Differential Group Composition and Children's Problem Solving Using Logo

E. Claire Chadwick

A Thesis

in

The Department

of

Education

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Arts at
Concordia University
Montréal, Québec, Canada

March 1986

ISBN 0-315-30642-4

ABSTRACT

Differential Group Composition and Children's
Problem Solving Using Logo

E. Claire Chadwick

This study examined what bearing group composition had on efficiency and programming style of 11-year-olds using Logo Turtle Graphics for problem solving. The 36 subjects were divided into 18 groups comprising one, two or three members of the same or mixed sexes. Instructional materials were developed and used to provide all groups with the requisite level of skills. Each group was then free to choose its problem-solving strategy to achieve the three given, fixed goals involving making the Logo turtle draw a copy of a simple line drawing. Observational methods were used to measure efficiency, defined as final score divided by time to completion, and to investigate programming style which included the type of programming, the mode(s) used, the level of planning, and the sequence in which the graphic components were programmed. Results indicate that neither group size nor sex composition has a significant effect on efficiency or programming style. It appears that students can successfully work together in a cost-effective group setting and at the same time enjoy the cognitive and social benefits if offers. Some suggestions for further research in the area of group work are included.

ACKNOWLEDGEMENTS

I would like to acknowledge all the people whose help and encouragement have made the completion of this thesis possible.

I would first like to thank the principal, staff and students of Roslyn School in Westmount, Quebec, for allowing me to conduct the study, and Edgar Guntermann for his help during the project.

My sincere thanks and appreciation go to Professor Jesus Vazquez-Abad for his valuable help and advice throughout his supervision of this thesis.

Lastly, I would like to thank Patricia Chwartkowski for her moral support, and my family for their understanding and tolerance of all my extended absences.

TABLE OF CONTENTS

PICTURE

CHAPTER 1

Introduction

Over the last decade, the field of education has experienced the effects of considerable social and commercial pressures to make more use of microcomputers. We have witnessed an influx of these valuable resources and it appears that the continued presence of this innovation is inevitable. What is now required is an attention to the educational and sociological factors that affect our ability to fully exploit these resources (Rushby, 1981). A major concern is how this new technology may be successfully integrated into the schools. Is it being used in a way that offers optimal benefit to those concerned? Students at all levels are being initiated to computers; this introductory experience must be successful if it is to foster a positive attitude towards computing (Coburn, Kelman, Roberts, Snyder, Watt & Weiner, 1982). Of the methods of introducing students to a computer, one that has attracted widespread attention and sustained its popularity is the programming language Logo, in particular Logo Turtle Graphics.

The many uses for computers in schools have been classified in several ways, one of which reflects whether the student is controlled by the computer as with traditional types of computer assisted instruction (CAI) such as drills or tutorials, or is himself in control of the computer. The latter can be the case when working with Logo. The student interacts with the computer which provides him with immediate feedback to his programming. Several claims have been made that such programming with Logo facilitates the development of problem-solving skills (Watt, 1982; Papert, 1980). Problem solving, in a broad sense, can be viewed as performing a series of actions to reach a goal for which there is no immediately accessible solution. One problem-solving strategy that is extremely effective involves breaking problems into small parts and solving these sub-problems in a systematic way. Such a strategy may be called procedural programming; the small parts of the problem are solved in subprocedures that collectively form a superprocedure to solve the whole problem. Although Logo Turtle Graphics easily facilitates procedural programming, problems may also be solved by another strategy which involves more linear or unstructured programming. Whichever approach is adopted, the student is usually involved in programming the computer to draw a representation of a predetermined but not necessarily fixed goal.

It has been suggested that tasks involving problem solving (Durling & Schick, 1976) and particularly CAI problem solving (Lathrop & Goodson, 1983) lend themselves well to group work. Most research on the Logo environment indicates that students usually work in small, collaborative groups to jointly produce a project or goal (Krasnor & Mitterer, 1983). This has been identified as a very positive element in the students' learning of cognitive and social skills (Krasnor & Mitterer, 1983; Papert, 1980;

Watt, 1982; Hawkins, 1983). Many other claims have been made for Logo, not the least of which is that, as a tool, it offers great potential for our educational system. In recognition of such, Logo has been readily adopted by schools on a very large scale. However, as Leron (1984) points out, the eventual success of Logo depends on how we use it. Many of the issues concerning the everyday practice of Logo have yet to be addressed. Given that we know that most Logo sessions involve group work, this study investigates the question of how groups can be organized to offer optimal benefit to the group members. Does a particular group composition enhance performance over that of another? Research to date on Logo has given little if no indication as to optimal group size or composition and there has been no investigation into group size and composition relative to certain kinds of problems. This issue should be addressed if we are to determine how teachers may best manage their Logo sessions, a decision that has, up to now, been left largely to the individual teacher's intuition.

This study investigates the effects of a group's composition on certain aspects of its performance and behaviour.

The Statement of the Problem

The study focusses on how 11-year-old children solve a given problem using the Logo language on a computer. Both performance (in terms of the product) and behaviour (in terms of the process) are considered:

1. Do children work more efficiently alone or in groups of two or three and does the sex composition within the group affect efficiency?

2. Does a group's composition affect the programming style of its members when there is no direct intervention?

The possible disadvantages of group work such as reduced efficiency (Freedman, Carlsmith & Sears, 1970) and potential personality conflicts seem to be outweighed by the advantages. Members can benefit from the group's pool of abilities (Freedman et al., 1970) and from other members' feedback. Group work can also lead to the generation of new ideas (Torrence, 1970).

Of course where work with computers is concerned, cost-effectiveness has dictated that students work in groups. However, with the improved ratio of computers to students we now need to consider what the ideal group situation is. Boyd et al. (1983) have discussed the importance of students working together to develop social skills, irrespective of the number of computers available.

Research in this area with CAI is limited mostly to studies that have shown no significant differences in achievement, be it adults working in groups of one to four (Cartwright, 1972; Okey & Majer, 1975) or high school students working alone or in pairs (Lebel, 1982; Love, 1969) or in groups of one to three (Karweit & Livingston, 1969).

Some differences were found in these studies; Karweit and Livingston (1969) concluded that boys worked more quickly than girls on a simulation game. Okey and Majer (1975) found that learning efficiency increased with group size so that groups of three or four required the least time. Trowbridge and Durnin (1984) however, found that a group of four seemed to be too large while Lathrop and Goodson (1983) claim that three students is the ideal number for working at a computer. Would this be the case in a non-CAI situation where the students are working with Logo and hence have more latitude in terms of tackling the problem solving task? A group of three could benefit from the opportunity to collaborate.

On the other hand, working with Logo generates a lot of communication among the students (Hawkins, 1983; Nelson, 1981). While this could cause distraction for some students and general disruption in the classroom, Durling and Schick (1976) demonstrated that an opportunity to vocalize to another person was crucial to the success of problem solving in a college level population. How would the level of efficiency be affected if the group were to include a third member? Would the additional collaboration be detrimental to an efficiency measure or would the possible disadvantage of extra discussion time be outweighed by the superior achievement scores resulting from the sum of more students' input?

The second question to be considered is to what extent the group's collaboration affects the process by which the group arrives at its goal? Observations of 12-year-olds' Logo programming behaviour show that most students, including those who have previously written procedural programs, when given a choice, use a linear approach; this is characterized by trial-and-error, semi-random work at the keyboard. There is no pausing to look back or plan ahead (Leron, 1984). Indeed, it is very common for students to proceed with very little or no pre-planning (Tetenbaum & Mulkeen, 1984; Leron, 1984; Hillel, 1985). Hoyles (1985) suggests that planning and programming style appear to depend on the level of abstraction or definition of the goal. When the goal is not clearly defined and fixed, for example, students who experience the least difficulties in achieving the goal will often scrap what they have done and restart or modify the goal, rather than revise the plan (Tetenbaum & Mulkeen, 1984; Hillel, 1985; Pea, 1983). When students have a clearly defined task they often do not follow a strict sequence of planning, hands-on, then debugging (Hoyles, 1985).

Another common trend in Logo programming is the overwhelming use of direct mode during which students continuously revise their plan, contingent on the immediate feedback provided by Logo's direct mode (Hillel, 1985). This involves alternate 'planning-in-action' and 'debugging-in-action'. The latter refers to revision made on a very local basis (Hillel, 1985).

This study investigates how students in different group compositions go about programming. In the particular task environment chosen, 11-year-old students in a school setting are provided with the necessary tools to solve a problem that is well defined and fixed. They are then given a free rein to programme as they choose. The investigation focusses on both performance (product) and behaviour (process). Does group size and composition have an effect on the efficiency of the problem-solving activity? Do students use procedural or linear programming. Does their work involve pre-planning or planning-in-action and does any planning occur on a local or global basis? Lastly, does group composition affect the sequence in which the task components are programmed?

CHAPTER 2

Review of Related Literature

The discussion of related literature is presented under the headings: Logo programming language, computer assisted learning, problem solving, grouping, grouping and Logo, efficiency, programming style and observational research study design.

Logo Programming Language

Logo is a computer language developed by S. Papert and W. Feurzeig in the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology in the late 1960s (Papert, 1980). Logo is reknowned for its procedural nature and its powerful editing facilities but for many the primary advantage is the subset of Logo called Turtle Graphics. This offers a visual dimension to programming in an interactive environment. The user manipulates the cursor, known as the turtle, on the screen by typing in commands. The turtle can draw lines to create geometric shapes. Feedback is immediate so any errors can be corrected as they occur.

Computer Assisted Learning: Frameworks for the Application of Computers in Education

Rushby (1979) characterizes CAL as a "flow of rapidly changing, very detailed information with the computer playing a prominent role as mediator" (p. 22). He then discusses computer assisted learning (CAL) using the framework proposed by Kemmis, Atkin, and Wright (1977). Four paradigms are given to relate computer assisted learning to the field of education: the **instructional** paradigm includes instructional dialogue and drill and practice; the **revelatory** paradigm is characterized by the student's learning through a process of discovery, as with simulations. The computer acts as a mediator between the student and a 'hidden model of some real-life situation' which allows the student to discover the rules that govern the model. Whereas instructional CAL focusses on the subject matter and the student's mastery of it, revelatory CAL focusses much more on the student and his interaction with the subject matter as presented by the computer. The **conjectural** form of CAL is that which helps the student to manipulate and test his ideas and hypotheses. Knowledge is said to be created through the student's experience while he explores information on a certain topic. He also controls the learning and is involved in instructing or programming the computer; Logo is an example of this form of CAL. Finally, the **emancipatory** paradigm is concerned with helping the student reduce his workload of 'inauthentic' work in his learning. For example, he can use the computer to browse through information, to retrieve it or to generate accurate calculations.

Taylor (1980) suggests a framework for the application of computers to education that differentiates between using the computer in one of three modes. The computer functions as a tutor, a tool or a tutee.

The **tutor** mode has its roots in programmed instruction and is often known as CAI. The student is tutored by the computer which executes a program. The computer presents material to which the student responds then, from its evaluation of the student's response, the computer determines what to present next. Some programs are designed to store, analyze and act upon student responses, enriching the nature of the tutorial. The use of the computer as a tutor requires students to play a rather passive role as compared to the student's more active involvement in, for example, programming in Logo.

The **tool** mode includes the use of the computer to perform functions such as word processing, statistical analysis or accounting. It is often used to save time and intellectual energy in that many tedious, time consuming jobs can be transferred to the computer.

To function as a **tutee**, the computer has to be tutored by its user. One very common example of this is Logo. According to Papert (1980), when children are programming the computer they are teaching it how to think and can therefore explore and extend their own thinking.

In his analysis and categorization of CAL, Boyd (1982) identified four modes. He placed working at a computer with Logo in what he calls an **auto-elaborative CAL** mode. This mode encompasses developing problem-solving skills, learning abstract thinking and organizing one's own learning more than it emphasizes learning facts. An essential element is the question of how much and what kind of guidance should be provided to the student. With Logo, because no built-in guidance is available, it has to be provided by the teacher so it varies depending on the situation.

One of Boyd's other categories is **personally-guided small group CAL**. The teacher, as leader and subject matter expert, holds discussions or 'post-mortems' with groups of learners, guides them in their choice of work and helps them to discuss any difficulties. The type of programs used are usually modelling, simulation, problem solving or data analysis. Boyd states that "the main point to note is that the leader and group members all learn from each other's non-verbal and verbal communication and through the stages of group-identity-formation, as well as from the computer programs" (p. 307). It seems that an element of this CAL mode is present in many Logo-learning situations. The teacher may not necessarily be an expert on Logo but because guidance to the student is not built in, he may hold small group discussions, guide students in their choice of projects and help them discuss their difficulties.

The **pre-packaged computer assisted instruction (CAI)/computer managed** instruction (CMI) is Boyd's term for materials that develop factual knowledge and reproductive skills. The fourth mode is called **gated co-operatively-elaborated CAL**. Students are involved in a cooperative effort of producing studyware. The term 'gated' is used to indicate that those wishing to participate must show they have the requisite skills before they are accepted as a contributor.

The frameworks discussed so far appear to contain common elements. Only the gated co-operatively-elaborated CAL mode (Boyd, 1982) has no counterpart in the other frameworks presented. It is difficult to estimate the extent of this mode's use in education but Boyd provides some examples of its use. University students have produced materials for other students; professors and teachers have produced CAI materials, some of which have been shared with others so that the authors could build on each other's work.

The common element of the emancipatory paradigm (Kemmis et al., 1977) and the tool mode (Taylor, 1980) is that of helping the student to reduce the amount of time and intellectual energy expended on tedious work. The most widely exploited educational application would be in the use of word processing. The also widely used drill and practice type of application can be placed in what Taylor (1980) calls the tutor mode, Boyd (1982) the pre-packaged CAI/CMI mode, and Kemmis et al. (1977) the instructional paradigm. They are all characterized by a focus on the student's mastery of the subject matter and by the computer's control over the student's learning.

Anderson (in Rushby, 1981) discusses the role of control in providing the student with what he calls the 'right' kind of experience of computers. He suggests that we should discourage computer development in which the student's learning is constrained or paced by the computer because we deny the student the experience of control. The learner should be allowed to make "self-selecting decisions and to manipulate the informational substance of the discipline of knowledge being studied" (Rushby, 1981, p.8). This element of the student's controlling the experience can be found in the revelatory CAL (Kemmis et al., 1977) where the student learns through interaction with and discovery of the subject matter, but more so in conjectural CAL (Kemmis et al., 1977), the tutee mode (Taylor, 1980) and the auto-elaborative CAL mode (Boyd, 1982). In these modes the learner can explore his own thinking, test his own hypotheses and actively participate in and control his own learning. It is this type of application of computers that forms the basis of this study, with Logo, currently the most widely used example, serving as the main focus.

Problem Solving

Logo has been used for various purposes in the educational setting. It has served to introduce students to a programming language and computer literacy, and has provided a basis for learning mathematics and other subjects, including music, language arts, fine arts, physics and biology (Watt, 1982). It has also attracted particular attention due to its so-called facilitation of the development of problem-solving skills (Watt, 1982; Feurzeig

& Papert et al., 1969).

A problem has been defined as:

> a situation in which an individual or group is called upon to perform a task for which there is no readily accessible algorithm which determines completely the method of solution (Lester, 1978, p.29).

The process of solving a problem can be broadly defined in the following ways which are similar.

Ausubel defines problem solving as:

> any activity in which both the cognitive representation of prior experience and the components of a current problem situation are reorganized in order to achieve a designated objective (Ausubel, 1968, p. 533).

According to Hartley and Lovell:

> problem solving implies a novel situation for the student; he has the requisite knowledge and sub-skills to solve the problem but has to sequence his reasoning and/or develop heuristics which take him from the initial to the goal state (Hartley & Lovell, 1984, p. 38).

Of the cognitive based definitions of problem solving, Polya's seems to be of the most direct use for the purpose of this study. In his terms problem solving means:

> to search consciously for some action appropriate to attain a clearly conceived but not immediately attainable, aim. To solve a problem means to find such action (Polya, 1962, p. 117).

It is often the case, however, that students working with Logo modify their aims (Cathcart, 1985; Hillel, 1985) or indeed set out with no fixed goal (Solomon, 1982; Pea, 1983). This situation does not arise in the present study due to the nature of the task. It is therefore necessary to modify Polya's term "clearly conceived" with "fixed" aim.

Polya's work is also of particular interest in describing the process of problem solving with Logo. He states that problem solving involves four phases: understanding the problem, devising a plan, carrying out the plan, and looking back (Polya, 1945). He notes that the problem solver would likely progress in this order but it is possible that he might skip a phase he found unnecessary, or he may decide to recommence at phase one. This choice of action is open to the Logo problem solver and is one of the general elements of interest in this study. In short, the aim is to place students with the requisite skills in a novel problem-solving situation and investigate their choice of action to achieve a given fixed goal.

Grouping

The question of how groups compare with individuals in problem solving has attracted a great deal of attention.

Depending on the conditions, having another person present may be a disadvantage or an advantage: when a group is working on a problem together, both distraction and stimulation may occur.

In their discussion on groups and problem solving, Freedman, Carlsmith and Sears (1970) claim that under most circumstances groups are less efficient than individuals who work alone. Group members "distract, inhibit and generally tend to interfere with one another" (p. 139). On the other hand, the fact that they are better at spotting errors may outweigh this relative inefficiency. Another disadvantage of group work is that depending on the particular characteristics of the individuals, conflict may arise among members and have a destructive effect on the group. Boyd et al. (1983) found this was not the case in their observation of 15-to 17-year-olds working in pairs at a computer. Even though students were randomly assigned a partner, they were sufficiently involved in the problem-solving activity to put aside any personal differences; no conflicts occured.

According to Freedman et al. (1970), group work provides two major advantages. Individually, members can check each other's work, and collectively, they can provide a much wider range of abilities than one person could. The authors also point out that these characteristics of group work are more advantageous to certain types of problems than others, for example, problems that involve a large number of separate operations such as complex mathematics. In this type of situation, the group members really work alone and the group works together to check each other's work.

There are other benefits derived from group work. Feedback from other members of the group can help in the formulation and reshaping of ideas. Members may also discover new information they would otherwise have missed if working alone. There are also social benefits to working in a group. One is the cooperation that the situation fosters. Also some students find the group setting less threatening.

Boyd, Douglas and Lebel (1983) discuss the importance of a type of 'sociostructure' for educational CAL where "social cooperativeness and responsibility goals are always as important as the specific individual skills acquired" (p. 1). The authors point out that inattention to the educational impact of the sociostructure leads to a 'closet computer queen' phenomena: "...the intelligent pupil whose social exchange skills are negligible.... is sent down the hall to work on the Apple or PET in the broom closet and ... develops ever more technical competence at the expense of social competence" (p. 1).

In a discussion about elementary school children using a variety of CAL programs, Conlin (1981) also argues the importance of children not working alone on a computer. She stresses the linguistic benefits. Working together generates an enormous amount of language, even in quiet children.

Research findings in general on group versus individual work seem to indicate no detrimental effects for those students who work together. On the contrary, in many instances group work seems to be advantageous. This is the case for students who are working on problem solving.

Torrence (1970) conducted two experiments on the influence of pair interaction on creative functioning. In one study, 5-year-old children working alone and in pairs were asked to 'hitchhike' on one another's ideas. They were scored for fluency, flexibility and originality. Torrence noted that the children in pairs "were only occasionally sparked by one another's ideas and continued for the most part along their own tracks of thinking" (p. 393). However, the 'sparking of ideas' did occur with sufficient frequency to make a difference in originality.

In the second study with college educational psychology students, the tasks involved: asking questions, guessing causes, guessing consequences and improving the product. Again, the students in pairs were asked to 'hitchhike' on one another's ideas. Responses were scored in the same way as with the children. The results for the adults followed the same pattern as those for the children. There were differences in favour of subjects working in pairs on all three variables but the differences were most apparent on originality. In both studies, "results strongly support the hypothesis that dyadic interaction stimulates individuals to produce more original responses than they are able to produce working alone" (Torrence, 1970, p. 393). Also, both the child and adult subjects working in pairs seemed to persevere longer than those working alone. Torrence further notes that the adult pairs seemed to be having fun whereas subjects working alone seemed to be fatigued and ready to stop.

For Johnson and Johnson (1974), their research clearly showed that a cooperative goal structure is the most desirable for increasing group productivity and for promoting achievement in problem-solving tasks. It results in higher achievement than a competitive one. They further point out that when working together on a problem-solving task, students learned not only how to problem solve but also how to cooperate and work with other individuals to address a common problem or accomplish a common task.

From research findings related to groups or individuals working on computers, there is little, if any, evidence to suggest that groups perform differently to individuals. Most of the research in this field has focussed on adults or high school students working with tutorial CAI. Few studies have considered differences in younger students working alone or in a group on CAI and there are no research findings available that relate directly to Logo and differential grouping.

Cartwright (1972) conducted a study on the use of group work in CAI. The subjects were students in an introductory educational psychology course. They worked alone, in pairs, or in groups of three or four. Subjects in each condition worked on a series of CAI lessons and then later responded individually to a multiple choice test. Cartwright found no significant differences in means or variances of learning scores among the four treatments. He suggested that "the group use of CAI may be implemented as a legitimate (and less expensive) alternative to individual CAI, without affecting performance" (Cartwright, 1972, p.401).

In Love's study (1969) of learning achievement differences between students working individually and in pairs, students from grades 9 to 12 were instructed in Boolean algebra using CAI. The pairs were selected by mutual choice. Results showed that the paired subjects performed as well as the individuals. Both members of the pair learned during CAI.

In Karweit and Livingston's study (1969) 6th-grade students played a simulation-type computer game that required them to make decisions about the number of employees they would hire and the price they would charge for a product in a surfboard manufacturing company. The subjects were all of high academic ability and familiar with computer games. They were divided according to sex and randomly assigned to one of four conditions: a group of three, a pair, individual and a control. The criterion used to measure performance during the game was based on the player's net assets. All subjects took a post-test that measured learning of certain economic relationships, accounting concepts and price-setting strategies. Results showed no significant differences on either criterion between the students who played in groups and those who played individually. There was, however, a significant difference between the speed with which the students played the game: the boys were faster than the girls.

One study of 7th-and 8th-grade students showed that small group usage of interactive computer-based learning material had certain advantages over individual usage. Trowbridge and Durnin (1984) investigated interactivity as a function of group size. The students worked individually or in groups of two, three or four. They were asked to manipulate pictures on the screen with the objective of discovering the concepts regarding current flowing through a complete circuit. Video, audio and key push components of the group activity were recorded. The students took a pre- and immediate post-test quiz, then a delayed post-test that consisted of the previous post-test plus a brief interview that required a practical demonstration. Results showed no evidence of any detrimental effects among students working in pairs or groups of three. The groups of four, however, seemed to be too large for all members to maintain high levels of interactivity with either the program or with other group members. Students working in groups tended to interpret questions presented in the program as the author had intended more often than the individuals. There was often discusssion about different interpretations which led to the correct version. Students working alone were more likely to misinterpret the questions and continue on an incorrect path, however, they often went back to review troublesome material. The authors suggest this may explain why the individual's level of performance was not lower than that of the other groups.

Lathrop and Goodson (1983) state that "programs dealing with logic and problem-solving skills lend themselves well to computer learning centers, because much of the learning takes place as a result of student interaction" (p. 6). They then discuss the merits of working in three different conditions. They claim that three students is the ideal number for working at a computer as they "often watch each other and discuss the process, developing their communication and social skills as well as their thinking and reasoning skills. These types of interactive exercises enrich the usual classroom environment" (Lathrop & Goodson, 1983, p. 6). Working alone may encourage the student to concentrate too much on the computer rather than on the process, while students working in pairs tend to take turns at the computer.

Group or Individual Setting and Logo

Although Papert (1980) has not explicitly advocated that children work in groups with Logo, he does present several pages of screen graphics and text accompanied by the "hypothetical conversation between two children who are working and playing with the computer" (p. 77). He then states that "these and other experiments can happen every day - and they do".

Krasnor and Mitterer (1983) also state that "in most implementations of the Logo curriculum, children share equipment and are encouraged to cooperate with peers to complete small group projects. This social arrangement can be considered another major component of the Logo environment" (p. 19).

In a discussion of 5-and 6-year-olds working together with Logo, Nelson (1981) notes that "children share their Logo experience with each other. They explain exactly what they are doing, tell what they will do next and show how to do it" (p. 15). Nelson also discusses the interaction of a group of 3rd-grade children who, in order to extend their computer time, decided to pool together to work in fours: "There are definite advantages to this arrangement. Even though they may take turns at the keyboard, the interaction, showing of ideas and group problem solving give the students an experience far richer than private 45-minute sessions could possibly be" (p. 16).

An observational study by Hawkins (1983) on the social features of working with Logo found linguistic benefits very similar to those presented by Conlin (1981) mentioned earlier. Hawkins concluded that children working together talked to each other more about their work particularly when doing computer programming tasks as compared to non-computer tasks. Also when the children were free to choose to work either individually or collaboratively, they favoured the latter when working on the computer.

Of the many studies mentioned that regard the subject of grouping, there are few that involve the use of Logo. The results from all studies, however, indicate that there are numerous advantages of group work while no evidence of disadvantages has yet emerged. With Logo, does a group of a certain composition work better than others? In order to investigate the role of group number it seemed appropriate to observe students working alone, in pairs and in triads. It was thought that four students in a group would be too many to allow them all access to the computer and that disruptive behaviour might result. Children usually work in groups of different sex composition yet there is little mention of how this factor may affect the way the children work. A variation in sex composition and the number of students in the group constitute the two major factors of interest for this study.

Efficiency

The students in Hawkins's study (1983) seemed to appreciate the benefits of collaborative Logo work yet many also felt that working as a group can hinder getting the work done efficiently. Results from a questionnaire showed that "preferences for solitary work were dominated by a concern for getting something done with speed and efficiency, for not wanting to take time to explain to or negotiate with someone else" (p. 46).

Durling and Schick (1976), however, demonstrated that, for their college population, the opportunity to vocalize was crucial to the success of problem solving. Efficiency was measured by the number of card choices to solution and time to solution was also considered. Results showed that vocalization was a more important factor than group size in influencing successful problem solving. Efficiency in problem solving was superior for students vocalizing with a partner than for students working with a partner without vocalizing.

Okey and Majer (1975) investigated the differences in college students working alone, in pairs, or in groups of three or four at a PLATO CAI terminal. Time and achievement scores were the dependent measures. There were no significant differences found in achievement but there was a significant difference in the time required to complete the instruction. It was the pairs that required the most time as they had more frequent discussion; the groups of three or four required the least. The learning efficiency measure was calculated as the total achievement score divided by the total time at the computer. Efficiency was found to increase with group size.

Group size may or may not make a difference to efficiency but what about the other factor of a group's composition - the sex of its members? Results from a Logo mathematics case study (Hoyles, 1985) showed no observed sex differences in levels of achievement for 11 to 12-year-olds working in pairs. Students of this age range were studied in the Karweit and Livingston study (1969). Students worked on a computer in groups of differential composition and rate of play was used as the efficiency criterion. The computer task involved a simulation and game-months per hour were recorded. Here again, there were no significant differences in learning. There were, however, significant differences in the speed with which the students played. The boys were faster than the girls.

The efficiency measure used by Okey and Majer (1975) seemed the most appropriate for the present purpose; the task here was dissimilar to both the simulation discussed by Karweit and Livingston (1969) and the card choice activity in the study by Durling and Schick (1976). In this study the problem-solving process involved creating a product and the interest was focussed on the relationship between the quality of the product and the time required to produce it. This is because many factors, difficult to discern (let alone measure and control), could have intervened in the process. Indeed, the final score was obtained from a product and may therefore not reflect the difference in efforts spent on planning, discussion, correcting errors in a procedure and so forth. Thus it was felt that, for lack of a better measurement, the ratio between score obtained and time spent developing the solution would be most appropriate, and certainly better than looking at score or time alone. It is realized that such a ratio cannot truly be called "efficiency", as the efficiency of a system refers to a relationship between input and output. However, in order to remain consistent with the literature on the topic, for the purpose of this study, it is referred to as "efficiency", and the measure chosen for it was the task (product) score divided by the time to task solution.

Programming Style

Another aspect of different grouping effects on children's collaborative work with Logo that is perhaps of more interest is programming style.

Solomon (1982) proposes a model of what she calls learning style of students working with Logo. The first is labelled the planner because the student always works with a coherent, formulated plan. He may then build structured programs from bottom level up or from top level down. The second style is manifest in the macro-explorer student who "likes to mess about with subprocedures or building blocks to arrive at a product, rather than starting out with a specific goal" (p. 202). When the student has made something interesting from his 'wandering' he could be encouraged to make a procedure for it. Finally, the student as micro-explorer would explore the turtle environment on a micro level before planning or beginning more directed exploration. Solomon notes this type is often timid and needs to explore in a gradual, conservative mannner. This difference in level of style would appear not to be a function of age; Solomon described a 6-year-old as a macro-explorer and an 11-year-old as a micro-explorer. She further notes that students can be encouraged to switch modes, from perhaps micro-explorer to planner. Also any one student may use all three styles. It is not clear whether a student must progress from the micro to macro-explorer style to become a planner or if he fluctuates between the three, particularly if he were not influenced by intervention.

It would appear that, to a certain extent, style is dependent on the context in which the task is conceived and carried out. For example, students who have readily demonstrated an ability to plan and execute subprocedures often prefer not to do so. The reason Hoyles (1985) suggests for this is that when students feel the task involves using turtle graphics "as an extension of their drawing arm they do not perceive the need to divide their picture into subprocedures" (p. 42). If students were asked to draw a figure by hand, they would typically begin at the interior and work out to the exterior, adding details (L. Weisbord, art education specialist, personal communication, February 6, 1986). The student's choice of style may also depend on the amount of freedom he is given. Leron (1984) notes that 11 and 12-year-old students who can write neat, procedural programs and who know the theoretical advantages of doing so will tend to revert to linear programming if there is no direct intervention. This type of programming may be planned but it appears this is usually not the case, when students are left to establish their own style.

Linear programming is typically characterized by its trial-and-error 'hacking' approach which is described as "indiscriminate, semi-random striking on the keyboard, without pausing to plan ahead or to look back" (Leron, 1984, p. 6). According to Leron working in this 'hacking' style may be beneficial; it may serve as a natural, intermediate phase of exploration of Logo. Another common trend in Logo programming is the overwhelming use of direct mode (Hillel, 1985). Few children pre-plan prior to programming (Tetenbaum & Mulkeen, 1984) and any planning is usually what Hillel (1985) calls 'planning-in-action'. This is no doubt what Pea (1983) refers to as 'on-line' programming.

The Logo programming behaviour of the paired kindergarten children Munro-Mavrias (1985) studied showed these characteristics. The majority of the children did not pre-plan before they began typing and their programming was, for the mostpart, contingent on the feedback their saw on the screen.

The case study by Hoyles (1985) of paired 11-and 12-year-olds working on Logo projects concluded that students' work styles do not fit into the categories put forward by Solomon (1982). Hoyles indicated that planning and programming style appear to be not only context specific but also dependent on the level of abstraction or definition of the goal. Students do not follow a strict sequence of planning, hands-on, then debugging when there is a clearly defined task. One strategy that seemed effective involved the students' "continuously refining their plan by interspersing planning with 'hands-on' activity" (p. 59). This appears to coincide with one of the models discussed by Statz (1973). She states that in Karl Duncker's view, the "activity of revision is the key to the solution of a problem" and that the "entire process of solving a problem involves the continual restructing of a problem while seeking the solution" (Statz, 1973, p. 26).

In her study of 10-year-olds problem solving with Logo, Statz (1973) used a model of problem solving derived from Polya's work (1945). Polya's model includes the following essential steps:

1. defining the problem
2. devising a plan
3. gathering information
4. executing the plan
5. revising the plan
6. evaluating the results

Statz also notes that the steps need not be followed in this order and one or more may be omitted.

Working with Logo enables the student to execute or test an intermediate plan, get immediate feedback then use that information to revise the plan. If the students do not like the outcome it is at this point that they often scrap what they have done and restart or modify the goal, rather than revise the plan (Tetenbaum & Mulkeen, 1984; Hillel, 1985; Pea, 1983). Hillel (1985) points out that changing goals allows students to avoid their errors so they can always achieve something. However, he further notes that constant use of the avoidance option can limit the opportunity to learn from one's Logo errors. Consequently, students should occasionally be asked to deal with them. This was the case in the present study. The students were placed in a situation where they:

a. could not modify their goal (it was imposed upon them)

b. could not use CS (the clearscreen command that erases the screen) to scrap their work and restart as they knew there was a time constraint.

It seems that only under these circumstances would students necessarily be involved in the 'activity of revision' or 'restructuring of a problem' that Duncker (cited by Statz, 1973) proposes.

Although the goal is provided by the researcher, the process of reaching it is initiated by the students. One can therefore anticipate that the majority of the commands would be executed in direct mode, a process that Hillel (1985) suggests encompasses activities that include:

- exploration
- planning
- verification of programs
- adjustment of numerical inputs to commands

It is generally true that students work together in groups in this process. One of the aims of this study is to investigate how differential composition of groups affects programming style. Does the opportunity to collaborate on problem solving and vocalize one's thoughts affect work style? One would expect an individual's style to be influenced by the input from another group member. If this is so, it could explain why Hoyles (1985) did not observe the defined learning styles in her student pairs that Solomon (1982) claims were present in her students who worked individually. The students in a study by Cathcart (1985) also worked alone and the author notes that being able to talk through the problem (a response to a student's talking or question was seldom provided by the researcher) would perhaps have been a helpful debugging strategy.

Of the analyses of programming style discussed so far, those of Hillel (1985), Pea, (1983), Hoyles (1985), and to a certain extent, Statz (1973) appear to be the most relevant for the purposes of this study. The three learning styles discussed by Solomon (1982) emerged in a situation that involved teacher intervention and where the goal was chosen by the student and was not necessarily fixed. This was not the case in the present study. Although the concepts of Solomon are of interest, the research situations are dissimilar therefore it was considered that making further reference to her study would be inappropriate.

The model proposed by Statz (1973) involved six essential steps. While it was not the intention of this study to analyze programming style into discrete steps, the fifth step appears to be of particular interest to the type of problem-solving process anticipated in this study. Step five involves revising the plan where "a problem solver might test a solution by submitting it to constant revision until one of a set of successive approximations works" (Statz, 1973, p. 22).

It can be expected that when left to establish their own style of programming, students would use direct mode and few subprocedures. Would the composition of a group affect the level of use of subprocedures? If the group does not use indirect mode at all, of what does their programming behaviour consist in direct mode? Also can we assume that groups of varying composition would all choose not to plan their work or would some groups plan more than others? Does the students' discussion refer only to the task component the group is presently working on or do students verbally consider more global aspects of their goal? These questions are addressed from the point of view of how the particular aspect of programming style is affected by differential group composition.

Observational Research Study Design

In their extensive review of literature on Logo, Krasnor and Mitterer (1983) discuss the nature of the abundance of recent Logo studies. The authors conclude that most work has resulted in detailed anecdotal descriptions and "in general, there has been a tendency to overlook the need for objective and systematic examination of the Logo experience and its effects. Without this appraisal, there is a distinct possibility that exciting learning opportunities may be lost" (Krasnor & Mitterer, 1983, p. 24).

The research study design in the present case aimed to address this call for a more systematic study of the Logo experience while using observational methods.

The study may be described as one employing contrived observation and more precisely, situational testing. Borg and Gall (1983) define situational testing as a situation in which the subjects are aware that they are playing a role (as opposed to naturalistic contrived situations where intervention in a natural setting by a reseacher cannot be detected by the subjects, so preserving the 'naturalness' of the situation).

The advantage of using this form of contrived observation lies in the greater degree of control of the situation and the resulting focus on the behaviour that is of interest. One can criticize the artificial nature of such a type of testing and claim that the results would not represent behaviour in a natural setting. Borg and Gall (1983) point out, however, that when subjects are in a situation that leads to emotional interaction (which is often true in a Logo setting) "it appears that most subjects become deeply involved in the situation, and many seem to forget, at least for the moment, that the situation is an artificial one" (Borg and Gall, 1983, p. 502).

One of the components of this artificial setting is the presence of the observer. It is possible to reduce any effect of this on the behaviour of those being observed. After several sessions of observations the subjects become accustomed to the observer's presence and behave as if he or she were not present (Borg and Gall, 1983). It is mainly for this reason that the second of the three experimental problem-solving tasks was selected for across-group content analysis; it also seems appropriate from the grouping's interpersonal viewpoint. By the beginning of the second experimental task the group members were accustomed to one another and the notion of working collaboratively.

The type of data generated by this study can be examined using content analysis. "Content analysis is a research technique for the objective, systematic and quantitative description of the manifest content of communication" (Bernard Berelson cited in Borg and Gall, 1983, p. 511). The raw material for a content analysis may take one of many forms, from written material such as books, newspapers or speeches to other types of communication such as music or pictures. In this case the content of the communication includes transcripts of audio tapes (an example given by Borg and Gall) and protocols of Logo programming, both of which were recorded during the second experimental task.

Summary

Students working with Logo in a school situation usually do so in groups. What are the implications of working in different types of groups? Might one particular group work more efficiently than others on a given problem-solving task and if so, in what way would that group differ from the others - in size or sex composition, or both?

Apart from considering efficiency outcomes, this study's main focus is on whether group composition had any bearing on programming style. Of specific interest were the types of programming, the modes used, the level of planning and the sequencing of programming the components of the task.

In order to best address its purpose, the study involved a contrived observational design within a school setting. Subjects worked either alone or in groups of two or three. They were first provided with Logo programming skills. They were then placed in a situational testing environment and were asked to solve a problem which involved making the Logo turtle draw an approximate copy of a given, fixed line drawing. The subjects' programming performance was measured while their programming behaviour was observed. For the purpose of comparing the groups' efficiency, all 18 groups were compared across the instructional post-test task and the three experimental tasks. For the more in-depth analysis of programming style, the behaviour of each group was examined across only one task.

Based on the review of available literature, it was expected that there would be little difference in efficiency and programming style among groups but in this particular setting the following general findings were anticipated:

a. that programming would be mostly of the linear type,
b. that almost all programming would be done in direct mode,
c. that any planning would be contingent and mostly at the local level and lastly,
d. that the sequencing of components of the task would involve starting with the major, interior components and then working out towards the exterior details.

CHAPTER 3

Method

The purpose of this research was to investigate whether differential group composition of 11-year-old children working on a computer using Logo Turtle Graphics would have a bearing on their level of efficiency and programming style in problem solving.

In particular, the study observed the behaviour of groups varying in their number and sex of group members; in each grouping the subjects worked alone or in a group of two or three and sex composition was homogeneous or heterogeneous.

Operational Definitions

The following definitions are used only as a framework for describing the study. The definition of problem solving entails the population's ability to solve the tasks, hence the need for the instructional phase of the study prior to the actual experimental one.

Grouping: one of 18 conditions to which subjects were assigned; the groupings varied in size from one to three members and in sex composition.

Problem solving: "to search consciously for some action to attain a clearly conceived but not immediately attainable, (fixed) aim" (Polya, 1962, p. 117). Subjects were equipped with requisite skills then, when presented with the problem (which represents the novel situation) they could establish their own style to find a solution.

Fixed problem: to develop a Logo program that reproduced a given graphic on the computer screen.

Task: used synonymously with 'problem', even though in some cases, 'task' may denote a process that is well known and established as opposed to more than one process or no known process.

Establish their own style: imposing no constraints on the subject's problem-solving style; neither specific instructions nor help was given about the process of achieving the given, fixed goal.

Instructional package: the materials specifically designed to ensure that at least one member of any grouping was able to perform the given tasks. The package consists of two components: a diagnostic pretest and the instructional materials, both of which use a directive approach.

Directive approach: the researcher gave the subjects specific instruction that required them to make the turtle perform specific actions or produce a given shape; there was no opportunity for free exporation.

Logo program: a series of Logo commands typed on the keyboard that may or may not achieve the given, fixed goal.

Programming in Logo: the action of creating a Logo program with the possibility of using all Logo primitives and two procedures created by the researcher in direct and/or indirect mode(s). Subjects were free to collaborate with other group members and consult a list of Logo commands.

Experimental programming session: a 45-minute period during which a group of subjects were able to choose their own style of programming in Logo in order to achieve a given, fixed goal.

Efficiency: measured by dividing the grouping's achievement score by their time to completion. More precisely, the final score from the most complete version of each graphic (calculated by using the evaluation instrument in Appendix 9) was divided by the total completion time for that particular version of the graphic.

Programming style: measured in four ways:

1. the type of programming
   a. procedural - commands are structured into subprocedures
   b. linear - a list of step-by-step commands

2. mode(s) used
   a. direct

      This is the step-by-step, draw mode in which the user executes one command at a time (Hillel, 1985)

b. indirect

"when a sequence of commands which describes a specific entity is typed prior to execution" (Hillel, 1985, p. 13)

The following categories have been added to further qualify the use of direct mode:

- the use of direct mode attempted and abandoned
- direct mode used successfully to create a procedure that runs
- direct mode used to create a procedure that contains at least one error
- direct mode used to create a superprocedure containing subprocedures

3. level of planning

a. pre-planning

The criterion considered appropriate was a discussion of a plan which included reference to 2 or more components before any group member began typing

b. contingent planning

This term refers to:

- planning-in-action (Hillel, 1985)
- on-line programming (Pea, 1983)
- interspersing planning with hands-on activity (Hoyles, 1985)
- continual restructuring of a problem while seeking a solution (Dunker cited in Statz, 1973, p.26)

- "testing a solution by submitting it to constant revision until one of a set of successive approximations works" (Stats, 1973, p.22)

It was decided to further investigate this level of planning by determining to what extent the contingent planning was global or local:

global - includes reference by any group member to any component looking back (B) or ahead (A) other than the component currently being worked on

local - includes reference by any group member to the component currently being worked on

4. sequence in which components of the graphic were programmed:

   a. interior to exterior

      - head and body components are programmed before the exterior details

   b. other

      - components are programmed in what appears to be an indiscriminate sequence

Subjects

The 36 subjects who participated in the study were from an English speaking public elementary school which operates within the Protestant School Board of Greater Montreal. The 18 males and 18 females were mostly members of grade five mixed-ability classes; four were from a grade five/six split class. The subjects volunteered to participate and were assigned to groupings according to their availability within their lunch-time and afterschool extracurricular activities schedules. There were a total of 18 groups which comprised nine types of groupings: those where the subjects worked:

1. alone (male M, or female F)
2. in pairs (all males MM, all females FF, or mixed FM) or
3. in a group of three (all males MMM, all females FFF, two females and a male FFM, two males and a female MMF).

Materials

The Computers

The three computers used were all Apple-II plus compatible. Two had colour monitors while the third had a monochrome monitor. Each computer had one disk drive. The equipment was set up in an ante-chamber adjacent to the computer classroom. The computers were arranged in such a way that subjects working at one work place would not see the video monitors of the other computers. The room had one window but was otherwise artificially well lit.

The Instructional Module

In order that all subjects would have equivalent entry skills when starting work with the experimental tasks, an instructional module was developed. To further ensure equal treatment among subjects, it was intended that the module be as researcher-independent as possible. In designing the instruction, the author used the model developed by Dick and Carey (1978) as a guide. This is based on a systematic approach to designing instruction. It presents a 9-step set of procedures that can be seen in Appendix 1. The discussion of the application of the model is in Appendix 2.

The components of the instructional package and the numbers of the appendices in which they may be found are as follows:

1. researcher's instruction manual (Appendix 3)
2. diagnostic checklist (Appendix 4)
3. truck task (Appendix 5)
4. instructional materials and tests (Appendix 6)

Pilot testing and formative evaluation of the module was carried out following the suggestions by Dick and Carey (1978), and are discussed in Appendix 2.

Support Materials

A reference list of Logo commands and their corresponding functions (see Appendix 7), blank paper and a pencil were supplied to all subjects at the beginning of each session.

The Experimental Tasks

The pretest in the form of the diagnostic checklist, the instructional materials and the Truck task post-test were designed to ensure that each grouping could demonstrate mastery of certain Logo concepts. The subjects had made various shapes using Logo graphics. The three experimental tasks required that some of these shapes be generated in different compositions. The subjects were presented with a simple line drawing of a boat, a snowman and city and were asked to make the Logo turtle draw a copy of them. (A copy of the task pictures may be found in Appendix 8.) The choice of these easily identifiable drawings was intended to generate thoughtful programming.

Although the subjects had made various shapes with Logo during the instructional phase, the tasks were designed to include some challenging concepts, in particular, the angle of the Boat sail, the combination of inputs to REPEAT required for Boat's semi-circle and the bird in City, and the turtle state when making Snowman. The procedures CIR and POL (see Unit 6 in Appendix 6), created by the researcher, were available for the subjects' use; CIR was the only means of making a circle but the use of POL was optional for making the squares and triangles.

Instruments

Means of Recording the Subjects' Logo Commands

For the purpose of keeping a record of the Logo commands the subjects typed in, the program TEACH on the

Apple Logo Tool Kit Disk was used. This program was loaded into the computer once Logo had been booted. It then stored the commands typed until, at the end of the session, they were defined as a procedure and saved on disk.

Means of Recording the Subjects' Conversation

A portable cassette recorder containing a 90-minute blank tape was placed on top of each computer. The machine was turned on when the subjects had been given their instructions and were ready to begin their problem solving. Four sessions were recorded for each grouping: the Truck task and the three experimental tasks.

Means of Recording the Time

A supplementary tape recorder was placed equidistant from the three other recorders. This one contained a prerecorded tape that emitted a beep signal at 1-minute intervals. The tape recorders on the computers simultaneously recorded the beep signal and the subjects' conversation.

The Experimental Task Evaluation Instrument

In order to evaluate the graphics generated for each task, a hard copy was produced from the protocols of the Logo commands that had been saved on disk. If more than one version of a task was made by a grouping, the most complete version was used.

The evaluation instrument was developed by the author.

A broad sample of the subjects' graphics were assessed to determine in what way and to what degree they differed from the task version. Five categories of discrepancies were identified and each was assigned a scale of points to be deducted based on the degree of discrepancy. The categories were: 1. incomplete form of a component, 2. component not joined, 3. a component poorly placed or aligned, 4. lack of neatness, and 5. poor shape. For each category the amount of points that could be deducted ranged from five to 20. The evaluation criteria can be seen in Appendix 9.

Each graphic was evaluated by the author and one other researcher; the evaluations were made on an independent basis and inter-scorer reliability was found to be high ($r > .90$).

A copy of the evaluation instrument may be found in Appendix 10. The charts were completed for each grouping's work in the following manner. The presence or absence of each component of the graphic was checked on the list marked 'components'. The evaluation criteria for the completed components were then applied to the graphic; points to be deducted in the five categories were noted and the total deductions were entered. The final score was calculated as 100 minus the total points deducted then multiplied by the fraction of the components completed. For example, the City task contained eight possible components but if one were missing the fraction would be 7/8.

Application of these criteria in this manner allowed for a total possible score of 100. A lower score was awarded if the components did not correspond to those of the task version and/or a component was missing.

The total completion time (TCT) for the most complete version was entered from the researcher's observation sheet. The total number of commands typed to produce that version was then established from the Logo protocol.

Procedure

The study was conducted over a period of 4 1/2 weeks. All but two groups attended a total of five sessions. The other two required an additional session; one for remedial instruction, the other because the subject worked slowly. Each session lasted approximately 45 minutes.

There were time constraints placed on the testing procedure. The only possible times the subjects could be available were after they had eaten lunch during the 1-hour break or after school. However, many of those who volunteered were not available at all times; they had responsabilites in the school, extracurricular activities or family commitments. The subjects were therefore assigned to their group according to their availability. On this basis, the duration of a group's testing varied from 2 1/2 to 8 days.

Three reseachers were available to conduct the study so during any one session there was from one to three groups being tested. The number was dependent upon the group's timetable. Each researcher was asked to follow the instructions in the Researcher's Instruction Manual (see Appendix 3).

The experimental sessions were conducted as follows. At the beginning of each session every group of subjects was provided with a reference list of Logo commands and their corresponding functions (see Appendix 10). There was also blank paper and pencils available.

The first session for every group began with the diagnostic checklist (see Appendix 4). If there were 2 or 3 subjects in the grouping they were asked to stand and take turns typing at the keyboard. If no remedial instruction was required from Units 1 to 5, the next step was to give the instruction from Unit 6. This was mandatory for all groups as it presented concepts that were unfamiliar to the subjects.

For the second session, the subjects were asked to do the instructional post-test which involved making a picture of the Truck. They were given a copy of the picture and specific instructions as to what strategies they should use (see Appendix 5 for the picture and instructions). From that point on, the role of the researcher was one of an unobtrusive observer. The commands the subjects typed in were saved on disk at the end of the session.

Sessions three to five were similar; the experimental tasks differed in terms of the picture given but they were administered in the same way. The subjects were given very few instructions (see Appendix 8) before they began and they were free to choose their working strategies. Again, the commands typed in during the session were saved on disk.

CHAPTER 4

Results and Discussion

The results and discussion are presented in two parts. The efficiency data on all 18 groups across four tasks is presented first followed by the results on programming style of all 18 groups across one task only.

Efficiency

Efficiency results are presented in Table 1 and Table 2 which represent grouping by number and grouping by sex respectively.

In order to analyze effects of group size and composition two separate ANOVAs were run. Analyses were made using the program BMDP2V (Dixon, 1981) on Concordia University's Cyber 835. (A single 3X3X4 ANOVA was rejected due to the small sample sizes.) First, a 3X4 ANOVA was run to investigate the effects of group size on efficiency across the four tasks; results are shown in Table 3.

It was found that there was no size by task interaction and that size had no significant effect on efficiency; the only significant effect was the tasks.

Table 1

Efficiency Results: grouping by number

	GROUP	TASK 1 Score	TASK 1 Time	TASK 1 Eff.	TASK 2 Score	TASK 2 Time	TASK 2 Eff.	TASK 3 Score	TASK 3 Time	TASK 3 Eff.	TASK 4 Score	TASK 4 Time	TASK 4 Eff.
INDIVIDUALS	M1	100	17	5.9	80	45	1.8	95	40	2.4	85	40	2.1
	M2	90	33	2.7	80	26	3.1	80	32	2.5	83	46	1.8
	M3	60	19	3.2	63	40	1.6	100	29	3.5	95	24	4.0
	F1	100	24	4.2	90	37	2.4	80	30	2.7	75	42	1.8
	F2	100	13	7.7	90	13	6.9	85	20	4.3	80	40	2.0
	F3	60	27	2.2	95	48	2.0	90	24	3.8	70	27	2.6
	Average M	85.0	22.2	4.3	83.0	34.8	3.0	88.3	29.2	3.2	81.3	36.5	2.4
	SD	19.7	7.3	2.1	11.5	13.9	2.0	8.2	6.9	0.8	8.6	8.8	0.8
PAIRS	MM1	100	17	5.9	75	15	5.0	90	39	2.3	90	41	2.2
	MM2	100	13	7.7	95	32	3.0	73	25	3.0	83	39	2.1
	FF1	64	46	1.4	57	31	1.8	90	48	1.9	64	37	1.7
	FF2	100	14	7.1	90	27	3.3	95	19	5.0	95	20	4.8
	MF1	100	24	4.2	100	23	4.4	85	21	4.1	80	45	1.8
	MF2	80	30	2.7	95	44	2.2	95	26	3.7	68	41	1.7
	Average M	90.7	24.0	4.82	85.3	28.7	3.3	88.0	27.7	3.3	80.0	37.2	2.37
	SD	15.3	12.6	2.5	16.3	10.0	1.2	8.2	10.9	1.2	10.4	8.8	1.2
TRIADS	MMM1	38	13	2.9	95	21	4.5	95	23	4.1	95	28	3.4
	FFF1	72	27	2.7	80	29	2.8	80	23	3.5	70	44	1.6
	MFF1	100	32	3.1	90	30	3.0	95	20	4.8	80	31	2.6
	MFF2	54	16	3.4	90	42	2.1	90	35	2.6	90	41	2.2
	MMF1	90	30	3.0	67	48	1.4	75	37	2.0	97	47	2.1
	MMF2	80	29	2.8	64	42	1.5	80	22	3.6	70	41	1.7
	Average M	72.3	24.5	3.0	81.0	35.3	2.6	85.8	26.7	3.4	83.7	38.7	2.3
	SD	12.9	7.4	0.3	11.5	10.1	1.2	8.6	5.0	1.0	13.1	7.5	0.7

Table 2

Efficiency Results: grouping by sex

	GROUP	TASK 1 Score	TASK 1 Time	TASK 1 Eff.	TASK 2 Score	TASK 2 Time	TASK 2 Eff.	TASK 3 Score	TASK 3 Time	TASK 3 Eff.	TASK 4 Score	TASK 4 Time	TASK 4 Eff.
FEMALES	F1	100	24	4.2	90	37	2.4	80	30	2.7	75	42	1.8
	F2	100	13	7.7	90	13	6.9	85	20	4.3	80	40	2.0
	F3	60	27	2.2	95	48	2.0	90	24	3.8	70	27	2.6
	FF1	64	46	1.4	57	31	1.8	90	48	1.9	64	37	1.7
	FF2	100	14	7.1	90	27	3.3	95	19	5.0	95	20	4.8
	FFF1	72	27	2.7	80	29	2.8	80	23	3.5	70	44	1.6
	Average M	82.7	25.2	4.21	83.7	30.8	3.21	86.7	27.3	3.5	75.7	35.0	2.4
	SD	19.4	12.0	2.1	14.0	11.9	1.4	6.1	5.9	0.7	10.9	10.2	0.9
MALES	M1	100	17	5.9*	80	45	1.8	95	40	2.4	85	40	2.1
	M2	90	30	2.7	80	26	3.1	80	32	2.5	83	46	1.8
	M3	60	19	3.2	63	40	1.6	100	29	3.5	95	24	4.0
	MM1	100	17	5.9	75	15	5.0	90	39	2.3	90	41	2.2
	MM2	100	13	7.7	95	32	3.0	73	25	2.9	83	39	2.1
	MMM1	38	13	2.9	95	21	4.5	95	23	4.1	95	28	3.4
	Average M	81.3	26.0	4.7	81.3	29.8	3.2	88.8	31.3	4.3	88.5	36.3	2.6
	SD	26.3	10.6	2.6	12.3	11.4	1.9	10.3	7.1	1.1	5.6	6.6	1.2
MIXED	MF1	100	24	4.2	100	23	4.4	85	21	4.1	80	45	1.8
	MF2	80	30	2.7	95	44	2.2	95	26	3.7	68	41	1.7
	MFF1	100	32	3.1	90	30	3.0	95	20	4.8	80	31	2.6
	MFF2	54	16	3.4	90	42	2.1	90	35	2.6	90	41	2.2
	MMF1	90	30	3.0	67	48	1.4	75	37	2.0	97	47	2.1
	MMF2	80	29	2.8	64	42	1.5	80	22	3.6	70	41	1.7
	Average M	84.0	26.8	3.2	84.3	38.2	2.4	86.7	26.8	3.5	80.8	41.0	2.6
	SD	17.2	5.9	0.5	15.1	9.6	1.1	8.2	7.4	1.0	11.2	5.5	0.4

Table 3

Group Size By Task Analysis of Variance

	Sum of Squares	Degrees of Freedom	Mean Square	F	Tail Prob.
Size	4.987	2	2.493	.68	.5237
Error	55.357	15	3.690		
Task	27.326	3	9.109	6.78	.0007
TS	7.761	6	1.294	.96	.4610
Error	60.453	45	1.343		

The second 3X4 ANOVA was run to investigate the effects of sex composition on efficiency across the four tasks; results are shown in Table 4.

Table 4

Sex Composition by Task Analysis of Variance

	Sum of Squares	Degrees of Freedom	Mean Square	F	Tail Prob.
Sex	5.382	2	2.691	.73	.4963
Error	54.961	15	3.664		
Task	27.326	3	9.109	6.64	.0038*
TS	6.468	6	1.078	.79	.5461*
Error	61.747	45	1.372		

* After Greenhouse-Geisser correction - (as sphericity test was found significant)

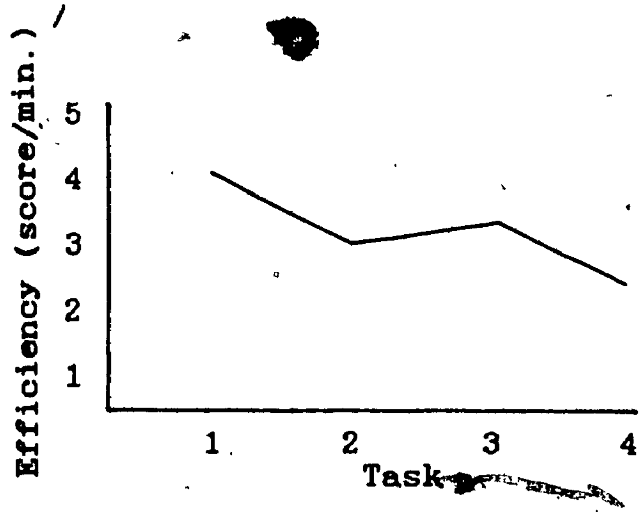
It was found that there was no sex composition by task interaction and that sex composition had no significant effect on efficiency; again, only the tasks were found to have a significant effect.

Effect Of The Task

It can be seen from Diagram 1 that the level of efficiency varied according to the task which would indicate that the tasks differed in level of difficulty. The effect of practice may also be present and perhaps even interact with task difficulty. While this is interesting from the point of view of task analysis (e.g. what geometric concepts are involved, what transfer of skills learned is present, etc.) it will not be discussed further as this aspect of efficiency (i.e. the intrinsic characteristics of the tasks) was not the main focus of the study.

Diagram 1

Effect of The Tasks on Efficiency



It should be noted, however, that the effect of the tasks may be due only to the overall high scores for Task 1 which could be attributed to a slightly different treatment.

Task 1 was the post-test for instruction involving specific indications on the programming style to be used. If the results from Task 1 were eliminated and those of only Tasks 2, 3, and 4 were examined, the effect of the tasks may disappear.

Although the overall scores for the post-test for instruction (Task 1) were high, it should be noted that some groups obtained low scores on this task (see Table 1 or 2). However, it should be further noted that most of those groups went on to obtain much higher scores. Of the eight groups that scored 80 or less on Task 1, six scored 90 or over on one or more of the subsequent tasks; the two other groups obtained at least one further score of 80.

Programming Style

A profile sheet showing the results of the programming style analysis for each group can be found in Appendix 11. To present an account of the overall findings across the second experimental task (which corresponds to the third in the overall sequence of four tasks), each of the four measures of programming style is discussed in turn.

The Type of Programming

In no case was there any evidence of procedural programming; each group's work comprised one long list of unstructured commands that varied in number from 90 to 226.

__Mode(s) Used__

There was an overwhelming use of direct mode. All except three groups used direct mode only and even the exceptions used direct mode to create an initial complete version of the graphic before entering indirect mode. Two groups, FM2 and MM2, did not hesitate to enter the indirect mode; the FFM2 group asked if they should do so and then proceeded. In all three cases, the subjects entered one long list of commands without stopping to verify that the results met with their approval. When it was suggested to the FM2 group that they may want to check what they had entered so far, they insisted on entering all the commands together, but at the same time showed a lack of confidence that their record of commands was accurate or that their procedure would run. The MM2 group showed even greater pessimism. Both groups' predictions were confirmed; their procedures contained at least one error that resulted in a graphic image that far from met their approval. Both groups rationalized their lack of success by saying either the computer was at fault (for example, "the editor made it so warm that it melted"), or the snowman itself ("he killed himself with the spike"). Only the FFM2 group proceeded without mention of the outcome; they seemed to assume their procedure would run, as it did.

When asked at the end of the session, why they chose not to put their work in the editor, the FFF1 group also expressed a lack of confidence that an attempt to use indirect mode would work. Their justification included remarks about the inordinate amount to writing, typing and time that would be involved and their inability to identify what would amount to numerous mistakes.

Level of Planning

It should first be noted that only the groups of two and three subjects can be included in this category because the level of planning was determined from the verbal content of the audio tapes. The tapes of subjects working alone contain mostly vocal reactions to the results of their programming hence they offer no reliable evidence as to the subject's level of planning.

Results show almost no evidence of preplanning. Only in the MFF2 group did a subject refer to two components before beginning typing. In two other cases, FF2 and MF2, a group member made a suggestion as to which component to start with (which did not meet the criterion for pre-planning, as used here).

Contingent planning was mostly on a local level. The extent of the global planning varied from one to five references to any component other than the one the group was working on, in all but two cases; the FF2 group had missing data but the MFF2 group made a total of 14 references. (It was this group that subsequently successfully used indirect mode to create a procedure that ran.) The number of instances of making references ahead (29 A) was significantly higher than that of those made looking back (16 B).

Sequence in which Components were Programmed

The components of the snowman task were identified as: body (B), head (He), hat (Ha), left arm (LA), right arm (RA), and fork (F). Data was available for 14 out of the original 18 groups. The data for the groups M1, F2, MMM1, and FFF1 was lost because of problems with the TEACH program. The order in which the 14 different groups attempted the components was compiled; results ~~for~~ the entire sample and subsample are presented in Tables 5 to 11.

Table 5

Frequencies of Rank Order of Components for Entire Sample

	component					
rank	B	He	Ha	LA	RA	F
1	10	2	2	0	0	0
2	1	8	1	3	1	0
3	3	0	5	2	4	0
4	0	3	1	7	2	1
5	0	1	4	2	3	4
6	0	0	1	0	4	8

n=14

Table 6

Frequencies of Rank Order of Components for Singles

rank	B	He	Ha	LA	RA	F
1	3	0	1	0	0	0
2	0	3	0	1	0	0
3	1	0	1	0	2	0
4	0	1	0	3	0	0
5	0	0	1	0	1	1
6	0	0	0	0	1	3

n=4

Table 7

Frequencies of Rank Order of Components for Pairs

rank	B	He	Ha	LA	RA	F
1	4	1	1	0	0	0
2	0	3	1	1	1	0
3	2	0	1	2	1	0
4	0	1	1	2	1	1
5	0	1	1	2	1	1
6	0	0	1	0	1	3

n =6

Table 8

Frequencies of Rank Order of Components for Triads

rank	B	He	Ha	LA	RA	F
1	3	1	0	0	0	0
2	1	2	0	1	0	0
3	0	0	3	0	1	0
4	0	1	0	3	0	0
5	0	0	1	0	1	2
6	0	0	0	0	2	2

(column heading: component)

n=4

Table 9

Frequencies of Rank Order of Components for Males

rank	B	He	Ha	LA	RA	F
1	4	0	0	0	0	0
2	0	2	0	1	1	0
3	0	0	1	2	1	0
4	0	2	1	1	0	0
5	0	0	2	0	1	1
6	0	0	0	0	1	2

(column heading: component)

n=4

Table 10

Frequencies of Rank Order of Components for Females

rank	B	He	Ha	LA	RA	F
1	2	0	2	0	0	0
2	0	4	0	0	0	0
3	2	0	1	0	1	0
4	0	0	0	3	1	0
5	0	0	1	1	1	1
6	0	0	0	0	1	3

n=4

Table 11

Frequencies of Rank Order of Components for Mixed Groups

rank	B	He	Ha	LA	RA	F
1	4	2	0	0	0	0
2	1	2	1	2	0	0
3	1	1	3	0	2	0
4	0	0	0	3	1	1
5	0	1	1	1	1	2
6	0	0	1	0	2	3

n=6

In order to investigate the existence of a (significant) order, a series of analyses was performed on the entire sample. First, a Friedman two-way rank analysis of variance (Cohen and Holliday, 1979, pp. 166-8) points out the existence of some preference in the sequence of components ($\hat{\chi}^2_r=40.12 > 10.85=\chi^2_r[n=14;k=6;p=0.05]$). Although more appropriate tests exist for paired comparisons (cf. Sachs, 1982, p. 555), they are not easily available and it was felt that, due to the exploratory nature of the study, a series of Friedman tests at the p=0.01 level as way of post-hocs would suffice for determining the nature of the order. Thus, it was found that the order between hat, right and left arms was not significant ($\hat{\chi}^2_r=2.71 < 9.14=\chi^2_r[n=14;k=3;p=0.01]$); that the order between the last three components and the fork was significant ($\hat{\chi}^2_r=18.08 > 10.89=\chi^2_r[n=14;k=4;p=0.01]$); that the order between body, head and hat was significant ($\hat{\chi}^2_r=9.5 > 9.14=\chi^2_r[n=14;k=3;p=0.01]$); and that the order between body and head was significant ($\chi^2_r=2.57$). Given that there are no tables for the case k=2, the test was performed via Spearman's $\hat{r}_s=-0.80 > -0.60=r_s[\text{two-sided};n=14;p=0.01]$; see Sachs, 1982, p. 555). Hence, we could establish that the six components constitute an order of the four following elements:

Body > Head > (Hat or Right Arm or Left Arm) > Fork

This general order represents the anticipated sequence if subjects were to consider the task one in which they could use Logo as an extension of their drawing arm. The subjects worked from the interior by beginning with major components (Body and Head) and then moved outwards to make adjacent components (Left Arm, Right Arm and Hat), finishing with the most extreme detail, the Fork.

Next, in order to explore different behaviour among subgroups, a series of Kolmogorov-Smirnov One Sample tests (Cohen and Holliday, 1979, PP. 133-6) was conducted at the p=0.05 level, comparing the particular group's sequencing of components with the previously found general order of the four elements; the results are shown in Table 12.

Table 12

$\hat{D}$ values of the Kolmogorov-Smirnov tests

GROUP:	SINGLE	PAIRS	TRIADS	MALES	FEMALES	MIXED
CHOICE:						
First element	0.25	0.33	0.25	0.0	0.5	0.33
Second element	0.25	0.5*	0.25	0.5	0.0	0.5*
Third element	0.25	0.33	0.0	0.0	0.5	0.16
Last element	0.25	0.33	0.5	0.25	0.25	0.5*

(*) These are the only tests near significance (D[n=6;p=0.05]=0.52); columns other than PAIRS or MIXED are compared to D[n=4;p=0.05]=0.62.

From this we can conclude that there were no significant differences among groups for the order in which the four elements were programmed. Only in the case of PAIRS and MIXED groups did results approach significance, but the small sample sizes (and particularly the loss of data in all other categories) may be responsible for this apparent finding.

CHAPTER 5

Conclusions

The study aimed to investigate the role of group composition on 11-year-olds' problem solving with Logo. The children's programming of a series of given, fixed graphic goals was analyzed to look for differences in efficiency and style of programming. The two aspects will be discussed separately, as they were in Chapter 4.

Efficiency

Findings from the various studies on the effects of group work on performance led to the prediction that group composition would have no bearing on efficiency as indeed was the case; neither sex composition nor group size had a significant effect on efficiency as it was defined here (score divided by time). Had the efficiency measure been different, for example, score divided by group size or time divided by group size, there may have been a different outcome. This may also have been the case if the task had been different in nature; students could have been provided with a given but less fixed goal that asked them, for example, to create a graphic of a means of transportation or a city scape of their choice. A task that left room for more creativity or originality may have generated a different type of collaboration from which varying levels of efficiency may have emerged. It would be interesting to study the effect of these variables.

In the present study the absence of differences in performance raises the question of why the presence of one or two additional members in a group did not result in a level of performance different from that of a student working alone. We could conclude that the possible reduction in the speed of work due to the additional communication from more members (Freedman et al., 1970) was counteracted by the beneficial effect of the opportunity to vocalize (Durling & Schick, 1976) and the input of a wider range of abilities as well as improved feedback provided by an increased group size (Freedman et al., 1970). We could, however, also challenge this conclusion with the argument that perhaps not all members provide a consistent level of active participation, thereby reducing the sum of the contributions to the group. This may be particularly true for a group of three. Further research should be focussed on obtaining information on the particular behaviour of each group-member in order to clarify this point.

Programming Style

It should be remembered that conclusions on the different aspects of programming style are based on the analysis of one task only across all groups.

Type of Programming and the Mode(s) Used

That there was no evidence of procedural programming is consistent with the findings of Hillel (1985) and Leron (1984). Both report that their students showed resistance to adopting this style. Hillel (1985) states that the 8 - 9-year-olds in his study mostly used procedures as a way of saving a picture and not as a means of simplifying the construction of a complex task. Leron (1984) offers an interesting discussion about this; perhaps we expect to see the use of procedures in structured programming for tasks whose level of complexity does not particularly warrant it. As adults, we appreciate the 'beauty' of structured programming and enjoy the benefits that a procedural programming language offers; children, it seems, have a different appreciation of the benefits of Logo (Leron, 1984). They enjoy the gratification from the immediate feedback in the very visual interactive setting and to such an extent that, if given a choice, they will not only avoid using procedural programming but will also avoid procedures altogether by not using indirect mode. Comments from some of the students in this study indicated that they did not feel confident enough that an attempt to work in indirect mode would be successful. Perhaps they felt, as those children in Pea's study (Pea, 1983) did, that it is better to rewrite a program from scratch, if necessary, than to systematically check through the commands to locate any errors. In that case, their general feeling could be that the time and effort invested in using direct mode does not pay off for them, even though they may have been told it should. Presumably, then, when the child quoted by Pea (1983) said that it was easier to do it 'the hard way', he meant that what was perhaps easier for an adult was harder for him. In general, it appears that children's natural style is to use a linear approach and perhaps expecting them to use structural programming (particularly in a situation where the need is not apparent) is imposing an adult perspective they are not yet ready to consider.

Level of Planning

It should be remembered that data on the level of planning was collected from the pairs and triads only. It was found that group composition had no bearing on the level of planning and next to none on the amount of planning.

The level of planning was consistent with that anticipated; in terms of pre-planning the results showed that across the groups examined, it was almost non-existent. This result was similar to that found by Tetenbaum and Mulkeen (1984). Although a high level of pre-planning was not anticipated, the extremely low incidence may, to a certain extent, be attributed to the experimental testing situation. Firstly, the students knew they had a time constraint; secondly, they started the testing session infront of a computer that was already turned on. Given that the students had no prompting on what style of programming to adopt, they perhaps saw the flashing cursor as an invitation to begin typing immediately. It should, however, be noted that the subjects were reminded they could use the paper provided to write anything down.

Almost all planning was contingent, that is, 'planning-in-action' (Hillel, 1985) characteristic of 'on-line' programming (Pea, 1983). This planning was mostly on a local basis. The amount of global references made was quite consistent across groups, although the incidence of looking ahead in the planning was much higher than that of looking back over the task. Looking back is one of the four possible phases involved in problem solving discussed by Polya (1945) although he does point out that the problem solver may skip a phase he found unnecessary. This was largely true in terms of looking back. The three other phases were involved in the problem solving but from what might be called a very shallow perspective. There seems to have been a continual cycle of understanding the problem, devising a plan and carrying out the plan, but all on a very local basis. This strategy is very similar to the one that Hoyles (1985) thought was effective when students had a clearly defined task.

Of the groups examined, only one showed a much higher level of contingent global planning; it was this group that had the most success in reaching their goal.

<u>Sequencing of Components</u>

Results on the sequencing of components showed that group composition had no significant effect on the order in which the components of the snowman task were programmed. The general order followed the sequence or 'natural order' that one would typically use to draw a picture of the snowman by hand. (According to Hoyles (1985), this can contribute to the students' resistance to divide their picture into subprocedures.)

It is interesting to speculate on the source of this general order. There was almost no evidence of groups pre-plannng which meant there was no initial open discussion of a proposed sequence. Even the few instances of group members making reference to the next component did not generate a discussion. There was no evidence of disagreement or a proposal for an alternative plan. Perhaps this indicates there is an unstated general concensus of opinion because all members of the group assumed they would follow the 'natural order' one would use in drawing by hand. It could, on the other hand, indicate a willingness to follow the suggestions or directions of another group member who may (perhaps temporarily) have assumed the role of group leader. Again, it could be interesting to study the role of group size and also sex composition on the roles played.

Group size or composition appears to make no difference to those aspects of children's programming that were the focus of this study, for the type of task used here. Thus, it seems we can justifiably reap the benefits of implementing cost-effective group work, so making scarce resources available to more people (Boyd et al., 1983). At the same time students can benefit from the social setting which fosters group cooperation skills (Johnson & Johnson, 1974; Boyd et al., 1983) and provides cognitive (Krasnor & Mitterer, 1983; Papert, 1980; Hawkins, 1983) and linguistic benefits (Conlin, 1981). A further advantage of the social setting is that the student avoids becoming party to the 'closet computer queen' phenomena that Boyd et al. (1983) mention.

While it might be naive to assume that a group's ability could equal the sum of the range of abilities of its members, it is interesting to consider whether, when there are three in a group, there is room for all three to make their optimal contribution. It is possible that one group member becomes uninvolved in the problem solving. If this happens, what are the consequences; does he become bored so that his behaviour becomes disruptive? General impressions of the groups' working relationships were that this was the case for certain groups. Working as a group of three did not seem to allow for all members' involvement over the 45 minutes of problem solving. The most positive working relationships were observed in the all female group, and of the heterogeneous groups, those comprising two females and one male. Of the groups of two, heterogeneous groups seemed to work better than pairs of girls and much better than the pairs of boys. In general, where there were two or more boys at a computer, disruptive behaviour occured. The students' working relationship is surely a factor that influences programming behaviour. Although it was not a focus of this study, it is an important aspect of group work that warrants further research if the computer experience is to offer optimal benefits to all those in the group.

REFERENCES

Ausubel, D. P. Educational psychology: a cognitive view. New York: Holt, Rinehart and Winston, 1968.

Borg, W. R. & Gall, M. D. Educational research: an introduction. (4th ed.) New York: Longman, 1983.

Boyd, G. M. Four ways of providing computer assisted learning and their probable impacts. Computers and Education, Vol. 6, No. 3, 305 - 10, 1982.

Boyd, G. M., Douglas, L., & Lebel, C. Learner Support Options in Computer-Assisted Learning. Paper presented at the Fourth Canadian Symposium on Instructional Technology, Winnipeg, 1983.

Cartwright, G. F. The Use of Groups in Computer Assisted Instruction. Paper presented at the Canadian Symposium on Instructional Technology, Calgary, May 24 - 26, 1972.

Cathcart, G. M. Logo programming bugs and debugging strategies of grade six students. Master of Education Thesis, University of Alberta, 1985.

Coburn, P., Kelman, P., Roberts, N., Snyder, T., Watt, D., & Weiner, C. Computers in education. Reading, Massachusetts: Addison-Wesley, 1982.

Cohen, L., & Holliday, M. Statistics for education and physical education. London: Harper and Row, 1979.

Conlin, C. A computer in an Infant School. CAL News, March, 1981.

Dick, W. & Carey, L. The systematic design of instruction. Glenview, Illinois: Scott, Foresman and Company, 1978.

Dixon, W. J., & Brown, M. B. (Eds.) Biomedical computer programs p-series. Berkeley: University of California Press, 1981.

Durling, R. & Schick, C. Concept attainment by pairs and individuals as a function of vocalization. Journal of Educational Psychology, 1976, 68 (1), 83 - 91.

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. Programming language as conceptual framework for teaching mathematics. Report 1889, Nov. 30, 1969.

Freedman, J. L., Carlsmith, J. M. & Sears, D. O. Social psychology. Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1970.

Hartley, J. R. & Lovell, K. Principles underlying the design of computer-based instructional systems. In D. F. Walker & R. D. Mess (Eds.), Instructional software: principles and perspectives for design and use. Belmont, Cal.: Wadsworth Publishing Co., 1984.

Hawkins, J. Learning Logo together: the social context. Centre for Children and Technology, Bank Street College of Education, Technical Report No. 13, 1983.

Hillel, J. Mathematical and programming concepts acquired by children, aged 8 - 9, in a restricted Logo environment. Research Report No. 1. Department of Mathematics, Concordia Univerity, Montreal, 1985.

Hoyles, C. A preliminary investigation of the pupil-centered approach to the learning of Logo in the secondary school mathematics classroom. University of London Institute of Education, London, 1985.

Johnson, D. W. & Johnson, R. T. Instructional goal structure: cooperative, competitive or individualistic. Review of Educational Research, 1974, 2 (44), 213 - 240.

Kemmis, S., Atkin, R. & Wright, E. How do students learn? Working Paper on Computer Assisted Learning. Occasional Paper No. 5. (Centre for Applied Research in Education, University of East Anglia, United Kingdom, 1977).

Karweit, N. & Livingston, S. A. Group versus individual performance and learning in a computer game: an exploratory study. John Hopkins University, Baltimore, Md. Centre for the Study of Social Organization of Schools, 1969. (ERIC Document No. ED 032 789)

Krasnor, L. R. & Mitterer, J. O. Logo and the development of general problem-solving skills. Paper produced at Brock University, Ontario, April, 1983.

Lathrop, A. & Goodson, B. Courseware in the classroom, Addison-Wesley: Menlo Park, 1983.

Lebel, C. Cooperation between adolescents in computer-assisted algebraic problem solving. M. A. Thesis, Concordia University, Montreal, 1982.

Leron, U. Some problems in children's Logo learning. Proceeding of the Seventh International Conference for the Psychology of Mathematics Education, Israel, 1983.

Leron, U. Current issues in Logo learning. Department of Science Education, Israel Institute of Technology, Haifa, June, 1984.

Lester, F. K. Mathematical problem solving in the elementary school: some educational and psychological considerations. In L. L. Hatfield C.D.A. Bradbard (Eds.), Mathematical problem solving: papers from a research workshop. Columbus, Ohio: ERIC/SMEAC, 1978.

Love, W. P. Individual versus paired learning of an abstract algebra presented by computer assisted instruction. Florida State University, Tallahessee. Computer Assisted Instruction Centre, 1969. (ERIC Document No. ED 034 403)

Munro-Mavrias, S. Personal computers and the problem-solving skills of young children. M. A. Thesis, Department of Education, Concordia University, Montreal, March, 1985.

Nelson, H. Learning with Logo. on Computing, summer, 1981.

Okey, J. R. & Majer, K. Individual and small group learning with computer assisted instruction. Paper presented at the Annual Meeting of the American Educational Research Association, (Washington, D. C., March 30 - April 3, 1975).

Papert, S. Mindstorms. New York: Basic Books, 1980.

Pea, R. D. Logo programming and problem solving. Center for Children and Technology, Bank Street College of Education, Technical Report No. 12, 1983.

Polya, G. How to solve it. Princeton: Princeton University Press, 1945.

Polya, G. Mathematical discovery. New York: Wiley, Volume I, 1962; Volume II, 1965.

Rushby, N. J. An introduction to educational computing. London: Croom Helm, 1979.

Rushby, N. J. (Ed.) Selected readings in computer-based learning. London: Kogan Page, 1981.

Sachs, L. Applied statistics: a handbook of techniques. New York: Springer-Verlag, 1982.

Solomon, C. Introducing Logo to children, Byte Magazine, August, 1982.

Statz, J. A. The development of computer programming concepts and problem-solving abilities among ten-year-olds learning Logo. Unpublished doctoral dissertation, Syracuse University, York, 1973.

Taylor, R. P. (Ed.) The computer in the school: tutor, tool, tutee. New York: Teachers College Press, 1980.

Tetenbaum, T. J. & Mulkeen, T. A. Logo and the teaching of problem solving: a call for a moratorium. Educational Technology, November, 1984, 16 - 19.

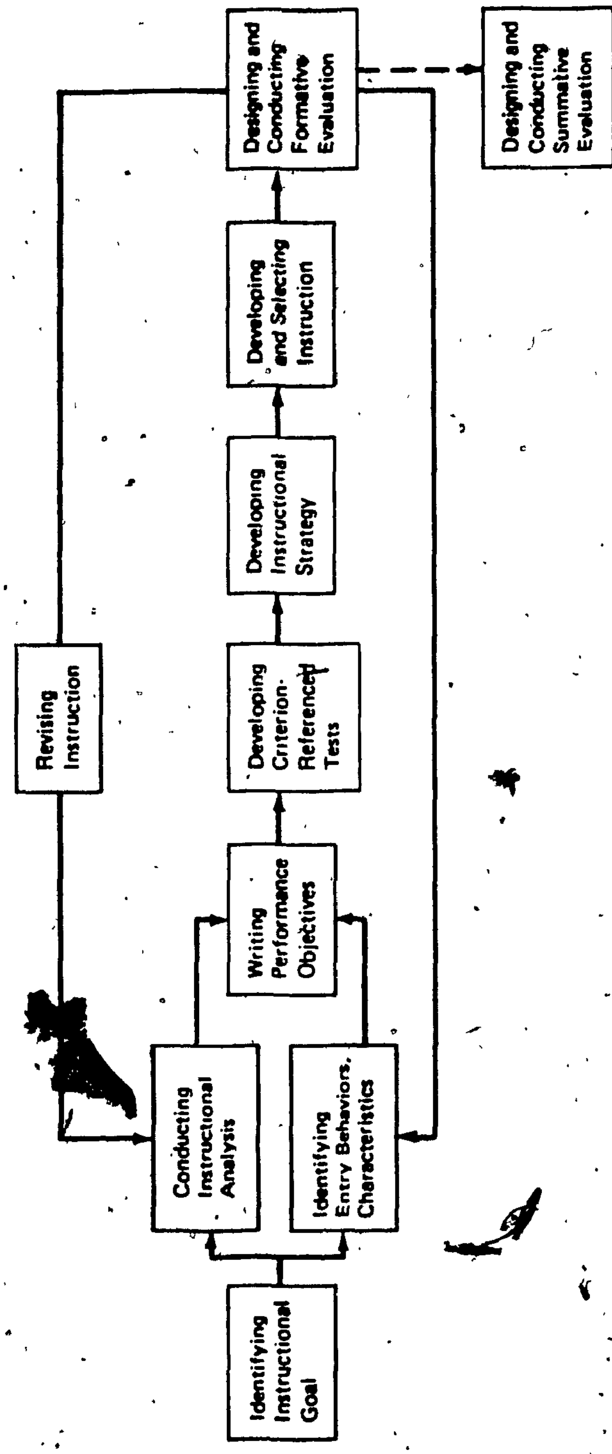
Torrence, E. P. Influence of dyadic interaction on creative functioning. Psychological Reports, 1970, 26, 391 - 394.

Trowbridge, D. & Durnin, R. Results from an investigation of groups working at the computer. California University, Irvine. Educational Technology Centre, 1984. (ERIC Document No. ED 238 724).

Watt, D. Learning with Apple Logo. New York: McGraw-Hill, 1984.

Watt, D. Logo in the schools, BYTE Magazine, August, 1982.

APPENDIX 1



A Systems Approach Model for Instructional Design

Dick and Carey, 1978.

APPENDIX 2

Discussion on the Design of the Instructional Module

The model used to develop the instruction (Dick and 1978) involves a 9-step process. (see Appendix 1) Only eight of the steps are discussed here; the ninth concerns summative evaluation which is not an integral part of this thesis. Also, Developing an Instructional Strategy and Developing and Selecting Instruction are discussed together under the heading Developing the Materials.

Identification of the Instructional Module

This was identified as: Given a copy of the truck picture, the subjects will write and run a programme that will draw an approximate copy of the picture; it should be the same in shape but not necessarily the same size.

The Instructional Analysis

The instructional analysis diagram (Diagram 2) shows the skills, concepts and information that needed to be taught. It also shows the learning dependent relationship among the subskills in the hierarchy. This provided the sequence of instruction to be respected in the instructional module.

Identification of Entry Behaviours and Characteristics

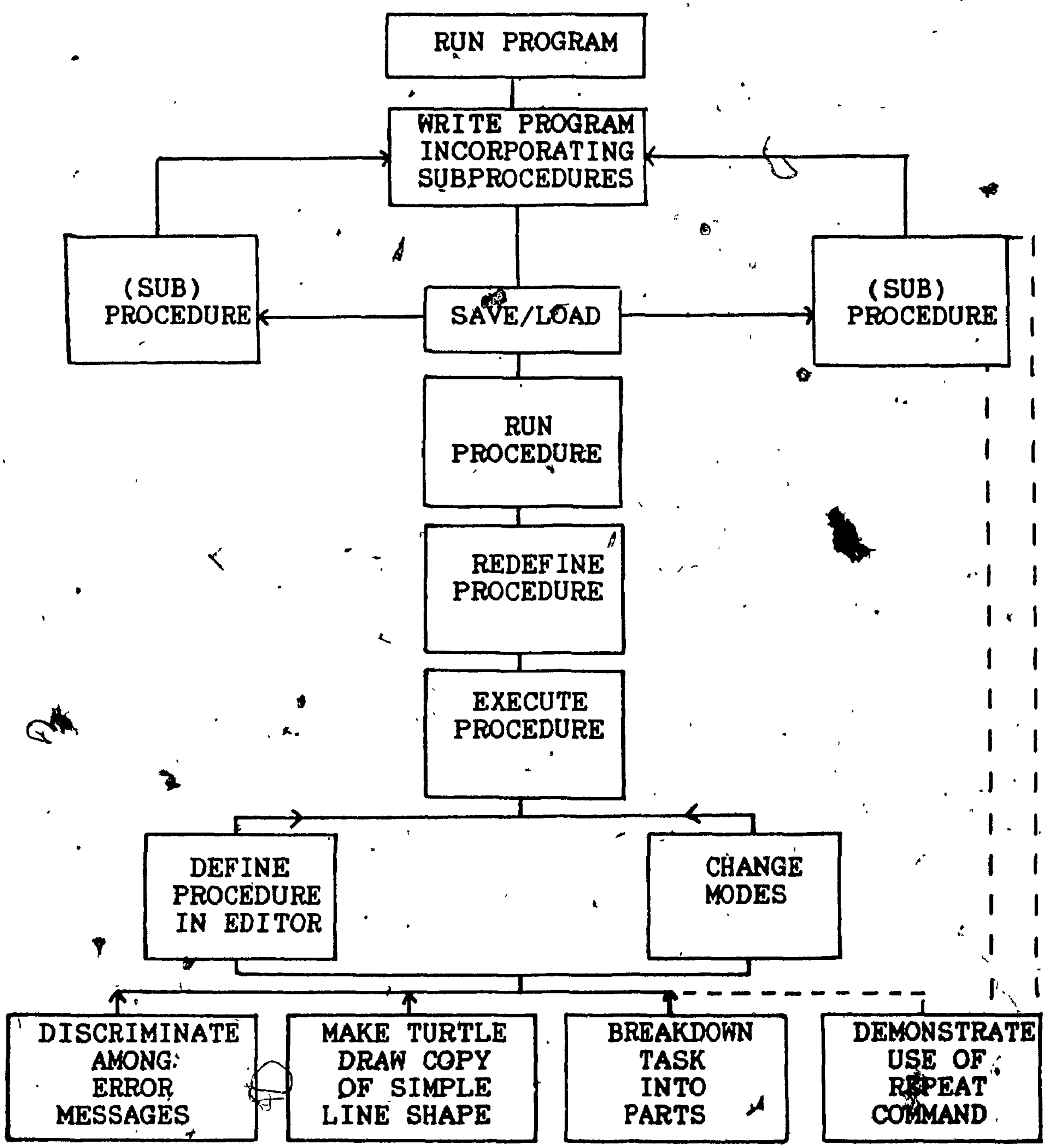
The population was of mixed intellectual capabilities and was drawn from several grade 5 classes and one grade 5/6 split class.

In the initial stage of the development process the designer consulted a teacher working with children of the target population. It was then possible to identify the population's level of English language skills and their specific entry skills in Logo so that these could be matched in the materials. According to the teacher, the population had variable entry skills. They had been taught by different teachers at different times and some had attended supplementary Logo workshops, however, the case of a child having no Logo skills was very rare. It seemed that on an individual basis members of the target population would perform at many different levels in the hierarchy of the instructional design diagram. The children were to be given the instruction individually or in a group of two or three. The possibly heterogeneous nature of any group's entry skills did not present a problem; the children had experience in working in groups on the the computer and they were encouraged to help one another in a group effort.

The maximum time period proposed for the children's work session was 45 minutes, this being the length of their school class period.

Diagram 2

Instructional Analysis Diagram: the hierarchical analysis of writing and\running a program in Logo



Writing Performance Objectives

The performance objectives for each unit were derived from the instructional analysis which identified the behavioural component. The conditions under which the behaviour had to be performed and the acceptable performance criterion were then added.

Developing Criterion-Referenced Tests

The test items were designed to measure the behaviours described in the behavioural objectives.

Developing the Materials

The purpose of the package was to establish that the population had attained a given level of entry behaviour in the language Logo. It required three parts: a pretest, instructional materials and a post-test. The pretest was a diagnostic checklist which was used to determine whether instruction was required and if so, at what stage of the learning sequence. The post-test was the Truck task. The subjects were to use their knowledge of basic turtle commands to make the computer draw a picture of a truck. (This particular problem was designed to encorporate the use of various basic Logo turtle graphics skills, for example, making simple shapes with the option of using the REPEAT command, using the PENUP command to draw one shape inside another, drawing a circle adjacent to a line, using subprocedures.)

In order to cover the eventuality that an individual or group had zero entry skills, the instruction was designed to cover the whole instructional analysis. The sequence for presenting the information during the instruction was as laid out in the analysis.

The need for a pretest and corresponding instructional materials that would meet the specified objectives could not be met by commercially available materials. The author developed materials specifically for use in the study. For the purpose of subject matter verification, the workbook by Watt (1984) was used.

The instructional materials, with the exception of Unit 6, were designed to be used only if it was necessary to bridge the gap between the subjects' previous level of skill and that required to achieve the instructional goal. Unit 6 involved new concepts and was presented to all groupings.

The materials were designed to employ a directive teaching approach, however, the subjects were provided with instant feedback from the researcher and the computer due to the interactive nature of Logo Turtle grahpics.

The components of the instructional package were:

1. researcher's instruction manual
2. diagnostic checklist
3. truck task
4. instructional materials and tests

Conducting a Formative Evaluation

A formative evaluation of the module was conducted to gather information on the effectiveness of the materials. Revisions were made on this basis. There were two stages to the evaluation: one-to-one and small group.

One-to-one Evaluation

The first draft of the materials was used with a colleague who had some knowledge of Logo and then later with two children. The first child was from the same grade level as that of the target group, but he was of above-average general ability and had a good knowledge of Logo. The second child was two years younger, also above average in ability, but he had little knowledge of Logo.

The children were seen on an individual basis in a quiet, non-school setting. The designer sat with each child and worked through the materials as suggested in the researcher's instruction manual. Audio tapes were made of each session to provide a record of difficulties, solutions, comments and reactions.

The Revision Process

Only Units 4, 5, and 6 were revised. Use of units 1 to 3 was not required in the pilot testing so there was no basis for revision. From the testing of Units 4 to 6, the audio tapes were analyzed to establish what revisions were necessary. Changes were made relating to content and the procedures used in presenting the materials. These included omission and addition of information and examples and corrections of minor mathematical or logistical errors.

Revision were made in the printed instructional materials and on the accompanying Logo disk files. The manner of presentation was also changed. The spontaneous language from the audio recording was used to create a more natural wording for the instruction and the result was an instructional script from which the researchers could read verbatim.

Small-group Evaluation

The model used to develop the materials suggests that between 10 and 20 students participate in a small group evaluation. For the purpose of this study, it was necessary to consider constraints on time and the total number of subjects available. It was decided that 5 students would be sufficient. They, among others, volunteered to participate, but they were selected so as to be a representative target sample. The children were all from grade five classes. They were of mixed general ability and had various levels of experience with Logo. Two evaluations were conducted; one with two boys, another with two girls and a boy.

The revised materials were used in the school that the children were attending in a setting very similar to that of the experimental study. Again, audio recordings were made, however, during these sessions there was little intervention from the designer.

The revised materials were found to be successful and only very minor changes were necessary.

APPENDIX 3

Researcher's Instruction Manual

Introduction

The materials you will be using were designed for use in an observational study of 11-year-olds working on a computer with the language Logo. The subjects will be given a pre-test, at least a minimal amount of instruction, then a post-test before they are asked to complete the three experimental tasks.

The subjects will be administered the appropriate parts of the module in the grouping to which they have been assigned i.e. they will be alone, in a pair or in a group of three. Whereever the term 'subject' is used, it refers to any number of subjects in a given grouping, that is, a subject working alone or two or three working together. Similarily, the term 'he' or 'him refers to both male and female subjects.

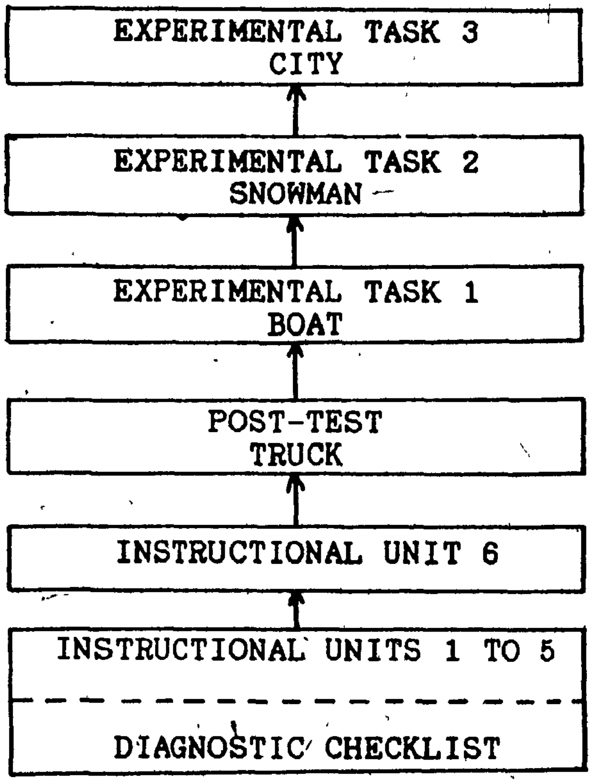
The materials to be used include:

- diagnostic checklist
- instructional materials units 1 - 6
- truck task
- experimental tasks

and are to be used in that order. See Diagram 3 for the sequence of instruction and testing.

Diagram 3

Sequence of Instruction and Testing



In Preparation for Each Session

1. Make sure the computer and monitor are working and Logo is booted and functioning.

2. Read the 'Instructions to the Researcher' on the cover page of the unit to be used. These provide information on

   a. files to be loaded or erased

   b. materials for the subject

   c. review of materials previously covered

At the Beginning of Every Session

1. Remind the subject he can refer to the Logo commands sheet and that he can use the blank paper and pencil provided, if

he wants to write something.

The Diagnostic Checklist

Seating

If there is one subject, sit next to him at the computer; if there are two or three subjects, have them stand in front of the computer. The researcher should sit at the side of the computer so the subjects and the monitor can be seen.

Administration of the Checklist

Ask the subject to perform the actions using the exact wording as shown in the column WORDING OF RESEARCHER. Where subjects are working in a group, ask each subject in turn to perform one of the actions. The subject should respond by typing the commands shown on the same line in the column COMMANDS. Begin with unit 1 and continue through to the end of unit 5 until either:

a. the subject working alone shows he is having difficulty (i.e. he is not able to attempt the action or fails to complete the action after three attempts),

b. all subjects working in the group show they are having difficulty (even if only one group member is able to complete the actions, continue working through the checklist) OR

c. The 5th unit has been completed

If the subject is not able to complete Unit 2 on Using the

Repeat Command, he should be given instruction immediately from the Instructional Materials Unit 2. When instruction is completed, go back to the Diagnostic Checklist and recommence with Unit 3.

If the subject is not able to complete an action in any unit other than Unit 2, stop using the checklist. Refer to the instructional material of the same name and number and commence instruction at the beginning of the unit.

Instructional Materials

Units 1 to 5 are to be used only if the subject cannot complete the actions on the checklist. For example, if unit 3 of the checklist cannot be completed, the instruction should commence at the beginning of unit 3 and should continue till unit 5 is finished.

Unit 6 is to be used with all subjects.

Truck Task Post-test

Once the subject has completed one of the following:

a. all five units of the diagnostic checklist
b. all six units of the instructional materials
c. a combination of a. and b.

turn to TRUCK TASK and follow the INSTRUCTIONS TO THE RESEARCHER. Note that this session will be recorded on audio tape.

Experimental Tasks

The three tasks will be admininstered on three consecutive sessions in the following order: boat, snowman and city. Turn to the EXPERIMENTAL TASKS and read the INSTRUCTIONS TO THE RESEARCHER. The instructions remain the same for all three tasks. These sessions will also be recorded.

APPENDIX 4

Diagnostic Checklist

Instructions to the Researcher

1. Consider seating arrangement
2. Ask the subject if he knows this kind of keyboard or if it's different in any way to the one he uses. Show him the [ ] keys.
3. Give him the sheet of Logo commands and read through them. He may know all or some - don't teach him now - tell him he'll be learning the new ones soon.
   Say he can use the sheet of commands whenever he likes if he forgets a command.
4. Ask the subjects to share the typing, if there are more than one.

MATERIALS

Logo commands sheet

Blank paper and pencil

ERASE

BOHI

Unit 1

Using Basic Turtle Commands

COMMANDS	WORDING OF RESEARCHER
ST/CS	First of all, can you make the turtle show on the screen?
FD 80	Can you make the turtle go forward 80 steps?
BK 40	and can you make it come back 40 steps
RT 90	Now make it turn right 90 steps
FD 35	and then forward 35 steps
LT 90	O.K. now make it turn left 90 steps
FD 40	then go forward 40
BK 80	and back 80
PU	Now I want you to make the turtle lift up its pen so it won't draw when it moves
RT 90	O.K. now make it turn right 90
FD 35	then go forward 35 steps
LT 90	and turn left 90
PD	Now we need to have the pen down again so the turtle can draw
FD 30	Go forward 30 steps
PU	then lift the pen up again
FD 7	and now take just 7 steps forward
PD	Now put the pen down for the last time
FD 5	and go forward 5 steps
HT	Now that we've written Hi can you hide the turtle so we don't see it on our picture?
CS	And if we want to erase what we've drawn what do we do?

ST
subject
types
commands
to make
square

Now I want so see if you can do something on your own. Do you think you can make me a square? You need to get the turtle on the screen.

I want you to type in each command, so don't use REPEAT

O.K. good, now can you clear the screen

Unit 2
Using the Repeat Command

Note

If the subject is not able to attempt or complete the following, give him/her instruction immediately from the Instructional Materials Unit 2 then proceed to Unit 3 of the Diagnostic Checklist.

COMMAND	WORDING OF RESEARCHER
	Now we're going to use the REPEAT command
"one"	When we're using REPEAT do we put all the commands on one line or on different lines?
e.g. REPEAT 3 [FD 10 PU FD 10 PD]	Can you make the turtle draw a dotted line like this. (draw on paper) Before you begin, you need to decide how to make it. What are you going to repeat and how many times?
CS	Good, now clear the screen
e.g. REPEAT 4 [FD 80 RT 90]	Now can you make the turtle draw a square using the repeat command?
CS	Good, now you can clear the screen again.

Unit 3

Defining a Procedure in the Editor and Executing it

COMMAND	WORDING OF RESEARCHER
ED "BOX or ED TO BOX	We're going to teach the computer how to do something that we can save on the disk in case we want to use it later. Let's call this a procedure. We're going to write the procedure in the editor. Let's call the procedure BOX. O.K. how do we start?
REPEAT 4 [FD 50 RT 90]	I want you to tell the computer how to make a square box using the repeat command. Make the square with its sides 50 turtle steps.
END	How do you tell the computer you've finished writing the procedure?
CTRL C	The turtle doesn't work when you're in the editor so can you get back to the turtle screen?
BOX	Now how can you make the turtle draw our box?

Unit 4

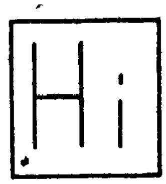
Editing a Procedure

COMMAND	WORDING OF RESEACHER
ED "BOX	O.K. now let's say we want to make the box bigger. We have to change the number of turtle steps that we put in our procedure. How can we do this?
CTRL N	Can you move the cursor down one line?
⟶	Now move it across to the 50
CTRL D12 or ⟵12	and change 50 to 120
CTRL C BOX	Now let's see the turtle draw the bigger box.

Unit 5

Saving and Loading a Procedure and using Subprocedures

COMMAND	WORDING OF RESEARCHER
	That's the size of box I wanted
CATALOG	I've written some procedures on the disk but I can't remember their names. Can you ask the computer to show me the list of the names (of the procedures I've saved)
LOAD "HI	The one I want is that one called HI so can you load it from the disk into the computer so we can use it?
HI	Will you show me what HI does?
ED "BOHI	O.K. but I'd like to put your BOX and my HI together so that when I type just BOHI the turtle will draw BOX and then draw Hi like this (on paper teacher draws the square first then Hi) Can you tell me what I want you to do?



BOX	Can you put BOX at the beginning?
HI	and then Hi next
CTRL C BOHI	Now let's see what the BOHI does.
SAVE "BOHI	Good, now will you save it for me please.

APPENDIX 5

Truck Task

Instructions to the Researcher

MATERIALS

Logo commands sheet

Blank paper & pencil

Truck task picture

Blank tape and beep signal tape

LOAD

POL

CIR

TEACH

DIRECTIONS

1. Make sure the blank tape and rewound beep signal tape are loaded in the tape recorders.
2. Remind the subject of the use of POL, CIR and REPEAT to make semi-circles and arcs.
3. Read the instructions for the Truck Task.
4. When the subject is ready to begin working, turn on the tape recorders, one with the beep signal, the other with the blank tape.
5. If the subject asks you a question concerning Logo, encourage him to try what he thinks might work, or refer him to the commands sheet.
6. Save the group's procedure at the end of the session.

TRUCK TASK

(Researcher's wording of instructions to the subject)

I'd like you to make the turtle draw a picture of a truck like this one (show picture).

Your picture should be the same shape but you don't have to make it the same size.

Decide how you can break down the design into pieces. You can use the turtle screen to try out your ideas but you have to put the procedure in the editor after. I'll give you some paper so you can write the commands down if you want.

Also, I want you to write the procedure in the editor so that it has just one procedure to draw the truck but it would be a good idea to put several small procedures in the big truck procedure.

You can call the procedure TRUCK and then when you've finished and you type TRUCK, the turtle should draw the whole truck.

Don't forget to use the list of commands I've given you if you need them.

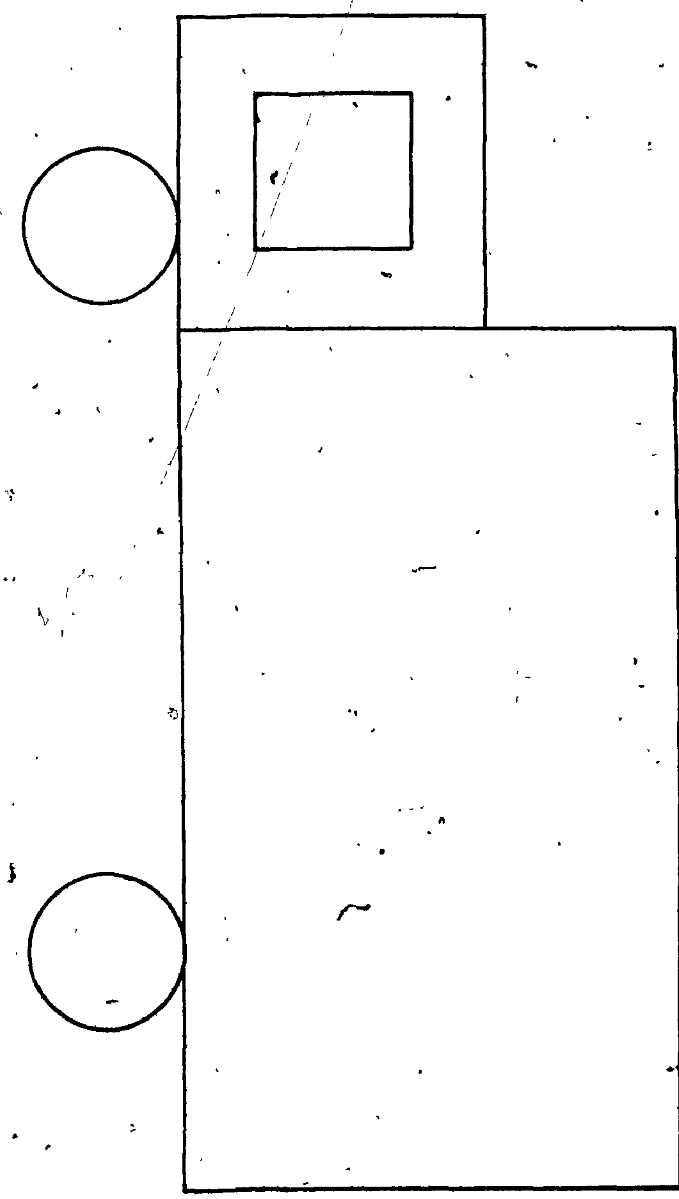
(if there are 2 or 3 subjects working together -)

You should help each other to do this and share typing in and writing down the commands.

O. K. Before you begin, can you tell me what you're supposed to do.

(Note: all procedures must be saved!)

TRUCK PICTURE



APPENDIX 6

Instructional Materials

Unit 1

Drawing Simple Line Shapes using Basic Turtle Commands

Instructions to the Researcher

MATERIALS

- Logo command sheet
- Blank paper and pencil
- Examples
- Exercises

Drawing Simple Line Shapes Using Basic Turtle Commands

Performance Objective

Given a series of three simple design exercises, the subject will make the turtle draw an approximate copy of each design in direct mode. The turtle designs should be the same in shape but not necessarily in size. The subject may refer to the list of Logo commands provided.

Content of the Unit

a. use of full and short commands

FORWARD	FD	+ INPUTS
BACK	BK	+ INPUTS
RIGHT	RT	+ INPUTS
LEFT	LT	+ INPUTS
CLEARSCREEN	CS	
PENUP	PU	
PENDOWN	PD	
HIDETURTLE	HT	
SHOWTURTLE	ST	

B. Discrimination among error messages

"I don't know how to ---"

"Not enough inputs to ---"

Introduction to the Keyboard

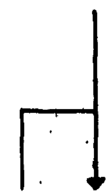
- you can see the keyboard for the computer is a lot like a typewriter
- when we press a key, the letter we pressed appears on the screen instead of on paper
- the blinking light is called a cursor - it shows you where you're going to type next (type "hello")
- you can erase "hello" by using ⟵

Introduction to the Turtle

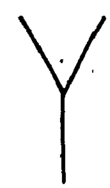
- "Welcome to Logo" means the computer is ready
- ? means it's ready for us to type
- we're going to meet a little turtle that will do a drawing for us
- first we have to make the turtle show on the screen
- type CLEARSCREEN and press return to have the turtle show on a blank screen
- now we're ready to make the turtle draw
- whatever we type in will show down here where the ? is
- these are the designs we're going to make the turtle draw. (give subject a copy of the EXAMPLES)

Demonstrate Basic Turtle Commands

Examples

Researcher types	Points to note	on screen
FORWARD 50	- command	
RIGHT 90	- space bar	
FORWARD 50	- number of amount of turtle steps.	
RIGT 90 ←		
RIGHT 90	- press return	
FORWARD 50	- use ← key to erase mistake	
BACK 100		
CLEARSCREEN		

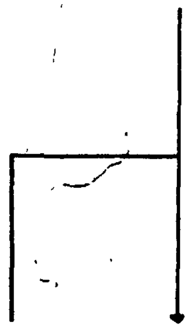
The researcher asks the subjects to take turns in typing EXAMPLES B. and C.

B.		
FD 70	- short commands easier to use	
RT 30.	- error message	
FD 40		
BK40		
BK 40		
LT 60		
FD 40		
HIDETURTLE		
CS		

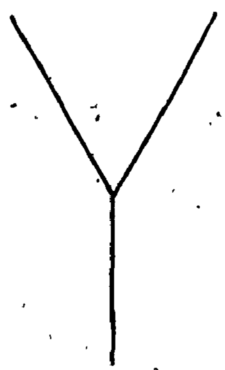
C.		
SHOWTURTLE	- error message	
HT		
ST		
RT 90		
FD 25		
PENUP		
FD 25		
PENDOWN		
FD25←		
FD 25		
PU		
RT 90		
FD 30		
RT 90		
FD 25		
PD		
FD 25		
HT		

Examples

1.



2.

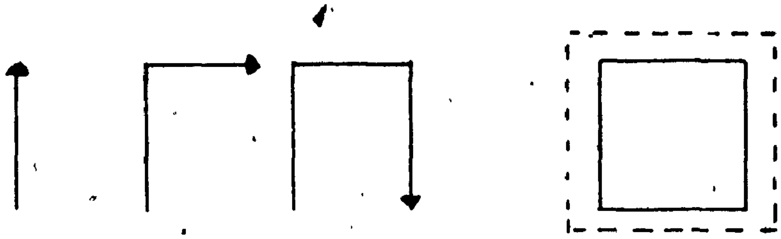


3.

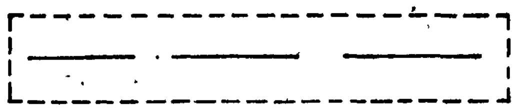
Exercises

Make the turtle draw the designs you see inside the dotted boxes.
The other lines will show you how to do it, step by step.

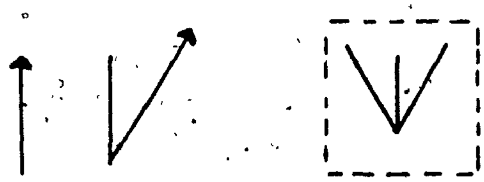
1.



2.



3.



Unit 2

Using the Repeat Command

Instructions to the Researcher

MATERIALS

Logo commands sheet

Blank paper and pencil

Exercises

Using the Repeat Command

Performance Objective

Given a series of three simple design exercises, the subject will use the REPEAT command to make the turtle draw an approximate copy of each design in direct mode. The turtle design should be the same in shape but not necessarily in size. The subject may refer to the list of commands provided.

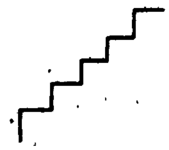
Content of the Unit

REPEAT [inputs]

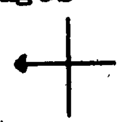
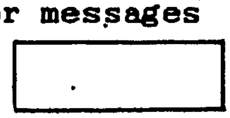
Introduction

If we want to make a square we have to type in a long list of commands to make the turtle go forward and turn and forward and turn etc. four times. We can make this a lot easier if we use the REPEAT command. We can make the turtle draw something as many times as we want.

Examples

Researcher types	Points to note	on screen
A. FD 10 RT 90 FD 10 LT 90 FD 10 RT 90 FD 10	- long list of commands to make 2 steps	 
REPEAT 5 [FD 10 RT 90 FD 10 LT 90]	- to make 5 steps - shorter to use REPEAT - need to know how many times to repeat (note spacing) and what you want to repeat (in brackets) - all commands on one line	

The researcher asks the subjects to take turns in typing EXAMPLES B. and C.

Researcher types	Points to note	on screen
B. CS FD 20 BK 20 RT 90	- doing this 4 times will make a cross (show on paper)	
CS REPEAT 4 [FD 20 BK 20 RT 90]	- with REPEAT - forgot input for number of repeats - respond to typing error messages	
REPEAT 4 [FD 20 BK 20 RT 90]		
C. CS REPEAT 2 [FD 20 RT 90 FD 40 RT 90]	- to draw a rectangle - respond to typing error messages	

Exercises*

Use the REPEAT command to make these designs.

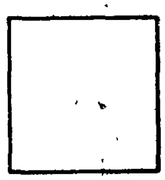
Your designs should be the same shape as these but they don't have to be the same size.

Before you begin each one, tell your teacher how you're going to do it. What are you going to REPEAT and how many times?

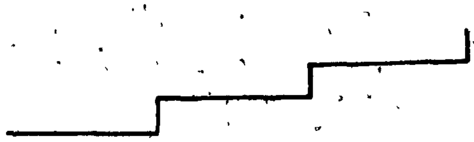
You need to clear the screen each time before you do a new design.

CS
1.

CS
2.



CS
3.



Unit 3

Defining a Procedure in the Editor and Executing it

Instructions to the Researcher

MATERIALS

- Logo commands sheet
- Blank paper and pencil
- Examples
- Exercises

Defining a Procedure in the Editor and Executing it

Performance Objective

Given a series of three simple design exercises, the subject will write a procedure in the edit mode then execute it. Two of the exercises will incorporate the use of the REPEAT command. (Whether the procedure runs of not at this stage is of little importance)

Content of the Unit

EDIT -----	'procedure'
ED	'editor'
TO ----	
END	
CTRL C	

----(execute procedure)

Introduction

We're going to teach the computer how to do something that it can remember. The commands that we type for that are called a procedure.

When we're writing a procedure we have to do it in a place called the editor. When you're in the editor you can't see the turtle.

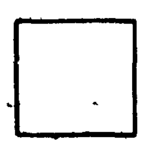
Here's a copy of the designs we're going to teach the computer to make.(give a copy of examples)

Examples

Researcher types	Points to note	on screen
A.		
EDIT "TEE	- use of shift 2 for "	
RT 90	- LOGO EDITOR at bottom of screen	
FD 25	- TO	
BK 50	- type slowly and carefully	
HT	- check command is correct before pressing return	
END		
CTRL C	- get back to drawing screen	
TEE		
CS		

The researcher asks the subjects to take turns in typing

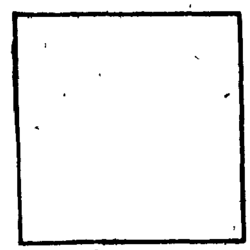
EXAMPLES B. and C.

Researcher types	Points to note	on screen
B.		
ED "BOX	- use of ← to erase error before pressing return	
FD 50		
RT 90		
FD 50		
RT 90		
FD50		
←		
FD 50		
RT 90		
FD 50		
END		
CTRL C		
BOX		

C.

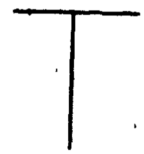
```
ED                    - to make a bigger square
TO BIGBOX             - using REPEAT
REPEAT 4
[FD 110
RT 90]
END

CTRL C
BIGBOX
```

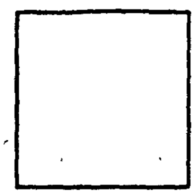


Examples

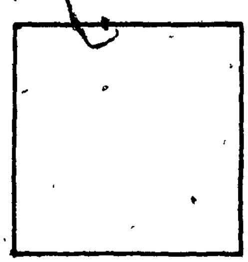
A. TEE



B. BOX



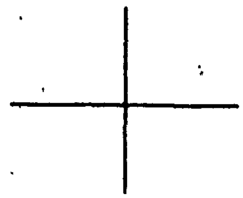
C. BIGBOX



Exercises

Write a procedure to draw each design then try it out to see what happens. Don't worry if it doesn't work! - we'll learn how to fix a procedure soon.

1.



Use the REPEAT command to make 2. and 3.

2.

3.

Unit 4

Editing and Running a Procedure

Instructions to the Researcher

MATERIALS

Logo commands sheet

Blank paper & pencil

Examples

Exercises

LOAD

UNIN4

Editing and Running a Procedure

Performance Objective

Given three procedures that contain bugs, an exact indication of the bug locations and a copy of the designs the procedure are intended to draw, the subject will edit each procedure so that it will run.

Content of the Unit

CTRL N

CTRL P

CTRL D

CTRL B

CTRL O

Introduction

(For both examples and exercises, the subject should see the design that the procedure is intended to draw. Also the procedure should be called before editing is attempted.)

Examples

I've written some procedures but they didn't do what I wanted them to do because they had some mistakes in them.

Here's a list of the commands in the procedures and the designs they're supposed to draw (give the subject Examples Unit 4)

A.
Let's look at the first one. It's called I.
Type I and let's see what it does.

That's what it does but what I wanted it to do is this.(show the subject the design on his Example paper).

Look at the procedure. The place marked with an X is where the mistake is.

Do you know what I've done wrong? (subject may respond that you forgot to put the pen down.)

(show the subject on his Example paper what the commands produce.) Look, we did forward 60 then penup then forward 20. WE need to put the pen down after the FD 20.

Can you get back into the editor so we can fix it? (subject types EDIT "I)

O.K. we need to move the cursor down so let's look at your list of commands.(show on LOGO commands sheet)CTRL N will move the cursor down to the next line, N is for next.

Now move it down to FD 15 and then type CTRL O to open a line, O is for open(show on LOGO command sheet)

Now you can type PD.

O.K. let's get out of the editor (CTRL C) and try I again.

Good, you've fixed it.

B.
Let's look at another procedure that has a mistake.

Type RECT. That's not what we want, is it?

What's the problem? (REPEAT 1 should be REPEAT 2)

So let's edit RECT (subject types EDIT "RECT)

We need CTRL N to get to the next line then move the cursor over (with the arrow key)

Now we need to delete 1 (show subject on command sheet) so we use CTRL D, D is for delete.

O.K. now you can type 2

Alright, let's see it (subject types CTRL C and RECT to get a rectangle on the screen)

C.
The last one is called EL

Can you show me it first?(screen will say LOGO doesn't know how to F in EL)

Can you fix it for me? Type EDIT "EL

O.K. now move the cursor down to F and put the D that's missing. Good.

O.K. let's try it (subject types CTRL C and EL)

Now I've got some procedures I want you to fix on your own.(give the subject the EXERCISES sheet and read through the instructions with him.)

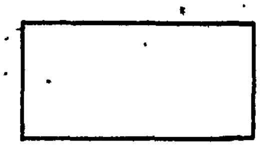
Examples

The X shows you where the mistake is:

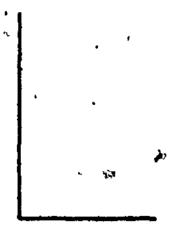
The procedure | the design

```
A.
TO I
FD 60
PU
FD 20
(Line missing)
FD 15
HT
END
```

```
B.
TO RECT
REPEAT X [FD 30 RT 90 FD 60 RT 90]
HT
END
```



```
C.
TO EL
LT 90
F 60
RT 90
FD 100
END
```



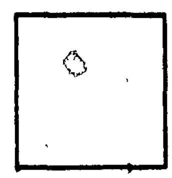
Exercises

Here are 3 procedures with the designs they are supposed to draw, but they don't work properly. I'd like you to fix them.

1. Type the name of the procedure to see what it does.
2. Find the X in the procedure. That means there's a mistake.
3. Fix the mistake in the editor.
4. Make sure the procedure draws a picture like the one here.

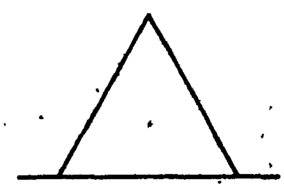
1.BOX

```
TO BOX
REPEAT 4 [FRD 40 RT 90]
HT
END
```



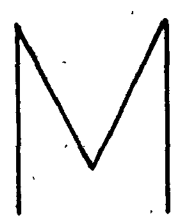
2.TENT

```
TO TENT
RT 30
FD 80
RT 120
FD60
RT 120
BK 20
FD 110
HT
END
```



3.EM

```
TO EM
FD 80
RT 135
FD 30
LT 90
FD 30
RT 135
FD 40
(line missing)
END
```



Unit 5

Saving and Loading a Procedure and Using Subprocedures

Instructions to the Researcher

MATERIALS

Logo commands sheet

Blank paper & pencil

Exercises

Saving and Loading a Procedure and Using Subprocedures

Performance objective

Given a) the names of 4 procedures and the designs they draw (saved on disk)

b) a series of 3 design exercises

the subject will

1) determine the names of the procedures.
2) use the catalog then load the procedures needed
3) encorporate the procedures (as subprocedures) into a new procedure

Content of the unit

SAVE "----

LOAD "----

CATALOG

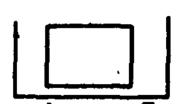
Introduction

(The notion of procedures and subprocedures should be explained)

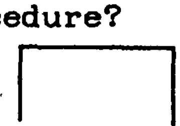
-we're going to teach the computer to do something new from what it already knows. What it already knows is called a procedure, right? We write procedures in the editor and save them on a disk so we can use them again. Well, we can load some small procedures from the disk into the computer and put them together to make a bigger procedure that can do more than one thing.

Examples

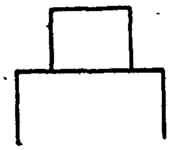
A.

Researcher types	Wording of researcher	on screen
EDIT TO SQUARE REPEAT 4 [FD 40 RT 90] END	First of all, I'm going to write a procedure called SQUARE in the editor	
CTRL C SQUARE	Now let's look at it	
Subject types		
CATALOG	I've written some other procedures that are on the disk but not loaded. We're going to use those and our square to make a big procedure that contains 2 small procedures. Will you show me the list of my procedures.	
LOAD "U	I want you to load that one called U so you have to type LOAD "U	
U	Let's see what it does.	
	O.K. we know what SQUARE does so now let's put them together.	
EDIT TO SQUAREU	We're going to call the new bigger procedure SQUAREU so can you get us into the editor. Now type TO SQUAREU.	
SQUARE U	Now I want the turtle to draw the square first so move down a line and type SQUARE then down another line and type U	
END	Now we need to put END	
CTRL C SQUAREU	Now let's see what SQUAREU does	

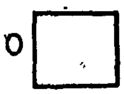
B.

CATALOG	Let's look at the catalog again. I need you to load N for me. Do you remember how to load a procedure?	
LOAD "N	LOAD "N	
N	Let's see what it does.	

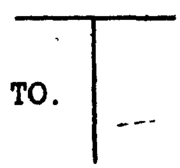
	We're going to put N and SQUARE together to make a new procedure called NSQUARE
EDIT TO NSQUARE	So let's get into the editor and start a procedure called NSQUARE
N SQUARE END	We want N first then SQUARE and now END
CTRL C NSQUARE	O.K. let's see it



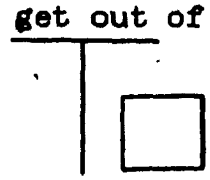
C. CATALOG	For the last one we'll use 2 procedures you haven't seen. Look in the catalog and I'll show you what we need.
LOAD "O LOAD "T	We want to load O and T
O	Now let's look at them. Type O first



CS T.	O.K. clear the screen and then type T.



EDIT TO TO.	Let's call the new procedure TO. We have to put a . after TO
T O END	We need the T first then O and END
CTRL C TO.	So now let's get out of the editor and see TO.



Exercises

(The procedures the subject will need for these exercises are saved on the disk but should not be loaded by the teacher.

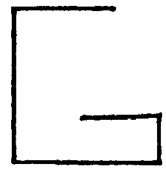
Give the subject the EXERCISES sheet and read over the instructions with him.)

Exercises

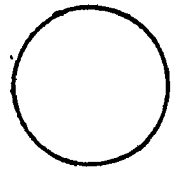
the names of the procedures saved but not loaded

the design drawn by procedure

G

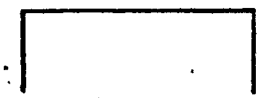


PENNY



LONG

FRAME

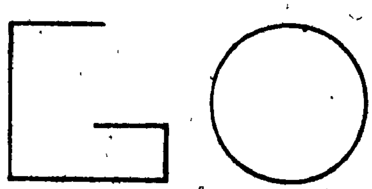


Make 3 new procedures that will draw the designs below.

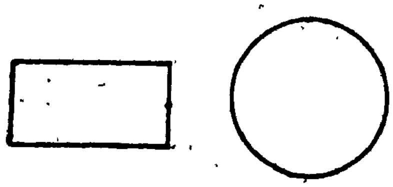
Use the CATALOG then load the procedures you need.

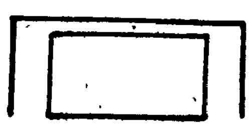
Look at each procedure when you load it.

Give the 3 procedures any names you want.

1. 

don't call this one GO - it won't work!

2. 

3. 

Unit 6

Drawing Polygons, Circles and Arcs

Instructions to the Researcher

MATERIALS

Logo commands sheet

Exercises

List 1

Blank paper & pencil

LOAD

POL

CIR

Drawing Polygons, Circles and Arcs

Performance objective

Given a) 2 procedures called POL and CIR on the disk

b) a series of 3 design exercises

the subject will

1) determine the procedure (POL or CIR) and inputs needed to create 2 of the designs

2) determine the input to REPEAT and the angle needed to create the third design

Content of the unit

CIR + input

POL + input

Introduction

Do you know how to make circle and arcs? (subject will likely respond Yes, with CIRCLER and ARCR)

Well, I'm going to show you a different way of making them.

Can you check that the pen is down and the turtle is showing.

Can you type:

REPEAT 3 [FD 20 RT 120]   How many sides does this shape have?

Now I want you to type in some more REPEAT commands and we'll make different shapes .

I don't want you to erase them each time so we can see all the shapes on the screen at the same time.

REPEAT 4 [FD 20 RT 90]

REPEAT 6 [FD 20 RT 60]

REPEAT 10 [FD 20 RT 36]

REPEAT 15 [FD 20 RT 24]

(Give the subject List 1)

-If we have a look at all the commands we've given the computer they all look like REPEAT, then a number of sides and forward 20 was always the same, then right a certain angle.

Well the angle is the same as 360 divided by the number of sides in the shape.(Show on paper)

e.g. look at the square with 4 sides. If we divide 360 by 4 we get 90. Also for the shape with 6 sides the angle is 360 divided by 6 which equals 60.

I think the easiest one is the shape with 10 sides. 360 divided by 10 is 36, right?

I've written a procedure that will let us make a shape with any number of sides we want. It's called POL.

You can type in POL 3 and you get a shape with 3 sides. See?

Try some other POL numbers but don't use more than 22.

Also don't clear the screen then we can look at all the shapes together and compare them.

So tell me again, what is this number that you're typing after POL? (subject should say it's the number of sides).

You can see that when the number of sides is bigger, the shape looks more like a circle.

So with the procedure POL we can make a triangle, a square and other shapes that look more and more like a circle.

Now we're going to look at how to make circles of different sizes.

If we have a small number of forward steps before the turtle turns, we get a small circle, right?

And if we have a bigger number of forward steps before it turns, what happens? (the circle will be bigger)

I've written another procedure that lets us change the forward steps to get the size of circle we want.

Now let's try it. Type CIR 5, but don't clear the screen then we can look at them all together.

The number 5 is the steps it goes forward before it turns each time.
Now do CIR 8
CIR 10

You see when the CIR number we use is bigger, so is the circle, right?

CIR 14 is the biggest that will fit on the screen.

Let's just try CIR 16 to see what it does (it wraps)

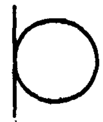
O.K. leave them on the screen.

All those circles are on the right side of the screen. What about drawing some on the other side. We'll turn the turtle around first so type

RT 180
now type CIR 10

What I want to do now is draw a circle that touches a line so can you clear the screen and type

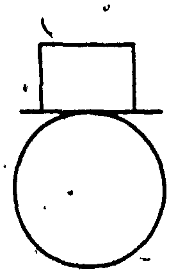
FD 100
BK 50
CIR 5



(on screen)

Let's put all the things we've just learned together to make a picture. It's a round face with a little square hat on it. O.K. so type

CS
RT 90
FD 100
BK 50
CIR 8
BK 10
LT 90



(on screen)

and if you type POL 4 what will it give us? (a square) That's right, POL gives us the number of sides we want but CIR makes a circle the size you want.

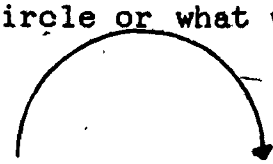
POL
HT

O.K. we've looked at different shapes and circles of different sizes. Now I want to show you how to make just part of a circle.

Remember that the number of sides times the angle equals 360. Let's look at what happens when it equals only 180.

type CS
ST
REPEAT 15 [FD 10 RT 12]

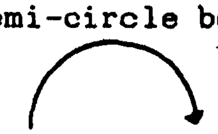
so we get half a circle or what we call a semi-circle (on screen)



(show on paper) if you multiply 15 repeats by 12 turning steps it only makes 180 which is half 360

Now let's see what happens when we change the FD 10 to FD 5.
Type CS
REPEAT 15 [FD 5 RT 12]

We get a smaller semi-circle because the number of forward steps was smaller.
(on screen)



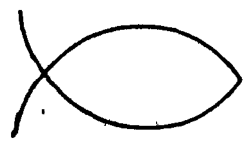
This time we'll just repeat 10 so type

```
                CS
then            REPEAT 10 [FD 10 RT 12]
(on screen)
```

you see we get an arc because the line doesn't go even half way round a circle. Ten times 12 is 120.

We can make a design that looks a bit like a fish so type

```
                RT 90
     and        REPEAT 10 [FD 10 RT 12]
(on screen)
```



These are both arcs because they're not complete half-circles.

Remember for the half-circle we had 15 repeats but here we only have 10 so it doesn't go round as far.

O.K. that's the end of the things I wanted to show you but now I want to see if you can do something on you own.

Can you use what we've just learned to make these designs? (show the subject the Exercise page)

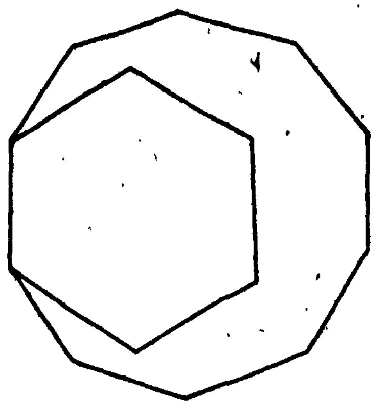
LIST 1

```
REPEAT 3 [FD 20 RT 120]
REPEAT 4 [FD 20 RT 90]
REPEAT 6 [FD 20 RT 60]
REPEAT 10 [FD 20 RT 36]
REPEAT 15 [FD 20 RT 24]
```

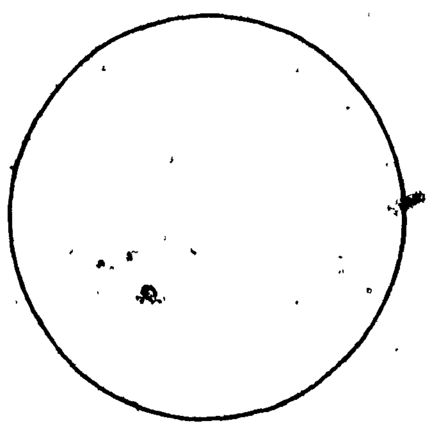
Exercises

Make these designs using the commands you just learned.

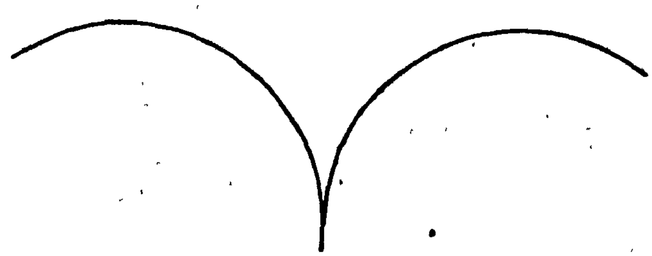
1.



2.



3.



APPENDIX 7

Logo Commands

FULL	SHORT	
Turtle commands		
FORWARD	FD number	- move the turtle forward
BACK	BK number	- move the turtle back
RIGHT	RT number	- make the turtle turn right
LEFT	LT number	- make the turtle turn left
PENUP	PU	- lift the turtle pen up
PENDOWN	PD	- put the turtle pen down
HIDETURTLE	HT	- hide the turtle
SHOWTURTLE	ST	- show the turtle
CLEARSCREEN	CS	- erase your pictures and give a blank screen
REPEAT number [---]		- repeat the commands in the [ ] a number of times
Making a Procedure		
EDIT "--- or EDIT TO ---	ED "	- start writing a procedure called ---
END		- the procedure is finished
CONTROL	CTRL C	- bring back the drawing screen
	CTRL G	- stop what you are doing

Editing commands

CTRL N	- move the cursor down one line (to the next line)
CTRL P	- move the cursor up one line)to the previous line)
CTRL D	- delete (erase) a letter or number where the cursor is
CTRL B	- move the cursor to the left without erasing
	- move the cursor to the right
	- delete (erase) the letter or number on the left
CTRL O	- make a new empty line (open a line)
SAVE "---	- save the procedure called --- onto a disk so you can use it again
LOAD "---	- bring the procedure called --- from the disk to the computer so you can use it
CATALOG	- give a list of the names of the procedures you have saved on your disk

APPENDIX 8

Experimental Tasks

Instructions to the Researcher

MATERIALS

Logo commands sheet

Blank paper and pencil

Copy of the appropriate task picture

Blank tape and beep signal tape

LOAD

POL

CIR

TEACH

DIRECTIONS

1. Make sure the blank tape and rewound beep signal tape are loaded in the tape recorders.
2. Remind the subject of the use of POL, CIR and REPEAT to make semi-circles and arcs.
3. Read the instructions for the Experimental Tasks.
4. When the subject is ready to begin working, turn on the tape recorders, one with the beep signal, the other with the blank tape.
5. If the subject asks you a question concerning Logo, encourage him to try what he thinks might work, or refer him to the command sheet.
6. Save the grouping's procedure at the end of the session.

Experimental Tasks

(Researcher's wording of instructions to the subject)

This time I'd like you to make the turtle draw a picture like this. (Give the subject a copy of the appropriate task picture.)

Your picture should be the same shape but you don't have to make it the same size.

Don't forget to use the list of commands I've given you if you need them.

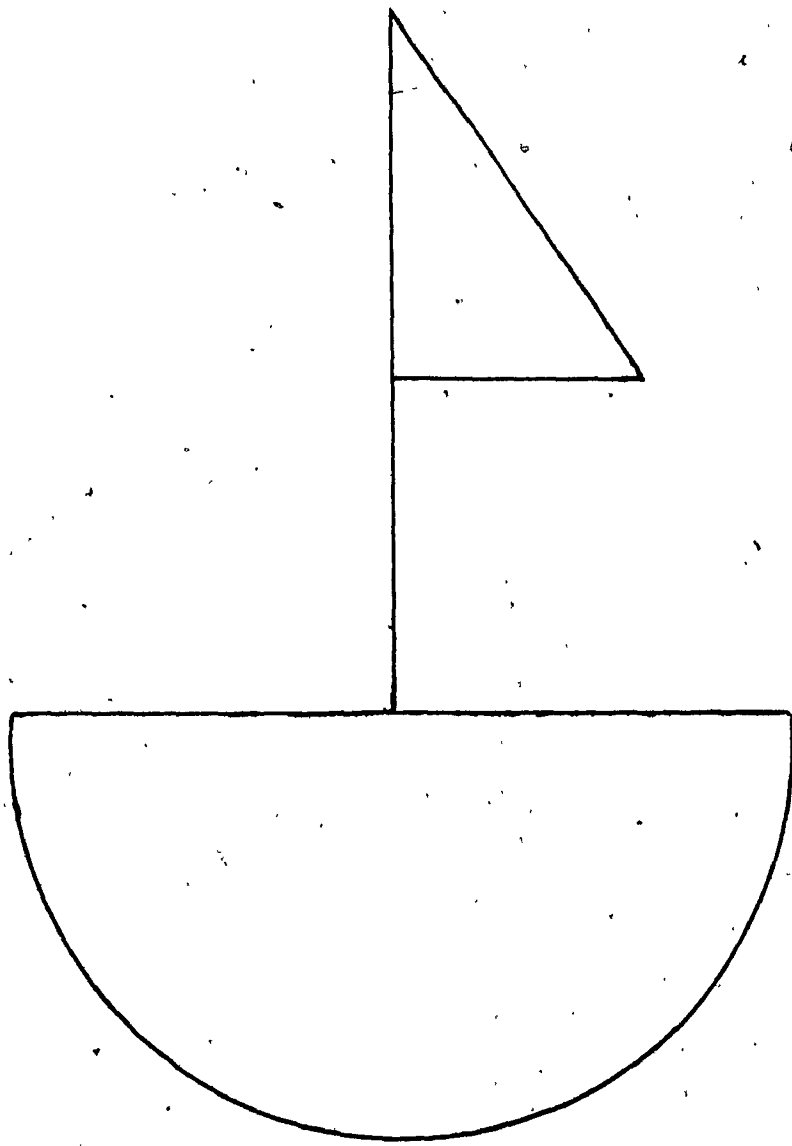
O.K. before you begin, can you tell me what you're supposed to do?

Note

DO NOT ask the subject to write procedures or write in the editor; let him use his own strategies for these tasks.

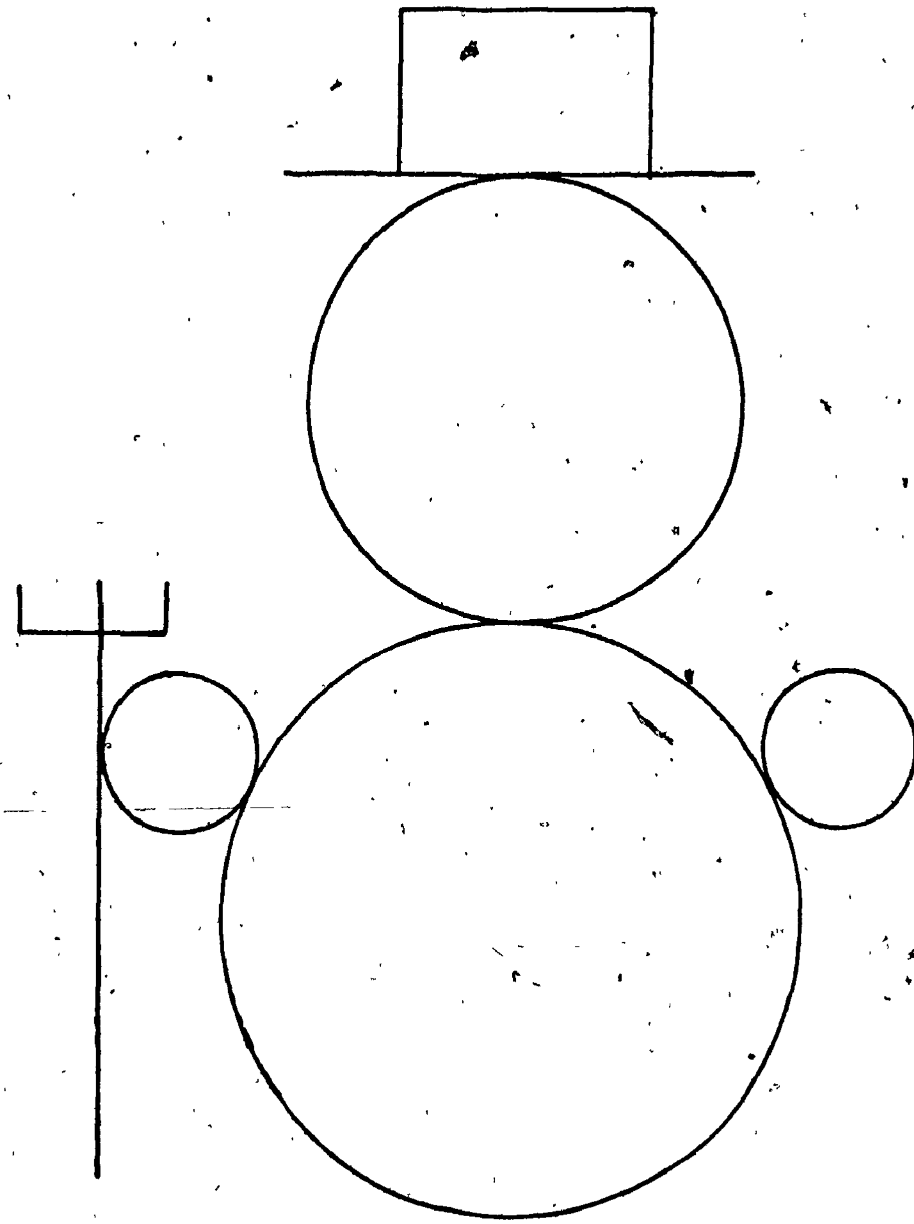
Picture 1

BOAT



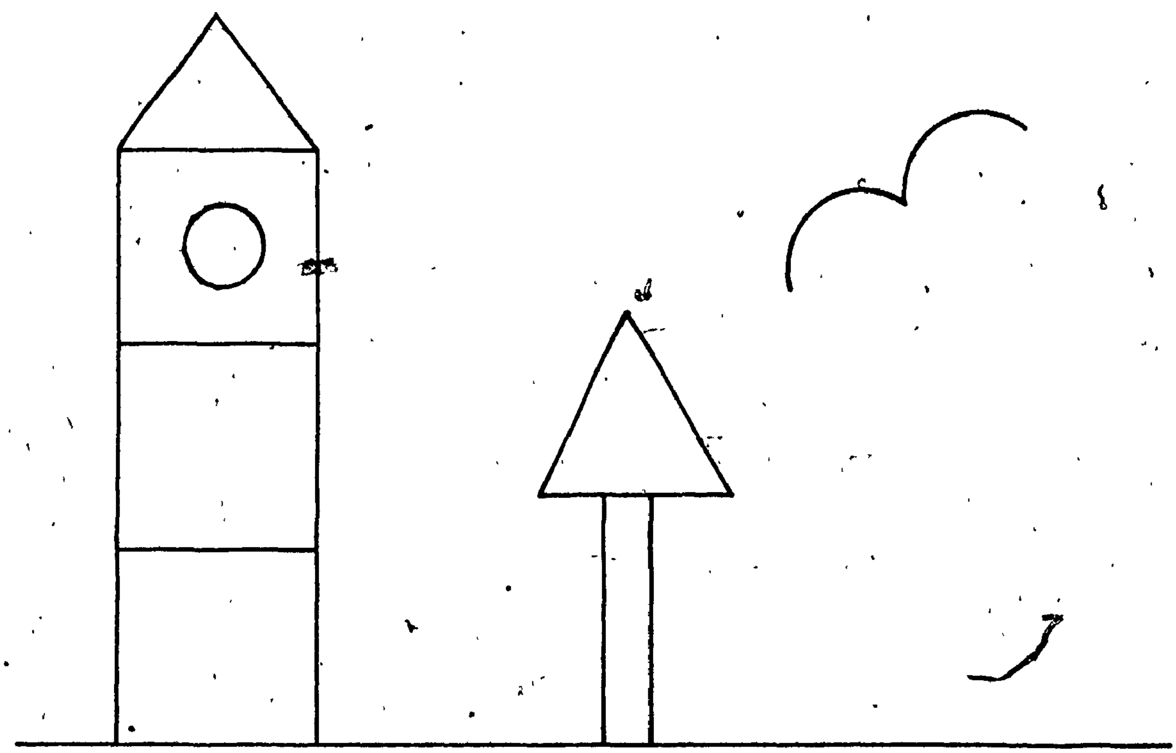
Picture 2

SNOWMAN



Picture 3

CITY



APPENDIX 9

Experimental Task Evaluation Criteria for Completed Components

CATEGORY	DESCRIPTION	RANGE OF POINTS TO BE DEDUCTED
A	incomplete form of component - lines need to be extended	5 - 20
B	components not joined - they should be touching	5 - 20
C	components poorly aligned or placed	5 - 20
D	lack of overall neatness - extraneous lines	5 - 20
E	poor shape	5 - 10

APPENDIX 10

Experimental Task Evaluation Instrument

GROUP ________

TCT: Task Completion Time
CMMDS: Number of commands for most complete version

TRUCK

components	
cab	
window	
box	
wheel lt	
wheel rt	

points deducted	category
	A
	B
	C
	D
	E
total deductions	

TCT ________

CMMDS ________

FINAL SCORE 100 - $\frac{X}{5}$ ____ =

Comments:

BOAT

components	
mast	
bottom	
sail	

points deducted	category
	A
	B
	C
	D
	E
total deductions	

TCT ________

CMMDS ________

FINAL SCORE 100 - $\frac{X}{3}$ ____ =

Comments:

GROUP ________

SNOWMAN

components

hat	
head	
body	
armlt	
armrt	
fork	

points deducted	category
	A
	B
	C
	D
	E
total deductions	

TCT ________

CMMDS ________

FINAL SCORE 100 - X $\frac{\quad}{6}$ =

Comments:

CITY

components

bird	
treetrnk	
treetop	
towerbtm	
towermdl	
towertop	
towerclk	
spire	

points deducted	category
	A
	B
	C
	D
	E
total deductions	

TCT ________

CMMDS ________

FINAL SCORE 100 - X $\frac{\quad}{8}$ =

Comments:

APPENDIX 11

Group Profiles of the Analysis of Programming Style

GROUP M1

1. TYPE OF PROGRAMMING	1a. linear ..✓.. 1b. procedural .....
2. MODE(S) USED	2a. direct .data missing 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent ..... n/a local ..... global ....
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ...... 4b. other ..data missing

GROUP M2

1. TYPE OF PROGRAMMING	1a. linear .✓... 1b. procedural .....
2. MODE(S) USED	2a. direct .✓.. 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent ..... n/a local ..... global ....
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior .✓... 4b. other .....

GROUP M3

1. TYPE OF PROGRAMMING — 1a. linear .✓... / 1b. procedural .....

2. MODE(S) USED — 2a. direct ..✓.. / 2b. indirect ..... / ....

3. LEVEL OF PLANNING — 3a. pre-planning ..... / 3b. contingent ..... / local ..... n/a / global ....

4. SEQUENCE OF PROGRAMMING COMPONENTS — 4a. interior to exterior ...... / 4b. other .✓...

GROUP F1

1. TYPE OF PROGRAMMING — 1a. linear ..✓.. / 1b. procedural .....

2. MODE(S) USED — 2a. direct ..✓.. / 2b. indirect ..... / ....

3. LEVEL OF PLANNING — 3a. pre-planning ..... / 3b. contingent ..... / local ..... n/a / global ....

4. SEQUENCE OF PROGRAMMING COMPONENTS — 4a. interior to exterior ..✓... / 4b. other .....

GROUP F2

1. TYPE OF PROGRAMMING	1a. linear . data missing 1b. procedural .....
2. MODE(S) USED	2a. direct .. data missing 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent ..... local ..... n/a global ....
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ...... 4b. other ... data missing

GROUP F3

1. TYPE OF PROGRAMMING	1a. linear .✓.. 1b. procedural .....
2. MODE(S) USED	2a. direct .✓.. 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent ..... local ..... n/a global ....
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ...... 4b. other ..✓..

GROUP MM1

1. TYPE OF PROGRAMMING	1a. linear ..✓.. 1b. procedural .....
2. MODE(S) USED	2a. direct ..✓.. 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent ..✓.. local ...✓. global ...2B 3A........
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ..✓.. 4b. other .....

GROUP MM2

1. TYPE OF PROGRAMMING	1a. linear ..✓.. 1b. procedural .....
2. MODE(S) USED	2a. direct ..✓.. 2b. indirect ✓.... used to create subprocedure that..... contained at least one error
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent .✓.. local ..✓. global ...1A..........
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ..... 4b. other ..✓..

GROUP FF1

1. TYPE OF PROGRAMMING	1a. linear ..✓.. 1b. procedural .....
2. MODE(S) USED	2a. direct ..✓.. 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent .✓... local ..✓.. global ✓...2.6 ....
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ...... 4b. other ..✓..

GROUP FF2

1. TYPE OF PROGRAMMING	1a. linear .✓... 1b. procedural .....
2. MODE(S) USED	2a. direct .✓... 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning data missing 3b. contingent ..... local ..... global ....
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ..✓... 4b. other .....

GROUP MF1

1. TYPE OF PROGRAMMING	1a. linear . ✓ . . . 1b. procedural . . . . .
2. MODE(S) USED	2a. direct . ✓ . . . 2b. indirect . . . .
3. LEVEL OF PLANNING	3a. pre-planning . . . . . 3b. contingent . ✓ . . . local . . ✓ . . global . . 2B . . . .
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior . . . . . . 4b. other . ✓ . . .

GROUP MF2

1. TYPE OF PROGRAMMING	1a. linear . . ✓ . . 1b. procedural . . . . .
2. MODE(S) USED	2a. direct . . . ✓ . 2b. indirect . ✓ . . . used to create subprocedure that . . . . contained at least one error
3. LEVEL OF PLANNING	3a. pre-planning . . . . . 3b. contingent . . ✓ . local . . . ✓ . global . . . 2A . . 2B . . . . . .
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior . . . . . . 4b. other . . ✓ . .

GROUP MMM1

1. TYPE OF PROGRAMMING	1a. linear ..✓..
	1b. procedural .....
2. MODE(S) USED	2a. direct ..✓..
	2b. indirect .....
	....
3. LEVEL OF PLANNING	3a. pre-planning .....
	3b. contingent ..✓..
	local ...✓.
	global ..4A..1B.......
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ..✓...
	4b. other .....

GROUP FFF1

1. TYPE OF PROGRAMMING	1a. linear ..✓..
	1b. procedural .....
2. MODE(S) USED	2a. direct data missing
	2b. indirect .....
	....
3. LEVEL OF PLANNING	3a. pre-planning .....
	3b. contingent ..✓..
	local ...✓.
	global ....3A..........
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ..✓...
	4b. other .....

GROUP MFF1

1. TYPE OF PROGRAMMING	1a. linear ..✓.. 1b. procedural .....
2. MODE(S) USED	2a. direct ..✓.. 2b. indirect ....
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent .✓... local ..✓.. global ....2 A..........
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ...... 4b. other .....

GROUP MFF2

1. TYPE OF PROGRAMMING	1a. linear ..✓.. 1b. procedural .....
2. MODE(S) USED	2a. direct ..✓.. 2b. indirect ✓... used successfully to create........ procedure that runs
3. LEVEL OF PLANNING	3a. pre-planning ..... 3b. contingent ..✓.. local ...✓ global ...7A..7B......
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior .✓... 4b. other .....

GROUP MMF1

1. TYPE OF PROGRAMMING	1a. linear ..✓..
	1b. procedural .....
2. MODE(S) USED	2a. direct ..✓..
	2b. indirect .....
	....
3. LEVEL OF PLANNING	3a. pre-planning .....
	3b. contingent ..✓..
	local ...✓.
	global ...1A..1B......
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior ..✓...
	4b. other .....

GROUP MMF2

1. TYPE OF PROGRAMMING	1a. linear .✓...
	1b. procedural .....
2. MODE(S) USED	2a. direct .✓...
	2b. indirect .....
	....
3. LEVEL OF PLANNING	3a. pre-planning .....
	3b. contingent .✓...
	local ..✓.
	global ...1A..1B......
4. SEQUENCE OF PROGRAMMING COMPONENTS	4a. interior to exterior .✓...
	4b. other .....