for Logo and Mathematics Education*, London, 186 – 192.
- Lesh, R. and Herre, J. (1987). Dienes Revisited: Multiple Embodiments in Computer Environments. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 211 – 220.
- Lowenthal, F. and Vandeputte, C. (1989). Manipulations of Cartesian Graphs: A First Introduction to Analysis. *Focus on Learning Problems in Mathematics*, 11, 2, 89 – 97.
- Markovits, Z. and Sowder, J. (1991). Students' Understanding of the Relationship between Fractions and Decimals. *Focus on Learning Problems in Mathematics*, 13, 1, 3 – 11.
- Markovits, Z. and Sowder, J. (1994). Developing Number Sense: An Intervention Study in Grade 7. *Journal for Research in Mathematics Education*, 25, 1, 4 – 29.
- Marshall, S. P. (1989). Retrospective paper: Number Sense conference. In J. T. Sowder & B. P. Schappelle (Eds.), Establishing foundations for research on number sense and related topics: Report of a conference (pp. 40-42). San Diego State University Center for Research in Mathematics and science Education.
- Mason, J. (1987). What do Symbols Represent? In Janvier, C. (Ed.), *Problems of Representation in the Teaching and Learning of Mathematics*. NJ: Lawrence Erlbaum Associates, 73 – 81.
- Mason, J. (1988). Imagery, Imagination and Mathematics Classrooms. In Pimm, D. (Ed.), *Mathematics Teachers and Children*. London: Hodder and Stoughton, 297 – 302.
- Mason, J. (1993). Less may be More on a Screen. In Jaworski, B. (Ed.), *Proceedings of the Sixth International Conference on Technology in Mathematics Teaching (TMT93)*, Birmingham, England. University of Birmingham, 367 – 374.
- McIntosh, A., Reys, B. and Reys, R. (1992). A Proposed Framework for Examining Basic Number Sense. *For the Learning of Mathematics*, 12, 3, 2 – 8 & 44.

- Myers, B. (1986). Visual Programming, Programming by Example and Program Visualisation: A Taxonomy. *CHI '86 Proceedings*, April, 59 – 66.
- Nesher, P. (1985). Changes in Decimal Reasoning. *Proceedings of the Ninth International Conference for the Psychology of Mathematics Education*. Holland, 334 – 341.
- Nesher, P. (1986a). Are Mathematical Understanding and Algorithmic Performance Related? *For the Learning of Mathematics*, 6, 3, 2 – 9.
- Nesher, P. (1986b). Shifts in Reasoning (The case of Extending Number Concepts). *Educational Studies in Mathematics*, 17, 67 – 79.
- Noss, R. (1985). *Creating a Mathematical Environment through Programming: A Study Of Young Children Learning Logo*. Doctoral dissertation. Chelsea College, University of London. London: Institute of Education, University of London.
- Noss, R. (1988). The Computer as a Cultural Influence in Mathematical Learning. *Educational Studies in Mathematics*, 19, 251 – 268.
- Noss, R. and Hoyles, C. (1987). Structuring the mathematical environment: The dialectic of process and content. *Proceedings of the Third International Conference of Logo and Mathematics Education, Plenary Lecture*, Montreal, 27 – 39.
- Novak, J. (1986). The importance of Emerging Constructivist Epistemology for Mathematics Education. *Journal of Mathematical Behaviour*, 5, 181 – 184.
- Nunes, T., Schlieman, D. and Carraher, W. (1993). *Street Mathematics and School Mathematics*. Cambridge: Cambridge University Press.
- Owen, D. (1987). Decimal multiplication in Grade Seven. *Proceedings of the Twelfth International Conference for the Psychology of Mathematics Education*. Montreal, 423 – 429.
- Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. Brighton: The Harvester Press.
- Papert, S. (1993a). *The Children's Machine – Rethinking School in the Age of the Computer*. Hemel Hempstead: Harvester Wheatsheaf.
- Papert, S. (1993b). Situating Constructionism. In Harel, I. and Papert, S. (Eds.), *Constructionism*. New Jersey: Ablex, 1 – 12.
- Papert, S., Watt, D., diSessa, A. A. and Weir, S. (1979). *Final Report of the Brookline Logo Project, Logo Memos 53 and 54*. Cambridge, Mass: Massachusetts Institute of Technology.
- Pea, R. D. (1986). Language-Independent Conceptual "Bugs" in Novice Programming. *Journal of Educational Computing Research*, 2 (1), 25 – 36.
- Piaget, J. (1965). *The Child's Conception of Number*. London: Routledge and Kegan Paul.
- Picciotto, H. (1989a). A Teacher's Perspective of Boxer. Paper presented to the *American Educational Research Association* annual meeting.

- Picciotto, H. (1989b). Teacher-Created Educational Software: Basic, Logo, Boxer. *Boxer Technical Report No. E4*. University of California, Berkeley.
- Picciotto, H. and Ploger, D. (1991). Learning about Sampling with Boxer. *Journal of Mathematical Behaviour*, 10, 91 – 113.
- Pimm D. (1987). *Speaking Mathematically*. Blackwell: Oxford
- Pimm D. (1995). *Symbols and Meanings in Mathematics*. London: Routledge.
- Pirie, S. and Kieren, T. (1989). A Recursive Theory of Mathematical Understanding. *For the Learning of Mathematics*, 9, 3, 7 – 11.
- Pirie, S. and Kieren, T. (1992a). Creating Constructivist Environments and constructing creative Mathematics. *Educational Studies in Mathematics*, 23, 505 – 528.
- Pirie, S. and Kieren, T. (1992b). Watching Sandy's Understanding Grow. *Journal of Mathematical Behaviour*, 11, 243 – 257.
- Pirie, S. and Kieren, T. (1994). Beyond Metaphor: Formalising in Mathematical Understanding within Constructivist Environments. *For the Learning of Mathematics* 14, 1, 39 – 43.
- Ploger, D. (1990a). Reasoning and Learning about Mechanisms in Biology. *Draft Report*, Boxer Group. University of California, Berkeley.
- Ploger, D. (1991). Learning about the Genetic Code via Programming. Representing the Process of Translation. *Journal of Mathematical Behavior*, 10, 55 – 77.
- Ploger, D. (1990b). Problem Solving with a flexible Computer System: Reconstructing a Discovery in Biology, *Draft Report*, Boxer Group. University of California, Berkeley.
- Ploger, D. and diSessa, A. A. (1987a). Rolling Dice: Exploring Probability in the Boxer Computer Environment. *Boxer Experimental Report No. E1*. University of California, Berkeley.
- Ploger, D. and diSessa, A. A. (1987b). Hyper-Programming in Boxer. *Boxer General Report No. G2*. University of California, Berkeley.
- Ploger, D. and diSessa, A. A. (1989). *Boxer Research: An Annotated Bibliography*. *Boxer General Report No. G1*. University of California, Berkeley.
- Ploger, D. and Lay, E. (1990). The Structure of Programs and Molecules. *Draft Report*. Boxer Group, University of California, Berkeley.
- Plunkett, S. (1979). Diagrams. *Mathematics Education for Teaching*, 3 (4), 3 – 15.
- Plunkett, S. (1981). Decomposition and all that Rot. In Floyd, A. (Ed.), *Developing Mathematical Thinking*. London: Addison-Wesley, 177 – 185.
- Presmeg, N. C. (1986). Visualization and Mathematical Giftedness. *Educational Studies in Mathematics*, 17, 297 – 311.

- Presmeg, N. C. (1986). Visualization in High School Mathematics. *For the Learning of Mathematics*, 6, 42 – 46.
- Presmeg, N. C. (1989). Visualization in Multicultural Mathematics Classrooms. *Focus on Learning Problems in Mathematics*, 11, 1, 17 – 23.
- Raeder, G. (1985). A Survey Of Current Graphical Programming Techniques. *IEEE Computer*, 8, 11 – 25.
- Resnick, L., Nesher, P., Leonard, F., Magone, M., Omanson, S and Peled, I. (1989). Conceptual Bases of Arithmetic Errors: The Case of Decimal Fractions. *Journal for Research in Mathematics Education*, 20, 1, 8 – 27.
- Rowland, T. (1982). Teaching Directed numbers: an experimental report. *Mathematics in School*, 11, 1, 24 – 27.
- Ruddock, G., Mason, K. and Foxman, D. (1984). Assessing Mathematics 2. Concepts and Skills: Decimal Place Value. *Mathematics in School*, 13, 1, 24 – 28.
- Scardamalia, M., Bereiter, C., McLean, R., Swallow, J. and Woodruff, E. (1987). Computer Supported Intentional Learning Environments. In Collins, A. (chair), *Strategies for teaching thinking skills with Interactive Technologies*. Symposium conducted at the meeting of the American Educational Research Association, 2 – 10.
- Self, J. (1985). *Microcomputers in Education: A Critical Appraisal of Educational Software*. Brighton: The Harvester Press Limited.
- Sendov, B. and Stanchev, I. (1985). *Children in an Information Age: Tomorrow's Problems Today. Selected papers from the International Conference*, Varna, Bulgaria, 6 – 9 May 1985. Pergamon Press: Oxford.
- Sfard, A. (1991). On the dual nature of Mathematical Conceptions: Reflections on Processes and Objects as different sides of the same coin. *Educational Studies in Mathematics*, 22, 1 – 36.
- Sfard, A. (1994). Reification as the Birth of Metaphor. *For the Learning of Mathematics*, 14, 1, 44 – 55.
- Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer*, 16 (8), 57 – 69.
- Shuard, H. (1982). Differences in Mathematical performance between Boys and Girls. In Cockcroft, W. *Mathematics Counts*, Appendix 2, 273 – 287.
- Shuard, H. and Rothery, A. (1984). *Children Reading Mathematics*. London: John Murray.
- Shweiker, H. and Muthig, K. (1987). Solving Interpolation Problems with Logo and Boxer. In Gorny, P. and Tauber, M. J. (Eds.), *Visualization in Programming*, Fifth Interdisciplinary Workshop in Informatics and Psychology, Lecture notes in Computer Science, 282. Berlin: Springer-Verlag, 164 – 177.

- Silver, E. (1986). Using Conceptual and Procedural Knowledge: A Focus on Relationships. In Hiebert, J. (Ed.), *Conceptual and Procedural Knowledge: The case for Mathematics*. Hillsdale, NJ: Lawrence Erlbaum, 181 – 198.
- Simon, M. (1995). Elaborating Models of Mathematics Teaching: A Response to Steffe and D'ambrosio. *Journal for Research in Mathematics Education*, 26, 2, 160 – 162.
- Simon, M. (1995). Reconstructing Mathematics Pedagogy from a Constructivist Perspective. *Journal for Research in Mathematics Education*, 26, 2, 114 – 145.
- Sinclair, H. (1987). Constructivism and the Psychology of Mathematics. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 28 – 41.
- Skemp, R. (1971). *The Psychology of Learning Mathematics*. Harmondsworth: Penguin.
- Smith, D. (1925). *History of Mathematics, Volume II, Special Topics of Elementary Mathematics*. Boston: The Athenum Press.
- Smith, F. (1988). How Education backed the wrong Horse. In *Joining the Literacy Club: further essays into education*. Portsmouth, NH: Heinemann, 109 – 123.
- Soloway, E., Ehrlich, K., Bonar, J. and Greenspan, J. (1982). What do Novices know about Programming? In Shneiderman, B. and Badre, A. (Eds.), *Directions in Human-Computer Interaction*. Norwood NY: Ablex Publishing, 27 – 54.
- Sowder, J. (1989). Estimation and Related Topics. In Grouws, D. A. (Ed.), *Handbook of Research on Mathematics Teaching and Learning*. New York, NY: Macmillan, 371 – 389.
- Steffe, L. and D'ambrosio, B. (1995). Towards a working Model of Constructionist Teaching: A Reaction to Simon. *Journal for Research in Mathematics Education*, 26, 2, 146 – 159.
- Steinbring, H. (1989). Routine and Meaning in the Mathematics Classroom. *For the Learning of Mathematics*, 9, 1, 24 – 33.
- Stewart, I. and Tall, D. (1976). *The Foundations of Mathematics*. Oxford: O.U.P.
- Sutherland, R. (1988). *A Longitudinal Study of the Development of Pupils' Algebraic Thinking in a Logo Environment*, Ph.D. Thesis. Institute of Education, University of London.
- Sutherland, R. (1992). What is Algebraic about Programming in Logo? In Hoyles, C. and Noss, R. (Eds.), *Learning Mathematics and Logo*. London: MIT Press, 37 – 54.
- Tall, D. (1983). Introducing Algebra on the Computer: Today and Tomorrow. *Mathematics in School*, 12, 5, 37 – 40.
- Tall, D. (1985). Understanding the Calculus. *Mathematics Teaching*, 110, 49 – 53.

- Tall, D. (1991). Intuition and Rigour: The Role of Visualisation in the Calculus. In Zimmermann, W. and Cunningham, S. (Eds.), *Visualization in Teaching and Learning Mathematics*, Providence, RI: MAA Notes Series, Vol. 19, 105 – 119.
- Tall, D. and Thomas, M. (1989). Versatile Learning and the Computer. *Focus on Learning Problems in Mathematics*, 11, 2, 117 – 125.
- Tall, D. and Thomas, M. (1991). Encouraging Versatile Thinking in Algebra using the Computer. *Educational Studies in Mathematics*, 22, 125 – 147.
- Tall, D. and Vinner, S. (1981). Concept Image and Concept Definition in Mathematics, with particular reference to limits and continuity. *Educational Studies in Mathematics*, 12, 151 – 169.
- Thipkong, S. and Davis, E. (1991). Pre-service Elementary Teacher' Misconceptions in Interpreting and Applying Decimals. *School Science and Mathematics*, 91(3), 93 – 99.
- Thom, R. (1973). Modern Mathematics: Does it Exist? In Howson, G. (Ed.), *Developments in Mathematical Education. Proceedings of the Second International Congress on Mathematical Education*. Cambridge: Cambridge University Press, 194 – 209.
- Thompson, P. (1984). The Relationship of Teachers' Conceptions of Mathematics and Mathematics Teaching to Instructional Practice. *Educational Studies in Mathematics*, 15, 105 – 127.
- Thompson, P. (1992). Notations, Conventions, and Constraints: Contributions to Effective uses of Concrete Materials in Elementary Mathematics. *Journal for Research in Mathematics Education*, 23, 2, 123 – 147.
- Thompson, P. and Dreyfus, T. (1988). Integers as Transformations. *Journal for Research in Mathematics Education*, 19, 2, 115 – 133.
- Tompa, K. (1988). The Role of Audiovisuals in Mathematics Teaching on Visualisation in Mathematics Education. *Proceedings of the Twelfth International Conference for the Psychology of Mathematics Education*. Hungary, 609 – 616.
- Turkle, S. and Papert, S. (1990). Epistemological Pluralism: Styles and Voices within the Computer Culture. *Signs*, 16 (1). The Chicago University Press, 345 – 377.
- Turkle, S. and Papert, S. (1993a). Epistemological Pluralism and the Revaluation of the Concrete. In Harel, I. and Papert, S. (Eds.), *Constructionism*. New Jersey: Ablex, 161 – 192.
- Turkle, S. and Papert, S. (1993b). Styles and Voices. *For the Learning of Mathematics*, 13, 1, 49 – 52.
- Van Couvering, D., Klotz, L. and Ploger, D. (1987). Toolboxes: A Method for Managing Computational Tools. *Boxer Technical Report No. T2*. University of California, Berkeley.
- Vance, J. (1986). Ordering Decimals and Fractions: A Diagnostic Study. *Focus on Learning Problems in Mathematics*, 8, 2, 51 – 59.

- Vance, J. (1992). Understanding Equivalence: a number by any other name. *School Science and Mathematics*, 92 (5), 263 – 266.
- Vergnaud, G. (1978). The Acquisition of Arithmetical Concepts. *Proceedings of the Second International Conference for the Psychology of Mathematics Education*. Osnabruck, 344 – 355.
- Vergnaud, G. (1982a). A Classification of Cognitive Tasks and Operations of Thought involved in Addition and Subtraction Problems. In Carraher, T. P., Mossier, J. M. and Romberg, T. A. (Eds.), *Addition and Subtraction: A Cognitive Perspective*. Hillsdale, NJ: Lawrence Erlbaum, 39 – 59.
- Vergnaud, G. (1982b). Cognitive and Developmental Psychology and Research in Mathematics Education. *For the Learning of Mathematics*, 3, 2, 31 – 41.
- Vergnaud, G. (1983a). Why is an Epistemological Perspective a Necessity for Research in Mathematics Education. *Proceedings of the Fifth International Conference for the Psychology of Mathematics Education*. Montreal, 2 – 20.
- Vergnaud, G. (1983b). Multiplicative Structures. In Lesh, R. and Landau, M. (Eds.), *Acquisitions of Mathematics Concepts and Processes*. New York: Academic Press, 127 – 174.
- Vergnaud, G. (1984). Didactics as a Content–Oriented Approach to Research on the Learning of Physics, Mathematics and Natural Language. *A.E.R.A. Proceedings*, New Orleans, 1 – 23.
- Vergnaud, G. (1987a). About Constructivism. *Proceedings of the Eleventh International Conference for the Psychology of Mathematics Education*. Montreal, 43 – 54.
- Vergnaud, G. (1987b). Conclusion. In Janvier, C. (Ed.), *Problems of Representation in the Teaching And Learning Of Mathematics*. London: Lawrence Erlbaum Associates, 227 – 232.
- Vergnaud, G. (1987c). Problem Solving and Concept Development in the Learning of Mathematics. *E. A. R. L. I. Second Meeting*, Tubingen, September, 1 – 15.
- Vergnaud, G. (1988a). Multiplicative Structures. In Hiebert, J. and Behr, M. (Eds.), *Number Concepts and Operations in the Middle Grades Vol. 2*. London: Lawrence Erlbaum Associates, 141 – 161.
- Vergnaud, G. (1988b). Theoretical Frameworks and Empirical Facts in the Psychology of Mathematics. *Proceedings of the Sixth International Commission for Mathematics Instruction*. Budapest, 2 – 23.
- Vergnaud, G. and Errecalde, P. et al (1980). Some Steps in the Understanding and the use of scales by 10 – 13 year-old students. *Proceedings of the Fourth International Conference for the Psychology of Mathematics Education*. Berkeley, 285 – 291.
- Vergnaud, G., et al (1990). Epistemology and Psychology of Mathematics Education. In *Mathematics and Cognition: A Research Synthesis by the International Group for the Psychology of Mathematics Education*. Cambridge: Cambridge University Press, 14 – 30.

- Vinner, S. (1989). The Avoidance of Visual Considerations in Calculus Students. *Focus on Learning Problems in Mathematics*, 11, 2, 149 – 156.
- Von Glaserfeld, E. (1985). Representation and Deduction. *Proceedings of the Ninth International Conference for the Psychology of Mathematics Education*. Holland, 484 – 489.
- Von Glaserfeld, E. (1987). Preliminaries to any Theory of Representation. In Janvier, C. (Ed.), *Problems of Representation in the Teaching and Learning of Mathematics*, NJ: Lawrence Erlbaum Associates, 215 – 225.
- Vygotsky, L. (1978). *Mind in Society. The Development of Higher Psychological Processes*. London: Harvard University Press.
- Wardle, M. (1990). A Visual approach to Arithmetic. *Micromath*, 6, 1, Spring, 24 – 25.
- Wearne, D. (1990). Acquiring Meaning for Decimal Fraction Symbols: A One Year Follow-Up. *Educational Studies in Mathematics*, 21, 6, 199 – 219.
- Wearne, D. and Hiebert, J. (1988). A Cognitive Approach to Meaningful Mathematics Instruction: Testing a Local Theory using Decimal Numbers. *Journal for Research in Mathematics Education*, 19, 5, 371 – 384.
- Wearne, D. and Hiebert, J. (1994). Place Value and Addition and Subtraction. *Arithmetic Teacher*, 41, 5, 272 – 274.
- Weizenbaum, J. (1984). Another View from MIT. *Byte*, 9, 6, 225.
- Wheatley, C. and Wheatley, G. (1979). Developing Spatial Ability. *Mathematics in School*, 8, 1, 10 – 11.
- Wilensky, U. (1993). Abstract Meditations on the Concrete and Concrete Implications for Mathematics Education. In Harel, I. and Papert, S. (Eds.), *Constructionism*. New Jersey: Ablex, 193 – 204.
- Williams, D. and Rochelle, J. (1993). Mathematical Problem Solving in a Boxer Computational Environment. In Jaworski, B. (Ed.), *Proceedings of the Sixth International Conference on Technology in Mathematics Teaching (TMT93)*, Birmingham, England. University of Birmingham, 491 – 498.
- Wood, D., Bruner, J. and Ross, G. (1976). The role of tutoring in problem-solving. *Journal of Child Psychology and Child Psychiatry*, 17, 89 – 100.
- Woods, P. (1977). The Ethnography of the School. Units 7 – 8, Block II, E202, *Schooling and Society*. Milton Keynes: The Open University Press.
- Yershishalmy, M. and Chazan, D. (1990). Overcoming Visual Obstacles with the aid of the Supposer. *Educational Studies in Mathematics*, 21 (3), 199 – 219.
- Young, R. (1983). Surrogates and Mappings: Two Kinds of Conceptual Models for Interactive Devices. In Gentner, D. and Stevens, A. (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum, 15 – 33.

Zawojewski, J. (1983). Initial decimal Concepts: Are they really so Easy? *Arithmetic Teacher*, 30, 7, 35 – 52.

Zazkis, R. (1992). Theorem-out-of-Action: Formal vs. Naive Knowledge in solving a Graphic Programming Problem. *Journal of Mathematical Behaviour*, 11, 179 – 192.

Zazkis, R. and Khoury, H. (1993). Place Value and Rational Number: Representations: Problem Solving in the Unfamiliar Domain of Non-decimals. *Focus on Learning Problems in Mathematics*, 15, 1, 38 – 51.

Zimmermann, W. and Cunningham, S. (1991). What is Mathematical Visualization? In Zimmermann, W. and Cunningham, S. (Eds.), *Visualization in Teaching and Learning Mathematics*. Providence, RI: MAA Notes Series, Vol. 19, 1 – 9.

Zloof, M. (1984). Classification of Visual Programming Language. *Proceedings of the 1984 IEEE Workshop on visual languages*, Hiroshima, 232 – 235.

**Appendix 1:  
The Line-Numbers  
Representation**

## The Microworld as seen by the students

line-numbers-1

The image shows a Scratch script titled "line-numbers-1". The stage features a horizontal line with two arrows at its ends. The left arrow is labeled "0.19" and the right arrow is labeled "0.213". A script on the stage contains the following code:

```
menu
  cs
  start rt 90
  line num1
  line num2
  compare num1 num2
end
```

A "scale" block is connected to the "line num1" and "line num2" blocks. A "start" block is connected to the "menu" block.

A "compare" block is shown on the stage, with its "num1" slot pointing to a "scale" block and its "num2" slot pointing to another "scale" block. Both "scale" blocks have "Data" blocks below them containing "0.19" and "0.213" respectively.

A "start-number" block is also present on the stage.

A "help" button is located on the right side of the screen.

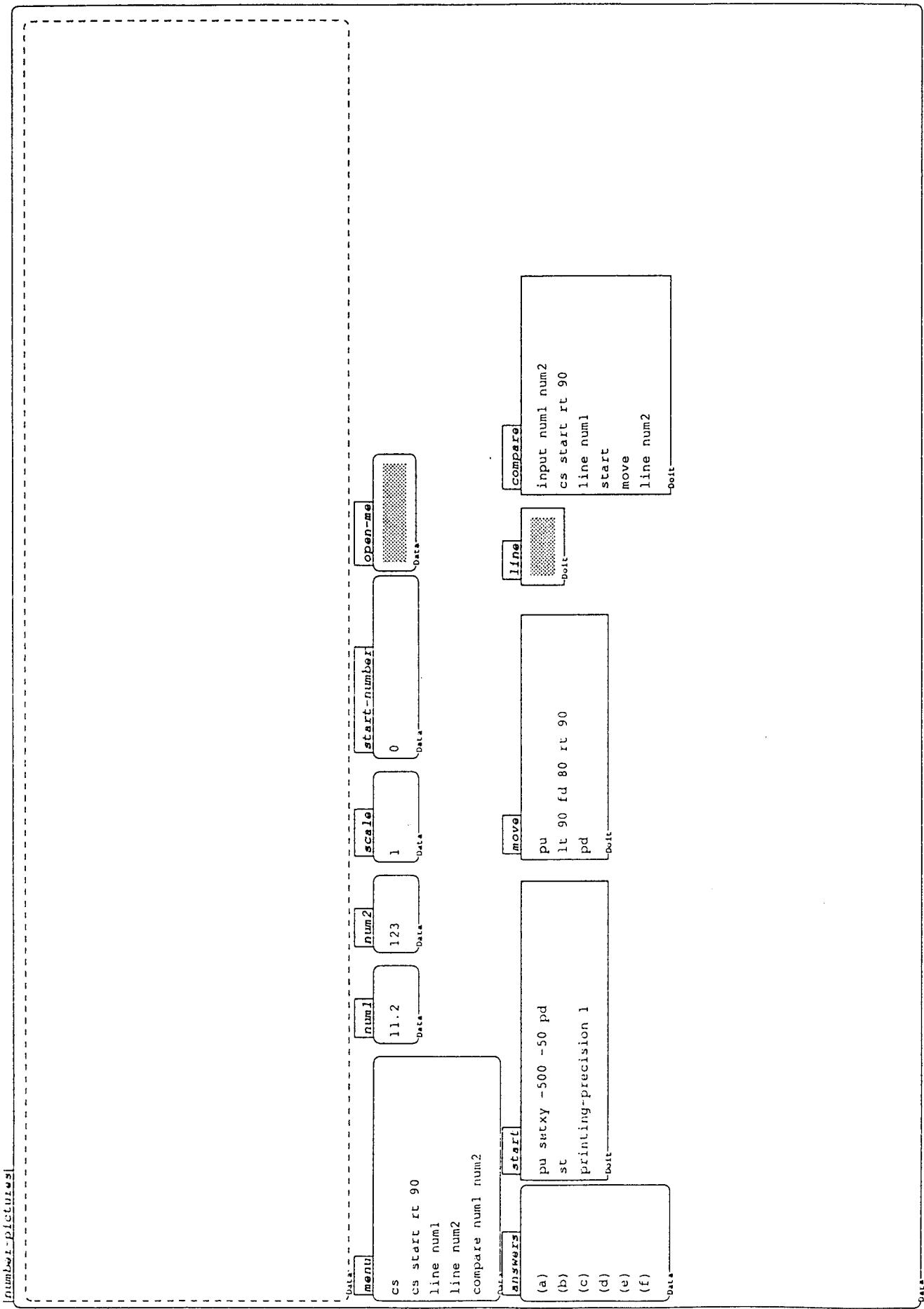
A "Answers" box contains the following list of options:

- (a) 1.29 or 213;
- (b) 346 or 428;
- (c) 1.5 or 1.23;
- (d) 0.7 or 0.54;
- (e) 1052 or 1039;
- (f) 3456 or 5123?

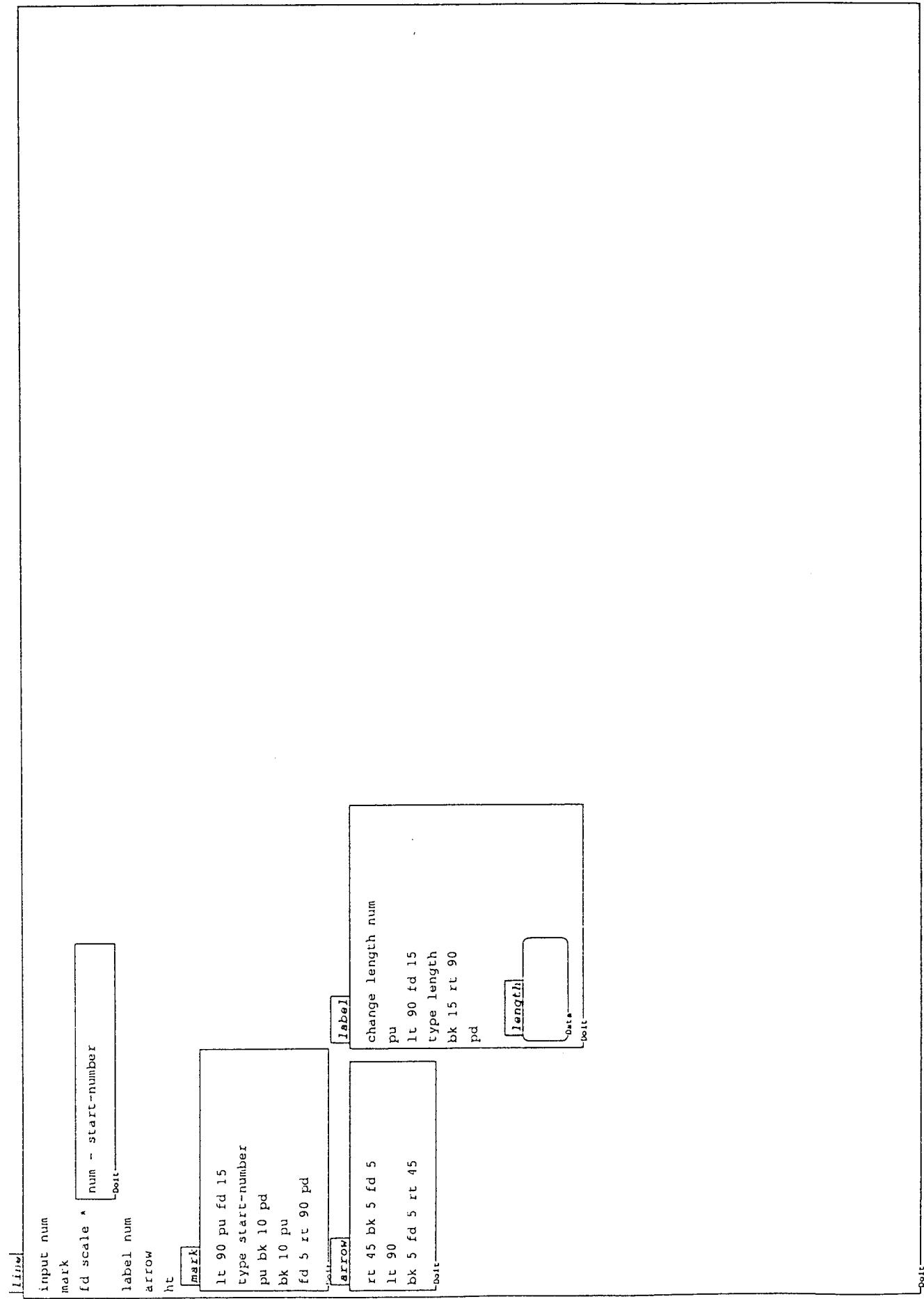
A text box on the right side of the screen contains the following instructions:

Put the numbers in num1 and num2 boxes and press linefeed on 'compare num1 num2'.

## The start, move and compare programs

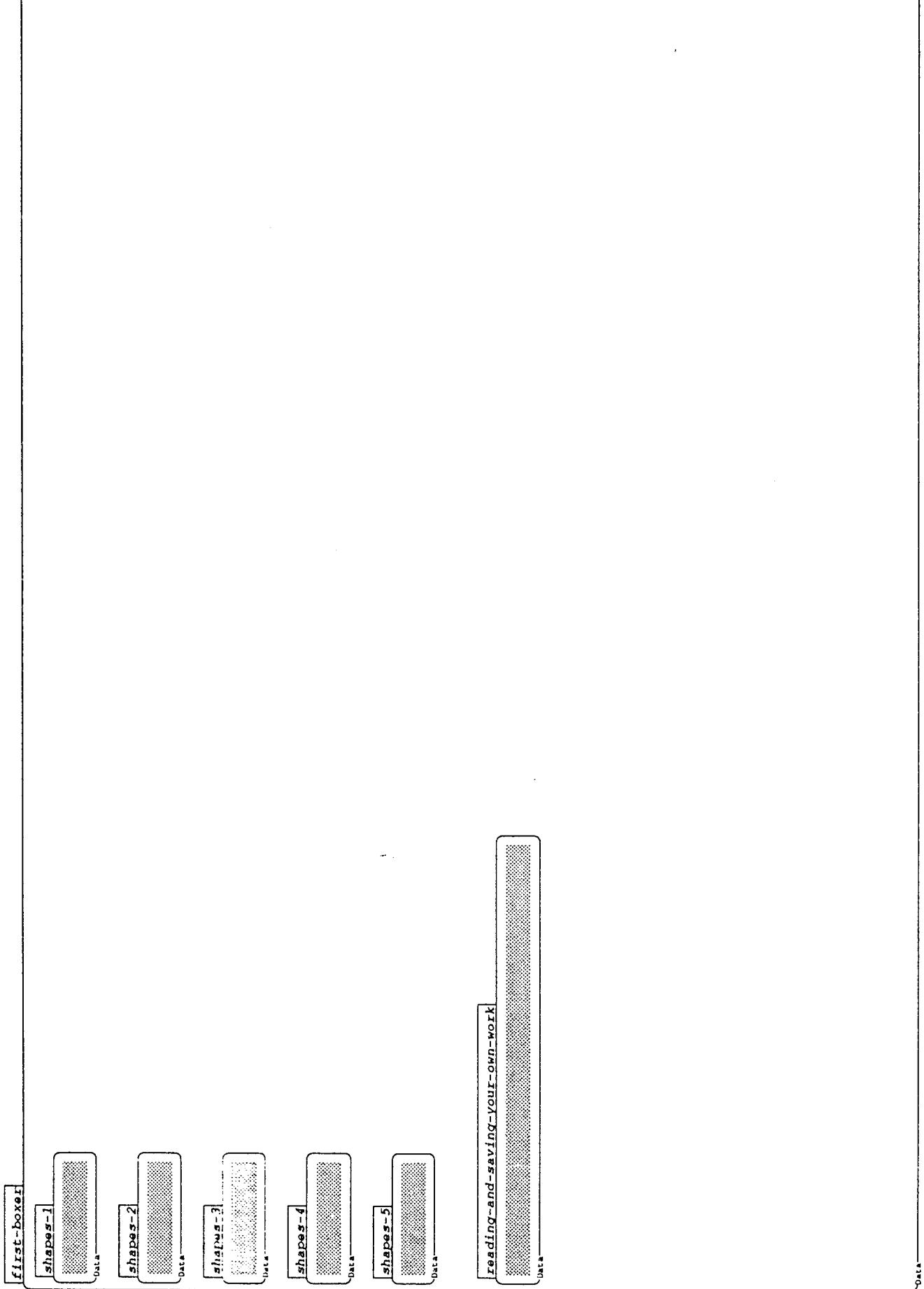


## The line program



**Appendix 2:**  
**First Boxer**

## First-Boxer Overview

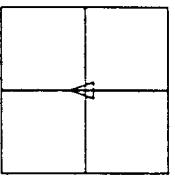


## Shapes-1:

read-me

shapes-1

1. Do each of the lines above the dotted line in the menu  
(Click on the line and press LINE-FEED).  
Data
2. Open the first DoIt box in the menu.  
Click on that line and press Line-feed.  
Do it until you get a square.  
Data
3. challenge  
Make this pattern!  
Data



△

Data

menu

```
forward 126
right 90
left 90
clearscreen
```

Data

Data

Data

Data

Data

## Shapes-2:

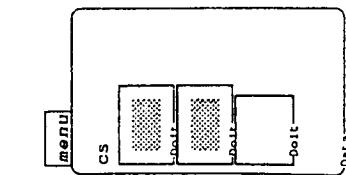
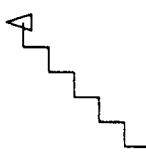
1. Putting a Doit box round lots of commands lets you do them all at once.

Do the Doit boxes in the menu.

2. Try putting some of your own commands in the empty Doit box and try doing it.

3. **challenge**

Make this pattern



## Shapes-3:

read-me

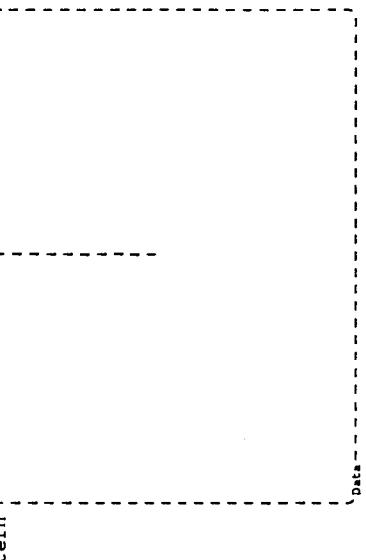
1. Using REPEAT with Doit boxes saves a lot of typing.

Do the Doit boxes in the menu.  
Data

2. Try some of your own in the empty Doit boxes in the menu.  
Data

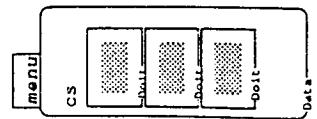
3. challenge

Make this pattern



You will need to use pu (for penup) and pd (for pendown).  
Data

Δ



shapes-3

## Shapes-4:

read-me

shapes-4

1. This time the DoIt boxes have names.  
Do each of the commands above the dotted line.  
**Dot**
2. We can make even more patterns using REPEAT with these names.  
**Dot**

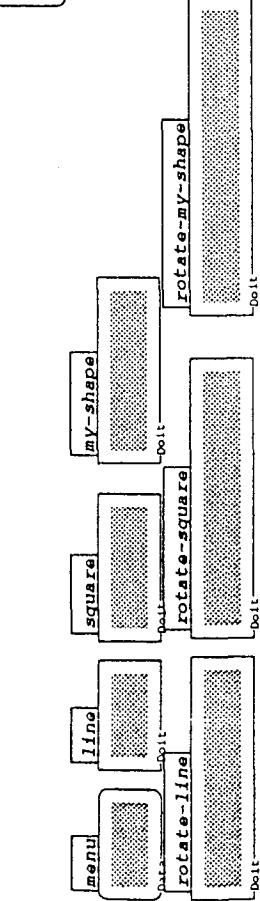
Δ

3. Try rotate-line and rotate-square in the menu.  
**Dot**

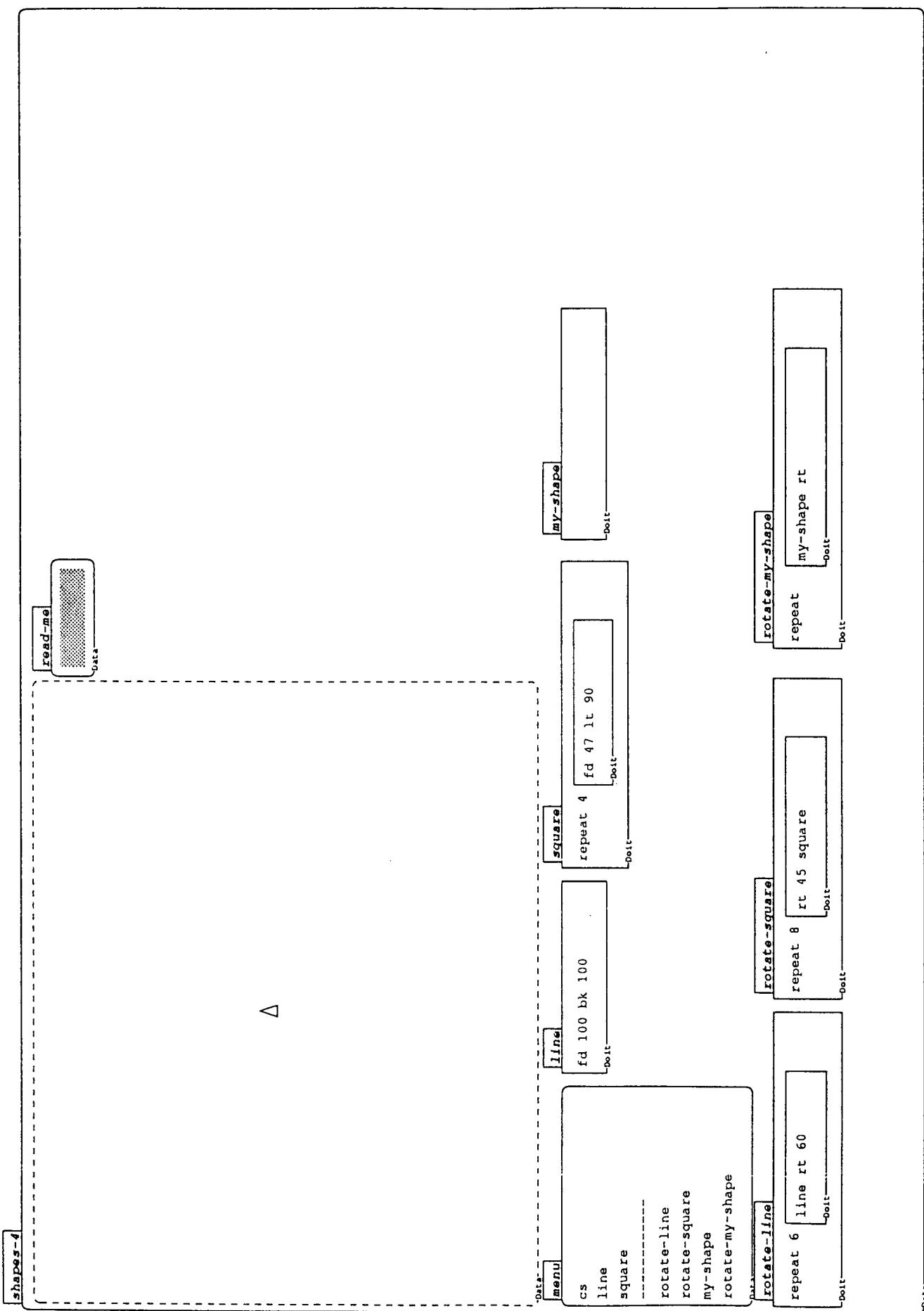
3. Try making patterns of your own by filling in the DoIt boxes my-shape and rotate-my-shape.  
**Dot**

challenge

4. Make this pattern  
**Dot**



## Shapes-4 showing the doit boxes opened



## **Shapes-5:**

માનવસત્તુ - 5

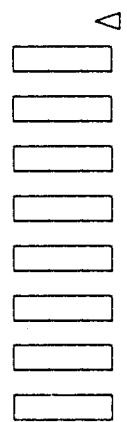
Read-me

1. Try doing each of the commands in the menu.
  2. Make different sized fences.

Challenge

3. Make this pattern (

1



卷之三

1

Date \_\_\_\_\_

menu      cs start rectangle move  
              cs start

卷之三

```
repeat 2 [ fd 100 rt 90  
          fd 25 rt 90 ]
```

1

etxy -200 -150

Date \_\_\_\_\_

menu  
cs  
start  
rectangle  
move  
cs start fence

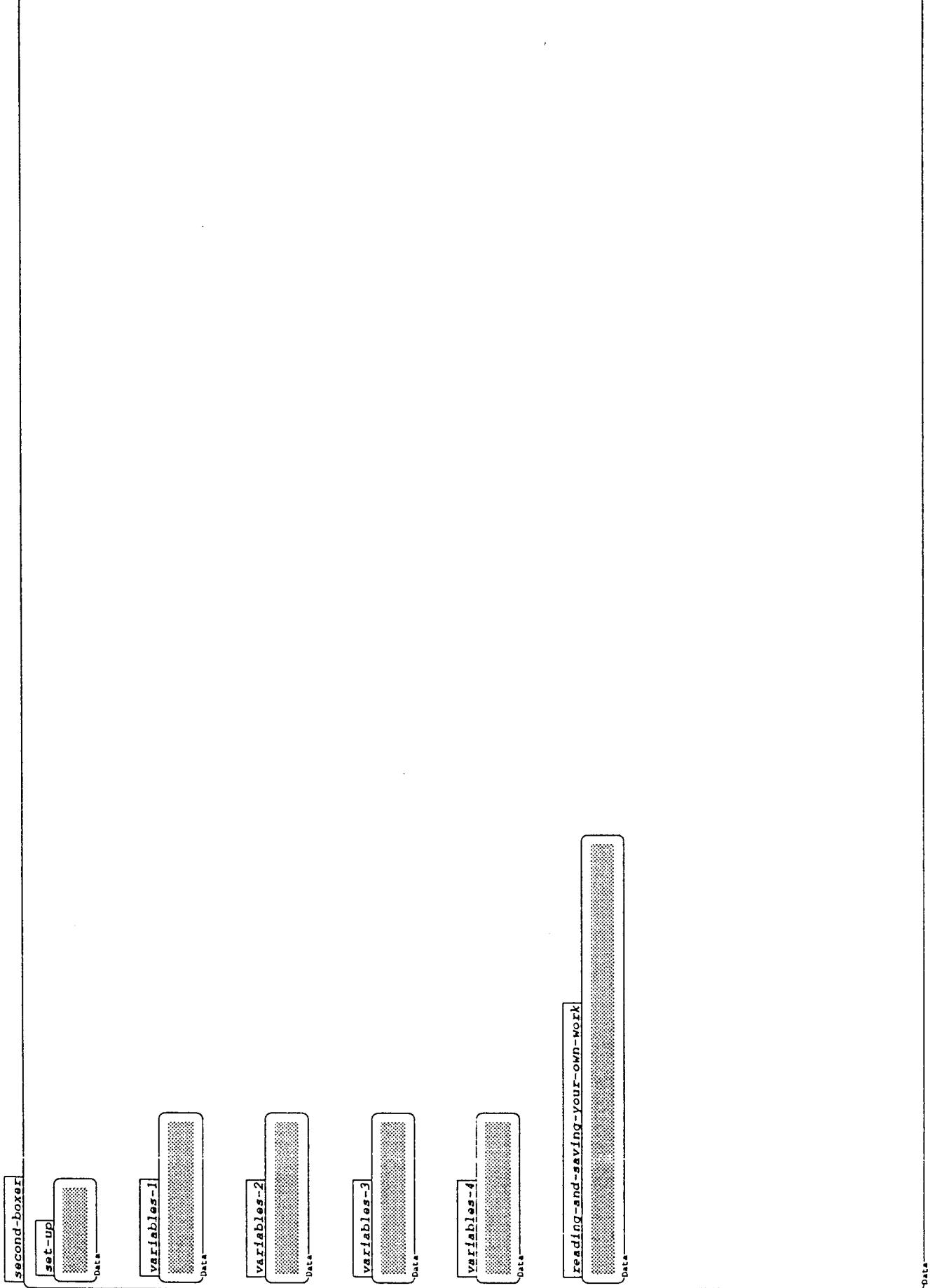
10

pu  
rt 90 fd 25 lt 90  
pd

rectangle move  
rectangle move  
rectangle move

**Appendix 3:  
Second Boxer**

## Second-Boxer Overview



## Set-up showing how to create boxes

**set-up**

**Read-me**  
In second-boxer, you have to make all the boxes on your own.

**challenge**

**Data**

**help**

L3            - makes a Turtle box

Control and L3    - makes a bigger Turtle box

L9            - makes a Data box

F3            - makes a name box

I10          - makes a Doit box

**more-help**

Control and L        - refreshes the screen

Control, diamond and PrSc - prints the screen

Control and K        - kills everything to the right on that line

Control and Y        - yanks back the last thing that was killed

R7            - jumps into a box

R9            - jumps out of a box

**Data**

Now get a print-out of this box

## Set-up showing challenge

The image shows a Scratch script for a challenge setup. The script consists of the following blocks:

- A **repeat [3]** control block.
- Inside the loop:
  - A **repeat [4]** control block.
  - Inside the inner loop:
    - fd 40 rt 90** movement blocks.
    - A **Do It** control block.
  - A **Do It** control block.

## Variables-1:

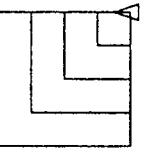
**variables-1**

**read-me**

1. Look at the DoIt box called 'square'. Inside it is a Data box called 'side'.  
Data
2. Do 'square' in the menu. Then change the number in 'side' and do 'square' again.  
Data
3. Do a shape of your own. ('my-shape' has been started for you.)  
Data

**challenge**

4. Make this pattern  
Data



**my-shape**

```
repeat 4 [ fd side lt 90 ]
```

**square**

```
repeat 4 [ fd side lt 90 ]
```

**menu**

```
cs
square
side
```

**variables-1**

## Variables-2:

variables-2

read-me

1. Do each line in the menu.

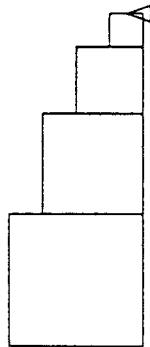
(Question: what does 'start' do?  
Try some plus and minus numbers)

2. Try doing 'square' and 'move' more than once.

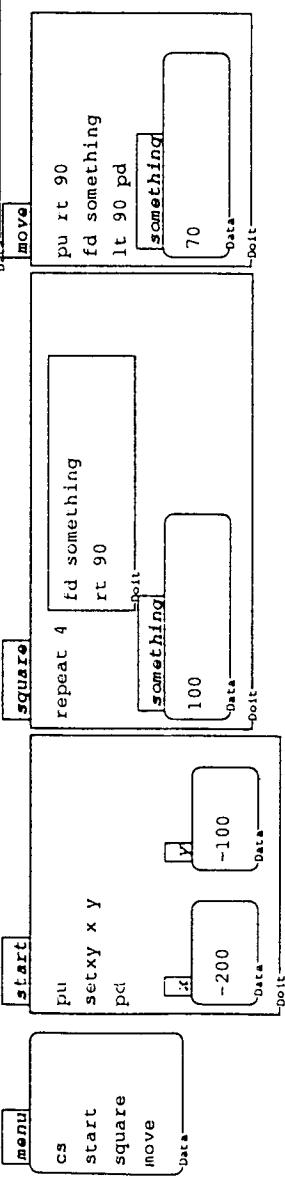
3. Change each of the numbers in the 'something' boxes and do 'square' and 'move' again.

challenge

4. Make this pattern



data



## Variables-3:

variables-3

Lead-me

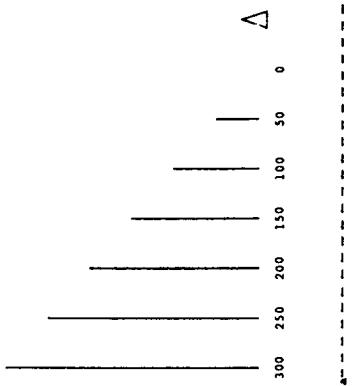
1. Do each of the commands in the menu.

2. Try doing 'line' and 'move' more than once.

3. Change the numbers in the 'length' and 'amount' boxes and do 'line' and 'move' again.

challenge

4. Make this pattern



Data

line

```
pu rt 90
fd length
mark
```

mark

```
pu bk 25
type length
```

dot

move

```
pu rt 90
fd amount
lt 90 pd
```

amount

70

data

start

```
pu setxy x y pd
x -200
y -100
```

x

y

data

menu

```
cs
start
line
move
```

data

## Variables-4:

Variables-4

- Do each of the commands in the menu.

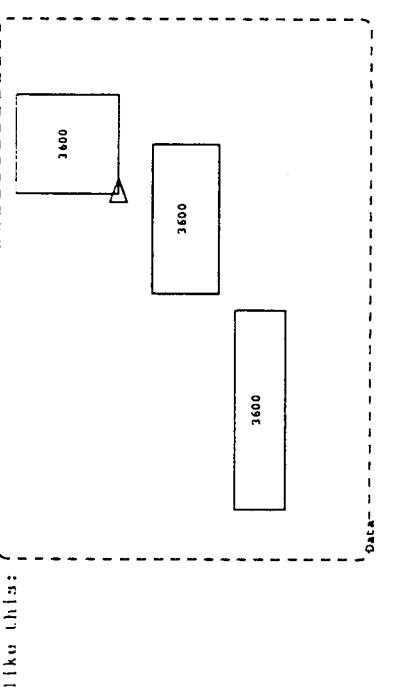
Data

- Make different sized rectangles.

Data

challenge

3. Make 3 rectangles with the same area  
like this:



Δ

menu

```
cs
start
rectangle
move
st
```

rectangle

```
rt 90
repeat 2
  fd x lt 90
  fd y lt 90
change area x * y
move
type area
```

start

```
pu
setxy x y
pd
x -120
y -70

```

move

```
pu fd x / 2
lt 90 fd y / 2
pd
```

move

```
pu
rt 90
fd bit
lt 90
fd bop
pd
bit
bop
50
18

```

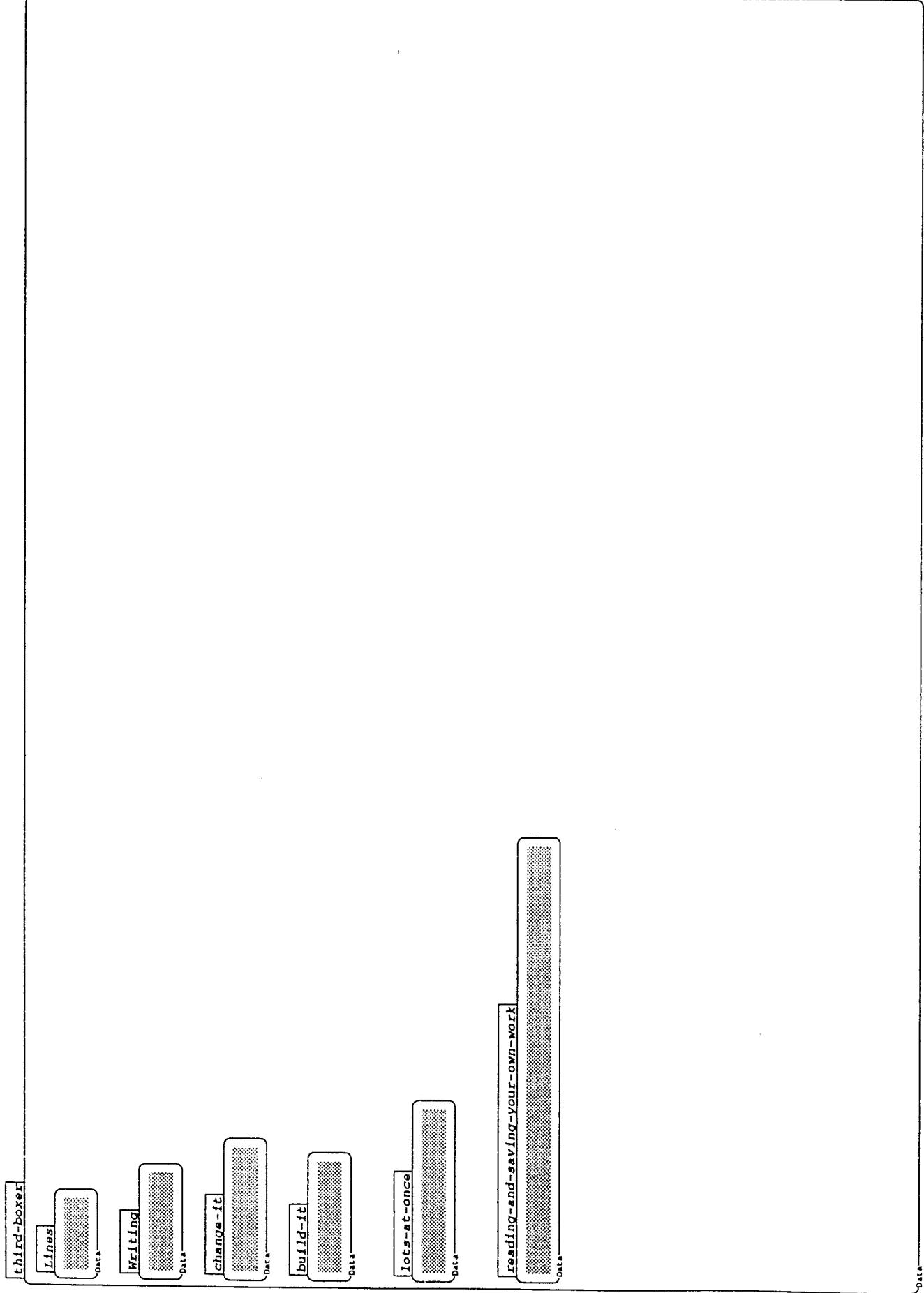
readme

```
pu
rt 90
fd bit
lt 90
fd bop
pd
bit
bop
50
18

```

**Appendix 4:**  
**Third Boxer**

## Third-Boxer Overview



## Lines:

Lines

menu

read-me

1. Try doing all the commands in the menu.

cs  
start  
line  
move

2. Now change the number in the Data-box called WIDTH inside LINE, and do line and move again.

challenge

3. Make this pattern

Δ

line

set-pen-width width  
fd length  
width  
length  
1  
50

move

pu  
rt 90 fd bit lt 90  
fd bit  
pd  
bit  
32

start

pu  
setxy x y  
pd  
x  
y  
100  
50

## Writing:

**read-me**

1. Try doing all the commands in the menu.
2. Now change the number in STYLE inside WRITE and do line, write and move again.
3. Try changing the word inside MESSAGE also.

**challenge**

3. Make this pattern

**menu**

```
cs
start
line
write
move
st
ht
```

**line**

```
pd set-pen-width width
fd length bk length
width
length
```

**start**

```
pu
setxy x y
pd
x
y
-200
-100
```

**write**

```
set-pen-font style
px type message
style
message
4
mo
note
```

The fonts go from 1 to 10 only. 1 is the biggest.

**move**

```
pu
rt 90 fd bit lt 90
fd bop
pd
bit
70
50
Data
```

## Change-it:

change-it

display

menu

change window faye  
change display joanna  
change store fajo  
change display 1  
change window 4  
change store window + display  
change store store + 20  
printing-precision 1  
counter  
clear-all

Read-me

challenge

1. Try doing all the commands in the menu.

2. Make the DoIt box called COUNTER count up in fours starting at 10 and ending at 74. Make it show the numbers in window.

change display  
change window  
change store

clear-all

counter

change display 0  
repeat 15  
  change display display + 2  
  sleep 0.5  
  redisplay

## Build-it:

The interface shows a Scratch script editor with the following components:

- Title Bar:** "build-it"
- Script Area:** Displays a Scratch script with the following code:

```
clear-all
change display [How [data] are you? [data]]
change window join-right display window
change store join-bottom store [is Bugs Bunny [data]]
counter
```

Below the script area, there is a **challenge** section containing two challenges:

  1. Try doing all the commands in the menu.
  2. challenge  
Make the Doit box called COUNTER count up in threes starting at 8 and ending at 89  
Make it show all the numbers in display.
- Scratch Stage:** Shows a stage with a window titled "menu" containing three buttons: "display", "window", and "store".
- Scratch Script Area:** Shows a script with the following code:

```
clear-all
change display [How [data] are you? [data]]
change window join-right display window
change store join-bottom store [is Bugs Bunny [data]]
counter
```

Below the script, there is a **challenge** section containing two challenges:

  1. Try doing all the commands in the menu.
  2. challenge  
Make the Doit box called COUNTER count up in threes starting at 8 and ending at 89  
Make it show all the numbers in display.
- Scratch Stage:** Shows a stage with a window titled "menu" containing three buttons: "display", "window", and "store".
- Scratch Script Area:** Shows a script with the following code:

```
clear-all
change display [How [data] are you? [data]]
change window join-right display window
change store join-bottom store [is Bugs Bunny [data]]
counter
```

Below the script, there is a **challenge** section containing two challenges:

  1. Try doing all the commands in the menu.
  2. challenge  
Make the Doit box called COUNTER count up in threes starting at 8 and ending at 89  
Make it show all the numbers in display.

## Lots-at-once:

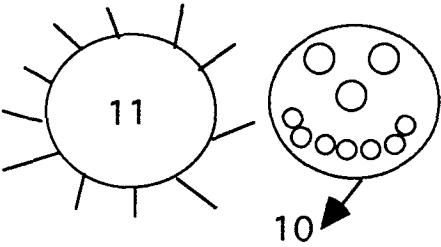
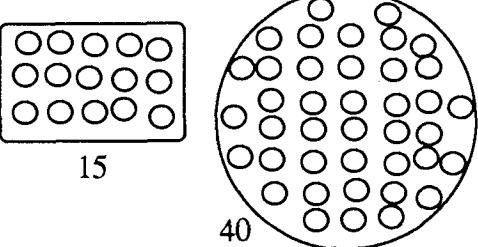
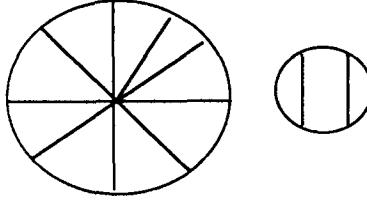
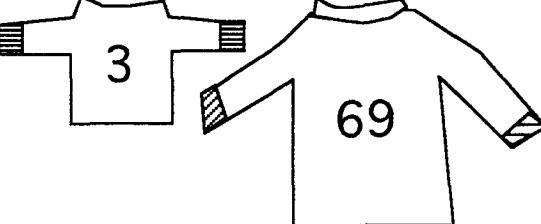
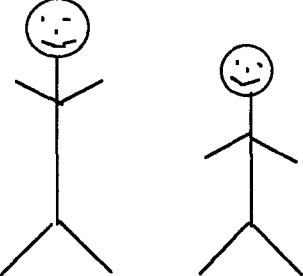
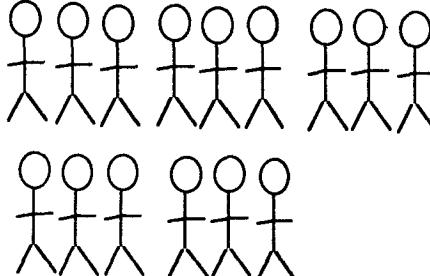
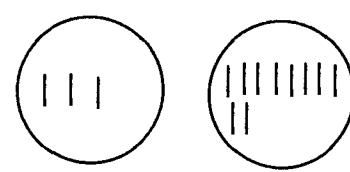
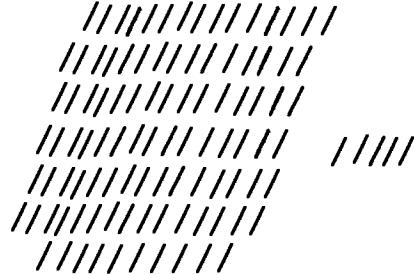
```
graph TD; Lots[Lots-at-once] --> display[display]; Lots --> menu[menu]; display --> Data1[Data]; menu --> changeNums1[change nums]; changeNums1 --> Data2[Data]; changeNums1 --> function[function]; changeNums1 --> clear[clear]; changeNums1 --> ReadMe[Read-me]; changeNums1 --> challenge[challenge]; challenge --> task1[1. Try doing all the commands in the menu.]; challenge --> task2[2. Put all of the numbers from 1 to 10 in 'nums.']; challenge --> doit1[Doit]; challenge --> doit2[Doit]; challenge --> Data3[Data]; challenge --> clear2[clear]; clear2 --> changeDisplay[change display]; changeDisplay --> Data4[Data];
```

Make the Doit box called function  
change each of these numbers to:  
(i) 1.3 2.3 3.3 4.3 5.3 6.3 7.3 8.3 9.3 10.3  
(ii) 0.9 1.9 2.9 3.9 4.9 5.9 6.9 7.9 8.9 9.9  
(iii) 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

```
function  
printing-precision 2  
clear  
for-each-item n nums  
    change display join-right display [n * 2]  
    doit  
    n  
    Data  
    doit
```

## **Appendix 5: Students' Representations**

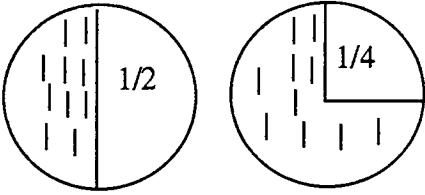
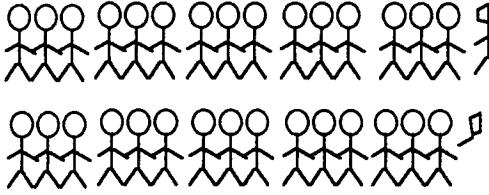
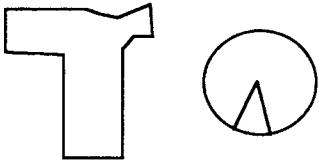
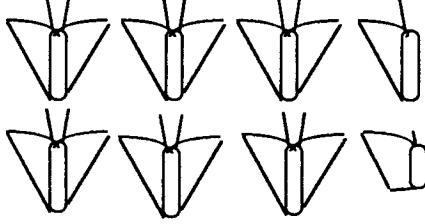
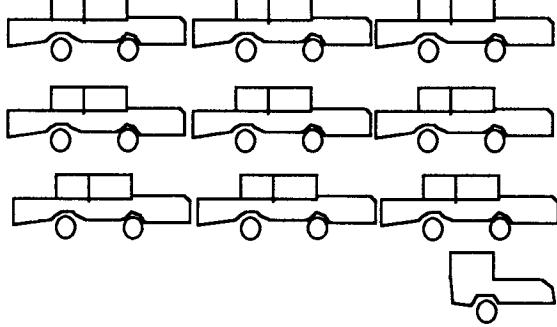
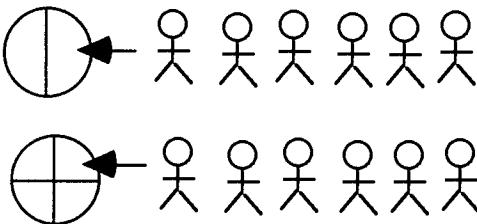
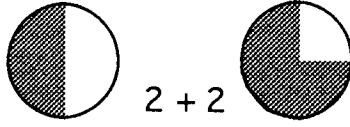
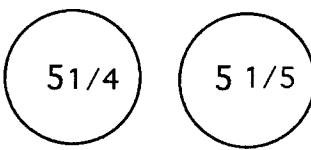
The Comparison Group's Representations of Natural Numbers

 <p><u>Ann-Maria's drawing to compare 11 and 10.</u></p>	 <p><u>Tahir's drawing to compare 15 and 40.</u></p>
 <p><u>Blake's drawing to compare 9 and 3.</u></p>	 <p><u>Hannah's drawing (of jumpers) to compare 3 and 69.</u></p>
 <p><u>Ahmet's drawing to compare 15 and 9.</u></p>	 <p><u>Cerisse's drawing to compare 9 and 6.</u></p>
 <p><u>Rebecca's drawing to compare 9 and 6.</u></p>	 <p><u>Dillon's drawing (of rain) to compare 100 and 5.</u></p>

## The Boxer Group's Representations of Natural Numbers

$\begin{array}{c c} T & U \\ \hline 1 & 5 \\ 0 & 8 \end{array}$	
<u>Joanna's 'drawing' to compare 15 and 8</u> 	<u>Nico's drawing to compare 3 and 9</u>
<u>Jason's drawing to compare 1 and 11.</u> 	
<u>Courtney's drawing to compare 5 and 18.</u> 	<u>Laura's drawing (of weighing scales) to compare 1 and 2</u>
<u>Jessie's drawing to compare 2 and 4</u> 	<u>Kathleen's drawing (of people) to compare 12 and 9.</u> 
<u>Rosie's 'drawing' to compare 6 and 12</u> 	<u>Sara's drawing (of people) to compare 2 and 3.</u> 

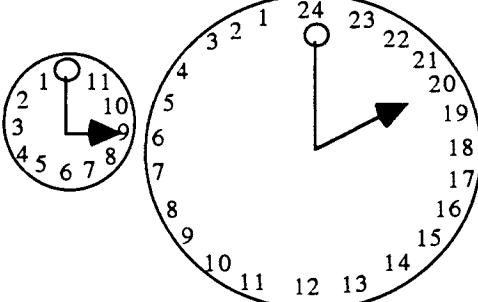
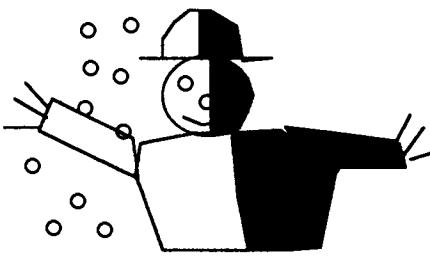
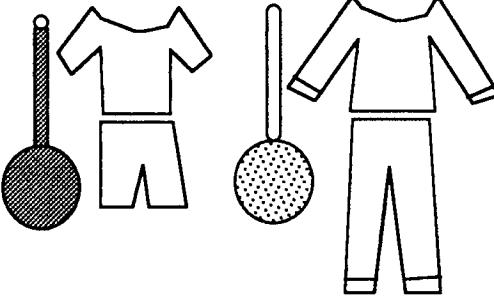
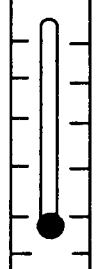
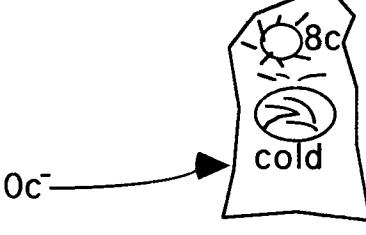
## The Comparison Group's Representations of Rational Numbers

 <p><u>Ann-Maria's drawing to compare <math>10\frac{1}{2}</math> with <math>10\frac{1}{4}</math></u></p>	 <p><u>Tahir's drawing to compare <math>15\frac{1}{2}</math> with <math>15\frac{1}{8}</math></u></p>
 <p><u>Blake's drawing (of half a vest) to compare <math>3\frac{1}{2}</math> with <math>3\frac{1}{8}</math></u></p>	 <p><u>Hannah's drawing (of butterflies) to compare <math>3\frac{1}{2}</math> with <math>3\frac{1}{4}</math></u></p>
 <p><u>Ahmet's drawing (of cars) to 'compare' 9 with <math>9\frac{1}{2}</math></u></p>	 <p><u>Cerisse's drawing to compare <math>6\frac{1}{2}</math> with <math>6\frac{1}{4}</math></u></p>
 <p><u>Rebecca's drawing to compare <math>4\frac{1}{2}</math> with <math>4\frac{1}{4}</math></u></p>	 <p><u>Dillon's drawing to compare <math>5\frac{1}{4}</math> with <math>5\frac{1}{5}</math></u></p>

## The Boxer Group's Representations of Rational Numbers

<p>8.0 8.25 8.23 8.10 8.05</p>	
<u>Joanna's 'drawing' to compare 8 and 8.25</u>	<u>Nico's drawing to compare 3 and 3.5</u>
<p><u>Jason's drawing to compare 1 and <math>1\frac{1}{5}</math></u></p>	<p>19 19.1 19.2 19.3 20</p>
<p><u>Courtney's drawing to compare 5 and 5.1</u></p>	<p><u>Liam's 'drawing' to compare 19 and 19.1</u></p>
<p><u>Jessie's drawing to compare 2 and <math>2\frac{1}{4}</math></u></p>	<p><u>Kathleen's (incomplete) drawing to compare 9 and <math>9\frac{1}{2}</math></u></p>
<p>6.00 6.10</p> <p><u>Rosie's 'drawing' to compare 6 and 6.1</u></p>	<p><u>Sara's drawing to compare 2 and <math>2\frac{1}{4}</math></u></p>

## The Comparison Group's Representations of Directed Numbers

 <p><u>Ann-Maria's drawing to compare temperatures of 9 and 20 degrees</u></p>	 <p><u>Tahir's drawing to compare temperatures of 1/2 and 1 degree</u></p>
 <p><u>Blake's drawing to compare temperatures of 3 and -1 degrees</u></p>	 <p><u>Hannah's drawings to compare temperatures of 0 and -20 degrees</u></p>
 <p><u>Ahmet's drawing to compare temperatures of '-0' and 'freezing'.</u></p>	 <p>how ever high the circle goes the colder it is.</p> <p><u>Cerisse's drawing to compare -1 and 0 degrees.</u></p>
$8 + 8 + 8 \\ + 4 + 4 \\ 2+2+2+2+2+2+2+2+2+2$ <p><u>Rebecca's 'drawing' to compare 8 and 24</u></p>	 <p><u>Dillon's drawing to compare temperatures of 8 and 0 degrees</u></p>