AN ANALYSIS OF THE DIFFERENCES IN PROBLEM-SOLVING OF GIFTED AND NON-GIFTED CHILDREN USING THE LOGO PROGRAMMING LANGUAGE

Eustathia Georgiou Maniatis

A thesis presented to the
Faculty of Graduate Studies and Research, McGill University
in partial fulfillment of the requirements for the degree of
Master of Arts

DEPARTMENT OF EDUCATIONAL PSYCHOLOGY AND COUNSELLING

ABSTRACT

The differences in the process of problem-solving by gifted and non-gifted children using the Logo programming language are analyzed. Working in pairs, sixteen 9-11 year old children were taught Turtle geometry for a ten-week period. A classification system of protocol analysis, developed in 1978 and extended for the present study, was employed to examine differences in problem-solving in terms of programming style (procedural nets), types and frequencies of errors (bugs), concomitant coping strategies (debugging actions), and the relationship between errors and strategies for the gifted and non-gifted children. A model of problem-solving for beginners in programming in the context of goal-directed projects is proposed.

RÉSUMÉ

Les différences entre les processus de résolution de problèmes appliqués par des enfants doués et des enfants normaux ont été analysées, selon le langage de programmation Logo. Répartis en paires, seize enfants de 9 à 11 ans ont suivi un cours de géométrie "Turtle" d'une durée de dix semaines. Un système de classification des protocoles d'analyse, conçu en 1978 et augmenté aux fins de la présente étude, a été utilisé pour l'examen des différences entre les processus de résolution de problèmes pour ce qui est du style de programmation (réseaux de traitement), du type et de la fréquence des erreurs (erreurs), des stratégies d'adaptation concomitantes (mesures de mise au point) et de la relation existant entre les erreurs et les stratégies commises et utilisées par les enfants doués et les enfants normaux. L'étude se complète d'un modèle de résolution de problèmes s'adressant à des débutants en programmation et s'inscrivant dans un contexte de projets axés sur un objectif.

ACKNOWLEDGEMENTS

This project would never have been completed without the help of many individuals. The author is grateful to the Laurenval School Board and especially to Gene Zinniger, Supervisor of Special Services, for his help and cooperation. The principal, Gary Norman, the teachers, parents, and students of Irving Bregman Elementary School supported the research with enthusiasm and their contribution will not be forgotten. Finally, many thanks to the students who took part in the study.

Special thanks to Judy Legalais, who helped me carry out the study and for her help in the design of the research. Many thanks to Kiyoko Okumura Monpetit, of Logo Computer Systems Inc., for her "crisis intervention" sessions when the computer was being obstinate, and to Sharnee Chait, whose thesis gave me the incentive to work with Logo and for taking time off in her busy schedule to offer valuable suggestions.

Special thanks to Bobrow and Fieldman Architects and especially to Dennis Benoit for his advice on some aspects of the graphic representation of the procedural nets.

To Donal McAnaney, Drs. Groen and Shore, many thanks for offering help each in their area of expertise. A special note of thanks goes to Dr. Socrates Rapagna for taking time off his sabbatical year to read and offer valuable suggestions to certain sections of the thesis.

Special acknowledgement goes to the Computer-Based Instructional Research Lab for its support and the loan of the computer so the research could be carried out in the school environment.

Lorraine Coffin has been a friend and an associate for the last few years and an inspiration throughout the duration of the thesis. Her artistic talents are shown on the graphic outputs of Appendix A, and the set up of the problem-solving model. With patience and perseverence, she formatted and typed the entire Appendix A. A very warm and sincere thank you goes to her.

Special acknowledgement goes to my advisor Dr. Glenn F. Cartwright. His patience and professional advice have given me the impetus I have needed in the last few years to continue my graduate work and to finish this project. His friendship I hope will continue, in the years to come, to stimulate and enrich my life.

Finally, to my entire family and especially my husband and two sons, Dimitri and Tom, a very special thank you for their patience and support in the continuation of my studies.

This thesis was supported in part by a Quebec research grant (FCAC).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

Overview of the problem

> Ideas cause ideas and help evolve new ideas.
>
> Roger Sperry, (1965) p. 82

> In the Logo environment new ideas are often acquired as a means of satisfying a personal need to do something one could not do before.
>
> Seymour Papert, (1980b) p. 74

The current growth of computer technology is applying colossal pressure to all facets of our society (Evans, 1980). In the field of education, microcomputer power is becoming available to children of all ages and "poses forcefully" the question of the computer's role in education (Lawler, 1982, p. 2). The answer to this question is not a simple one.

Though not incomplete technologically, the machine, to date, is "incomplete" both philosophically and psychologically. In fact, technology is advancing at a much faster rate than even experts have predicted. Alvin Toffler's 1970 account of computer technology in future societies failed to foresee the "inescapable" effects of the microchip (Evans, 1980). Following the threads of thought of his predecessor Charles Babbage (Hyman, 1982), Alan Turing was able to show as early as 1937, that for any precisely defined computational procedure a machine could be constructed to execute it in an automatic fashion. "Computational", of course, did not mean "numerical" in the restricted sense, but as long as a task could be processed in a specific fashion, then a machine could execute it.

> Computational concepts are not concerned with counting (as the word "computational" may suggest), but with any rule-governed symbol manipulation.
>
> Boden (1979), p. 136

Thus, Turing created a mental abstract of a computer which embodies the general principles of any computer, nine years before the first electronic computer was built. This so-called "Turing Machine" consisted of a long tape, divided into squares on which symbols could be written and erased. At the time, it was able to perform three operations only. It could either read a symbol off a square, print a symbol on a square or move one square right or left. Although the machine's repertoire seemed quite limited, Turing was able to prove to the academics of his time (Turing, 1937) that if this were possible, a machine could be constructed to calculate any other possible combinations. Thus, a computer may be defined as a universal "Turing Machine". In its present state, it can be as powerfully good as the procedure or program which modifies it. In essence it is an incomplete machine ready to be completed in an "infinite" number of ways, each producing essentially a new machine (Ellis, 1974).

In the present state of the technology, however, the number of ways in which the computer can be completed is not infinite. There are some restrictions that must be faced, some boundaries that have to be set in order to build a procedure that can be useful to the machine. First, the task must be "computational": following a finite set of rules, one set of symbols is manipulated to produce some other set of symbols (Ellis, 1974).

There are three discernible computational procedures that are used in processing information serially: algorithms, semi-algorithms and heuristics.

These three approaches to finding the solution to a specified problem (problem-solving) differ in several respects and are used not only by computer programmers but also by mathematicians, musicians, researchers and ordinary people solving problems.

Algorithms, are complete procedures because they stop when no solution exists. This type of procedure is very important to computer programmers. Because of the computer's speed, answers may be found in a reasonable amount of time even though the steps are long and tedious. An example of this type of procedure used in statistics is the computation of a correlation coefficient. This approach to problem-solving was used in the early days of computer programming to develop most of the programs available at the time (Hofstadter, 1979). However, this "brute-force" approach to problem-solving did not prove to be amenable to the sorts of problems which did not have a set answer (Ellis, 1974).

Semi-algorithms, on the other hand, are procedures that will search for a solution in a finite number of preconceived steps but they are unable to arrive at a solution if no solution exists. They are incomplete procedures in that if no solution exists, they may stop for the wrong reasons or search endlessly for a non-existent answer.

Heuristic procedures, in contrast, describe a process which is not as definite as an algorithm but still has some form. While these procedures may not lead to final solutions, steps can still be specified so that they may be computable and therefore programmable. Many of the computer games in the market today are heuristically programmed. A heuristic program is one which starts out with an appropriate method of solving a problem within the context of some goal and uses feedback from the effects of the solution to improve upon its own performance. Heuristic programming is one of the major contemporary artificial intelligence techniques and an extremely important strategy for problem-solving in general.

Although problem-solving activities of specific types of tasks are amenable to computer use, the matter is not really settled. Interpretation of these computational procedures is dependent on some person or agent (Minsky, 1967). In the case of the computer the machine becomes the agent which interprets these guidelines or rules by first treating a computational procedure as a blueprint and then, once the computer becomes the machine defined by the procedure, uses the procedure as a set of rules to follow. The Universal Turing Machine is clearly a flexible tool as powerful as the program which defines it; that is, it is a consequence of the procedure, not the determinant. Accordingly, procedural analysis is the first logical step in developing a "complete" computer.

Within the educational context, the framework suggested by Taylor (1980) for understanding the application of computing in education depends upon seeing its use in terms of three basic modes: "tutor, tool, and tutee" or student. Although the first two modes have important places in education, the present research will deal with an application of the third mode: the mode in which the student teaches the computer, as for example, in Papert's Logo. Rooted in the cognitive theory of Piaget, Papert has advanced a computer "microworld", where children gain problem-solving skills by interacting with a programming language called Logo.

Most of the research studies on Logo have used a variety of subjects and age groups but have remained basically anecdotal reports on behaviour (Austin, 1976, Weir & Emanuel, 1972). In 1978, Chait, refined a classification system of protocol analysis of the Logo experience in an attempt to understand it better. This study will use and modify, where warranted, Chait's system in order:

1. to examine in an exploratory fashion the "problem-solving" activities of "gifted" and "average" subjects in the Logo environment to determine what differences, if any, exist between them.

2. to evaluate the classification system and in the process propose a model of problem-solving.

Differences in problem-solving will be examined in terms of:

a) types of programming errors (bugs),

b) concomitant coping strategies (debugging strategies) and

c) structure and complexity of projects completed (procedural nets).

CHAPTER II

Review of the Literature

According to Bruner (1979), humans, in a never-ending search for "truth", are intrinsic "science-makers" with the capacity for theory-making. In a common-sense fashion, a theory of human behavior is ultimately based on this philosophical assumption: we know how to do things and know that we do them. To Bruner, theory is what makes us comprehensible to ourselves and psychologists must begin with a better systematic description of this knowledge, "this intricate sense of what Man is like" (Bruner, 1979, p. 189), or they will fail in their task.

In an effort to understand our environment, theories have been derived from many methods and have taken many forms. Within the field of psychology, many believe that a scientific, psychological theory is the only basis for understanding behavior. The search for "reality", however, has assumed many guises in modern life (Baldwin, 1980). To cite an example, although the most influential theory to account for human knowledge in Plato's time was that it was acquired through the senses or by experience (empiricism), Plato set out to prove that his test case of mathematical knowledge could not be acquired in this fashion (rationalism). He developed his theory of "ideas", a name given to the whole concept that the mind's "intuition of ideas" can be a source of human knowledge (Ackerman, 1965). Within the structure of modern-day psychology, the two basic and opposing tenets of empiricism and rationalism are still evident (Hilgard & Bower, 1975).

Thus, the various theoretical traditions that have developed since Plato's time, have based their assumptions on different models. The mathematical model used by Plato and his followers, including Descartes and Leibniz, led them to suppose that the "source" of genuine knowledge is found within, although an awareness of this knowledge may be suggested by experience (Scheffler, 1965). In the empirical tradition, natural science is taken as the basic model and since natural phenomena are revealed by experience, the mind may be explained as a "tabula rasa" at birth and experience draws knowledge upon it through the senses.

Both viewpoints are evident in today's developmental psychology: stimulus-response theory is empirically based, while psychoanalytic theory is concerned with an analysis of Man's inner thoughts. However, the pragmatic view, as expounded by Peirce, James, and Dewey has been more influential within child psychology and has manifested itself in the progressive education movement. This epistemological view stresses the experimental character of empirical science, putting great emphasis upon the active phase of experimentation (Scheffler, 1965). Since experimentation involves the active transformation of the environment, activity is the process of learning from experience. The mind is conceived as neither a deep well of ideas nor as a blank slate upon which experience writes. Rather, it is viewed as a capacity for active <u>generation</u> of ideas whose function it is to resolve the problems posed to an organism by its environment. Modern day functionalism and cognitive developmental theory are based on this theory of Man.

Kant's Apriorism

The philosophy of Immanuel Kant sought a reconciliation between empiricism and rationalism (Sahakian, 1975). In order to do this, he gave up the "realism" of what we perceive (Vuyk, 1981, p. 39).

> The mere, but empirically determined, consciousness of my own existence proves the existence of objects in space outside me.
>
> (Kant, 1787/1964, p. 245)

Kant found this reconciliation in the distinction between analytic and synthetic judgements, i.e., two different kinds of knowledge. Logical structures, syllogisms and deductive reasoning are not dependent on sensory experience but are basic or a priori to the internal structure of our mind. Synthetic judgements, on the other hand, are related to experience or "reality". "Judgements of experience, as such, are one and all synthetic" (Kant, 1787/1964, p. 49), but are also a priori. An example of synthetic knowledge is found in mathematical models. The proposition 7 + 5 = 12 is not merely analytic but also synthetic. The concept of the sum of 7 and 5 contains more than the union of the two numbers into one.

Kant also distinguishes between objects: phenomena (experience) and "noumena" (transcendental). The causal link between phenomena, objects in experienced world, is noumena, the underlying-thing-in-itself (Boden, 1980, p. 92). Phenomena are thus structured by features of the human mind such as forms and categories of space, time, cause, identity.

Mathematical judgements are fundamental to Kant's reconciliation of realism and empiricism. New knowledge is thus acquired through a close relationship between a priori and posteriori judgements. Kant's constructivist epistemology has been extremely influential to Piaget's genetic epistemology.

Genetic Epistemology

Jean Piaget's theories of child development reflect his philosophical and biological background, and his main concern with the theory of knowledge, as expounded by his genetic epistemology.

> Genetic epistemology attempts to explain knowledge, ...on the basis of its history, its sociogenesis and especially the psychological origins of the notions and operations upon which it is based.
>
> (Piaget, 1970, p. 1)

Although the term "genetic epistemology" was first coined by child psychologist, J. M. Baldwin in the early 1900's, it was Piaget who carried the view into contemporary psychology (Boden, 1980). He and his associates have investigated various concerns empirically and have described in great detail the development of the assumptions that the external world is stable and composed of objects perceived to be permanent and moving around in space. Childhood plays an important role in this, since it is at this time that the basic notions of logic and mathematics are conceptualized and form part of the reality.

> It is with children that we have the best chance of studying the development of logical knowledge, mathematical knowledge, physical knowledge and so forth.
>
> (Piaget, 1970, p. 13)

Although Piaget's search for knowledge is strongly influenced by Kant's apriorism, there are distinct differences in the two epistemologies. Piaget is concerned with the origin of Kant's "a priori categories" and postulates that they are genetic in nature. "For Piaget, the ontogenesis of knowledge reveals not so much its (contingent) history as its (essential) nature" (Boden, 1980, p. 85). The second difference is the nature of "noumenon".

According to Kant "reality" is not only unknowable but also unchanging (Vuyk, 1981). Piaget sees "reality" as constantly changing and moving away from the subject.

Consequently, since 1927, Piaget has been publishing his findings on the child's understanding of space, time, logic and mathematics, and his influence on child psychology has been unprecedented (Murray, 1979). A brief summary of some of Piaget's fundamental concepts will be followed by a closer examination of the educational implications of Piaget's theory in the study of mathematics.

Piaget's approach to the problems in the study of human behavior reflects his background of biological training (Boden, 1980). He parallels human behavior to the picture of a complex, mutually regulatory system in equilibrium. This biological system adapts or changes to best suit its present environment. The adaptation is not a function of its own nature but of the total system.

> The organism is not independent of the environment but can only live, act, or think in interaction with it.
>
> (Piaget, 1971, p. 345)

In human terms, the unit or structure that changes is called a "scheme" and the process of adaptation is characterized by two counterbalancing forces: assimilation and accomodation. These complementary processes are very complex but broadly speaking they keep the system in a state of equilibrium. Assimilation refers to the capacity of the organism to handle new situations without fundamental change, while accomodation describes the process of change within the organism in order to accept new situations (Baldwin, 1980). Equilibration is the related process whereby the two functions of assimilation and accomodation are described.

> Equilibration is in fact a continuous sequence of equilibrium-disequilibrium- equilibrium etc. and it is the nature of this process which is so difficult to explain in a satisfactory way.
>
> (Vuyk, 1981, p. 68)

Piaget offers many psychological examples of this process: a child who is under a year old may be unable to pick up a tiny object although his grasping scheme allows him to pick up large objects. To adapt to the demands of the situation, the scheme has to accomodate in order to pick up the tiny object. The gradual acquisition of this new ability is the accomodation process. Once the ability is acquired, the grasping scheme can assimilate the tiny object (Baldwin, 1980). If the situation is not completely assimilated by the existing scheme, Piaget considers this experience to be food or "aliment" for the scheme. Aliment constitutes the motivational aspect of the experience. The child is naturally attracted to these situations which are assimilable but not completely so. They are challenging and they tend to attract the child to activating and changing the scheme. As children grow up, they become equilibrated to a broader repertoire of situations which they achieve through a broader range of organized and systematic schemes.

In terms of biology, intelligence is thus defined as adaptation to the environment and its two essential functions consist of understanding (assimilation) and inventing (accomodation) (Groen, 1978). While understanding consists of generating transformations belonging to the structure that models reality, invention consists of developing the structure itself. Intelligence is the condition of equilibrium that results from the functioning of these two complementary processes.

It is these biological structures (schemes) that Piaget hypothesized exist in the mind which led to the development of his stage theory of human development.

The other influence was the structure of mathematics.

> The fundamental hypothesis of genetic epistemology is that there is a parallelism between the structures of knowledge and corresponding formative psychological processes.
>
> (Piaget, 1970, p. 13)

An example of this parallelism is Cantor's development of set theory, based on the fundamental operation of one-to-one correspondence. Where did Cantor discover this operation? According to Piaget (1970), Cantor found it in his own thinking since the most elementary examination of thought reveals that it is a primitive operation.

Another important example cited of the striking resemblance between mental thought and mathematical structure is the structuralism of the Bourbaki school. Bourbaki is a pseudonym for a group of French mathematicians who set out to isolate the fundamental structures of all mathematics. In consequence, they cast some light into the nature of number (Papert, 1980b). They established three "mother structures": an algebraic structure, a structure of ordering, and a topological structure. These were later modified to include the notion of categories (morphisms). These "mother structures" of the Bourbaki school have a close resemblance with the three structures of operational thought in children. Algebraic structures are found in the logic of classes and in the classification of various degrees as the child sorts groups of objects into piles. From the earliest ages, children are able to order things in their environment and the problem of seriation of sticks of different lengths is an example of this notion. Topological structure involves the operations of dividing and ordering space (continuity and proximity) which Piaget found to precede any notion of Euclidean or projective geometries.

The empirical investigation of these and other behaviors in children were found to be charactetistic of various stages of intellectual functioning and ultimately led to Piaget's stage theory. At a superficial level, development is seen as going through a sequence of stages: sensorimotor (up until two years), preoperational (2-8 years), concrete (8-12 years) and formal (12 years-adulthood). This aspect of Piaget has been misinterpreted: the stages are not monolithic entities, and transition from one to the other is a gradual process (Groen, 1978). The concept of stages is not empirical but rather rational and has developed from a consideration of what constitutes scientific knowledge.

According to Piaget (1970), knowledge is essentially active. "To know is to assimilate reality into systems of transformations" (p. 15), which gradually become better and better models of reality. Two aspects of knowledge can be distinguished: figurative and operative. The first gives us knowledge of "states" and includes such aspects of thought as perception, and mental imagery, which Piaget considers "interiorized imitation". The operative aspect deals not with "states" but with transformations from one state to another, and is characterized by actions which operate on either other states or operations. In logic or mathematics this aspect of knowledge can be seen without the figurative aspect. But in all other cases it subsumes the figurative though the figurative cannot occur without the operative (Vuyk, 1981).

This latter aspect of thought is what determines knowledge, although both figurative and operative thought processes can be called complementary (Piaget, 1970). Piaget distinguishes empirical abstraction (simple or Aristotelian abstraction) from reflective abstraction (Vuyk, 1981). Empirical abstraction involves abstraction from objects. Children will, by examining objects of varying weights in their hands, determine that usually larger objects weigh more than smaller ones. This they accomplish through empirical abstraction. Logico-mathematical knowledge involves reflective abstraction ("abstraction reflechissante"), where knowledge is derived, not from objects, but from coordinated actions. Reflection refers to the process of reorganizing at the level of thought. There are various ways that actions can be coordinated, and these ways are found at the root of logical and mathematical structures.

Educational Applications

Piagetian theory has been critically examined by many individuals from a variety of disciplines (see Vuyk, 1981). This suggests that certain aspects of his work have influenced more than just the fields of psychology or education. In no other area, however, are Piaget's "sins of omissions" more evident than in the lack of clarity in his own writings on education (Vuyk, 1981). As a result, although Piaget, himself may have asked the right questions (Johnson-Laird, 1977), many of his interpreters have not (Groen, 1978).

Elkind (1979) offers examples of some of the blatant misinterpretations of Piaget in educational practice. He labels them as indicators of "accomodation without assimilation" and feels that they stem from the failure of educators to distinguish between a school curriculum and a developmental curriculum. Many educators have tended to interpret Piaget's tasks on conservation as meaning that they should teach non-conservers "how to conserve" (Kamii & Devries, 1976). This is not what Piaget intended to be done with the various tasks in his experiments. Elkind (1979) feels the real task of educators is not to teach Piaget's "developmental curriculum" but to correlate it with the school curriculum. Put in another way, to turn Piaget into a theorist of a new curriculum is to "stand him on his head" (Papert, 1980b, p. 31). This section will examine some of what Piaget has written on education and the specific application of Piaget in Logo.

Various notions of educational relevance can be extrapolated from Piaget's writings: the role of activity, and the role of the environment. Generally, learning must be an active process and "the basic principle of active methods" is:

> to understand...to discover, or reconstruct by discovery, and such conditions must be complied with if the future individuals are to be formed who are capable of production and creativity and not simply repetition.
>
> (Piaget, 1973, p. 20)

"Activity", in this sense, doesn't only refer to physical manipulation of the environment, although it is part of it. Piaget differentiates between "knowing how to" and "knowing" (Copeland, 1979). Both concepts involve activity but of a different kind. The acquisition of logico-mathematical knowledge is a result of mental activity or "actions" that are interiorized. For the abstraction of the concept of number, an example of logico-mathematical knowledge, children must use reflective abstraction which involves "the creation of mental relationships between/ among objects" (Kamii

& Devries, 1976, p. 6).

Although some subjects, such as spelling, could be taught by traditional methods, Piaget feels that mathematics and science, disciplines developed through a process of discovery and research, must be taught in a similar fashion.

> An elementary truth in physics is verifiable by an experimental process.
>
> (Piaget, 1971, p. 26)

and is based on a rational, inductive and deductive approach. Thus, "learning by doing", as expounded by Dewey (1956), is further elaborated in this context.

The role of the environment is extremely important in the educational context.

> The essential aspect of interaction in Piaget's theory is its dialectical movement: S--O.
>
> (Vuyk, 1981, p. 51)

The subject influences the object (in this case the environment) at the same time as the object influences the subject. Clearly, the structure of the environment plays a key role in the interaction. Although the nature of activity is well elaborated in Piagetian terms, the nature of the environment is not. Thus, activity is signaled as crucial in the form of play; "assimilation in its purest form" (Piaget, 1971, p. 155), while the structure of the environment must complement (accomodate) this activity. The critical issue then becomes: how does one achieve a satisfactory balance between free activity and structure (which in itself implies restriction). This is not clarified by Piaget and is crucial in any direct application of the theory to education.

Three general psychopedagogical principles can be delineated from Piagetian theory (Piaget, 1972/1977). The first is that "real comprehension of a notion or a theory implies the reinvention of this theory by the subject" (p. 731). In other words, once the child is able to repeat certain notions in a particular learning situation this does not necessarily mean that he or she has understood them fully. Understanding, then, implies that the subject can at least partially reinvent the situation. The teacher's role then shifts from one who gives the lesson to one "who organizes situations that will give rise to curiosity or solution-seeking in the child" (Piaget, 1972/1977, p. 731).

Secondly, educators should consider that "the pupil will be far more capable of doing and understanding in actions than of expressing himself verbally" (p. 731). In other words, "awareness" of what is involved in a certain learning situation occurs long after the action. The subject in a learning situation possesses far greater intellectual powers than he actually consciously uses. In this context two factors are important - the teacher's knowledge and the dialogues between the child and the teacher or the child and his peers. Discussions between these groups will in turn encourage verbalization and subsequently "awareness".

The third important consideration of Piagetian theory deals with the teaching of mathematical concepts. "Formalization" of these concepts "should be kept for a later moment as a type of systematization of the notions already acquired" (p. 732). Teachers should be wary of presenting too early notions and operations in a formal structure. "Free" exploration of intuitive ideas should be encouraged before rules and axioms are discussed.

This will occur "in its own time and not because it is forced to by premature constraints" (p. 732). These principles will be further elaborated in the context of Seymour Papert's Logo.

Papert and the Logo Environment

Seymour Papert's theoretical conception of the Logo environment is strongly influenced by Piaget's knowledge-based theory of learning, learning outside the school curriculum.

> I take from Piaget a model of children as builders of their own intellectual structures. Children seem to be innately gifted learners, acquiring long before they go to school a vast quantity of knowledge by a process I call Piagetian learning or learning without being taught.
>
> (Papert, 1980b, p. 7)

In general, Piaget holds the belief that children acquire mathematical or scientific principles only by reconstruction of each new truth in a free problem-solving environment (Piaget, 1973). Only here does the child become the true epistemologist, the builder of knowledge. The roles of the environment and of the teacher are to direct and create situations where the child can work to find solutions. The environment enhances the process of learning which becomes an extremely personal and intrinsic part of the child.

Since the structure of mathematical knowledge is claimed to be the main source of stage theory, a separation between the learning process and what is being learned is not useful. In other words, in order to understand how a child develops the notion of number, one possible way might be to examine the structure of number.

This fundamental principle is at the root of the Logo environment.

When children work on Turtle geometry, they learn how to program a computer using a specific language. Within this framework, they are learning much more than the basics of programming: they are learning about the functioning of procedural knowledge. In understanding why a program doesn't work, they are generating transformations belonging to that structure. As they proceed to isolate and correct the errors (bugs) in their plan, they are debugging their own thought processes. This in turn is postulated to lead to the development of a general heuristic for organizing their thoughts. These general heuristics are called "powerful ideas" or "mathetic" principles by Papert. Step-by-step, they thus acquire through accommodation a progressive ability to assimilate reality (Groen, 1978). They become true hypothesis testers, builders of knowledge, intrinsic "science makers".

Since these procedural principles can be reduced to a concrete model, the child is able to grasp them more readily and subsequently to apply them to other areas of knowledge and problem-solving. Here, Papert extends Piaget and ascribes to culture (as part of the environment) greater powers than previously described. The reason why formal thinking does not develop until a child is almost twice the age at which he or she achieves concrete operations may not necessarily be due to the complex and abstract aspects of formal operations.

> Piaget writes about the order in which a child develops different intellectual abilities. I give more weight than he does to the influence of the materials a particular culture provides in determining that order.
>
> (Papert, 1980b, p. 20)

Two kinds of thinking are associated with formal operations: combinatorial thinking, where one has to reason in terms of the set of all possible states of a system (the colored beads problem described below) and self-referential thinking about thinking itself. In a typical experiment in combinatorial thinking, one which most children don't master until age 11 or 12, a child is asked to form all the possible combinations of various colored beads (Baldwin, 1980). Papert compares this task to constructing and executing a program.

> ...the most salient ingredients of the task are related to the idea of procedure --systematicity and debugging. A successful solution consists of following some such procedure as:
>
> 1. Separate the beads into colors.
> 2. Choose a color A as color 1.
> 3. Form all the pairs that can be formed with color 1.
> 4. Choose color 2.
> 5. Form all the pairs that can be formed with color 2.
> 6. Do this for each color.
> 7. Go back and remove the duplicates.
>
> (Papert, 1980b, p. 175)

Papert conjectures that children acquire this ability so late because contemporary Western culture provides little opportunity for procedural thinking, a lot less than it provides for quantity of number, for example.

> Of course our culture provides everyone with plenty of occasions to practice particular systematic procedures. Its poverty lies in materials for thinking about and talking about procedures.
>
> (Papert, 1980b, p. 223)

In this way, Papert feels the computer will surely alter and may ultimately enhance the way people learn and think. If given the opportunity early enough to experiment with procedure-rich environments, computer-based microworlds, children may learn to be systematic even as, or before, they learn to be quantitative.

Although Papert's Logo is based on several salient aspects of Piagetian theory, it also deals with many points of "reinterpretation". Papert is concerned directly with education; Piaget only indirectly. Papert is interested in intellectual structures that could develop, rather than in a description of those that already exist, and in developing environments that can support those new structures.

Computer-based Microworlds

The underlying premise of the proponents of the "tutee" mode of computer use in education is: the child learns more about the process of learning by teaching the computer than he or she does from being tutored by a program written by someone else. As Dwyer (1980 p. 92) explains:

> Student-controlled computing means that the student uses technology to develop and test his own models; that the student learns to deal with failures; that it is in his power to debug the procedures that caused these failures.

It is not incidental that Papert's vision of education focuses on computers and their potential. He and other cognitive scientists hold the belief that computer science has the potential to not only enhance the learning process but also to ultimately change the way people learn and think (Papert, 1980b). Lawler (1982, p. 138) further elaborates:

> Designing computer applications for education might be called cognitive engineering, for its objective is to shape children's minds.

This section will examine in more detail the environmental "engineering" used to provide that elusive balance between freedom and structure in a specified learning situation. Two paradigms of instruction, the standard form and the mode exemplified by the "microworld", will be summarized as they relate to computation and mathematics.

The standard form of instruction in the field of mathematics has been under a lot of criticism in recent years (Wavrik, 1980), although new directions in secondary school curricula are being reported (Doctorow, 1982). In general terms, mathematics, as it is taught in many schools today, is viewed as a static discipline, and one which can be broken down into a succession of small units. These units are presented to the student in an order dictated primarily by presumed difficulty and "pseudo"-logical sequence. Instruction includes:

1. the presentation of a specified skill, method or fact (content)
2. practice or drill
3. testing to determine if mastery has occurred.

(Wavrik, 1980).

To summarize, the standard framework reduces mathematics to a subject in which only fairly low-level cognitive skills are required. Ironically, this structure leaves virtually unlimited room for growth of higher mental processes (i.e., problem-solving skills), but doesn't have automatic mechanisms to provide encouragement for that growth (Burrill, 1982).

Papert is generally at odds with the fundamental principle of "packaging" knowledge in this fashion, but especially in the realm of mathematics. To him, mathematics is a dynamic discipline, a source of personal power in which context the learner assumes responsibility for the learning; he or she participates in the recreation of mathematical knowledge by actively exploring and extending its structure or the corresponding structures in his mind. The pedagogical consequences of this framework are varied. Logo exemplifies them: the learner is in control, the teacher assumes a secondary role. Mathematics is learned in cycles consisting of:

a) experience

b) abstraction of concepts derived from this experience

c) symbolic representation of concepts

d) (on higher levels) construction of theories to organize the concepts.

Computer technology is the vehicle through which mathematical knowledge can be acquired in this fashion.

Thus, the concept of the computer-based microworld enters education:

> A microworld is a subset of reality or a constructed reality whose structure matches that of a given cognitive mechanism so as to provide an environment where the latter can operate effectively.
>
> (Papert 1980b, p. 204)

Microworlds are not new either in education or in society at large. Examples of non-computer microworlds in education include the use of Cuisinaire rods to teach math concepts or the dynamic microworld of the Rubik's cube. Microworlds are in essence "task domains" or "problem spaces" designed for streamlined experience (Lawler, 1982, p. 139). What is new, however, is the use of the technology to structure microworlds of a specific kind "to stress the learner-directed process" (Papert, 1980a, p. 204) and to present the learner with "neat phenomena" (Lawler, 1982, p. 138) worthy of exploration.

Lawler's "neat phenomena" are "phenomena that are inherently interesting to observe and interact with" (Lawler, 1982, p. 139), and are intrinsically motivating to the learner. It is here that we find the "power of the idea".

There are several principles needed in order to formulate a well-designed microworld. First, it must be constructed around a "powerful idea": in the Piagetian spirit, this idea must be continuous with the child's own personal knowledge; it must empower the learner to act (physically and mentally) and this action must make sense in terms of the larger social context so that transference of learning from one situation to another can take place naturally and "painlessly".

> The idea is a powerful one because it is almost universally useful; it is crucial to the process of scientific investigation.
>
> (Lawler, 1982, p. 142)

An excellent case study is offered by Lawler (1980) of just such a powerful idea, the stepping of a variable, "changing one thing, a little at a time" (p. 18) and how this principle helped provide guidance to the subject in thinking about problems encountered outside the Logo microworld where it was originally explored and refined. As explained by Fischer (1981), the designers of microworlds must meet the ultimate challenge of providing simplicity as well as reality. A good example of a computer-based microworld is Papert's Logo environment.

The Logo Environment

Papert's computer-based Logo environment has a number of components of which the least important is the computer (Groen, 1978). It doesn't matter what computer is used so long as it is capable of creating and maintaining the environment. Most of the recent converts to the Logo philosophy may consider Logo to be a language available on microcomputers only. This, however, is far from the truth. Logo was originally developed at the Massachussets Institute of Technology as early as 1967 as a framework for teaching mathematics (Feurzeig, Papert, Bloom, Grant & Solomon, 1969). At that time, the language was available on large-scale computers only and because it required a large memory capacity remained accessible only on time-sharing systems and later on mini-computers. The advent of the microprocessor and the subsequent lowering cost of memory changed the situation quite rapidly. The present influx of Logo-worlds in microcomputers has raised many serious questions as to what language should children be introduced first in learning how to program a computer (Howe, 1981).

The Language

The implementation of Logo in 1967 at MIT had three components (Chait, 1978): the programming language called Logo, a teletypewriter, linked to a time-sharing computer, and the "Turtle". The language was by far the most important aspect of the environment then as it is today. The name, derived from the Greek word "word" or "thought", was coined by one of the originators of the language, Wallace Feurzeig.

The child of artificial intelligence research, Logo is an interpretive language using local variables to permit recursion, that is, a procedure can be a subprocedure of itself (Harvey, 1982). At the same time, one is able to define new commands and functions which can then be used like "primitive" ones (Papert, 1980b). Primitives are the built-in procedures, the indivisible chunks that Logo provides initially. In a sense, the structure of Logo is extensible in contrast to other common languages such as BASIC. This encourages programmers to build or chunk their own procedures into smaller parts.

In Logo, all commands can be executed directly without being embedded in a program (Harvey, 1982). This interactive aspect of the language allows for more errors to be corrected before they are written in a procedure. The language also gives immediate and helpful error messages. For example, if a child incorrectly types RIHGT instead of RIGHT, the computer responds with "YOU HAVEN'T TOLD ME HOW TO RIHGT". By retyping the part which is giving the computer difficulty, the inexperienced programmer can correct careless errors quickly and efficiently. Full list structures are also available for advanced programming and Logo procedures can construct, modify, and run other Logo procedures.

It is this type of hierarchical organization which mirrors Papert's conception of a discovery-rich procedural environment. Through the process of learning how to program in Logo, a child develops the ability to solve many problems related to heuristics - the study of methods and rules of discovery and invention (Polya, 1957). Polya describes four stages of solving problems: 1) understanding the problem, 2) devising a plan, 3) carrying out the plan, 4) looking back on the solution to see if it worked. In writing a program for the Turtle, a child goes through a similar process.

> Some of the strategies used in Turtle geometry are special cases of Polya's suggestion ...Such as: Can this problem be subdivided into simpler problems? Can this problem be related to a problem I already know how to solve?
>
> (Papert, 1980b, p. 64)

The idea of subproblem and subgoal are important notions in the world of artificial intelligence. "Divide and conquer is the first principle of scientific attack" (Winston, 1977, p. 251). All these aspects of the language make it a very useful and non-threatening language for programmers of all ages.

The Turtle

The third component of the Logo environment in 1967 was the "Turtle", named in honour of Grey Walter's mechanical tortoise (Papert & Solomon, 1972). Walter, one of the pioneers in cybernetics equipped his tortoise with certain light-detecting "capabilities": if light shone on photocells located in the front end, its motor would go into reverse until it was away from the light. Eventually, these "toys" became more sophisticated and could even recharge themselves (Evans, 1980).

Papert's Turtle was a second cousin to Walter's tortoises. Designed as a cybernetic toy, it was able to respond to certain commands by changing its orientation and its position. Later, a graphics screen was developed and the Turtle was represented as a trangular cursor in the center of the screen. Color graphics are now available on recent models, as well as "Sprites", color units extremely useful in the programming of motion. Although many modifications have been made since its inception, the two essential aspects of the Turtle's state have remained the same: its position and its orientation. Specific aspects of Turtle geometry as they relate to the present study will be further explained in the next chapter.

Research Studies

The previous sections have provided the theoretical connection between Piaget's genetic epistemology and educational practice and the field of artificial intelligence with respect to the Logo environment. This section will summarize the pertinent research undertaken in the last fifteen years to study the process of learning Logo and its effects on children or adults. Subjects have included students from preschool to graduate school, students with regular or outstanding ability in science or mathematics, students with special needs and severe handicaps; students with physical, mental and emotional difficulties. The Logo microworlds used have varied from the non-graphic varieties to color graphics and Sprites in Turtle geometry to the teaching of music (Bamberger, 1972), physics (diSessa & White, 1982), biology (Abelson & Goldenberg, 1977), and mathematics (Howe, 1981). The majority of studies have focused on elementary aged children in the learning of Turtle geometry.

The existing research has used one of two methodological approaches: case study or classical experimental, but the majority of studies have used the anecdotal case study in order to describe the process of learning Logo. One study (Chait, 1978) has provided a method of macro-analysis of the Logo experience which could be used across subjects. The studies will be grouped according to the methodological approach used, that is classical experimental, case study and protocol analysis. Studies using the case study approach will be grouped according to the populations examined.

Classical experimental

Several studies have used a classical experimental methodology in examining various aspects of the Logo experience: Milner (1973) conducted an extensive study using non-graphic aspects of Logo with elementary aged children. Logo graphics were studied in Syracuse University (Statz, Folk & Seidman, 1973) with an average population of elementary students. Gregg (1978) used aspects of Turtle geometry with preschoolers. Grade six students with low-math grades were taught Logo at Edinburgh (Howe & O'Shea, 1978). More recently, problem-solving techniques and social interaction were examined in elementary aged students at Bank Street College (Jewson & Pea, 1982). Each study has emphasized certain objectives at the expense of others. Teaching methodology has been the underlying theme in three of these studies while Gregg (1978) examined the development of a young child's mapping of symbols onto events.

Milner (1973) conducted a study with eighteen fifth graders in order to investigate how to teach programming, specifically whether or not the concept of variable could be acquired through programming in Logo. The acquisition of general problem-solving skills was also studied.

The study was designed in three phases. Phase I trained all students in the use of the Logo language which consisted of computer-assisted instruction (CAI) lessons. The rationale for teaching the primitives of the language independent of instructional methods, examined in the second phase, was to avoid confounding learning elements of the Logo language with the actual concept which was studied in detail. Therefore, Phase I teaching methodology was fairly consistent across all subjects: the CAI lessons were made up of brief tutorials followed by a period of independent work.

Phase II investigated the effects of method of teaching in the acquisition of the concept of variable. The students were first divided into two levels of ability, high and low, according to scores on the Stanford Achievement Test. They were then randomly assigned to one of three instructional methods, including an algorithm-method, an incomplete computer program method and a no-information (control) method. The students were then presented with the various programming problems: method of presentation was dependent on which method the student was assigned. Little help was offered by the experimenter, except to suggest various problem-solving strategies advocated by Polya.

In Phase III, the criterion phase, all students were given similar problems except that no information was offered. A pretest-posttest design was used to test the acquisition of the concept in question. Results demonstrated that the concept of variable was acquired; however, Phase II (method of instruction) had no significant effect on its acquisition.

The no-information method is likened by the experimenter to the "discovery" approach in teaching. Anecdotal evidence was presented on the affective components of the learning experience: motivation and enthusiasm continued throughout the duration of the experiment. The children's problem-solving techniques were also enhanced.

This project addresses several interesting aspects of teaching methodology and computer programming: the issue of expository versus discovery approaches to teaching and what degree of structure is optimal in this learning environment. It is interesting that in non-graphic Logo a structured method of teaching (i.e. a behavioral approach) is as effective as less structured forms of teaching. This conclusion, however, should not generalize across other Logo microworlds and for that matter across other tasks.

A similar study of longer duration conducted in Syracuse, examined similar issues in a behavioral approach (Statz, Folk & Seidman, 1973). Graphic Logo was taught in various learning environments to elementary school children (grades 4, 5, 6) for a period of two years. Issues examined included optimal teaching approaches to Logo, cognitive development, conditional reasoning and problem-solving. The children's performance was studied in terms of tests given throughout and at the end of the teaching period, as well as dribble file records. These ensure that a complete account is kept of all students' interactions with the computer (Feurzeig & Lukas, 1975). In contrast to Milner, this study included a teacher training component.

Although the various Logo curricula were geared to each individual child, several distinct teaching approaches were examined during the period from September 1972 through April, 1973. The approaches varied from a structured to an unstructured environment. "Structured" was characterized by a "concept-centered" approach while "unstructured" was termed "problem-centered" approach. The method which allowed for a combination of both approaches, more structure in the initial teaching sessions until certain basic concepts were developed and then less structure which afforded the students freedom in developing their own projects, was felt to facilitate good programming and good problem-solving in Logo.

> Children need a simple introduction to basic concepts, so that they understand what is essential to writing a program. Their experience can then bring out the intricacies avoided during the initial groundwork phase.
>
> (Statz, Folk & Seidman, 1973, p. 25)

This teaching approach was adopted in the current study and can be likened to Papert and Solomon's philosophy of how Logo should be taught.

Children were taught in groups ranging in number from 2 to 5, so that peer interaction and independence could be enhanced. Among the different settings that were tried was the classroom. Three variables were found to be important for success in this environment: flexible schedules of students, the interest and background of the classroom teacher and a self-contained class.

The current study adopted the teaching of Logo in pairs of subjects in order to foster peer interaction, verbalization activity and self-sufficiency. Although the computer was brought into the school setting , a separate computer room was used to teach Logo. The teaching schedule was developed to fit the children's classroom schedule. The classroom teacher's interest and cooperation was sought and encouraged at all times.

Conditional reasoning, characterized by the student's ability to use logic was examined in two experimental and two control groups but no results were given at the time of this report.

The issue of problem-solving was given a more detailed theoretical and practical focus. A battery of tests were administered before and after the Logo experience to an experimental group and to a control group which did not receive any Logo instruction. These tests consisted of four problem-solving tasks: a game of twenty questions, the Tower of Hanoi problem, a permutation puzzle, a set of word puzzles and an attitude questionnaire about problem-solving. The results showed significant effects on two of the tasks, measuring classification skills and isolation and control of variables. The attitude of children toward problem-solving, as measured by the instrument developed by the research team, did not prove significant. The researchers concluded that the reason for the lack of significant results could be attributed to the inappropriateness of the tasks for illustrating the specific problem-solving skills learned through Logo programming.

A model of problem-solving was developed, similar to Polya's and observational comparisons showed that this model may serve as a good description of the process of learning Logo.

The Syracuse project reported on many issues of interest to educators in the teaching of Logo. Psychologically, many interesting trends were observed, including the fact that developmental level may have a significant influence on the ability to learn Logo. However, no clear-cut conclusions were reached and the report has a unfinished quality about it (Chait, 1978). What it seems to lack is a description of the process of learning Logo.

The Department of Artificial Intelligence in Edinburgh has been conducting research in Logo since 1974. Originally, the focus was in using the computer as an artificial teacher to help improve the mathematical background of teachers and their teaching techniques (du Boulay, 1977). More recently, two projects have focused on whether children's math skills are affected by learning to program in Logo.

The first project, 1976-1978, a two year preliminary study used 11-13 year old subjects from a private school near the university (Howe, O'Shea & Plane, 1980). They were selected on the basis of their school marks in mathematics: they were students of near average ability and belonged to the lowest-level math group.

For two years the students worked at the university's Logo lab in groups of four for about one to one and a half hours a week from September to June. During the first year in the project, they learned Logo programming during normal school hours. The materials included a series of 33 worksheets, a Logo programming primer, designed for individual self-paced study of several key "computational metaphors" in Logo, including problem-solving tactics and debugging skills (Howe & O'Shea, 1978). The teaching during the first year was limited to building and interpreting Logo programs. The evaluation of the project during this time is termed by Howe as "monitoring" the progress of the students in a formative fashion and was in the form of anecdotal reports (Howe & O'Shea, 1978).

During the second year of the project several strategies were changed. First, the students came to the lab after school hours and they worked on maths worksheets designed to help them apply the programming notions of the previous year to the subject of mathematics. Activities were drawn from math topics in arithmetic, geometry and algebra. During this time, the evaluation process was more standardized and detailed.

The project was highly structured in several respects. The research was based on the classical experimental approach using two groups, experimental and control; standardized tests in mathematics as well as questionnaires designed to measure attitude towards mathematics were administered to both groups before and after the project. Great care was taken to monitor the results. The published report on the two year project showed a slight improvement of the experimental group on a basic maths test.

The reverse was true on a "maths attainment" test. Howe felt that the tests were not very sensitive to the topics that were covered in Logo. The most interesting findings dealt with the anecdotal evidence presented on the changes in motivation, self-confidence and verbalization on the part of the students (Howe & O'Shea, 1978). Teachers also found that the student's were able to "argue sensibly about mathematical issues" and to "explain their mathematical difficulties clearly" (Howe, O'Shea & Plane, 1980, p. 16.)

The second project was recently completed (Howe, Ross, Johnson, Plane, & Inglis, 1982). The setting was a state secondary high school in Edinburgh where six Terak microcomputers were installed in a mathematics classroom in August 1980. Ninety first year students (aged 12-13) were introduced to programming in Logo, the methods closely related to the previous study (Howe, 1981). This project was designed as a first phase of a longitudinal evaluation study. The principal reason for the new investigation was to strengthen the previous experimental design by offsetting several uncontrolled variables, such as the Hawthorne effect of "coming to the university" which may have influenced the previous results.

This study was also divided into two parts. The first part was concerned with providing the necessary materials to support the programming activities in the classroom. It was carried out in 1978-80 and included resolving issues of hardware implementation, development of curriculum materials and the training of the teachers. The teaching approach used in the previous study was modified somewhat. Logo programming skills (Logo worksheets) were taught in parallel with the teaching of related math topics instead of in two distinct phases as previously described.

The second part was carried out during the school year 1980-81 and focused on the evaluation of the effects of programming on the performance of classroom mathematics. The formal and informal measures as well as the design issues used were similar to those employed in the previous study. Dribble file records were also kept of all interactions with the computer. No group results proved significant. However, when the groups were broken down by sex the performance of the experimental groups' females, as measured by a standardized Mathematics Attainment Test, improved significantly over the control group. Attitude questionnaires about mathematics and Logo remained stable throughout the study. The examination of the records of pupils' transactions with the computer showed that progress is governed by ability and amount of exposure to Logo-related work.

Although great care was exercised in monitoring the results of both studies, no conclusive evidence was obtained on whether Logo programming did promote the learning of mathematics. Clearly, Howe feels conclusive results can only become apparent in a classroom situation over a long period of time.

The teaching strategy used by Edinburgh is more structured than the one advocated by Papert. Howe describes it as a strategy "related more directly to the school curriculum" (Howe, 1981, p. 111). He argues that structure and systematization are not in conflict with Piaget's notions of an activity-based approach: "there is need for rational, deductive activity to make sense" (Howe, 1981, p. 112). The argument, however, may be contradictory with the learner as an experimenter in a free problem-solving environment. The key issue here again is what is the right amount of structure for an optimal learning environment.

Within this teaching framework, Howe (1978) describes three stages in learning to program in Logo. The first stage is called "product-oriented" and is characterized by the students tendency to copy and execute procedures without making any attempt to understand them. The second is called "style-conscious" programming and relates to the students' concern to work in what they perceive as correct programming. In this stage, they may use sophisticated techniques but may not be in complete understanding. The third is called creative problem-solving where the student is creating original programs and is in complete control of his environment. These three stages are postulated to be present in all programming. However, they are conspicuously related to the teaching strategy in the Logo worksheets developed by Howe. These start with an explanation of the worksheet's purpose (Howe , O'Shea, & Plane, 1980). A specimen Logo procedure is then presented which the student is supposed to type in and execute. This could be termed the product-oriented stage. The worksheet then is followed by exercises which involve modifying the procedure in minor ways (the style-conscious stage) and finally adapting it to some novel problem area (versatile stage). What is not clear is whether these worksheets were developed after these stages were observed or before? Is Howe seeing developmental stages or are his observations related to the method of teaching he has adopted?

Gregg (1978) conducted an experimental study with preschoolers on the processes and stages of a child's mapping of symbols onto events. The floor Turtle and a button box which modified the keyboard (see Perlman, 1974) were used to teach the children spatial concepts related to Turtle geometry. The button box presented to the children three symbols (buttons that could be pressed): one button made the robot Turtle turn 90 degrees left; another made it turn 90 degrees right and the third made the Turtle go forward.

Eighteen children, 4-5 years old, were pretested on their knowledge of certain spatial concepts, colors and shapes; they were then brought to the Turtle lab where they were given the specific problem sets after a period of free play and demonstration. Statistical evidence over the period of a year indicated that children varied in their ability to solve certain problems. Concepts such as left-right button differentiation and their identification with the Turtle's starting orientation were difficult to understand. At the same time, the children were observed to go through identifiable stages in learning the Turtle tasks. It is interesting to note that in such a structured learning and teaching environment, many of the tasks were boring and unmotivating to the young children. They were also unable to explain to the experimenter why certain buttons worked and others did not.

Several research projects are now underway at the Children's School of the Bank Street College of Education in New York City (Jewson & Pea, 1982). Supported by the Spencer Foundation, computers have been placed in two classrooms of elementary aged children, in a private school in Manhattan. The children, ages 8 to 12, are learning to program in the Logo language alone or in groups "as active designers and programmers of their own projects" (Jewson & Pea, 1982, p. 332).

The research is focusing on problem-solving techniques and social interaction two areas highlighted by many other projects but not carefully studied so far. Most reports are being written at the time of this project and no results are available. In general, problem-solving techniques examined are planning strategies in programming and in everyday usage (Bank Street College, 1981-82). The goal of the research was also to construct a planning task in order to explore the issues of transfer. In-depth structured interviews were conducted with 26 children using a classical experimental design, prior to the experimental group's programming experience. Preliminary data on these interviews suggest that children may think globally but have little understanding of what activities and processes are involved in planning.

A planning task was developed which dealt with the children's everyday experiences and children from computer and non-computer groups were asked to carry it out early and later in the school year. The analysis will examine the product as well as the process involved in this task.

The second area of investigation involved the examination of the social effects of microcomputers in the classrooms where children were learning to program in Logo (Hawkins, Sheingold, Gearhart & Berger, 1982). Specifically, two studies using a before and after design investigated the type and amount of social interaction during regular classroom activities and during computer-based work, and children's perceptions of their peers as resources for help.

The first study recorded observations of task-related interactions among children who worked on both computer and non-computer tasks. These observations were coded as to type of interaction and the relationship between them in terms of peer teaching and collaboration. Great care was taken to train the observers and standardize the procedures used. Preliminary results of comparisons between non-teacher directed activities (work periods) and computer-based work were very interesting. There was more task-related interaction during computer activities than during regular work periods. There were also more episodes of collaboration (verbal and action) between students during computer periods. No differences were noted in either peer teaching or questions to teachers. Several factors could contribute to these results. First, the kind of work done on a computer may be more inducive to public viewing and discussion than most classroom work. The unique explicitness of some of the Logo instructions and outcomes may also facilitate joint involvement. It is also possible that the novelty of the computers and the degree of expertise of the students played some role in the increased interaction.

The second study examined the children's perceptions of who among their peers would be a resource when they needed help when engaged in a variety of classroom tasks or in computer-related work. A 10-item sociometric questionnaire was administered before the computer work started. Results were compared with a similar questionnaire given after the computer work was finished. Little consensus was found among children on their choices of helpers during non-computer activities. The same pattern emerged from pre to posttest. For computer activities, however, several children emerged as "experts" in the eyes of their peers. These children would be selected when help was needed but not necessarily for partners in the computer activities. There seemed to be a trend for the children to choose partners of the same level of expertise as themselves. In further examining the data, prior experience with computers did not seem to be a factor in the choice of a computer helper. Sex differences were also observed. Girls were only chosen as experts by other girls, while the majority of girls selected boys as experts. Teachers, on the other hand, identified several girls as knowledgeable about computers, but in general, collaborated with the children's perceptions of expertise.

The work at Bank Street College reflects a new and fruitful direction in research in computers and education in general, and Logo in particular. Indirectly, the implications of this type of research are exciting and varied. It may become clearer in the future the kinds of changes computers will impart on society and education. They may radically change certain aspects of the general organization of the classroom as well as the teaching and learning environment. However, these changes may not only become apparent in the cognitive domains but also in the social life of classrooms.

Although much has been written about the social implications of computer technology on our society (Zimbardo, 1980), and its effects on alienation, the trends described in these studies may show a different and opposite direction. The computer may instead of alienating human communication enhance and encourage it particularly in the classrooms of tomorrow. Only more extensive and longterm research in this area will answer some of these important questions.

Anecdotal Case Studies

Although Logo is considered by many to be the computer language for children, non-graphic and graphic aspects of the language have been used to teach programming to different populations of adults. This section will examine some of the "upper" limits of the Logo environment in respect to adult populations and some of the "lower" limits in respect to special populations of children.

Logo and Adult Populations: Brown and Rubinstein (1974) conducted a study with undergraduate humanities students with no previous programming experience and little background in higher level mathematics. Various non-graphic aspects of the language were emphasized including the concepts of recursion and programming styles such as top down and bottom up. Students were taught in a traditional (lecture) format but assignments allowed the students to devote some time to extensive projects. Logo was seen to be an ideal programming language to teach abstract reasoning to this type of population.

Logo has also been used to teach undergraduate students of very low mathematical ability some key mathematical ideas (Lukas & Feurzeig, 1972). In contrast to the previous study, the students were allowed to develop their own projects and the teachers' acted as consultants. Emphasis was placed on computational ideas, bugs and the use of procedures and subprocedures. Anecdotal evidence indicated that the students were able to use these ideas.

Turtle geometry was introduced to teachers at MIT (Austin, 1976) and at Edinburgh (du Boulay, 1977). The MIT experiment included other components of Logo such as physics and music. Initially, students were introduced to various primitives of the language and were then given the freedom to develop projects of their own choosing. The Edinburgh project used structured exercises (the Logo primer) to develop the various concepts but also allowed the students to work independently. Both studies presented anecdotal evidence on the type of work done by the teachers. In general, these studies suggest that adults with little mathematical background can be successful programmers in Turtle geometry.

To show the upper limits of the Logo language, a curriculum has been developed geared towards university students with a strong background in mathematics (see Abelson & diSessa, 1981). The formal aspects of Turtle geometry in this context suggest that adults of all ages and backgrounds can gain access to rich mathematical ideas from investigating this computer-based approach to geometry and physics.

"Special" Populations of Children: Special features of the Logo environment make it a language ideal for teaching programming skills to very young children, as well as children with physical, emotional and learning handicaps.

Since very young children might have trouble with typewriter keys and with the spelling of procedure names, special modifications of the keyboard and prototypic microworlds have been developed to allow children as young as 3 and 4 years old to interact with the computer.

Perlman (1974, 1976) developed a special hardware extension of the keyboard, called Tortis and used it to teach Turtle geometry with the robot Turtle to 4 year olds. Tortis is comprised of two parts, a series of button boxes with large buttons (each a command) labeled with pictures instead of words and a slot machine with several long plexiglass rows (each representing a procedure) with slots in the top for the user to place cards (commands). The system was designed for simplicity and modularity; salient features of the Turtle environment are emphasized (action box) for the initial interactions. However, other boxes can be added when the child is ready for the transition into more complex programming (Perlman, 1976). The key feature is that the children are in control of their learning and the system can grow with them as their knowledge increases. Anecdotal evidence suggests that children vary in their ability to map concepts onto the names on the button box. Right and left turn keys, in general, were harder for the children to understand and playing Turtle proved to be a good strategy in this regard. This evidence is also supported by Gregg (1978) using a completely different approach. Gregg, however, found that the children were easily bored in the early phases of the various structured tasks they were given while Perlman did not observe this in the free problem-solving environment. This may support the contention that the student should be in control of the learning.

Special "software" modifications have also been used in various prototypic situations. Lawler (1982) wrote a number of programs which allowed his own young daughter to learn how to read with minimal direct instruction. Called the beach microworld, the program utilized Sprites to create a special environment tailor-made to teach his daughter words and their meaning.

Based on the same principle, other software programs, developed at the Lamplighter School in Dallas (Gorman, 1981b) and part of several Logo versions on the Apple microcomputer allow young children to type one letter on the ordinary keyboard to command the Turtle.

All these environments allow young children to be in complete control of the learning that takes place. This seems to be the key to their success or failure, although no studies have verified these informal observations. The children seem to be learning "powerful ideas" but most of all they are learning how much fun learning independently can be.

The same principle of autonomy in a meaningful context (Goldenberg, 1979) apply to Logo microworlds developed for children with physical, emotional and learning handicaps. Researchers have been working with children who are physically handicapped at the Cotting School in Boston since 1978 (Weir, 1982). Anecdotal evidence is presented on students with cerebral palsy, unable previously to communicate effectively, who finally begin to realize some of their learning potential (Watt, 1982), with dyslexic children at the Carroll School in Lincoln Mass. (Weir, 1982), with autistic and other emotionally disturbed children (Weir, 1982), with profoundly deaf adolescents at the Lexington School for the Deaf (Bank Street, 1980-81). These studies are ongoing and no reports have been published to date.

Weir and Emmanuel (1976) report on using Logo to act as a catalyst for verbal communication with an autistic child in Edinburgh. A simplified version of Logo, a button box similar to Perlman's was used throughout the study as well as the floor Turtle. Data was collected in terms of dribble file records, direct observations and videotapes. The analysis of the results was more on an intuitive level but nevertheless proved to be dramatic. The child was able to identify strongly with the robot Turtle, and for the first time perform self-initiated and self-driven activities. These in turn helped him understand the relationship between action taken and response performed, the relationship between cause and effect. Eventually, he was able to communicate with the experimenter and other people for the first time.

Tessier (1978) also observed the therapeutic value of the Logo environment (the screen Turtle) in a preliminary study with an emotionally disturbed child in a residential school. Some positive effects were also seen on the child's school work during the study. Although the reports in this area are all basically descriptions of single cases, the potential of computer technology in the lives of "special" children seems unparalleled. Logo has shown that it is not only important in building basic intellectual skills but also in developing a stronger sense of self-confidence, self-worth and control over their environment.

Average Populations of Children: MIT has been the centre of Logo research since 1970. The influence of Papert has been outlined in previous sections; he has been a strong guiding force behind theoretical and practical research directions at MIT and other research centers. He developed a specific approach to teach Logo, elaborated in <u>Mindstorms</u> (1980) and first implemented by Solomon (1976). Much anecdotal evidence has collected over the years by Papert (1971, 1972, 1976) and his colleagues (Solomon, 1976; Papert & Solomon, 1972; Abelson, Bamberger, Goldstein, & Papert 1976) on the process of learning Logo, specifically how children acquire advanced mathematical, programming and problem-solving skills. Two major projects of recent vintage will be outlined in this section.

The first major project is summarized in two volumes and encompasses the work undertaken by Papert and his colleagues during the school year 1977-78 (Part II by Papert, Watt, diSessa & Weir, 1979, Part III by Watt, 1979). Project summary and data analysis is presented in Part II while Part III presents the detailed documentation of 16 students' learning experience in a case study format.

The setting was Lincoln School, a public elementary school in Brookline, Massachussetts, where a fully equipped computer lab was established. Sixteen students from the grade six classes were selected for the study based on teacher ratings and national achievement scores in order to ensure a variety of students including "average" and "exceptional" students at both ends of the achievement norms. The sample included six students of average academic achievement and six from the "non-average" groups. They had approximately 20 to 40 hours of hands-on experience programming four computers equipped with Logo and Turtle graphics.

The teaching approach followed Papert's adaptation of Piaget's free problem-solving environment in that children developed their own projects and were introduced to programming concepts and new material as the need arose. The teacher was experienced in teaching with this approach and no attempt was made to expose all students to a standard curriculum. Data collection included pre and post interviews, observations by the Logo teacher and outside observers, and complete records of the child's interaction with the computer in terms of dribble file records.

The students were taught in classes of four, so that there was always a one-to-one correspondance between them and the computers. This ratio was considered important by the research team in the results obtained, although the teacher/student ratio of 4:1 would not be necessary under normal operating conditions. Although each student followed a different path in learning Logo five general teaching objectives were outlined, a post-hoc classification based on observations of the children's work.

Final analysis of the data is presented in terms of what the students learned, what learning styles they used and what choices they made. These areas were discussed thoroughly in order to provide aid to educators in how to teach as well as what is learned in the Logo environment.

Most subjects (all but two) were able to achieve the objectives, as they were set up by the teacher. These two students had the lowest scores on the national achievement tests and although they underwent "significant observed learning" didn't achieve the criteria of core programming skills. Comparisons between the students' achievement in Logo and school achievement scores showed a close agreement between high scores and mastery of core as well as higher level concepts in programming.

Anecdotal evidence was presented on the relationship between programming and learning mathematics, as it related to the type and style of intellectual skill required. Logo tends to enhance individual differences in programming and in cognitive styles, a conclusion supported by several studies already discussed. Since the analysis was based on the process of learning rather than the product various identifiable styles were described.

Top down programmers, or "planners", start with a clear model of the end result at the outset and fill in the details of the program secondarily. Bottom-up or "modular" programmers build up the program from a set of modular subprocedures. Solomon (1982) elaborates further: some students followed an open-ended exploration of subprocedures to arrive at a product, "macro-explorers". Timid learners, on the other hand, "micro-explorers", tended to use the same commands and similar inputs repeatedly. Students displayed various degrees of these styles and many used combinations of them in various projects.

Evaluation of this study was conducted in an informal manner and although some issues of transfer into other curriculum areas were raised the evidence was inconclusive. Pre and post tests of heterogeneous material, developed by Hein and Dunning, were given to the subjects but the results were again inconclusive.

Although the Brookline reports can form the basis for an introductory Logo curriculum and are a rich source of Logo ideas for educators, the methodological issue has not been dealt with sufficiently. The report failed in obtaining "objective data" about learning gains and standardized tests were rejected as irrelevant to the ability to use Turtle geometry (Watt, 1982). The problem of developing standardized tests to reflect problem-solving skills and procedural thinking is still unresolved. Another limitation (Watt, 1982) of the project was that it required an extremely sensitive and experienced teacher. The teacher's role is depicted as unrealistic in that it demanded that he devote a great deal of time to the needs of each student. Recent research (see Bank Street section) has minimized this role and emphasized the role of peers in the learning process. Other Papert critics (see Rousseau

& Smith, 1981) feel that he does not use enough parallel research as support for his contentions. Whatever the limitations of this project may be, it provided an extensive analysis of what the students learned in computer programming and in Turtle geometry.

The second Brookline Logo project is now under way but no documentation is presently available. It follows closely from the first in that its goals are to develop a curriculum supporting classroom use of Logo (Watt, 1982). The setting then is shifted to the classrooms of grades four through eight where two computers were placed in a rotating basis. Each classroom had exclusive use of the computer for 8 to 12 weeks. The students were allowed to work individually or in pairs as the regular classroom work was taking place, although class lessons about once a week brought the students together to exchange ideas and introduce new concepts.

Teachers received limited formal training at the outset of the project. The curriculum materials developed by the project will be at two different levels: an introductory Logo curriculum and an advanced series of "Dynaturtle" games. The introductory course offers detailed instructions and project ideas for the students as well as information for the teachers. The advanced activities focus on the behavior of the Dynaturtle - a Logo Turtle designed to follow Newton's Laws of Motion. They are comprised of a series of games which simulate various aspects of the laws of motion. They could be used to extend programming as well as mathematical skills.

An aspect of this project worthy of mention is the way in which "student experts" from the first project emerged as valuable resources in the second. They gradually assumed leadership roles in teaching younger and less expert peers and in helping teachers when necessary. Student interaction was another key component of this project, since the classroom environment lent itself more to the exchange of ideas and strategies among the students than the computer lab.

The two special features of the second project resulted in the reduction of the teacher's role as source of authority and in the creation of support systems from the students themselves. The role of the students as tutors was not examined in any detail.

The current study worked with the students in pairs in order to foster student interaction and verbalization activity among them. This proved most successful but the students did not have the opportunity to exchange ideas with the other groups in the project until the end, when a demonstration was organized for the parents and the staff of the school. Although the opportunity did not arise, many students expressed a desire to see the other projects and interact with the other groups.

The most ambitious project to date on Logo and children was carried out jointly by the MIT Logo group, Texas Instruments, and the Lamplighter School in Dallas (Watt, 1982). This version of Logo uses square-shaped Sprites, a special implementation, which allows multiple moving Turtles on the screen. The goal of the project was to establish a setting where students would have easy access to computers and to see what they would learn from the experience. Although the computers have been in the school since 1979, no documentation is available to date. Most information is found in informal articles written by various teachers and the research that was to materialize never did.

A group of teachers, the Computer Group, were trained initially in the fall of 1978 and began to work with third and fourth grade students

(Gorman, 1981a). As the project continued these teachers were introduced to more elaborate Logo programming concepts which they in turn taught to their students.

By September 1980, 50 TI Logo prototypes had been installed at the school and the project had expanded to include all the students in Logo programming as well as the rest of the faculty. Preschool children were introduced to simplified versions of Logo and other modifications were made for the other grade levels. Programming was introduced as an enrichment activity during lunch hour, before and after school or when classroom work was completed. Eventually, teachers used Logo in their classrooms as well because they could see that it was relevant in their lessons (Gorman, 1981a). Presently, a half-time teacher-coordinator oversees the project and provides ongoing Logo tutorials for the teachers.

Gorman (1982) describes an informal study that took place in "conjunction" with the Lamplighter Project. The study involved third grade students randomly assigned to one of three homerooms where computers were located at the start of the school year. Two of the three homeroom teachers chose to allow one half hour of Logo instruction a week while the other teacher set a one-hour-a-week minimum for her students. These conditions existed from September until the end of April; at that time, students from all three classes were given a conditional rule-learning task. This test, adopted from cognitive psychology, requires students to solve a rule-learning task from a series of pictures of shapes in different sizes and colors. Students were told which were the relevant features of the task presented and were required to learn what combination of those features satisfied the binary rule chosen by the experimenter. The possible rules included conjunction, disjunction, conditional and biconditional. The two latter rules seemed difficult for the third grade students, who were all tested on the conditional. The students from the one-hour-Logo homeroom performed significantly better than the other two groups and better still than a sixth-grade level.

Gorman (1982) emphasizes that in this case the children were not taught to the test but rather their extra Logo sessions did improve a general problem-solving skill. Although the evidence is inconclusive on whether Logo improves all thinking, future research with other measures of problem-solving and other groups of subjects should prove fruitful.

The Lamplighter school Logo project was not intended to be a formal experiment and most observations are anecdotal reports on student attitude and behavior. The students show a keen interest in acquiring new Logo knowledge and are still motivated (after three years) to learn. They also show an unprecedented desire to share ideas and programs with their peers in a cooperative spirit. As they continue to explore the Logo environment, they have acquired sophisticated typing, programming, and problem-solving concepts.

Protocol Analysis

Most research studies on the Logo environment have used either the classical experimental or the case study approach. One study (Chait, 1978) in an attempt to bridge the gap between the two methodologies developed a classification system of protocol analysis which could be applied across subjects. The current study analyzed the Logo experience within the framework of this classification system. This section will describe in detail the development of this system.

The classification system was developed on the basis of a study conducted in 1978 in St. George's Secondary School, a private institution in Montreal, Canada. Five twelve year old subjects were taught Turtle geometry on an individual basis twice weekly for eleven weeks. The criteria for selection was their performance in mathematics; only those students with grades below "B" were included.

The teaching approach was consistent with the model developed by Solomon (1976) and Papert (1980b): the child was the experimenter and the teacher the research director. After the introduction of the Turtle geometry primitives and initial programming skills, new concepts were introduced as they related to the specific projects the subjects undertook. Projects generally were chosen by the subjects themselves and began with drawing on paper a picture of the intended finished product. Then, the subject was encouraged to break down the project so that subprocedures could be utilized. The study also used two special Logo commands, TEACH and CHANGE in order to simplify the writing as well as the editing of procedures. The commands were created by the researcher and became part of the Turtle geometry primitives, that is, they were indistinguishable from such commands as FORWARD or RIGHT to the subjects.

The present study adopted the same teaching strategy, with the exception that the students were taught in pairs and verbalization between the children was encouraged. The present study did not use the TEACH and CHANGE features of the Chait study. The features that were used as aids will be discussed in the methods section.

In the Chait study, a series of tests, developed by Hein and Dunning, were administered to the subjects before and after the study, but no significant results were reported. Data was collected in terms of dribble file records and daily supplementary observations by the experimenter. These records formed the basis for the classification system.

The Development of the Classification System

Influenced by Newell and Simon's (1972) approach to protocol analysis, the data (each subject's Logo experience) was structured sequentially into episodes. These episodes, however, remained at the level of macro-analysis, lasting from five to thirty minutes in duration, while Newell and Simon, in an effort to develop a model, analyzed their protocols at a much deeper level (micro-analysis): that is, their episodes lasted from one to two minutes in duration.

The classification system is described as mostly from the bottom-up, that is the system was developed to fit the data rather than the other way around (top-down) where the system could be developed to fit a specific theory (Miller & Goldstein, 1977; Miller, 1982).

An episode began when a subject was observed to be doing something new; this was determined by either the graphic output on the screen or when a new session began. Once the protocol was organized into sequential episodes, these in turn were classified into various categories which were logically consistent (mutually exclusive), objective, and involved a minimum of ambiguity.

Six types of activities were described, based on directly observable behaviour:

1) constructing (con) - this activity involved constructing (drawing) something with the Turtle either on the computer or on paper. It was characterized as goal-directed (the subject knew what the end product was going to be) and involved no errors or bugs.

2) generate and test (g&t) - this activity also involved drawing, was also goal-directed but differed from the previous category in that it involved the presence of bugs and of course debugging strategies. This category was found to be an important area where much debugging was accomplished before the actual procedures were written.

3) no overt goal (nog) - this category described activities where it was difficult to infer a goal from the subject's behavior.

4) procedure-writing (pw) - during this activity, the subjects wrote a program or procedure.

5) revising (rev) - this activity involved revising a procedure by editing.

6) adding (add) - this activity described the addition of instructions to a procedure by editing.

In order to provide meaningful data on each subject's program structures, a "Link" classification was developed. Four different ways of linking episodes to preceding episodes were described:

1) subprocedure (sub) - this link refers to a procedure which is called by another procedure, that is, is a subprocedure of the current episode.

2) **superprocedure (super)** - this is a procedure which contains other subprocedures. Since the link refers to previous episodes, this type of relationship was not common with the subjects in the study. They tended to link their episodes from the "bottom-up"; in other words they used "con" and "g&t" before writing a procedure.

3) **revision (rev)** - this linkage occurs when one procedure is a revision of the current episode.

4) **extension (ext)** - this possible link refers to special cases, when animation projects were attempted, and one procedure was an extension of the current procedure. It labels a linkage which doesn't clearly fall into the other three categories.

The major projects of each subject were then represented diagrammatically (except for extension) by procedural nets, following the usage of Brown & Burton (1978).

The current study also utilized this system in order to examine the program structure (horizontal or vertical) and program complexity among the various projects. However, the graphic representation of the procedural nets was radically modified while the graphic representation of a recursive procedure was excluded, since it was felt it was irrelevant to the analysis.

The third major classification, "Mode", offered an explanation of where the various activities described above were completed. Four different areas were described:

1) direct command (dc) - in this mode Logo commands were executed without their being written in a procedure. the "con" and "g&t" activities usually took place in "direct command" mode.

2) teach (t) - since TEACH was used as an aid, this mode was used when the subject was in the process of writing a procedure. Thus, the only activity which took place here was "pw".

3) change (ch) - this mode was used when the subject was editing a procedure. "Rev" and "add" were activities which took place in this area.

4) not on computer (noc) - this mode referred to any activity which did not take place on the computer. "Con" was one activity which could take place in this mode.

Finally, a system was developed to classify different types of errors (bugs) made by the subjects where (in what episode) they occurred and when they were eliminated. Bugs occurred in all activities except for "con" and "nog".

Debugging strategies, representing the subject's observable action following the occurrence of a bug were also classified as "debugging actions". An asterisk was used to indicate if the action was suggested by the experimenter.

The current study modified these two latter categories by adding new types of bugs and debugging actions in order to better describe the problem-solving activities of the particular students.

Although these headings represent the major aspects of the classification system, several other categories were included in the Descriptive Summary Tables, Appendix A, in order to facilitate the analysis (Chait, 1978, p. 92-143).

Results

Although a rough anecdotal outline of each child's work on Logo was reported, the main focus of the analysis was on procedural nets, types of errors, debugging actions and the relationship between bugs and debugging. Results are reported in a descriptive fashion.

Procedural nets of the subjects major projects revealed individual differences between programming styles. Horizontal program structures were observed when a subject undertook more revisions in a project. Vertical structures, on the other hand, contained fewer revisions and a deep hierarchical structure. Some changes in programming style were also observed over time, with an increased use of subprocedures. It was speculated that a vertical program structure may be optimal for Logo programming.

The bug type classification scheme also revealed individual differences between subjects in the frequency of occurrence of certain types of bugs. Bugs tended to increase with number of episodes since the latter were partially determined by bugs. Certain bugs were reported to be fairly common across subjects. These included:

Starting-orientation bugs (st. or.) - These were errors in the Turtle's starting heading, before a shape is drawn.

Trace bugs (trace) - Errors in the Turtle's trace state when the Turtle is moved.

Position bugs (pos) - Errors in Turtle's starting or ending position, before or after a figure is drawn.

Angle bugs (ang) - Errors in the input to turn commands.

Size bugs (size) - Errors in the size of a figure in proportion to the rest of drawing on screen.

Right-left orientation bugs (rlo) - Errors in direction.

Debugging strategies were also examined in terms of their frequency of occurrence. Certain strategies were reported to have been used frequently. These were:

Clearscreen-restart (cr) - This strategy cleared the graphic screen and allowed the students to start again. It was the most frequently used debugging action.

Edit (e) - This strategy was characterized by editing a procedure in order to correct an error.

Incremental adjustment (ia) - This action was characterized by the subject using small increments to the input to eliminate a bug.

The relationship between debugging action and different bug types was then analyzed. Several trends were reported. The angle (ang) bug was most frequently debugged by incremental adjustment (ia). Starting orientation (st.or), position (pos) and trace (trace) bugs were eliminated by clearscreen-restart (cr). Debugging strategies were further examined in terms of ones that were bug specific and those that were useful for a wide variety of bugs. Clearscreen-restart was shown to be a general purpose debugging action which could be used over a variety of bugs, while edit, incremental adjustment, restart without clearscreen and undo were bug specific actions.

In certain areas, the classification system proved useful and powerful. The method was not time-consuming and could be applied across individuals. Its power was reported to be in the isolation of the "g&t" activity as a rich area where problem-solving activities occur before the actual programs are written. The study, however, did not deal clearly with the issue of reliability in two areas: the collection of the data and the transcription of the protocols. The experimenter alone conducted both parts of the research with no reliability check.

The current study used tape recordings of each session in addition to dribble file notation and experimenter commentary in order to provide a more reliable method of collecting and subsequently of analyzing the data. At the same time, the transcription of the protocols was cross-checked by another researcher who completed the visual displays on all the projects.

The classification system isolated various areas of interest but several categories were not well defined. These included the evaluation column, the distinction between "con" and "g&t" activities and the definition of an episode. A more elaborate system of microanalysis of a single subject's protocols was recommended as the next valuable step before an actual model of the Logo learning "experience" could be drawn up.

The system could only be viewed "as a first approximation of a macroanalysis of the Logo experience" (Chait, 1978, p. 86) rather than a complete system of analysis. The present study modified and clarified several of the categories in order to further analyze the learning of Logo across different groups of subjects. These will be elaborated in the following chapters.

CHAPTER III

Method

The purpose of this study is to investigate in an exploratory fashion differences in problem-solving between gifted and average subjects in the process of learning Turtle geometry. A classification system was adopted and subsequently revised so that problem-solving can be analyzed in terms of a) types of errors (bugs), b) concomitant coping strategies (debugging actions) and c) procedural nets. This chapter will summarize the methods used first for the identification of the population, design considerations, and finally, the methods used to transcribe the protocols so that the data could be analyzed within this framework.

Subjects

Sixteen subjects, four males and four females from each of grades 4 and 5, participated in the study. All attended Irving Bregman Memorial Elementary School, a public school under the jurisdiction of the Laurenval School Board, located in Laval, Quebec, a middle-class suburb of Montreal. Identification procedures followed the school guidelines for selection of a gifted population, based on Renzulli, Reis and Smith's (1981) Revolving Door Identification Model for the Gifted and Talented.

Identification Procedures

The definition of giftedness used in this study most closely relates to the one set forth by Renzulli (1978) and further explained:

> Giftedness consists of an interaction among three basic clusters of traits: above-average general abilities, high levels of task commitment and high levels of creativity. Gifted and talented children are those possessing or capable of developing this composite set of traits and applying them to any potentially valuable area of human performance.
>
> (Ridge & Renzulli, 1981, p. 204)

Accordingly, the screening procedures for the identification of the gifted population are based on this multi-dimensional outlook and consisted of the following techniques:

Informal

1. Renzulli-Hartman Scales for Rating Behavioral Characteristics of Superior Students (SRBCSS, Renzulli, Smith, White, Callahan, & Hartman, 1976)

The SRBCSS are teacher rating scales recommended by many researchers for the identification of gifted students (Clark, 1979) and consisting of ten dimensions. Each dimension gives developmental information for the specific ability area it is designed to evaluate (Renzulli, Reis & Smith, 1981). The first four scales (Learning, Motivation, Creativity and Leadership) will generally yield sufficient information for candidates of most programs for the gifted and talented. In this case, however, the teachers were asked to complete all ten scales for each candidate they felt would benefit from a program for the gifted and talented to take place in the summer of 1982.

2. Parent inventory and nomination

A second source of developmental information that has shown to be valuable in the identification process (Martinson, 1977), is the parent rating scale. It generally consists of ratings and biographical information provided by parents who feel their child would benefit from a gifted program.

3. Peer inventory

This technique provides sociometric information about an individual by members of his or her peer group. In this case, each class which took part in the identification process was involved in peer nomination simulation activities. Appendix B provides copies of these three instruments.

Formal

4. Lorge-Thorndike Group IQ Test Form 1, Levels B and C.

Psychometric information about the candidates was obtained by the administration of the Lorge-Thorndike Intelligence Test. It is meant to test abstract intelligence, that is the ability to work with ideas and relationships (Clark, 1979). The test provides a verbal, a nonverbal, and a composite IQ score. Its administration takes about an hour for each of two batteries.

The criteria for selection of the gifted group (experimental) were nomination in three out of the four categories. Equal weight was placed in all four categories. The cut-off point for the IQ test was a composite score of 120 (Clark, 1979).

From a total population of 168 in both grades 4 and 5 of the school, 71 boys and 65 girls, whose parents gave written consent for the administration of the group IQ test, took part in the screening. The population ranged from 9.3 to 11.6 (Mean age = 10.5) years of age. From these, 15 boys and 10 girls met the criteria for placement in the experimental group.

TABLE 1

Summary Characteristics of the Subjects

GROUP	SUBJECT	SEX	AGE	INFORMAL MEASURES			FORMAL MEASURE
				Teacher	Parent	Peer	Comp IQ
Experimental Group							
A	1	M	9.9	x	x	x	123
	2	M	9.10	x		x	123
B	1	M	10.7	x	x	x	128
	2	M	10.11	x		x	131
C	1	F	11.0	x		x	120
	2	F	11.3	x		x	122
D	1	F	9.10	x		x	128
	2	F	9.9	x	x	x	124
Control Group							
a	1	M	9.7		x		100
	2	M	10.0				104
b	1	M	10.9		x		95
	2	M	10.9		x		94
c	1	F	11.2		x		97
	2	F	11.2				97
d	1	F	9.7				101
	2	F	9.9		x		101

Subsequently, four boys and four girls were matched according to their IQ score, chronological age and sex in pairs. This group's mean IQ was 124.9.

Four boys and four girls from the average group were then divided in pairs matched in the same fashion to each other and then matched to the experimental group in sex, and age. This group's mean IQ was 98.6. All subjects had no previous experience working with computers and were extremely eager to work on the study. Parental permission to participate was received for all subjects who were chosen for the study. Table 1 gives a summary of the characteristics of both groups.

Research Design

This is an exploratory research study examining differences in problem-solving using a blocked two-way design, blocking on group and crossing on the sex factor (Keppel, 1982). The subjects were first divided into two groups on the basis of the four criteria described above. These two groups were the experimental or gifted group (referred to in this and subsequent chapters by upper case letters A through D) and the control or average group (referred to in this and subsequent chapters by lower case letters a through d). There was an equal number of male and female subjects in each group. The subjects were grouped into teams of two. Within each team, the subjects were matched for sex, age and IQ. A t-test performed on the mean IQ's of these subjects showed a significant difference ($\underline{t}$ = 14.69, df = 14, $p < .001$).

Experimental Setting

The physical setting was a small, narrow corridor (converted into a storage room), adjoining the teacher's staffroom and the school's library. The room contained various cupboards, a tape recorder, a small table, a desk and three chairs. Paper and pencils were available to the subjects at all times.

Apparatus

Hardware: One free-standing computer graphics system, model TT3500, manufactured by General Turtle Corporation, was used in the study. The apparatus consisted of a main computer, and a special purpose mini-computer with 4 k-bytes of main memory driving a graphic screen system. An alpha-numeric screen, a moveable keyboard and a dual floppy disk unit completed the system. The storage capacity of each floppy disk was 256 k-bytes of 8-bit words and the main computer running Logo used a LSI/11 chip and 28 k-bytes of main memory. The vector graphic display screen consisted of a triangular cursor, representing the Turtle, which could move in response to the Logo commands.

Software: The Turtle geometry (Turtletalk) aspect of Logo was used in the present study as a means of programming the Turtle. This subset of Logo has provided the most used "entry route" into programming for beginners with no prior mathematical knowledge (Papert, 1980b).

Turtle geometry is a good example of a simple but powerful computer microworld. Its power lies in its simplicity: the essential aspects of an event are limited to the Turtle's position and orientation. Thus, the Turtle's state can be changed by:

1) the FORWARD command, which changes its position (the point on the plane where the Turtle is found), and

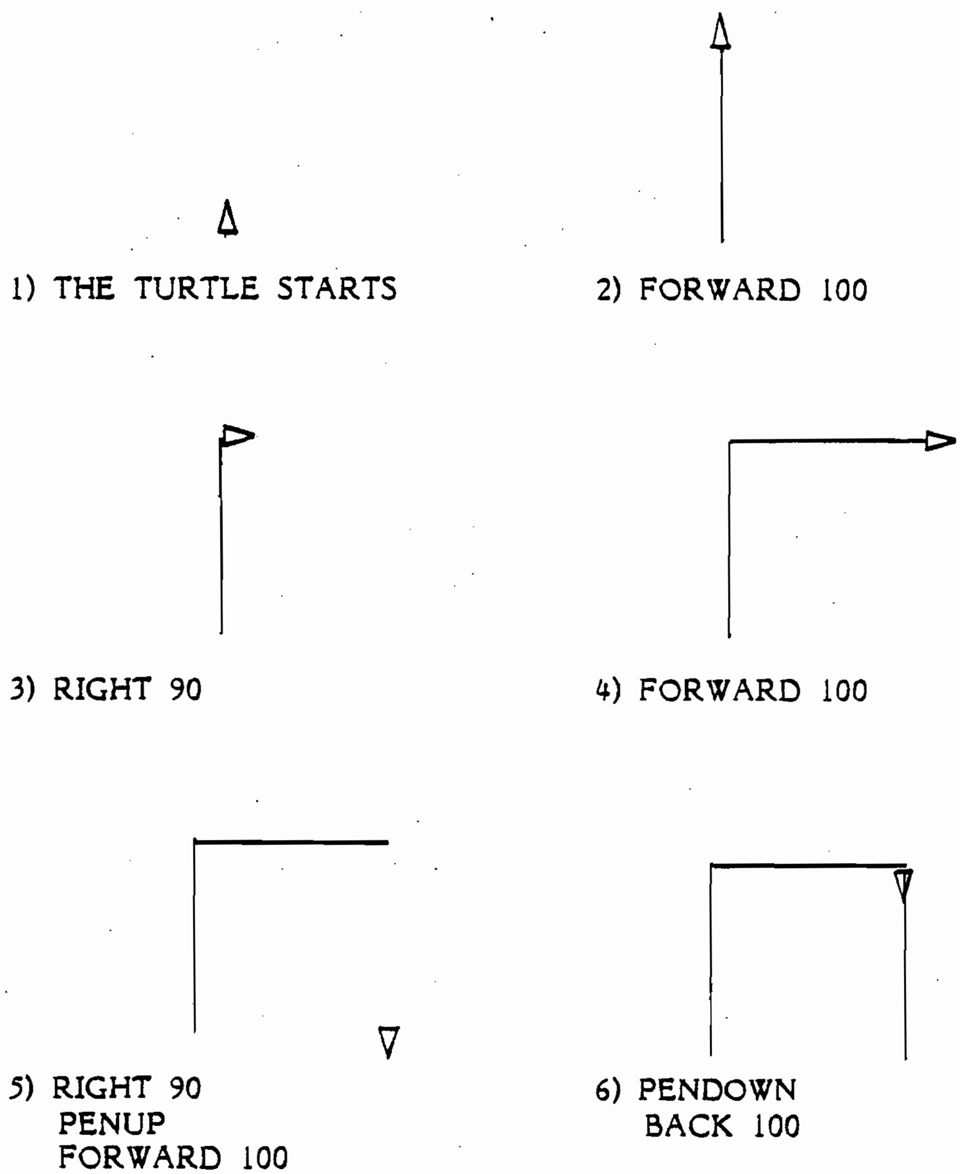
2) the RIGHT command, which changes its heading (the direction the Turtle is facing).

This simplicity may mask the sophistication of some of the concepts that are acquired in exploring Turtle geometry. Papert (1980b) describes these concepts in terms of two kinds of knowledge: mathematical and mathetic, although he conjectures that the two overlap. This section will describe some of the kinds of knowledge using specific examples from Turtle geometry.

A first encounter with either a floor Turtle or a graphic video screen Turtle usually begins by introducing the commands that change the Turtle's state. FORWARD 100 makes the Turtle move in a straight line a distance of 100 "Turtle" steps. Typing RIGHT 90 causes the Turtle to pivot clockwise in place through 90 degrees. BACK and LEFT cause the opposite movements. The Turtle can also leave a trace of the places it has been or not. The commands that control the Turtle's "trace" state are PENUP and PENDOWN.

FIGURE 1

A sequence of Turtle commands



The idea of programming is introduced through the metaphor of teaching the Turtle a new word (Solomon, 1976). The concept of teaching the Turtle "HOW TO..." is very simple and most children begin by writing programs which store SQUARE or TRIANGLE or any shape they invent and reproduce it at their own convenience.

There are many ways of creating a square or a triangle on the Turtle. Here's one version using abbreviations (FD for FORWARD and RT for RIGHT) which reduce the burden of typing and the likelihood of misspelling.

```
TO SQUARE          TO TRIANGLE
FD 100             FD 100
RT 90              RT 120
FD 100             FD 100
RT 90              RT 120
FD 100             FD 100
RT 90              RT 120
FD 100             END
RT 90
END
```

These figures have fixed sizes but they can also be made to have variable sizes. SQUARE and TRIANGLE can subsequently be used as subprocedures to create other programs (superprocedures) which invoke ("call") them directly.

Turtle geometry relates to the "intrinsic" properties of geometric figures, properties which describe the figure itself rather than the figure in relation to another reference (Cartesian geometry). The procedure SQUARE describes a figure with four equal sides and four equal angles. This description draws upon a person's intuitive geometric knowledge (Papert, 1973) and makes use of well established knowledge of "body geometry".

notions.

The Turtle procedures deal with the intrinsic property of constant curvature, i.e., the curve is the same everywhere. No other reference points are needed, such as radius or centre. These could be introduced at a later time, once one is ready for a more formal approach to Turtle geometry. The Turtle circle also illustrates the concept of "local" control. It deals with the figure a little piece at a time. This affords greater flexibility to the Turtle: arcs are simply pieces of circles. Eventually, the Turtle can extend easily out of the plane and to curved surfaces (Abelson & diSessa, 1980).

An important strategy of what a circle is is introduced to children who want to draw circles with the Turtle: play Turtle, or move your body as the Turtle on the screen must move in order to make the desired shape. Observe carefully what you are doing and then describe it in "Turtle" language. Most children discover the physical properties of a circle in this way. Other discoveries include the exploration of "The Total Turtle Trip Theorem" (Papert, 1980b). If the Turtle starts and ends in the same position, the sum of all turns will be 360 degrees.

These types of activities bring into focus two types of knowledge: one mathematical, the other mathetic, that is knowledge about learning. Mathematically, the Turtle circle episode illustrates to a child some of the basic principles of the algebra and geometry of finite differences. Although the idea of differential calculus or of a differential equation is not directly experienced, it may emerge as an extension (extrapolation) of the finite differences experienced to a limiting relation or value (Burrill, 1982).

> The Turtle program is an intuitive analog of the differential equation, a concept one finds in almost every example of traditional applied mathematics.
>
> (Papert, 1980b, p. 66)

Mathetically, the child is learning basic principles about learning itself. First principle: make sense of what you want to learn. The teacher encourages this by asking the child to verbalize something the child knows intuitively. The Turtle episode also illustrates "syntonic" learning (Papert, 1980b), learning which is in harmony with a person's sense of what is important. This type of learning is contrasted to "dissociated" learning, learning not in touch with one's personal knowledge. The term "syntonicity" is borrowed from clinical psychology and refers to several types: body syntonicity, ego syntonicity, and cultural syntonicity. The Turtle circle microworld could be described as related to the child's sense and knowledge about his or her own body (body syntonic) and satisfying in that it is coherent with his or her sense of self (ego syntonic). The phrase "ego-syntonic" is used by Freud to describe instincts or ideas acceptable to the ego. "Cultural syntonicity" refers to the fact that Turtle geometry makes sense in terms of the larger social context. Turtle geometry is learnable because it is syntonic in all three senses.

Some of the strategies used in the Turtle circle episode are special cases of Polya's ideas about problem-solving. The key to finding out how to draw a circle with the Turtle is to refer to something one knows - walking in a circle. This advice about heuristic strategies is not abstract - it is concrete, something a child can understand and use over and over again if the problem warrants it.

In the current study, several features were added to the Logo language in order to facilitate some aspects of Turtle geometry and to improve the storing and reading of files for the researchers. These were automatically loaded into each groups' workspace whenever a Logo session began and were used by some students as if they were Logo primitives. These were the following:

1) CIRCLER :SIZE or CIRCLEL :SIZE - These commands draw a circle of a specified size on the right or on the left of where the Turtle is facing.

2) ARCRIGHT :SIZE :DEGREES or ARCLEFT :SIZE :DEGREES - These commands draw arcs and require two inputs. An input is a number which goes with the command to specify how many times it should be executed. The first input is the radius which specifies the size of the arc, and the second input is the number of degrees of the arc.

3) REPEAT - This command requires two inputs. The first indicates how many times to repeat the second input, which is the list of instructions. These are enclosed in brackets.

Example: REPEAT 3 [RT 45 SQUARE]

The vector graphics of the TT3500 afford two special features of SPIN and MOVETURTLE, that were used extensively by the students in their projects. These allow the Turtle constant movement on the screen. Their features are listed below:

1) SPIN - This command gives the Turtle a spinning motion much like a spinning top, clockwise or counter-clockwise (negative input). It requires an input to tell how fast to spin by the degrees per second.

2) MOVETURTLE (MOVET) - This command gives the Turtle a push from

behind, causing it to keep moving in the direction it is heading. It requires an input to tell how fast to move the Turtle. The speed is indicated by the number of Turtle steps per second.

An added feature of the language used frequently for debugging purposes is the RUBDIS command. This command, a feature of the vector graphics screen, removes the effect of the last command given to the Turtle. If the last command was FORWARD 100, it erases the graphic output of that command and brings the Turtle back to where it was before it began going forward. It takes an input which tells the number of instructions to be rubbed out.

Finally, the STEP command was used by the students to debug errors which were difficult to isolate. Its action executes the procedure line by line in slow motion. The student controls the action of each line by pressing the carriage return. It takes the name of the procedure or ALL as an "input".

Teaching Procedure

Logo sessions were scheduled during the school day with the cooperation of the classroom teachers. The students came from their regularly scheduled classroom activities for sessions lasting from 40 minutes to one hour. They were taught in groups of two, twice weekly by one of two researchers. The study was scheduled to provide each group of students a total of twenty teaching sessions, but due to illness, class tests or other school activities, there were variations between the groups in the total number of sessions with Logo with nineteen sessions the maximum, and fifteen sessions the minimum.

The computer was brought to the school in January 1982. A three-week pilot study was undertaken in February in order to develop a consistent teaching procedure between the two experimenters. The teaching of the students in Turtle geometry started the week of March 15, 1982 and lasted for ten weeks, with a break of eight days for Easter vacation. A demonstration of projects was arranged for the staff, students and parents before the end of the school year.

The teaching procedure was consistent with the approach developed by Solomon (1976) and adopted by Chait (1978). This approach could be likened to the "combination approach" developed by Statz, Folk, and Seidman (1973). It was found to facilitate good programming and problem-solving in Logo and included more structure in the initial teaching sessions until certain basic concepts were developed, and then less structure once the concepts had been developed so that students could develop their own projects. In this fashion, the children were introduced to the basic Turtle commands and encouraged to explore the various effects, during the first session. Some students in their exploration made a free form design; others developed specific shapes such as squares or rectangles. The experimenters encouraged all students to "teach" those shapes to the computer, using the TO command. For instance, to teach the computer a new word SQUARE, the following sequence was followed:

? TO SQUARE	This means that you want to teach the computer how TO SQUARE.
> 5 FORWARD 100	Now we type in the instructions line by line numbering each line by fives or tens to allow some room between the lines.

>10 RIGHT 90	Second action is RIGHT 90.
>15 FORWARD 100	
>20 RT 90	You can use abbreviations.
>25 FD 100	
>30 RT 90	
>35 FD 100	
END	tells the computer that you have finished the definition.
SQUARE DEFINED	The computer answers.
?	The question mark shows that you are out of the definition mode.

If the students made any typing errors before the carriage return key was pressed, they used the backspace key to correct the line. If an error was noticed after the carriage return key was pressed, the students were shown that they could retype the line, using the same line number.

Examples of Typing Errors

>30 FD100	A space was left out between FD and 100.
>30 FD 100	The line is retyped correctly. It replaces the previous one.
>20RT 90	
YOU HAVEN'T TOLD ME HOW TO 20RT	The computer didn't understand this line.
>20 RT 90	The line is retyped correctly.

Once an initial procedure was written, most students further extended it to use it in a project. For example, those who had made a SQUARE might decide to make a house or a window or a windmill. These superprocedures used SQUARE as a subprocedure. The concepts of subprocedure and superprocedure were also explained to the students.

If the students made any errors in their procedure and realized them after the definition was ended (by running the procedure), the Logo editor was introduced as a means of either fixing those errors or any other changes they wanted to make in a procedure already defined. The students were shown how to type EDIT SQUARE if they wanted to correct the procedure SQUARE. Eventually, as the need arose, all students were introduced to the various editor commands and control keys which erased or edited the line in question. In the course of the project, the following commands and their abbreviated forms became familiar to all students. A list of these was made available to the students who needed to refresh their memories.

ERASE LINE 30(ERL 30)	This command erases the line from the procedure.
EDIT TITLE(EDT)	This command allows the student to change the title of the procedure.
EDIT LINE 30(EDL 30)	This command brings line 30 in a special place (buffer) so it can be changed by using special keys, the control key and a letter key from the keyboard.
CTRL-N	types out the Next word.
CTRL-C	types out the next Character.
CTRL-W	erases the last Word typed.
CTRL-R	types out the Rest of the line.

No standard curriculum was followed for the remainder of the study.

The children were encouraged to develop their own graphic projects, draw them on paper first, discuss their overall planning strategies, and then do them. The use of subprocedures and superprocedures was suggested as well as the verbalization of plans and their subsequent execution.

When bugs were encountered the students generally turned to the experimenters for guidance. They in turn asked the students to describe what had happened. Problem-solving strategies encouraged included playing turtle physically, or with paper and pencil or on the computer, by using the STEP command, or by printing out the procedure if one had been written. The editor was used to revise a procedure, if a bug had occurred after it was written. If a strategy chosen by the students was not successful in either locating an error or in correcting it, the experimenters intervened to suggest a new strategy. This advice was not necessarily followed by the teams.

The concept of using the system's procedure writing facility (TO) and the direct command mode at the same time (g&t activity) to write and try out a procedure was explained to some groups and is labelled as using the Editor-as-a-scratch-pad. The students were able to write the procedure title and follow either the direct command (dc) mode to try out the Turtle action on the graphic display screen by not numbering the line or to write the line in the procedure by numbering it. This approach gave the subjects greater flexibility in the writing of a procedure and could be seen as a more advanced form of programming. The classification system, however, did not prove adequate in describing this type of activity.

Although there was some variation among the various projects, similar commands and programming concepts were covered. Table 2 provides a summary of the Logo commands used by each group. Table 3 shows the programming concepts introduced during the course of the study within the context of the project work undertaken. For a summary of when in the course of the study the various groups were introduced to these commands and concepts, see the NEW MATERIAL column in Appendix A, p. 183.

TABLE 2

Turtle Geometry Commands Covered

COMMAND	EXPERIMENTAL GROUP				CONTROL GROUP			
	A	B	C	D	a	b	c	d
Forward	x	x	x	x	x	x	x	x
Back	x	x	x	x	x	x	x	x
Right	x	x	x	x	x	x	x	x
Left	x	x	x	x	x	x	x	x
Clearscreen	x	x	x	x	x	x	x	x
Hideturtle	x	x	x	x	x	x	x	x
Showturtle	x	x	x	x	x	x	x	x
Penup	x	x	x	x	x	x	x	x
Pendown	x	x	x	x	x	x	x	x
Repeat	x	x	x	x	x	x	x	x
Home	x	x	x	x	x	x		x
Heading		x	x	x			x	
Wrap		x	x	x	x	x	x	
Nowrap		x	x			x		
Moveturtle	x	x	x	x	x	x		x
Spin		x	x	x	x	x	x	
Rubdis	x	x	x	x	x	x	x	x
Step	x	x	x	x	x	x	x	x
Erase	x	x	x	x	x	x	x	x
Printout	x	x	x	x	x	x	x	x
Printout title	x	x	x	x	x	x	x	x
Print		x				x		
Circler		x			x		x	
Circlel		x			x		x	
Arcright			x	x	x		x	
Arcleft				x	x		x	
Edit	x	x	x	x	x	x	x	x
Edit title	x	x	x	x	x	x		
Edit line	x	x	x	x	x	x	x	x
Erase line	x	x	x	x	x	x	x	x
Level		x	x	x	x	x	x	
Stop		x	x	x	x	x	x	
Wait		x	x	x				

TABLE 3

Logo Programming Concepts Covered

Logo Material	Experimental Group A	B	C	D	Control Group a	b	c	d
Procedure Writing	x	x	x	x	x	x	x	x
Recursion	x	x	x	x	x	x	x	
Stop Rule		x	x	x		x	x	
Subprocedure	x	x	x	x	x	x	x	x
Superprocedure	x	x	x	x	x	x	x	x
Input	x	x	x	x	x	x	x	x
Circle	x	x	x	x	x	x	x	x
2 Inputs (Poly)		x	x	x	x	x	x	
Total Turtle Trip		x	x	x	x			
Input + or -		x	x	x				
2 Inputs +		x	x					
Editor as a scratch pad		x						

Collection of the Data

Data from each session were collected in terms of the following:

1. Dribble file notations- these were provided by the Logo system and included a complete record of each group's interaction with the computer.

2. Students' drawings of project ideas and plans drawn on paper, as well as any notes or calculations written on paper.

3. Tape recordings of each session.

4. Detailed teaching logs with personal observations by the Logo teachers during and after each session.

Transcription of the Protocols

The protocols were transcribed following the method developed by Chait (1978). Although the students were instructed in pairs, the protocols of each group were analyzed as one. Hard copies of the dribble files were obtained and the students' drawings and notes on paper were attached to each session. The tape recordings of each session and the experimenter observations were used to provide a more reliable method of conducting the first analysis. The transcription of the protocols was performed by one of the experimenters and cross-checked by another who completed the visual displays on the projects.

Although most of the classification system was not altered, some categories were modified and others were expanded. This section describes the changes that were made to the Chait classification.

1) Type of Activity - A new activity was isolated called "Test". It could be described as equivalent to the phase defined by Miller (1982) as "trying out the procedure" (p. 122). It refers to trying out the procedure after it is written to see if it works, and could also be related to Polya's fourth stage (see Chapter II). It invariably took place in "direct command" (dc) mode and proved useful in that it clearly separated one episode from another.

2) Evaluation - This category was radically changed. It was originally developed by Chait (1978) to show "the inferred evaluation of the episode by the subject" (p. 49). In the current study, it represents the evaluation of the episode by the observed action of the "Test" activity, that is when the procedure was tried out. If the procedure ran without any errors, then the evaluation was positive (+); if it did not then a negative (-) sign was used. A positive evaluation of the episode invariably led the subjects to work on the next phase of the project; a negative evaluation, on the other hand, was usually followed by revision of the same procedure to correct the error, or to generate and testing (g&t) in order to locate the bug.

3) Mode - Since this study did not use the aids developed by Chait for writing and editing a procedure, the following modification was made to this category. The "Edit" (e) mode was used to signify the use of the EDIT command to change or add to a procedure already written.

4) Error Type - This category was extended to include various new types due either to new commands introduced to the subjects, or commands which were further extended. Two errors were made when the REPEAT command was used. REPEAT/Syntax (R/syn) refers to errors made in the typing of the REPEAT statement, for example when a bracket was left out, while REPEAT/input (R/inp) refers to errors made in the number of times the commands in the brackets were repeated.

When a recursive procedure was written, some subjects made an error in the recursive line or the title line of the procedure. This error was labeled Recursion (rec).

The Stop bug was further explained to be either a Stop/R to show an error in the stopping of a recursive procedure, Stop/S to show an error in the stopping of a SPIN command, or Stop/M to show an error in the stopping of a MOVETURTLE command.

The out-of-bounds (ob) error type was further classified into the type of error which was observed to cause the out-of-bounds error message in the first place. Out-of-bounds could be caused by either the size of the figure being drawn "ob(s)", or the position of the figure being drawn "ob(p)".

A Typing error (typ) was signaled only if it caused the subjects to change episodes, that is the error caused the subjects to use revision or EDIT, or it occurred during the writing of a procedure.

5) Debugging Action - This category was also extended to include new strategies that the subjects used to correct certain bugs. If they used a calculation either on the computer or on paper to determine the correct response to a bug, the action was signaled by Calculation (C); the place where the calculation took place was either on the computer C(t) or not on the computer C(noc).

Decremental Adjustment (da) was used by the subjects when the input to a FORWARD command was too large and the computer responded with an out-of-bounds message.

PENUP (PU) was another strategy used by some groups to estimate the length of a certain line or the position of a certain figure before a decision was made on the correct number to be used.

6) Procedural Nets - The graphic representation of the subjects' major goal-directed projects was radically modified in order to glean more information from them and in order to tie them more closely to the Descriptive Summary Tables (Appendix A). Several conventions remained the same. First, only the projects the subjects worked on for more than one session (Major) were depicted in this fashion. The numbers refer to the episode within which the procedure was written. The names found underneath the procedure refer to the actual name given to that procedure by the students. This addition is also found in the Descriptive Summary

Tables under the Heading NAME OF PROCEDURE. These nets show the differences between bottom-up, mixed, and top-down programming styles. This will be further elaborated in Chapter IV, in Figures 2 through 9, where the procedural nets of all the subjects are found.

Summary of Chapter

The exploratory analyses of differences in problem-solving between gifted and average teams of subjects will be made using a classification system of protocol analysis developed in 1978, and revised for the present study. The subjects were identified as gifted and average in terms of four criteria based on a construct of giftedness, proposed by Renzulli (1978). The teaching procedure adopted was based on Papert's model of the child as the experimenter and the teacher as the director of the learning environment.

The classification system and the design of the present study allow for a more quantitative examination of differences between groups of subjects. However, the small number of teams per group limits the kind of meaningful statistical analyses that could be carried out. Therefore, the study will remain explorative and will examine problem-solving from various levels, some descriptive and others quantitative in order to best fit the data at hand. No attempt will be made to either generalize from these results nor to evaluate the Logo environment.

Once the raw data is systematically classified into the various activities of the classification system, the analyses will focus in examining differences in types and number of errors and concomitant coping strategies. The relationship between these two variables will also be examined. At another level of analysis, the procedural nets of the major projects of the subjects will be analyzed in order to investigate differences in programming style and project complexity between the two groups. Specific questions will be asked of each area of analyses in order to guide the exploration into a more focused direction. The results are presented in Chapter IV.

CHAPTER IV

Results

This chapter summarizes the results of the exploratory study. The focus will be on the problem-solving activities of gifted and average groups of subjects in the learning of Turtle geometry. Problem-solving in this context is seen in terms of three categories: the procedural nets employed by the students in their project work, the types and frequency of errors, and the types and frequency of debugging strategies. The relationship between the most common bugs and their debugging strategies will also be examined.

The protocols of each group are found in the Descriptive Summary Tables, Appendix A. They provide a complete description of the problem-solving experience of all the subjects according to the classification system adopted. They also provide the basis for the exploratory analyses in three key areas:

1) procedural nets,

2) bugs, and

3) debugging strategies.

Analyses of Procedural Nets

The following questions will be asked in this section of the analyses:

1) Do the procedural nets reveal differences in programming style between the gifted and the average groups? Programming style refers to the structure of the nets, horizontal or vertical (Chait, 1978) and top-down or bottom-up

(Papert, Watt, diSessa, & Weir, 1979).

2) Are there any discernible differences between the groups in the complexity of the projects? Complexity will be analyzed in terms of tree depth, number of procedures, number of revisions and number of bugs.

Figures 2 through 9 are the procedural nets for the subjects' major projects. Major projects are those which were more than one session in duration. The sequential order is from right to left: in other words, procedures written earlier in time are found on the right while ones written later are found on the left. Subprocedures, on the other hand, are always found below superprocedures on the tree structure.

Each relation is represented by different types of lines. Horizontal dotted lines indicate the revision of an episode structure. Vertical and diagonal interrupted lines indicate subprocedure to superprocedure relations, when the subprocedure is defined before the superprocedure. Vertical, horizontal and diagonal thin solid lines indicate the relation between subprocedure to superprocedure after the superprocedure is written; this is exemplified in the "add" classification. Thick, solid lines represent the relation between superprocedure to subprocedure, when the subprocedure is written in the superprocedure before it is defined.

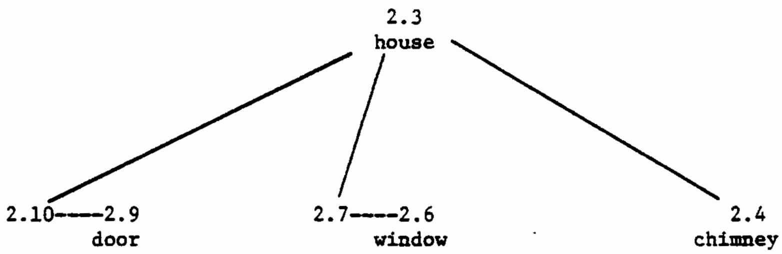
KEY

	Revision of Procedure
	Subprocedure to Superprocedure Link (Bottom-up)
	Subprocedure to Superprocedure Link by Adding after Superprocedure is written
	Superprocedure to Subprocedure Link (Top-down)

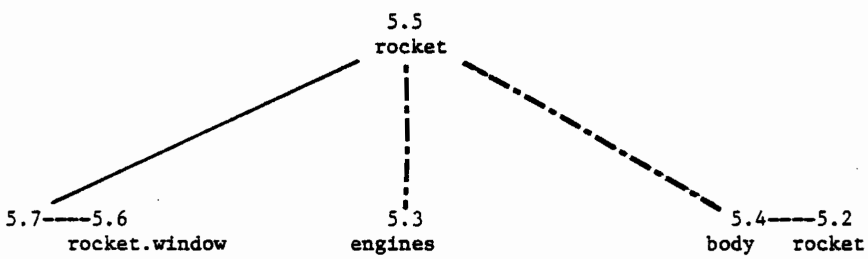
FIGURE 2

Procedural Nets of Group A's Major Projects

Project 2: HOUSE



Project 5: ROCKET



Project 7: SIDEWALK

7.12---7.11---7.10---7.9---7.8---7.7---7.6---7.5---7.4----7.3---7.2
sidewalk sidewalk

FIGURE 2 (cont'd)

Project 9:CITY

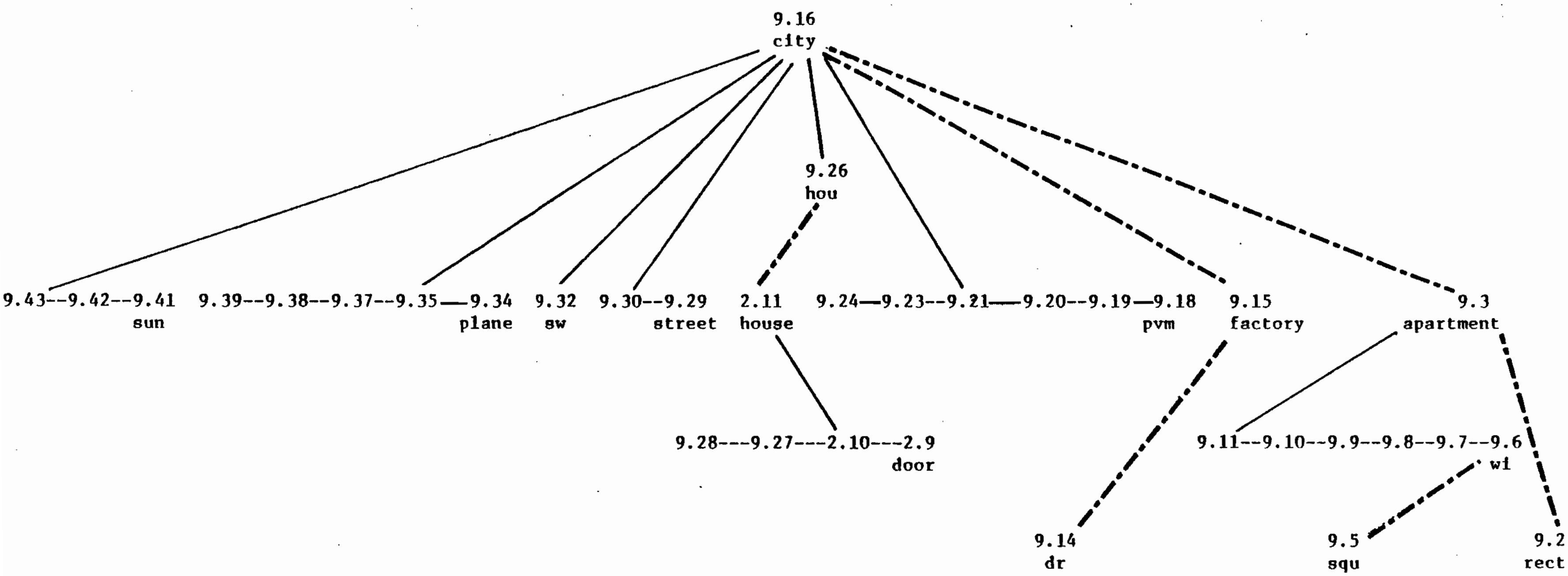


FIGURE 2 (cont'd)

Project 10:AIRPORT

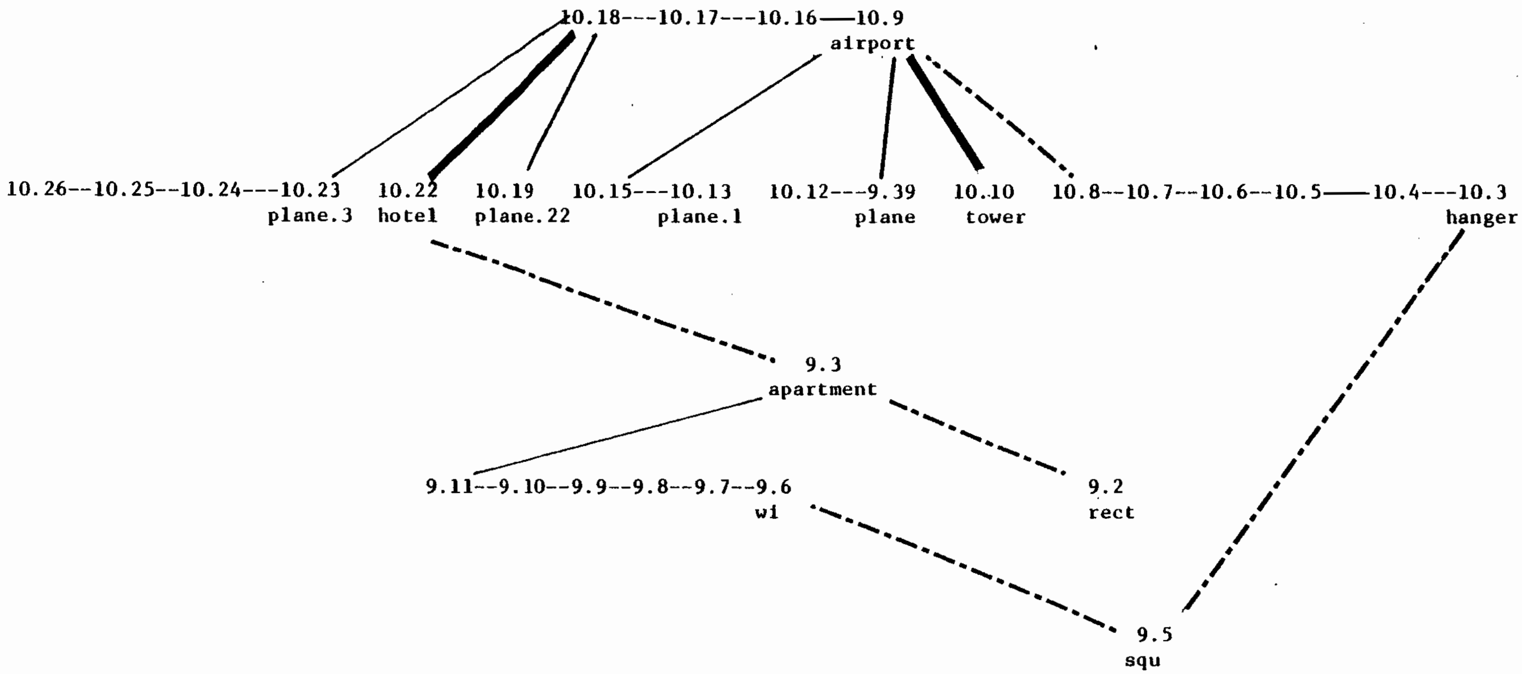
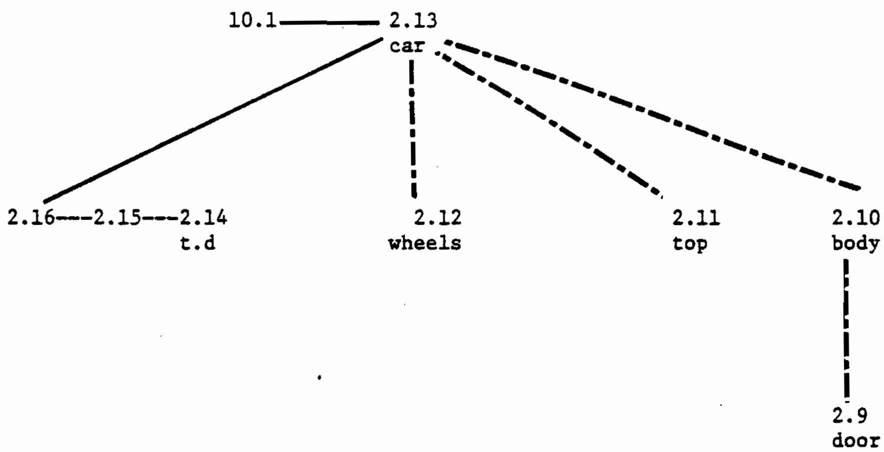


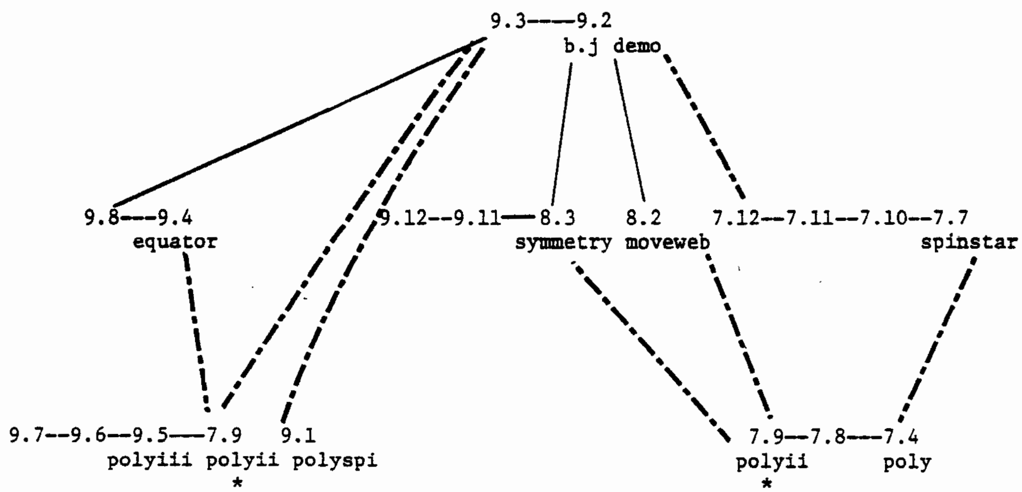
FIGURE 3

Procedural Nets of Group B's Major Projects

Project 2: CAR



Project 9: DEMO PROGRAM



*same procedure

FIGURE 3 (cont'd)

Project 5: CLOCK

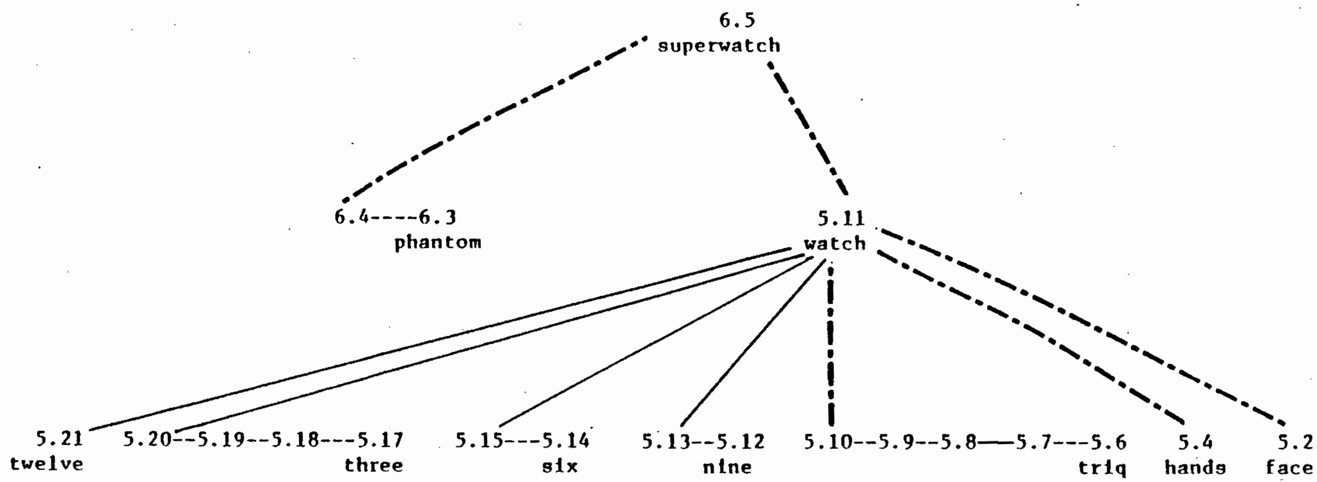


FIGURE 4

Procedural Nets of Group C's Major Projects

Project 2: LOGO

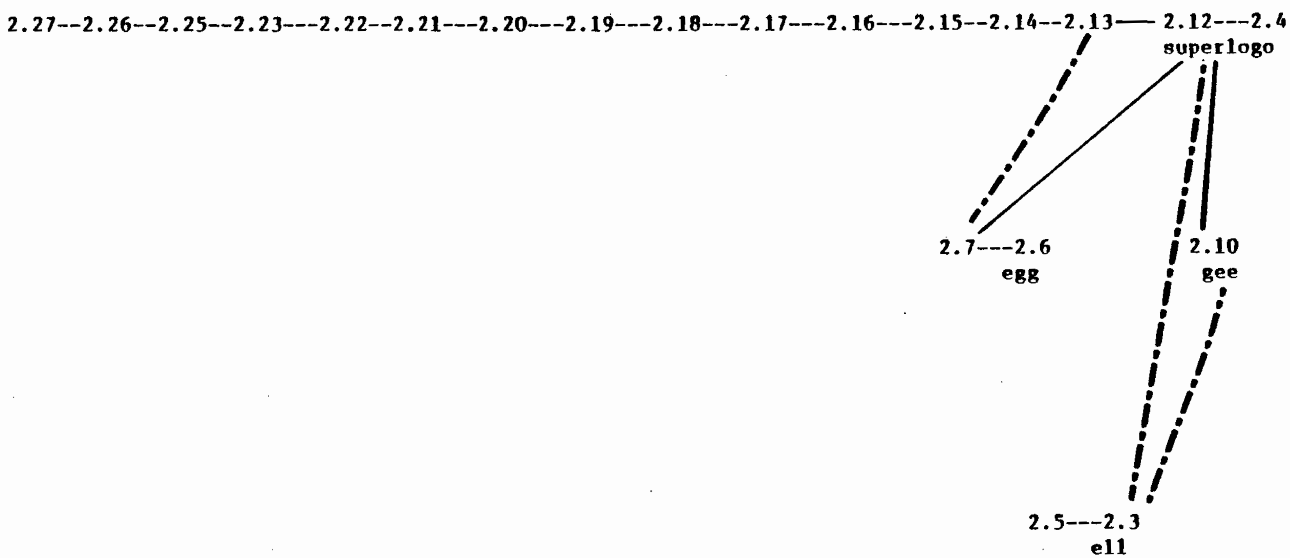


FIGURE 4 (cont'd)

Project 7: DECOY

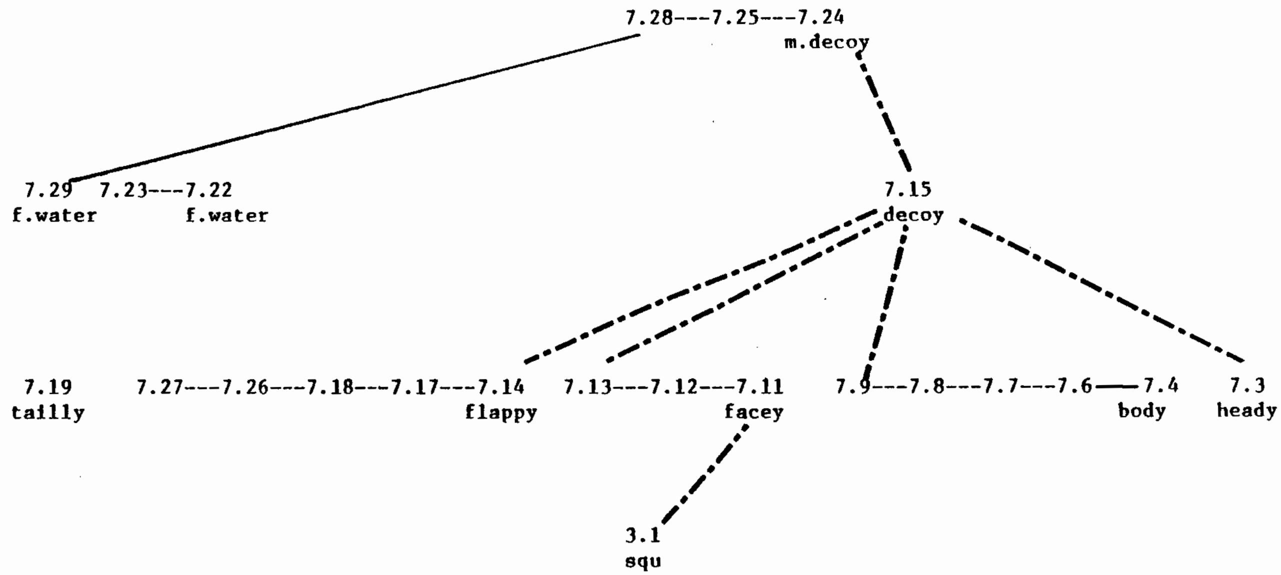
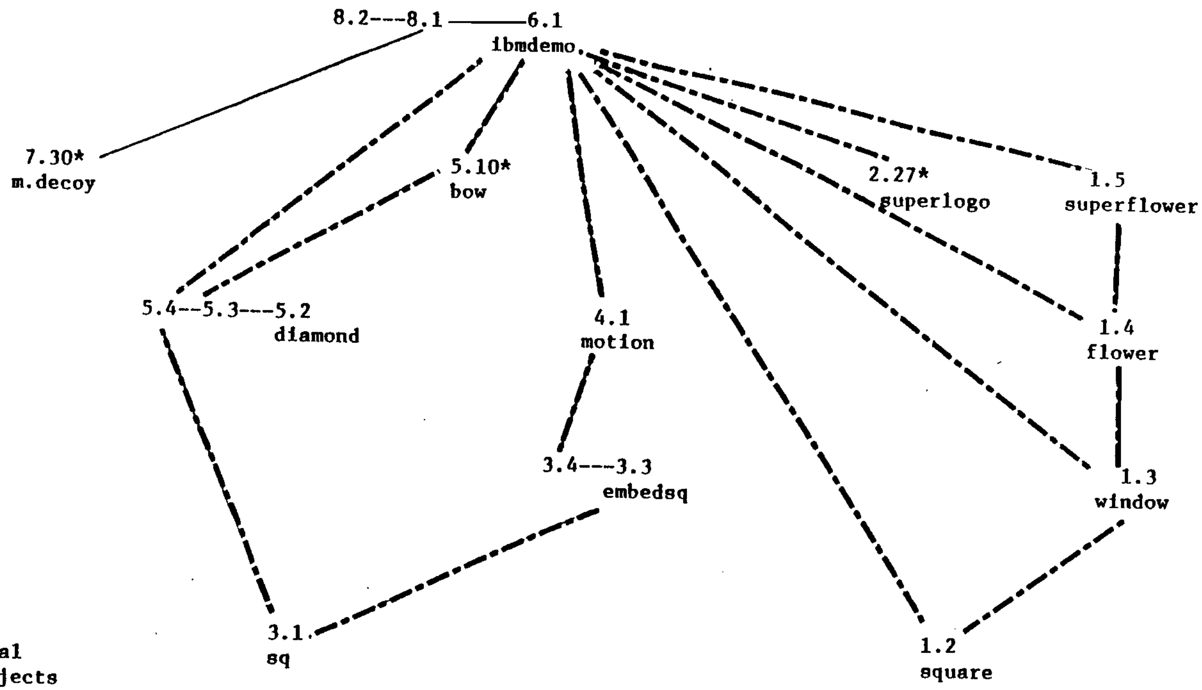


FIGURE 4 (cont'd)

Project 6:DEMO



* Complete Procedural Nets of these Projects are found in separate pages.

FIGURE 4 (cont'd)

Project 5: BOW

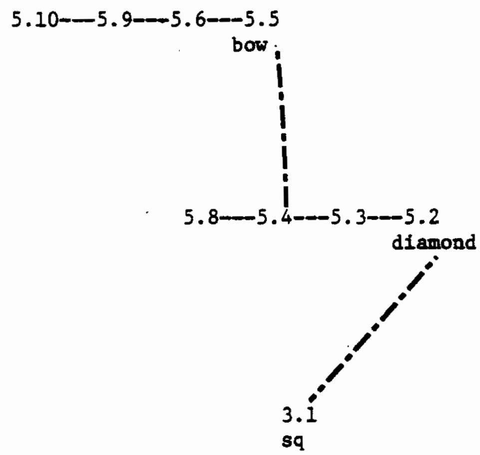


FIGURE 5

Procedural Nets of Group D's Major Projects

Project 2: WINDMILL

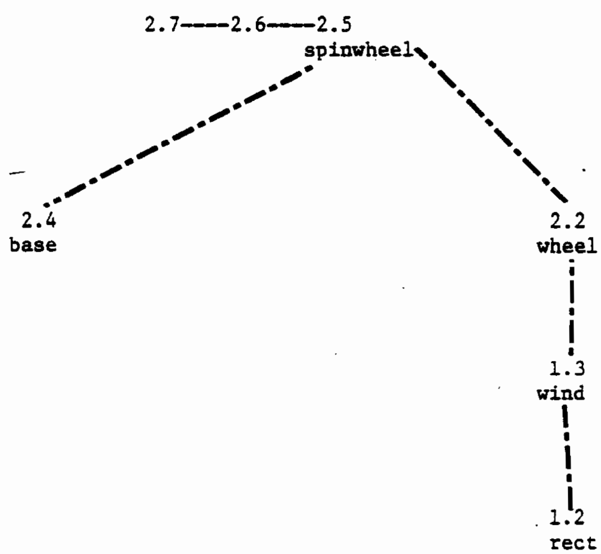
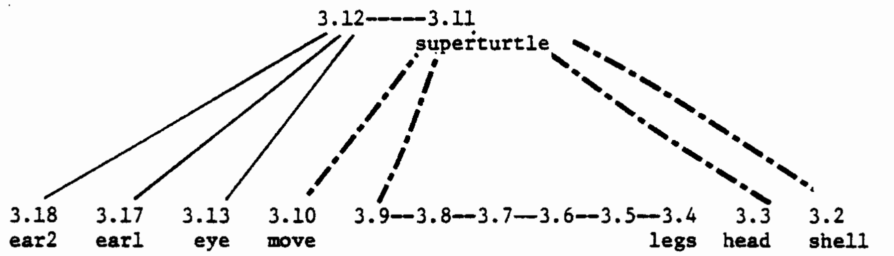
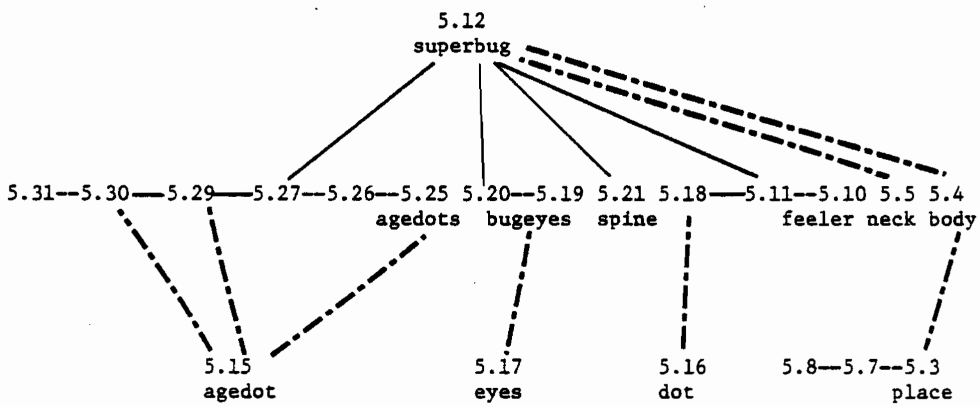


FIGURE 5 (cont'd)

Project 3: SUPERTURTLE



Project 5: LADYBUG



Project 7: DRAGONFLY

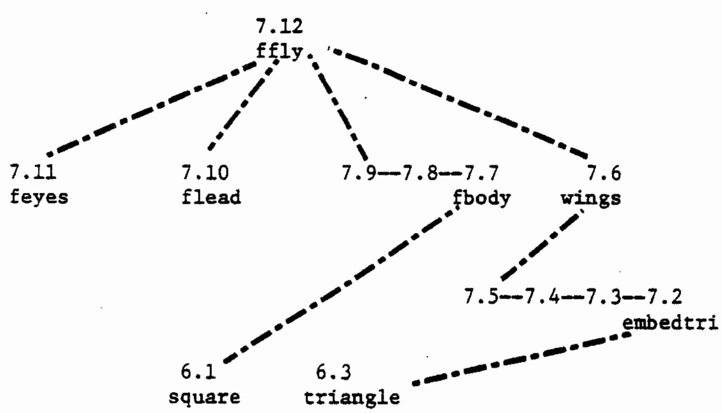


FIGURE 5 (cont'd)

Project 8: FARM SCENE

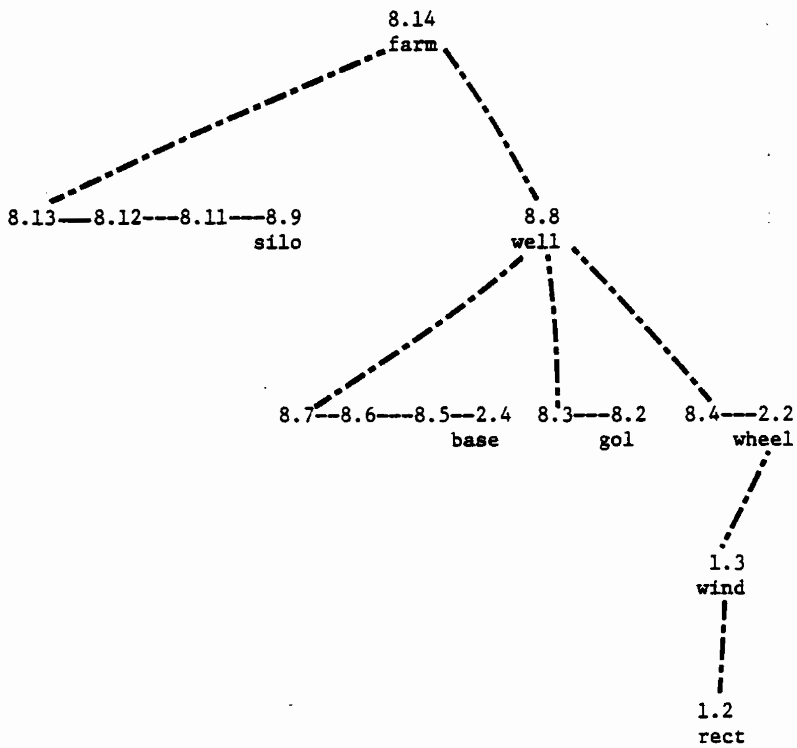


FIGURE 6

Procedural Nets of Group a's Major Projects

Project 3: WINDMILL

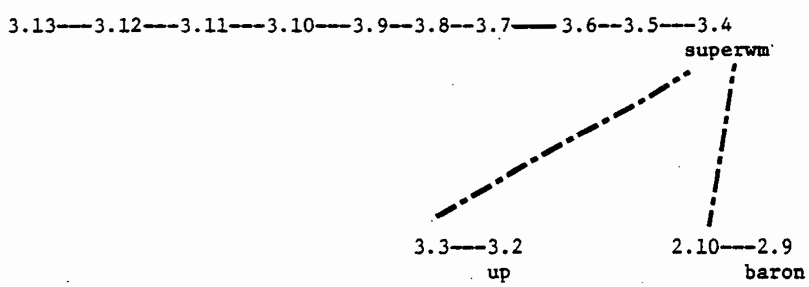


FIGURE 6 (cont'd)

Project 4: CLOWN

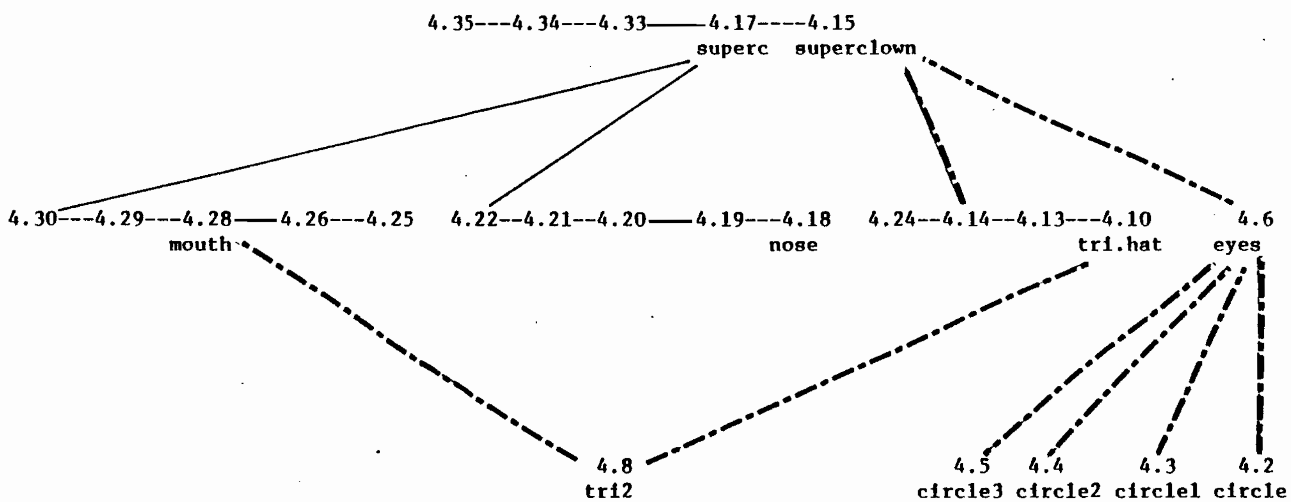


FIGURE 6 (cont'd)

Project 5: VAN

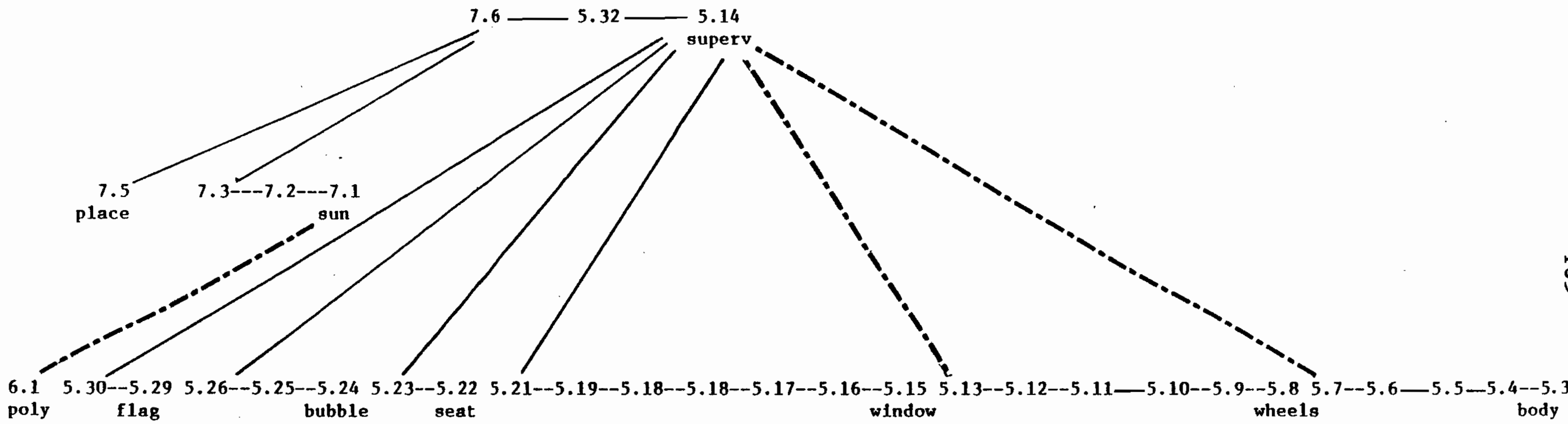
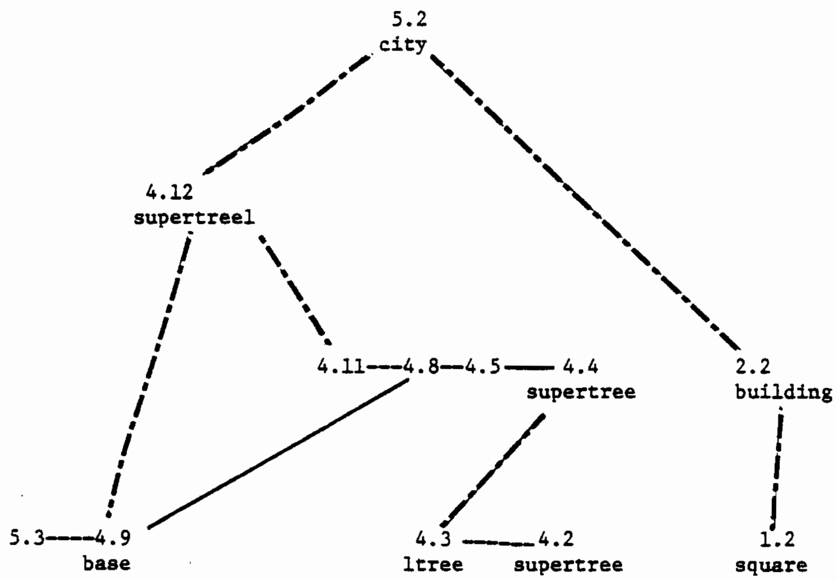


FIGURE 7

Procedural Nets of Group b's Major Projects

Project 5: CITY



Project 7: WINDMILL

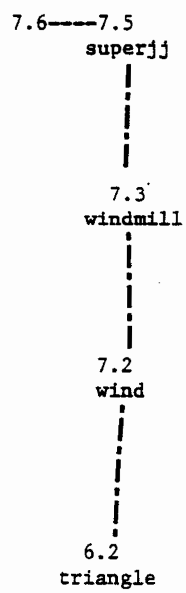
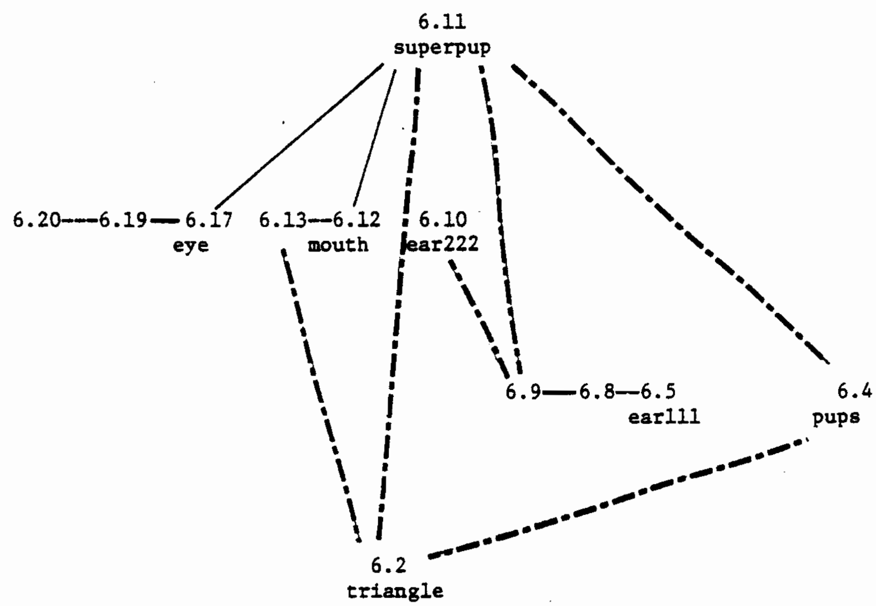


FIGURE 7 (cont'd)

Project 6: SUPERPUP



Project 8: ROCKET

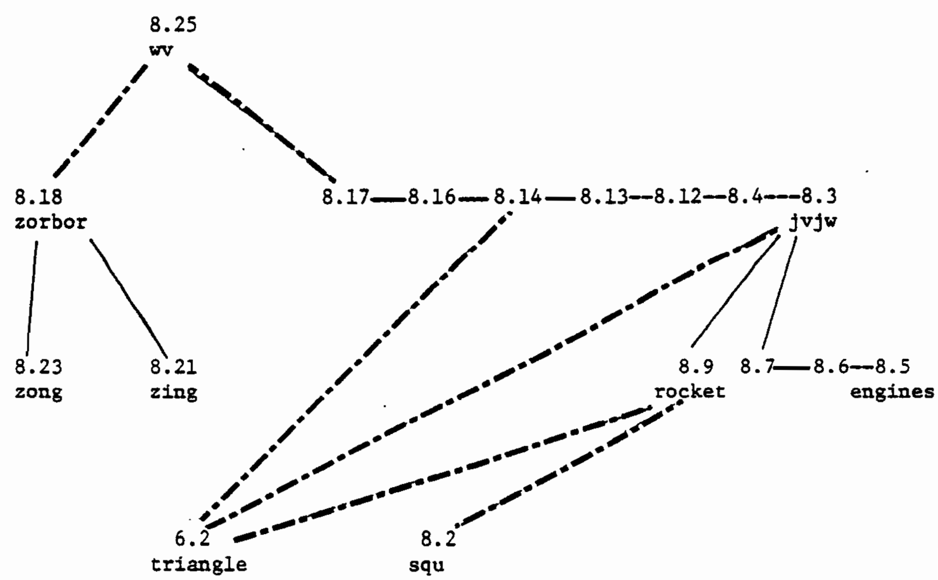
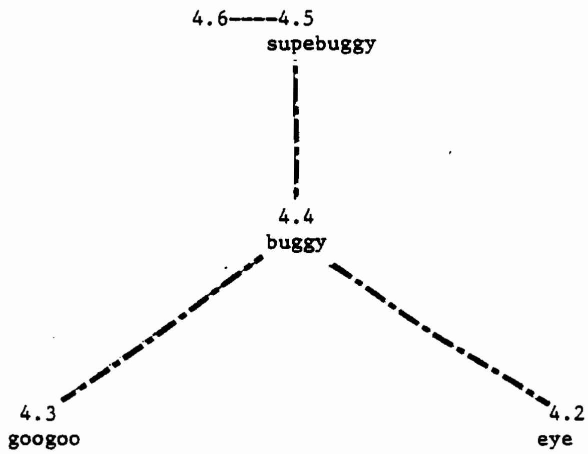


FIGURE 8

Procedural Nets of Group c's Major Projects

Project 4: BUG



Project 10: BUTTERFLY

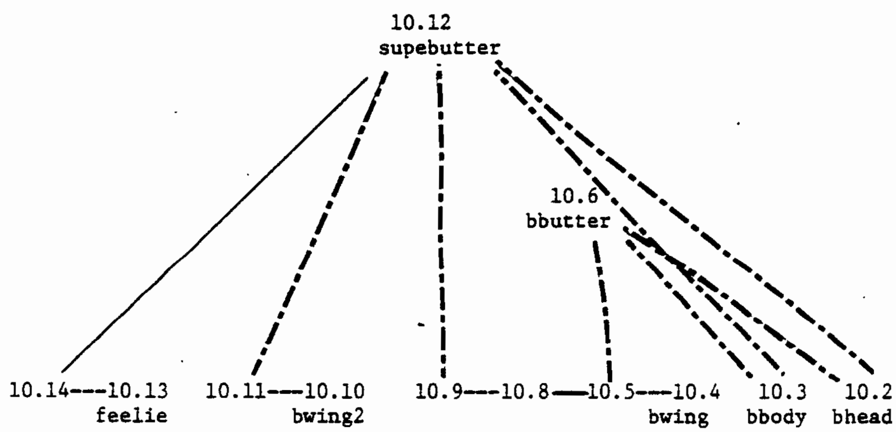
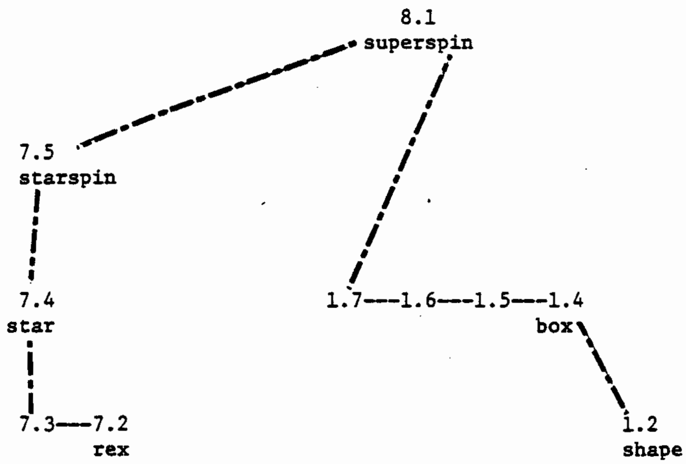


FIGURE 8 (cont'd)

Project 8:SERENDIPITY



PROJECT 12:CHAIN

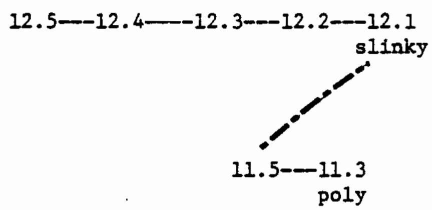


FIGURE 9

Procedural Nets of Group d's Major Projects

Project 2:HOUSE

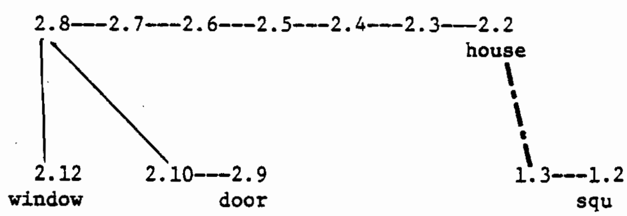
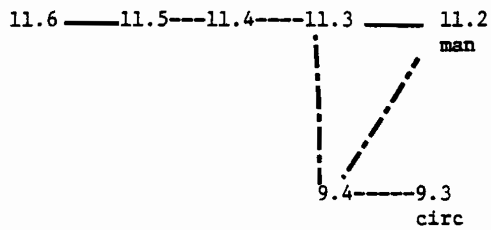
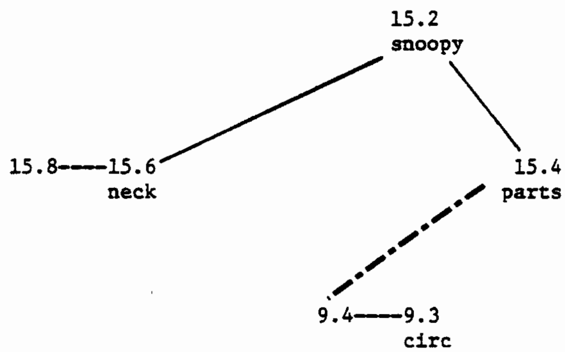


FIGURE 9 (cont'd)

Project 11: PERSON



Project 15: SNOOPY



Project 17: TRAILER

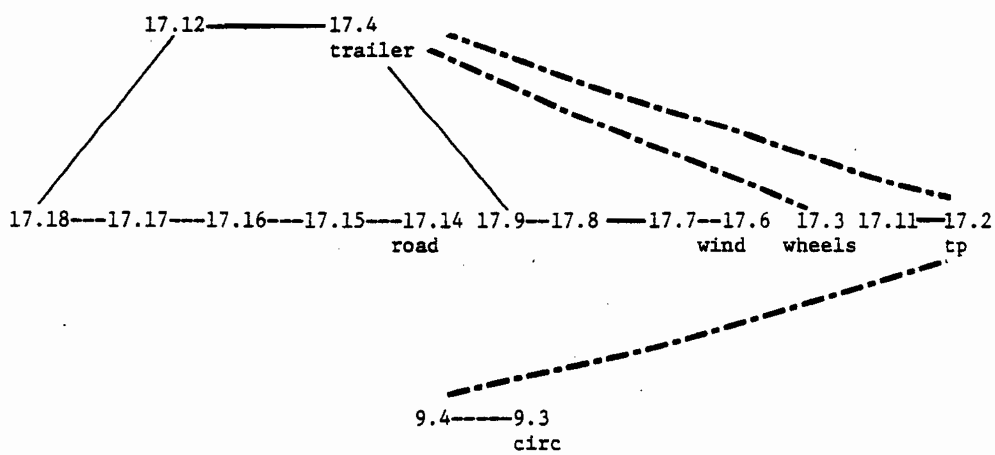
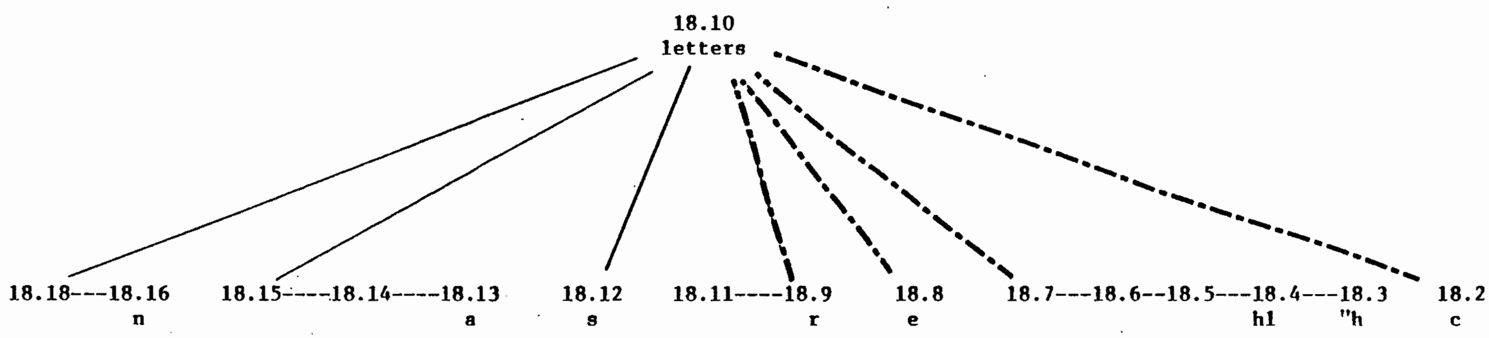


FIGURE 9 (cont'd)

Project 18: LETTERS



Thus, in Figure 2, Project 10: Airport, apartment (9.3) has two subprocedures, rect (written in episode 9.2) and wi (written in 9.6 and revised five times). Wi was defined using the subprocedure squ (9.5), which following the hierarchical order of the nets is also a subprocedure of apartment, hotel, hanger (sic), and finally of airport (the superprocedure). In the same figure, the superprocedure airport (written in 10.9) included at the time of definition, hanger (10.3). Hanger was revised once in episode 10.4, revised in order to add certain sections in 10.5 and revised in order to edit until 10.8. Subprocedures plane, plane.1, plane.22 and plane.3 were added to the superprocedure airport on subsequent episodes. These had been defined and debugged before their addition in airport. Procedures tower (10.10) and hotel (10.22) represent a special relationship to the superprocedure. They were written in airport before they were defined. This is termed top-down programming (Papert, Watt, diSessa & Weir, 1979). Conversely, the relationship between the subprocedure squ (9.5) and wi (9.6) and hanger (10.3) describes bottom-up programming, since the subprocedure was defined before the superprocedure was written. The addition of plane (9.39) to the superprocedure represents a mixed type of style. It is neither clearly bottom-up nor top-down. It could be termed as an intermediate style, since the superprocedure was already written and the subprocedure was added afterwards.

In general terms, the procedural nets demonstrate individual differences between subjects and changes within subjects in the course of the study. Projects that have the same name and may even have the same graphic output were not carried out in the same fashion. Although Group D and Group b both worked on Project Windmill, they each followed a different process to achieve their goal. On the other hand, Group D's procedural nets (Figure 5) show a progressively more sophisticated series of projects from the beginning to the end of the study. This may be determined, on the one hand, by examining the number of procedures defined per project. Project 2: Windmill had 5 procedures while Project 5: Ladybug had 11.

The procedural nets also show the hierarchical structure of each project (Chait, 1978), as indicated by the number of revisions. Figure 2, Project 7: Sidewalk is an extreme example of what may be termed a horizontal program structure, defined by many revisions and no general planning strategy. Figure 7, Project 5: City differs in that it has a more vertical structure, a more defined planning strategy and subsequently fewer revisions. Although there are individual differences between and within subjects, no global differences can be isolated between the gifted and the average groups. Most groups tended to favor a more vertical programming structure and a somewhat defined planning strategy.

Table 4 summarizes the procedural nets of the gifted groups in terms of duration of project (how many episodes each project lasted), tree depth of project, total number of procedures written, total number of revisions per project, and total number of bugs. This table also gives information on the programming style of the subjects in terms of bottom-up, top-down, or add linkages (intermediate programming style). The last column gives the total number of sessions each group took part in for the duration of the project. Table 5 summarizes the procedural nets of the average groups in terms of the same criteria.

TABLE 4

Summary of Major Projects

Experimental Groups

Group	Project	Dur (epis)	Tree depth	Total proc.	Total rev.	Total add	Total bugs	Total bot.up	Total ses.
A	2: House	11	2	4	2	3	16	0	16
	5: Rocket	8	2	4	2	1	10	2	
	7: Sidewalk	12	1	1	10	0	18	0	
	9: City	44	5	15	16	12	92	6	
	10: Airport	26	5	12	16	6	38	5	
B	2,10: Car	18	3	6	2	2	28	4	18
	5,6: Clock	28	3	10	9	5	56	5	
	7,8,9: Demo	31	3	9	11	4	26	7	
C	2: Logo	27	3	4	16	3	29	3	16
	5: Bow	10	3	3	6	0	9	2	
	1,2,3,4,5,6,8: Demo	12	5	12	3	1	5	12	
	7: Decoy	30	4	10	12	2	58	6	
D	2: Windmill	7	4	5	2	0	9	4	19
	3: Superturtle	19	2	8	6	3	27	4	
	5: Ladybug	31	3	11	7	7	64	8	
	7: Dragonfly	12	4	8	5	0	31	7	
	8: Farm scene	14	5	8	7	1	20	7	
Total		340	57	130	132	50	536	82	

TABLE 5

Summary of Major Projects

Control Groups

Group	Project	Dur (epis)	Tree depth	Total proc.	Total rev.	Total add	Total bugs	Total bot.up	Total ses.
a	3: Windmill	13	2	3	10	1	20	2	19
	4: Clown	35	3	10	12	5	54	8	
	5: Van	38	3	10	18	11	65	3	
b	4,5: City	16	4	7	4	2	21	6	17
	6: Superpup	20	4	7	3	4	54	6	
	7: Windmill	7	4	4	1	0	9	3	
	8: Rocket	25	4	9	4	8	52	6	
c	4: Bug	6	3	4	1	0	25	3	15
	1,7,8: Serendipity	14	4	6	4	0	20	5	
	10: Butterfly	15	2	7	4	2	36	7	
	11,12: Chain	8	2	2	4	0	12	1	
d	2: House	13	2	4	8	2	15	1	16
	11: Person	6	2	2	3	2	26	2	
	15: Snoopy	8	3	4	2	2	28	1	
	17: Trailer	19	3	6	7	5	39	3	
	18: Letters	18	2	9	8	3	54	4	
Total		261	47	94	93	47	530	61	

In terms of programming style, examination of the procedural nets of the subjects and Tables 4 and 5 gives rise to the following observations: the majority of projects were completed following bottom-up and mixed programming styles. Only one group used top-down programming for a section of a project. Several figures show that some projects were completed using bottom-up programming only. Figure 5, Project 2: Windmill, Project 7: Dragonfly, and Project 8: Farm scene are examples of this. Most figures show that the subjects tended to favor a combination of bottom-up and mixed programming for the more complex projects. Figure 2, Project 9: City, Figure 3, Project 5: Clock, Figure 5, Project 3: Superturtle, and Project 5: Ladybug are examples of this.

Some subjects showed a tendency not only to define subprocedures before writing the superprocedure but also to revise them until they were perfect. Figure 3, Project 5: Clock is an example of this kind of "cautious" programming style. Subprocedures triq (5.6), nine (5.12), six (5.14), and three (5.17) were all defined, subsequently revised until they met the criteria of the students for perfection and then inserted into the superprocedure, in this case, watch (5.11).

Other groups were more adventurous. Their subprocedures were inserted into the superprocedure before they were tested and, subsequently, if errors were found had to be revised afterwards. Figure 2, Project 9: City offers examples of this kind of programming style. Subprocedures pvm (9.18), plane (9.34) and door (2.9) were all defined, placed in the superprocedure and then tested, which led the subjects to revise them afterwards.

Some groups displayed characteristics similar to those described by

Solomon (1982) as "timid" learners or "microexplorers". Group d's procedural nets show the tendency of this group of girls to work in a self-created microworld of their own making: Circles. They repeatedly chose and developed project ideas that contained circular shapes, using in the process a subprocedure CIRC, which drew an arc of variable size. This group's work in Appendix A also shows that most of their project work remained at a simpler level than the other groups and lasted for shorter periods of time. Many of their projects were not represented diagrammatically because they did not last more than one session. For example, the team worked on 18 projects in the duration of the study and only 4 lasted more than one session. The final project (18) was the most ambitious of this team's work. Although individual differences were observed between the teams, no general differences were noted between the gifted and the average groups.

In terms of project complexity between gifted and average groups, several observations can be made.

1) The gifted teams chose and executed to completion more complex projects than the average teams. This is demonstrated, first, by the tree depth of the major projects. The tree depth of the gifted teams' could be termed as quite different in that they tended to use a deeper hierarchical programming structure than the average teams. The column tree depth of Tables 4 and 5 shows that the gifted teams wrote four projects that had a tree depth of 5, and three projects that had a depth of 4. The average teams did not write any projects with a depth of 5 and only five with a depth of 4. The remainder were projects of three or two. Figure 2, Project 9: City, completed by Group A is the most complex project undertaken by the subjects in the study.

2) At the same time, the gifted teams wrote more procedures (130) in contrast to the average teams who wrote 94. Since episodes are partially determined by number of revisions (Chait, 1978), the gifted made more revisions (132) in comparison to the average teams (93).

3) The average, on the other hand, had the same number of bugs in total (530) in contrast to the gifted's (536). These bugs were in part related to the revisions. For the most part, they were encountered during the generate and test (g&t) activity and were debugged before the procedure was written. Revisions were made only when bugs were encountered after the procedure was written and the Logo editor had to be used. It could be speculated, then, that the average teams made more errors in the g&t activity than the gifted, while the latter had to correct more errors after the procedure was written.

In summary, the examination of the projects undertaken by the gifted and average teams and their level of complexity indicate general differences between them. However, in terms of programming styles and structures, general differences were not found, although individual differences were observed.

Development of Problem-solving Model

Although the classification system did not demonstrate differences in programming styles, similarities were noted. One segment of the dribble file of a group of subjects will be described in detail, and a model of problem-solving in this context will be described based on the subjects' experience.

process to achieve their goal. On the other hand, Group D's procedural nets (Figure 5) show a progressively more sophisticated series of projects from the beginning to the end of the study. This may be determined, on the one hand, by examining the number of procedures defined per project. Project 2: Windmill had 5 procedures while Project 5: Ladybug had 11.

The procedural nets also show the hierarchical structure of each project (Chait, 1978), as indicated by the number of revisions. Figure 2, Project 7: Sidewalk is an extreme example of what may be termed a horizontal program structure, defined by many revisions and no general planning strategy. Figure 7, Project 5: City differs in that it has a more vertical structure, a more defined planning strategy and subsequently fewer revisions. Although there are individual differences between and within subjects, no global differences can be isolated between the gifted and the average groups. Most groups tended to favor a more vertical programming structure and a somewhat defined planning strategy.

Table 4 summarizes the procedural nets of the gifted groups in terms of duration of project (how many episodes each project lasted), tree depth of project, total number of procedures written, total number of revisions per project, and total number of bugs. This table also gives information on the programming style of the subjects in terms of bottom-up, top-down, or add linkages (intermediate programming style). The last column gives the total number of sessions each group took part in for the duration of the project. Table 5 summarizes the procedural nets of the average groups in terms of the same criteria.

TABLE 4

Summary of Major Projects

Experimental Groups

Group	Project	Dur (epis)	Tree depth	Total proc.	Total rev.	Total add	Total bugs	Total bot.up	Total ses.
A	2: House	11	2	4	2	3	16	0	16
	5: Rocket	8	2	4	2	1	10	2	
	7: Sidewalk	12	1	1	10	0	18	0	
	9: City	44	5	15	16	12	92	6	
	10: Airport	26	5	12	16	6	38	5	
B	2,10: Car	18	3	6	2	2	28	4	18
	5,6: Clock	28	3	10	9	5	56	5	
	7,8,9: Demo	31	3	9	11	4	26	7	
C	2: Logo	27	3	4	16	3	29	3	16
	5: Bow	10	3	3	6	0	9	2	
	1,2,3,4,5,6,8: Demo	12	5	12	3	1	5	12	
	7: Decoy	30	4	10	12	2	58	6	
D	2: Windmill	7	4	5	2	0	9	4	19
	3: Superturtle	19	2	8	6	3	27	4	
	5: Ladybug	31	3	11	7	7	64	8	
	7: Dragonfly	12	4	8	5	0	31	7	
	8: Farm scene	14	5	8	7	1	20	7	
Total		340	57	130	132	50	536	82	

TABLE 5

Summary of Major Projects

Control Groups

Group	Project	Dur (epis)	Tree depth	Total proc.	Total rev.	Total add	Total bugs	Total bot.up	Total ses.
a	3: Windmill	13	2	3	10	1	20	2	19
	4: Clown	35	3	10	12	5	54	8	
	5: Van	38	3	10	18	11	65	3	
b	4,5: City	16	4	7	4	2	21	6	17
	6: Superpup	20	4	7	3	4	54	6	
	7: Windmill	7	4	4	1	0	9	3	
	8: Rocket	25	4	9	4	8	52	6	
c	4: Bug	6	3	4	1	0	25	3	15
	1,7,8: Serendipity	14	4	6	4	0	20	5	
	10: Butterfly	15	2	7	4	2	36	7	
	11,12: Chain	8	2	2	4	0	12	1	
d	2: House	13	2	4	8	2	15	1	16
	11: Person	6	2	2	3	2	26	2	
	15: Snoopy	8	3	4	2	2	28	1	
	17: Trailer	19	3	6	7	5	39	3	
	18: Letters	18	2	9	8	3	54	4	
Total		261	47	94	93	47	530	61	

In terms of programming style, examination of the procedural nets of the subjects and Tables 4 and 5 gives rise to the following observations: the majority of projects were completed following bottom-up and mixed programming styles. Only one group used top-down programming for a section of a project. Several figures show that some projects were completed using bottom-up programming only. Figure 5, Project 2: Windmill, Project 7: Dragonfly, and Project 8: Farm scene are examples of this. Most figures show that the subjects tended to favor a combination of bottom-up and mixed programming for the more complex projects. Figure 2, Project 9: City, Figure 3, Project 5: Clock, Figure 5, Project 3: Superturtle, and Project 5: Ladybug are examples of this.

Some subjects showed a tendency not only to define subprocedures before writing the superprocedure but also to revise them until they were perfect. Figure 3, Project 5: Clock is an example of this kind of "cautious" programming style. Subprocedures triq (5.6), nine (5.12), six (5.14), and three (5.17) were all defined, subsequently revised until they met the criteria of the students for perfection and then inserted into the superprocedure, in this case, watch (5.11).

Other groups were more adventurous. Their subprocedures were inserted into the superprocedure before they were tested and, subsequently, if errors were found had to be revised afterwards. Figure 2, Project 9: City offers examples of this kind of programming style. Subprocedures pvm (9.18), plane (9.34) and door (2.9) were all defined, placed in the superprocedure and then tested, which led the subjects to revise them afterwards.

Some groups displayed characteristics similar to those described by

Solomon (1982) as "timid" learners or "microexplorers". Group d's procedural nets show the tendency of this group of girls to work in a self-created microworld of their own making: Circles. They repeatedly chose and developed project ideas that contained circular shapes, using in the process a subprocedure CIRC, which drew an arc of variable size. This group's work in Appendix A also shows that most of their project work remained at a simpler level than the other groups and lasted for shorter periods of time. Many of their projects were not represented diagrammatically because they did not last more than one session. For example, the team worked on 18 projects in the duration of the study and only 4 lasted more than one session. The final project (18) was the most ambitious of this team's work. Although individual differences were observed between the teams, no general differences were noted between the gifted and the average groups.

In terms of project complexity between gifted and average groups, several observations can be made.

1) The gifted teams chose and executed to completion more complex projects than the average teams. This is demonstrated, first, by the tree depth of the major projects. The tree depth of the gifted teams' could be termed as quite different in that they tended to use a deeper hierarchical programming structure than the average teams. The column tree depth of Tables 4 and 5 shows that the gifted teams wrote four projects that had a tree depth of 5, and three projects that had a depth of 4. The average teams did not write any projects with a depth of 5 and only five with a depth of 4. The remainder were projects of three or two. Figure 2, Project 9: City, completed by Group A is the most complex project undertaken by the subjects in the study.

2) At the same time, the gifted teams wrote more procedures (130) in contrast to the average teams who wrote 94. Since episodes are partially determined by number of revisions (Chait, 1978), the gifted made more revisions (132) in comparison to the average teams (93).

3) The average, on the other hand, had the same number of bugs in total (530) in contrast to the gifted's (536). These bugs were in part related to the revisions. For the most part, they were encountered during the generate and test (g&t) activity and were debugged before the procedure was written. Revisions were made only when bugs were encountered after the procedure was written and the Logo editor had to be used. It could be speculated, then, that the average teams made more errors in the g&t activity than the gifted, while the latter had to correct more errors after the procedure was written.

In summary, the examination of the projects undertaken by the gifted and average teams and their level of complexity indicate general differences between them. However, in terms of programming styles and structures, general differences were not found, although individual differences were observed.

Development of Problem-solving Model

Although the classification system did not demonstrate differences in programming styles, similarities were noted. One segment of the dribble file of a group of subjects will be described in detail, and a model of problem-solving in this context will be described based on the subjects' experience.

TABLE 6

Dribble File of Group B

Section of Project 5: Clock

```
1.  ?PU FD 70
2.  ?BK 70
3.  ?FD 75
4.  ?PD
5.  ?RT 90
6.  ?CIRCLER 140
7.  OUT OF BOUNDS
8.  AT LEVEL 3 LINE 30 IN REPEAT
9.  ?CS
10. ?FD 95
11. ?CS
12. ?PU FD 95
13. ?PD RT 90
14. ?CIRCLER 140
15. ?HT
16. ?ST

17. ?TO FACE
18.>5 PU FD 95
19. >10 PD RT 90
20.>15 CIRCLER 140
21.>END
22. FACE DEFINED

23. ?CS
24. ?FACE

25. ?RT 90
26. ?FD 95
27. ?RUBDIS 1
28. ?PU FD 100
29. ?FD 40
30. ?PD
31. ?HT
32. ?ST

      Session ends

33. ?FACE
```

TABLE 6 (cont'd)

34. ?RT 90
35. ?PU FD 140
36. ?RT 180
37. ?PD FD 110
38. ?RUBDIS 1
39. ?PD FD 100
40. ?HT
41. ?RUBDIS 1
42. ?RUBDIS 1
43. ?ST
44. ?PD FD 90
45. ?PU BK 90
46. ?PD RT 90
47. ?PU FD 70
48. ?BK 5
49. ?PD BK 65
50. ?POTS
51. TO TD
52. TO SQ
53. TO TOP
54. TO BODY
55. TO DOOR
56. TO FACE
57. TO CIRCLE2
58. TO EMBEDSQ :SIDE
59. TO SQUARE :SIDE
60. TO CIRCLE1
61. TO CIRCLE
62. TO BOOK
63. TO TRI
64. TO CAR
65. TO WHEELS
66. ?PO TRI
67. TO TRI
68. 5 FORWARD 150
69. 10 RIGHT 120
70. 15 FORWARD 150
71. 20 RIGHT 120
72. 26 FORWARD 150
73. 30 RIGHT 120
74. END

75. ?TO HANDS
76. >5 RT 90
77. >10 PU FD 140
78. >15 RT 180
79. >20 PD FD 90
80. >25 PU BK 90
81. >30 PD RT 90

TABLE 6 (cont'd)

82.>35 FD 65
83.>END
84. HANDS DEFINED

85. ?CS
86. ?FACE HANDS

87. ?HT
88. ?ST
89. ?LT 90
90. ?FD 5
91. ?RT 135
92. ?FD 5
93. ?RT 45
94. ?HT
95. ?RUBDIS 5
96. ?HT
97. ?ST
98. ?HT
99. ?FD 5
100. ?RT 135
101. ?FD 10
102. ?RT 90
103. ?FD 10
104. ?RT 135
105. ?FD 5
106. ?FD 5
107. ?ST
108. ?RT 90
109. ?HT

110. ?TO TRIQ :SIDE
111.>5 LT 90
112.>10 FD 5
113.>15 RT 135
114.>20 FD 10
115.>25 RT 90
116.>30 FD 5
117.>EDIT TITLE
118.>TO TRIQ
119.>END
120. TRIQ DEFINED

121. ?CS
122. ?FACE HANDS TRIQ

123. ?EDIT TRIQ
124.>PO
125. TO TRIQ

TABLE 6 (cont'd)

```
126. 5 LEFT 90
127. 10 FORWARD 5
128. 15 RIGHT 135
129. 20 FORWARD 10
130. 25 RIGHT 90
131. 30 FORWARD 5
132.>EDL 30
133.>30 FORWARD 10
134.>35 RT 135
135.>40 FD 5
136.>END
137. TRIQ DEFINED

138. ?CS
139. ?FACE HANDS TRIQ
```

The dribble file records of a section of a major project called Clock conducted by Group B is found in Table 6. Each line is numbered for easier reference and each activity is separated from the next one by a space. The section from Appendix A which corresponds to this dribble file is found in Table 7. The following are the accompanying explanations, with additional comments from the tape recordings of the sessions and the experimenter's logs.

This team of boys decided to work on project Clock; they drew it on paper and devised an execution plan. This activity is shown in Table 7 as constructing (con) but not on the computer (noc). The dribble file section (Table 6) starts with episode 5.2, lines 1 through 16. It is called the generate and test (g&t) activity of the first phase of the project, in this case the drawing of the clock face. The subjects discussed among themselves the size of the clock and decided to move the Turtle to the top left of the screen and then to draw the circle. They made three errors in this activity, all three associated with the movement of the Turtle to the top of the screen before they drew the circle. They corrected them immediately, since they were still in direct command (dc) mode. They used the CIRCLER procedure to draw the circle, were very pleased with the effect and proceeded to define the first procedure FACE which included the steps to move the Turtle where they wanted to position it and the actual drawing of the circle. This process of not separating the positioning elements from the actual drawing of the figure is characteristic of the beginning stages of programming (Papert, Watt, diSessa, & Weir, 1979).

Once the boys defined the procedure FACE, lines 17-22, they cleared the screen, line 23, Table 6, and tested the procedure. They were pleased with the output. "The Turtle is neat." Keeping the visual display on the screen (5.2 as a subprocedure, Table 7), they proceeded to the next phase depicted in Table 7 as episode 5.3 and in Table 6 as lines 25 through 32. Their goal here was to draw the minute and hour hands. In this episode, they succeeded only in moving the Turtle to the center of the circle. This they accomplished by turning the Turtle to face the center (RT 90) and then by moving FD 100. They immediately realized the circle had a radius of 140, CIRCLER 140, and that the center should be close to that distance from the outer edge and incrementally adjusted the distance by adding FD 40 to 100. The teacher intervened here to bring to the boys' attention the fact that the center may not be exactly 140, since the circle is not exactly a circle, but a many-sided polygon. The session ended at this point.

TABLE 7

Descriptive Summary of Section of Group B's Project Clock

FIRST ANALYSIS - GROUP B

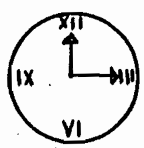
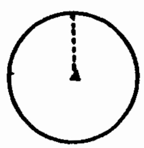
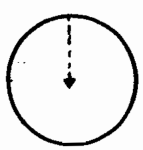
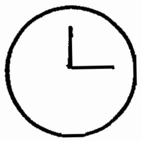
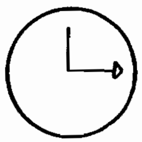
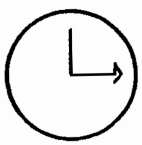
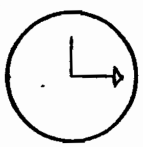
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 5: Clock										
	5.1			a)con			noc					
	5.2			a)g&t			dc			len	5.2	u
										ob(p)	5.2	cr
										trace	5.2	cr
			FACE	b)pw			t					
				c)test			dc	+				
	5.3			a)g&t	5.2	sub	dc			trace	5.3	rr*
										len	5.3	ia

TABLE 7 (cont'd)

FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 22 Ses.8	5.4			a)g&t			dc t			len len len	5.4 5.4 5.4	rr rr, rr PU PO, PO
			HANDS	b)pw			t					
				c)test			dc	+				
	5.5			a)g&t			dc			len ang	5.5 5.5	rr rr
			TRIQ	b)pw			t			typ cop	5.5 -	e na
	5.6			a)rev	5.5	rev	e		ctrl C, W, G			
				b)test			dc	-				
	5.7			a)rev	5.6	rev	e			cop	5.7	PO, e
				b)test			dc	+				

In the following session, the boys were eager to continue. Table 7 depicts this as episode 5.4, while in Table 6, lines 33 through 74, describe the dribble file records of this section. They re-drew the face and proceeded to draw the hands using the same strategy as before. They realized, as they were turning the Turtle in order to move towards the center of the circle, that they could have taken the Turtle backwards (BACK command); in this case, they would not have needed to turn the Turtle.

They were amused at this discovery but proceeded with the old strategy, remembering the discussion of the previous session and settling for FD 140 as the distance to the center. They turned the Turtle to the heading they wanted before putting the pen down and then tried FD 110 as the length of the minute hand. It was too large and they followed the rubout-retry (rr) strategy to eliminate that error, tried FD 100 and hid the Turtle (HT) in order to see the exact length of the line. A discussion followed between the boys on the size of the hand in relation to the clock numbers they were planning to draw later. Since they were not pleased with the length, they continued the discussion and settled on FD 90. This time they remembered the BK command, moved back down the minute hand to the center again, turned the Turtle RT 90 and discussed the length of the hour hand. They knew it would have to be less than 90, but how much less? One of the boys suggested that instead of going through the previous strategy (rr), they should put the Turtle's pen up (PU), try a forward 70 and see if it looks right. They tried this strategy out and were able to settle on 65 as the correct length fairly quickly.

A discussion followed as to whether they needed an arrow at the end of the hands. One of the boys considered it unnecessary but after a while they reached a consensus and tried it. They didn't remember if they had a triangle procedure that drew triangles of different sizes and used POTS and PO to check their workspace and the procedure they had on triangles. They didn't have one and decided to define the procedure HANDS before they continued any further. One of the boys wasn't sure about whether this was a good idea. "I think we have a bug for some reason." The teacher encouraged the writing of the procedure as far as they knew it and Table 6, lines 75 through 84, represent this section of the activity. One of the boys typed while the other one dictated from the text screen. They tried out both FACE and HANDS, lines 85 and 86, and both procedures were perfect. The boys seemed pleased.

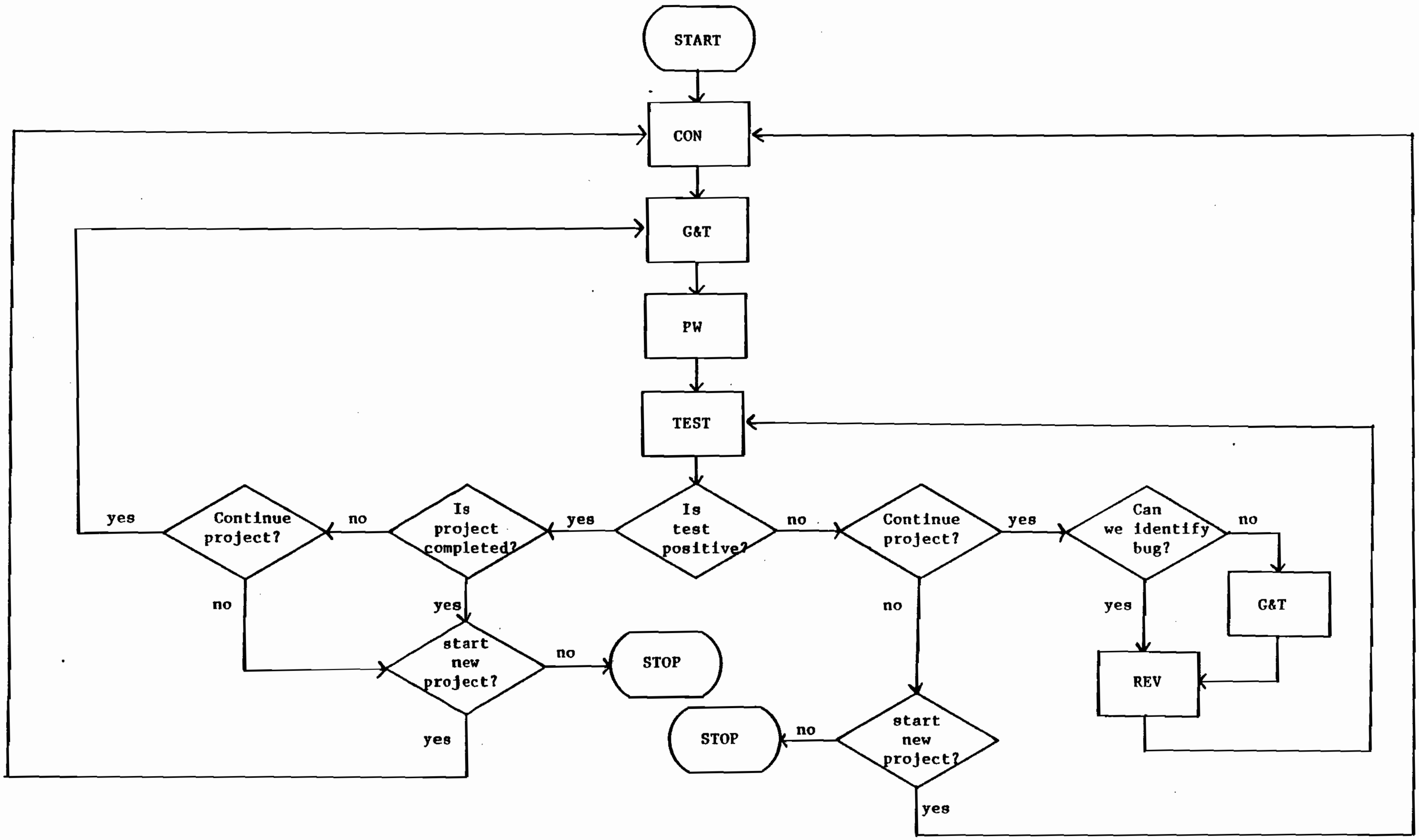
In the next episode, 5.5, they decided to draw the arrow for the clock hands but not to define a triangle procedure with variable sizes. This is shown in Table 6 by the lines 87-109. "We'll make a TRIQ one size." Since the Turtle was already at the end of the hour hand, they decided to draw that arrow first. One boy knew the first angle was RT 135 for the triangle and convinced his partner they should try this number for the first angle. The second angle (RT 45) gave them trouble. The teacher suggested they try out a new strategy: a triangle of different sizes; since the boys wanted a triangle which is centered on the hour hand, they rejected the idea. They eventually drew the arrow and defined the procedure TRIQ, lines 110-120. This presented some difficulty to them because they had forgotten to write down on paper the steps of the procedure and the text screen had rolled on top of the screen and out of their access. They asked the teacher if she had kept track and since she hadn't, they tried to remember the various steps themselves.

They made a typing error (perhaps stemming from the previous discussion of triangles with variable inputs) on the title of the procedure. They realized this before they had finished the definition and used the command EDT to correct that error. They then ended the definition and tried out the procedure, lines 121,122. They had made a copying error and the triangle was not finished. "We missed a forward." The boys were amused with this bug but were also pleased with themselves at discovering the error so quickly. They revised the procedure TRIQ (episode 5.6, lines 123-137) and were introduced to new editor control keys C,W and G in the process. They corrected line 30 (EDL 30) and added lines 35 and 40 to finish the triangle. The test this time was positive and the boys continued on the next phase of their project Clock.

This more detailed description of problem-solving behavior of a team of subjects illustrates what happened with most goal-directed projects of all the subjects in the study. A global model to reflect the novices' approach to problem-solving in this environment could be represented diagrammatically. Figure 10 illustrates the proposed model.

FIGURE 10

A Model of Problem-solving



The model emphasizes the importance of the generate and test activity before the procedure is defined and afterwards in order to isolate and debug errors. Many errors are eliminated before the procedure is written and one can infer that the students learn a great deal about different strategies for debugging and their efficiency in this activity. It should be noted, however, that this model would only represent the beginner programmer's style on goal-directed projects. As some groups gained proficiency in programming, they were able to work in a different manner. This is illustrated by Group B's work (see Appendix A) using the Editor-as-a-scratch-pad. The no-overt-goal activities of these students are also not represented by this model.

Analysis of Types of Bugs

The following questions will be asked in this section of the analysis:

1) What kinds of bugs do the gifted and the average teams encounter in problem-solving?

2) How many bugs do the gifted and the average teams encounter in problem-solving?

Tables 4 and 5 offer a summary of the total number of bugs per project for the gifted and the average groups. However, this number did not take into account the number of sessions each group worked on a project. This number varied from 15 to 19. In this section and the following sections on types of errors, frequency of debugging strategies, and the relationships between them, only the first fifteen sessions will be included in the analyses. This will control for the possible effect on the frequencies of these variables and for the number of sessions each team worked during the study. Tables

8 and 9 represent the most important types of bugs and their frequency of occurrence in the gifted and the average groups respectively. Idiosyncratic bugs or bugs which did not occur very often were not included. For a complete summary and brief explanation of each bug type, see the key in Appendix A. The tables also include besides the total number of each bug type, the median and the maximum. The median was used as a measure of central tendency since it is not influenced by discrepant scores as is the mean (Ferguson, 1976).

Although it is apparent that there is variation in the number of bugs made by the two groups of subjects, both groups made the same kinds of errors. In other words, no qualitative differences were observed, based on this classification system, between gifted and average groups in the kinds of errors made. Small exceptions are noted as the types of bugs are further explained. Secondly, although it could be hypothesized that the gifted would tend to make fewer bugs overall than the average, the next section will show that this may not necessarily be true. Bugs will be classified for the purposes of this discussion into several categories which will further explain the type of relationship each bug had in terms of the state of the Turtle.

1) Errors that are related to the rotation of the Turtle on the screen.

Angle bugs (ang), right-left orientation bugs (rlo) and starting-orientation bugs (st.or) belong to this group. These errors could be related to the ability to estimate the degree of rotation of the Turtle and the direction of the rotation (Papert, Watt, diSessa & Weir, 1979). All three were common among both groups of subjects. As shown on the tables, the gifted tended to make fewer of these kinds of errors than the average.

TABLE 8

Bug Types and their Frequencies
Gifted (Exp.) Groups

Group	ang	ar.	sh/d	cop	len	l/g	name	ob	pos	rlo	rec	size	st.or	stop	trace
A	20		0	14	37	0	4	2	28	33	2	17	4	12	26
B	12		0	8	29	1	4	3	8	4	2	5	1	12	10
C	21		6	5	20	4	3	4	18	2	1	2	3	5	12
D	25		2	4	28	2	6	3	28	11	0	5	6	6	19
Total	78		8	31	114	7	17	12	82	50	5	29	14	35	67
Md	20.5		1	6.5	28.5	1.5	4	3	23	7.5	1.5	5	3.5	9	15.5
Max	25		6	14	37	4	6	4	28	33	2	17	6	12	26

TABLE 9

Bug Types and their Frequencies
Average (Control) Groups

Group	ang	ar sh/d	cop	len	l/g	name	ob	pos	rlo	rec	size	st.or	stop	trace
a	21	2	15	27	2	6	3	19	7	0	9	7	8	13
b	28	0	3	34	2	3	1	26	8	0	10	8	4	14
c	21	4	6	35	1	3	3	19	10	0	5	0	7	19
d	47	0	16	52	0	2	4	9	22	0	21	10	1	19
Total	117	6	40	148	5	14	11	73	47	0	45	25	20	65
Md	24.5	1	10.5	34.5	1.5	3	3	19	9	0	9.5	7.5	5.5	16.5
Max	47	4	16	52	2	6	4	26	22	0	21	10	8	19

Angle bugs were most frequent with Group d and here we find a discrepancy between the median of the group (24.5) and this group's individual score (47). In examining the medians of the two groups, we still find a difference between the two groups in favor of the gifted.

Right-left orientation bugs also show differences among the teams in each group and between the groups. The average made fewer bugs (47) to the gifted's (50); Group A, in this case, however, made more than the other subjects combined (33). In examining the medians again we see the gifted overall made fewer "rlo" bugs (Md. = 7.5) than the average (Md. = 9).

In the starting-orientation category, the gifted (Md. = 7.5) again made fewer errors than the average (Md. = 9). In summary, although discrepancies between the teams in the two groups are found, the gifted made fewer errors in estimating the degree and the direction of the Turtle's rotation.

2) Errors related to the Turtle's distance on the screen.

Length bugs (len) and REPEAT/input (R/inp) belong to this category. As shown on the tables, the gifted tended to make fewer of this type of errors, (R/inp was not included on the tables because it was judged to be idiosyncratic) in comparison to the average. These types of errors could be related to the ability to estimate distance (Papert, Watt, diSessa & Weir, 1979).

3) Errors related to the Turtle's state in relation to the screen or the drawing on the screen.

Bugs in this category include the out-of-bounds (ob), size, position (pos) and the trace bug. The "ob" bug took place when the Turtle passed the outer (invisible) boundary of the screen. Both the gifted and the average made this error with the same frequency. The gifted made fewer size bugs (Md. = 5) than the average (Md. = 9.5). Size bugs relate to errors on the size of a figure in proportion to the rest of the drawing on the screen. In the case of the "pos" bug, however, the gifted made more than the average. This bug relates to errors made in the Turtle's starting or ending position before or after a figure is drawn. It could also be related to procedure modularity, state transparency, and project complexity. The more complex the visual array on the screen the more likely that this type of bug would occur.

The trace bug dealing with the Turtle's trace state (either up or down) when the Turtle is moved occurred more frequently to the average (Md. = 16.5) than the gifted (Md. = 15.5), although Group A made most of the errors of this type than the rest of the teams. This could be related to the complexity of the projects undertaken by this team (see section on procedural nets).

4) Errors that are related to the concept of recursion and the Turtle in movement.

Errors in this category include the stop classification (Stop/R,M,S), local-global (l/g) bugs and recursion errors (rec). The gifted made overall more "stop" errors (Md. = 9) which were of three kinds: "Stop/R" refer to recursive procedures which continue running because they lack a stop rule. "Stop/S" and "Stop/M" refer to the lack of stopping the SPIN or MOVETURTLE commands respectively. Both groups made the same number of errors in the "l/g" bug. This refers to errors in perspective when the Turtle is in movement. Subjects made this error when they expected the commands to move or spin the whole figure on the screen, while in fact, only the Turtle is affected. Recursion bugs occurred only in the gifted groups. This could relate to the fact that they spent more time on these kinds of procedures than the average.

5) Errors that could be characterized as careless.

Errors in this category include copying (cop), typing (typ), REPEAT/syntax (R/syn), and name bugs. Copying errors are errors in copying instructions from paper or from the text screen when defining a procedure. The gifted made fewer errors (Md. = 6.5) in this category than the average (Md. = 10.5), although Group A made considerably more than the rest of the teams in the gifted group. Typing errors were only identified when they caused the subjects to change episodes or when they were writing a procedure and were not included on the tables. They are included in the description of the protocols in Appendix A. Name bugs refer to confusion of procedure names and occurred with the same frequency for both groups. REPEAT/Syntax (R/syn) refer to errors made in the typing of the REPEAT statement mainly in the opening and closing of the square brackets. Since this task is also fairly idiosyncratic it is not included in the analysis but again is found in Appendix A.

Based on this classification system, we found that the gifted made the same kind of errors as the average. Differences were found in the number of bugs made by the two groups and differences (discrepant scores from the median) between teams were also noted. The gifted made overall fewer errors in estimating the rotation and the distance of the Turtle on the screen than the average. Errors related to procedure modularity, state transparency and project complexity (such as the position bug) were made equally by both groups and in some cases the gifted made more than the average.

Analysis of Debugging Strategies

Tables 10 and 11 contain the gifted and average teams' debugging actions respectively and the frequency of their use. These frequencies do not correspond to bug frequencies since more than one debugging action can occur for any one bug. The following questions will be asked in this section of the analysis:

1) What kind of debugging strategies do the gifted and the average use in problem-solving?

2) Do the gifted use more or less of certain types of debugging strategies than the average?

In general terms, the two groups used the same kinds of strategies to correct their errors, based on the classification system adopted. Several exceptions are worthy of mention. Two teams A and B used the command PENUP (PU) in order to verify the estimate they had on the distance between two points on the screen. For example, if they wanted to check if the input to a forward command would be 50, they first put the Turtle's trace up, drew the line without a trace and examined the results. Then, they rubbed out the last command either by RUBDIS or by going BACK, they put the pen down and then drew the line. Heading (H) was also another command that the gifted used to help them estimate the degree of rotation of the Turtle. This command, however, was not introduced to all the groups (Table 2) and its use reflects this rather than a preference by the subjects.

TABLE 11

Debugging Actions and their Frequencies
Average (Control) Groups

Group	ass	c	cr	e	da	H	ia	na	ns	p	PO	post	PT	rc	rp	rr	S	sp	u	*	PU
a	1	0	15	38	2	0	61	18	4	6	34	8	6	2	1	36	11	0	15	4	0
b	5	2	23	10	1	0	72	6	4	7	29	2	4	9	4	24	6	3	27	11	0
c	10	2	18	16	1	0	68	7	3	6	32	4	9	3	2	34	2	4	19	7	0
d	16	1	30	31	2	0	103	18	4	6	47	3	15	2	3	49	14	1	31	13	0
Total	32	5	86	95	6	0	304	49	15	25	142	17	34	16	10	143	33	8	92	35	0
Md	7.5	1.5	20.5	23.5	1.5	0	70	12.5	4	6	33	3.5	7.5	2.5	2.5	35	8.5	2	23	9	0
Max	16	2	30	38	2	0	103	18	4	7	47	8	15	9	4	49	14	4	31	13	0

TABLE 10

Debugging Actions and their Frequencies
Gifted (Exp.) Groups

Group	ass	c	cr	e	da	H	ia	na	ns	p	PO	post	PT	rc	rp	rr	S	sp	u	*	PU
A	37	0	21	40	9	0	78	20	3	11	57	2	4	2	2	23	20	1	41	5	1
B	13	0	13	16	0	1	27	8	4	6	45	2	2	4	0	22	4	0	15	0	2
C	5	0	15	28	0	5	37	11	6	7	58	4	83	0	1	26	4	1	9	6	0
D	5	2	26	27	2	1	53	6	7	6	31	4	10	6	0	36	8	0	6	3	0
Total	60	2	75	111	11	7	195	45	20	30	191	12	24	12	3	107	36	2	71	14	3
Md	9	0	18	27.5	1	1	45	9.5	5	6.5	51	3	6	3	0.5	24.5	6	0.5	12	4	1
Max	37	2	26	28	9	5	78	20	7	11	58	4	10	6	2	36	20	1	41	6	2

As shown in the frequencies of debugging strategies and their use for the two groups, Tables 10 and 11, individual differences are noted between and within the teams in each group. The gifted teams made most frequent use of print-out (PO), incremental adjustment (ia), edit (e) and rubout-retry (rr) in that order. The average on the other hand made most frequent use of "ia", "rr", "po", "e", undo (u) and clearscreen-restart (cr) in that order.

Incremental adjustment refers to using small increments in order to modify an error in the input of a command. Although the gifted teams made frequent use of this strategy (Md. = 45), Group A shows a discrepant score in this area (78). In the average teams (Md. = 70), Group d made the most frequent use of this strategy. The average made more use of "ia"; this may be related to the estimation of length or angle rotations. It is worth mentioning that this debugging action took place almost invariably in the g&t activity. This supports the Chait findings in that the subject relies on visual feedback from the screen to see if the bug was eliminated before the actual procedure was written. This will be further analyzed in the following section on the relationship between bugs and their debugging strategies.

Print-out refers to printing out a procedure or procedure titles in order to examine either the text of an already defined procedure or the titles of the procedures in the workspace. The gifted teams made most frequent use of this strategy than any other and this could be related to the fact that it tended to facilitate the location of an error after a procedure is written. The gifted had overall more revisions to their procedures than the average. However, "PO" was used as a matter of course even when errors were not present in order to add to an already defined program. (See Appendix A for when "PO" was used and in what context.)

The gifted teams made more frequent use of edit (Md. = 27.5) than the average (Md. = 23.5). This strategy refers to the revision of a procedure in order to correct an error. This could be related to the fact that the gifted wrote more procedures (project complexity) and had ultimately more revisions to make. Again, Group A made most frequent use (40) than the other teams. In the average teams, edit was used with most frequency by Group d. This team, however, worked on the least complex projects of the teams in the study. Edit, then, may not necessarily be an index of project complexity but also of project difficulty.

Rubout-retry (rr) was used more frequently by the average (Md. = 35) than the gifted (Md. = 24.5). It refers to the use of the RUBDIS command which allows one to erase the last command from the graphic display. It took place most frequently in the g&t activity.

Undo (u) refers to the elimination of an error by reversing a command. For example, if an error is made in the direction of the Turtle and the subjects turned it right instead of left, in this case, they would use the left command to debug. The average used this action more often (Md. = 23) than the gifted (Md. = 12), although Group A has an extremely discrepant high score in this category.

Several other actions will be discussed here since they occurred somewhat frequently to both groups. A summary of all the debugging actions and their explanation is found in the key to Appendix A. Clearscreen-restart (cr) refers to the action of clearing the graphic screen completely in order to eliminate a bug. The average teams made more use of this strategy (Md. = 20.5) than the gifted (Md. = 18). Group d made the most use of this action.

Not aware of a bug (na) is not necessarily a debugging action but is characteristic of a possible response to a bug. The average (Md. = 12.5) were more often unaware of bugs, especially when they were defining a procedure than the gifted (Md. = 9.5). This response generally took place when a copying bug was being made in the procedure writing mode.

Playing Turtle (PT) refers to playing turtle physically or with paper and pencil in order to understand where a bug occurs. Both teams made less use of this action in comparison to others.

In summary, both the gifted and the average used the same kinds of strategies but differences were observed in the frequency of occurrence of actions between the gifted and the average groups and within the teams in each group. The average groups made more frequent use of debugging actions that generally occurred in g&t activity ("ia", "rr", "u", "cr") while the gifted used actions that generally were used in the edit mode ("PO", "e").

Analysis of the Relationship between Bugs and Debugging Actions

After describing the differences between the occurrence of specific types of bugs and debugging strategies in terms of two groups of subjects, the current analyses will attempt to answer the following questions:

1) Do the gifted teams use different strategies than the average to debug certain bugs?

2) Which strategies do the gifted and the average teams find most useful to debug certain bugs?

The first analysis will describe the debugging strategies of the seven most

TABLE 12

Angle Bug

Group	ass	cr	e	da	ia	na	ns	p	PO	PT	rc	rp	rr	S	sp	u	*	H	PU	post	u,ia	ia,u	rr,ia
A		1	3		6					1				3		6	1				1	2	
B	2		1		5	1			2				2			3		1			1		
C			2		8		1	1	3	2			13		1	6		3					1
D		6	1		9		1	1		3			8	1		4		1			1		5
Total	2	7	7	0	28	1	2	2	5	6	0	0	23	4	1	19	1	5	0	0	3	2	6
a			3		11		1			5			2			1	2						2
b		2			20				1	3		1	3			2	5					5	
c			1		10					7			1			6	1	2				3	3
d		1	8		20	2	1	1	3	3	1			2		10	4				1	1	6
Total	0	3	12	0	61	2	2	1	4	18	1	1	6	2	0	19	12	2	0	0	1	9	11

frequently occurring bugs: in alphabetical order, these are angle, length, position, right-left orientation, size, starting orientation and trace. Tables 12 through 18 represent each bug and the frequency of the type of strategy each team and each group used to correct it. Several strategies were used in pairs and are included as separate headings. These are undo, incremental adjustment (u,ia), incremental adjustment, undo (ia,u), and rubout-retry, incremental adjustment (rr,ia). The asterisk denotes the direct intervention by the Logo teacher but not on specific strategies.

Angle bugs (Table 12) were most often debugged by incremental adjustment by both the gifted and the average. This finding supports the Chait results in this area. As previously described, it was used most frequently in the g&t activity. The action proved useful when the angle estimate made by the subjects was too small for the Turtle's heading. They incremented the input until the Turtle looked right. For example, if the goal was RT 90 and the subjects wrote RT 45, they adjusted the input until it was 90.

```
RT 45
RT 40
RT 5
```

The average used this action more frequently than the gifted groups perhaps due to the fact that first they made more angle bugs and secondly because they used smaller increments when they corrected their rotation.

Some groups made errors in the increments and corrected them by reversing the action. This is shown under the heading "ia,u" on the Table. The average made more of those kinds of errors than the gifted and used these dual actions to correct them.

Undo was also used by both groups to debug an angle bug. In most

TABLE 13

Length Bug

Group	ass	cr	e	da	ia	na	ns	p	PO	PT	rc	rp	rr	S	sp	u	*	H	PU	post	u,ia	ia,u	rr,ia
	Debugging Strategies																						
A		1	1	1	31								4			15						2	
B	1	1	1		20			2	6				6			7			2				
C		3			19			2	2							2		1					
D		1	2	2	32		1		5				1			5							
Total	1	6	4	3	102	0	1	4	13	0	0	0	11	0	0	29	0	1	2	0	0	2	0
a			3		43				2				13			3				1			
b		2			31		2	1	4				4			6	2					2	
c					45				3				3			4	2						
d	1	2	4		48	1			1				2			5	1			1		1	
Total	1	4	7	0	167	1	2	1	10	0	0	0	22	0	0	18	5	0	0	2	0	3	0

cases, the estimate was too large and it was corrected by reversing the action of the Turtle. For example, if the goal was RT 90, the subjects probably wrote RT 100 and then they used LT 10 to correct their error. Both groups used it with the same frequency although Group d used it more often than the other teams.

The heading "u,ia" shows the errors made in the reversing of the Turtle's action. Rubout-retry (rr) was used more often by the gifted, especially Group C, than the average. It also took place mostly in g&t activity. Some groups used "rr,ia" together to correct an angle rotation that was too small.

Play Turtle and edit as actions to correct an angle bug were used more frequently by the average than the gifted.

In general terms, although several trends are indicated the two teams used similar strategies to correct angle bugs. The Chait analysis found "ia", "cr", and "rr" as useful strategies for debugging angle bugs. "Cr" was not observed to be used frequently by these subjects.

The length bug was overwhelmingly debugged by "ia" (Table 13) by both groups. This trend may indicate that it took place more often in g&t activity. Similarly to the angle bug, this action shows that the majority of length bugs were short of the estimated distance and were corrected by adding increments to the input.

Undo was also used by both groups probably to correct a large estimate by reversing or substracting from the input. The gifted made more use of this strategy especially Group A.

Rubout-retry was used more frequently by the average groups especially by Group a.

TABLE 14

Position Bug

Group	Debugging Strategies																					
	ass	cr	e	da	ia	na	ns	p	PO	PT	rc	rp	rr	S	sp	u	*	H	PU	post	u,ia	ia,u	rr,ia
A		5	6		12	1	2	5	4	1		1	3	5			2			1			
B		2						2			1		5							1			
C	2	4	11			1	3		1	1			3	1						1			
D	2	3	11				2	5	6	3			8	2			1			2			
Total	4	14	28	0	12	2	7	12	11	5	1	1	19	8	0	0	3	0	0	5	0	0	0
a		2	5			2		3					5				2			3			
b	4	6	2		2		1	3	4	1	1	2	11	1		1	2			1			
c	1	7	2				2	2	4	1			5		1		3			2			
d		2	1					1	3	1		2	3	1			2						
Total	5	17	10	0	2	2	3	9	11	3	1	4	24	2	1	1	9	0	0	6	0	0	0

TABLE 15

Right-Left Orientation Bug

Group	ass	cr	e	da	ia	na	ns	p	PO	PT	rc	rp	rr	S	sp	u	*	H	PU	post	u,ia	ia,u	rr,ia
A			2		18					2						9					1		
B							1			1						4							
C					1																		
D		1	2		1					2				4		4							
Total	0	1	4	0	20	0	1	0	0	5	0	0	0	4	0	17	0	0	0	0	1	0	0
a													1			4						1	
b					2					2		1	2			2					4		
c		1	1		2					1			4								4		
d					5					9			4			1	1				9		
Total	0	1	1	0	9	0	0	0	0	12	0	1	11	0	0	5	1	0	0	0	17	1	0

In general terms, the trends indicate that both groups used similar strategies to correct a length bug. The Chait results found "ia", "cr", "u", and "rr" as important ones in this area. "Cr" was not found to be useful to the subjects in this study.

The position bug (Table 14) was the first bug where no real trends are indicated. Looking at the overall table, one sees that the subjects tried many more different strategies to correct this error than the other tables where trends seem more definite. Similarly, while the other two tables indicate bugs that generally occur in the g&t activity, the position bug shows itself to be more complicated and occurred in both g&t and edit.

Edit was used most frequently by the gifted groups to correct this error, indicating either that the position bug was discovered after the procedure was written, or debugging in g&t was not successful. The average groups used "rr" more frequently. They also used "cr"; both actions took place more in the g&t activity.

Table 15 shows the debugging strategies most commonly used to debug the right-left orientation bug. The gifted groups, in particular Group A, used "ia". This is an unusual strategy. If the goal is LT 90 and the subjects typed in RT 90 instead, they corrected it by incrementing the input by 180:

```
RT 90
RT 180
```

RT 270 is the same as LT 90 and the Turtle's heading is now facing in the right direction. Group A used this unusual way more often than the other groups. This finding is in line with the anecdotal description of the work of a gifted boy in the Brookline Project (Papert, Watt, diSessa & Weir, 1979).

TABLE 16

Size Bug

Group	ass	cr	e	da	ia	na	ns	p	PO	PT	rc	rp	rr	S	sp	u	*	H	PU	post	u,ia	ia,u	rr,ia
	Debugging Strategies																						
A		8					1	2	1		1		9						1				
B							1	1			2												
C			1						2														
D		2	1				1																
Total	0	10	2	0	0	0	3	3	3	0	3	0	9	0	0	0	0	0	1	0	0	0	0
a			2				1	1	2		1	1	4							1			
b		1						2			6		1		1								
c		2	2						2				6	1									
d		11		2			3		1	2			19										
Total	0	14	4	2	0	0	4	3	5	2	7	1	30	1	1	0	0	0	0	1	0	0	0

TABLE 17

Starting Orientation Bug

Group	ass	cr	e	da	ia	na	ns	p	PO	PT	rc	rp	rr	S	sp	u	*	H	PU	post	u,ia	ia,u	rr,ia
A	1	1											1			2							
B			1																				
C		1			2	1	1	1	1														
D		4	1		3		2		1				4			1							
Total	1	6	2	0	5	1	3	1	2	0	0	0	5	0	0	3	0	0	0	0	0	0	0
a		6											1			3							
b		4															4						
c																							
d		5					1			2			2			1	2						
Total	0	15	0	0	0	0	1	0	0	2	0	0	3	0	0	4	6	0	0	0	0	0	0

TABLE 18

Trace Bug

Group	ass	cr	e	da	ia	na	ns	p	PO	PT	rc	rp	rr	S	sp	u	*	H	PU	post	u,ia	ia,u	rr,ia
A	7	1	7			3		1	5		5		1	1		2	1						
B		2									1		8			1							
C		1	3							1	3		2				2						
D	3		1			1	1		1		5		8			1	1						
Total	10	4	11	0	0	4	1	1	6	1	14	0	19	1	0	4	4	0	0	0	0	0	0
a		1	4			2	1	1	1		4		1			2							
b		2									6		2		1	2							
c	1	3	1					4	1		2	1	8			3							
d	3	3	2			2		1	1		3		8			1							
Total	4	9	7	0	0	4	1	6	3	0	15	1	19	0	1	8	0	0	0	0	0	0	0

Debugging Strategies

This boy in correcting a "rlo" bug followed the same strategy of adding to the input the same way this group of boys did. The average groups used a more routine way of correcting the "rlo" bug. They reversed the command (u) and either incrementally increased the input (ia) or did it in one step (u).

The gifted used undo (u) in one step to correct the "rlo" bug more often than the average. If the goal was LT 90 and they typed in RT 90 instead, they immediately corrected it by LT 180. The average teams corrected the same bug more often in two steps. The first undid the previous command and the second incrementally added to the input:

RT 90
LT 90
LT 90

The Chait analysis in this area showed that the subjects had a strong preference for "u", and "rc" to debug the "rlo" bug. "Rc" was not used by these teams at all to correct this error.

The size bug (Table 16) was corrected mainly by "rr", and "cr". The average teams especially Group d made more use of these actions than the gifted. They generally took place in the g&t activity. The Chait results showed that those subjects used edit more than any other action to correct a size bug. In this case, then, the action must have taken place after the procedure was written.

The starting-orientation bug (Table 17) was more often debugged by "cr" by both groups and this finding is supported by the Chait analysis.

The trace bug (Table 18) was corrected by both groups using "rr", and "rc". The gifted teams, Group A especially, assimilated (ass) this bug more than the other teams. In other words, they were aware of the bug but incorporated it in the instructions. The Chait subjects used "cr", "e", and "u" more often than the other actions to correct a trace bug.

Team and group differences are found in the frequency of occurrence of specific actions to correct certain bugs. The gifted students used different strategies in some tasks ("rlo", "ang", "len", and "pos") while in others (trace, "st. or.", and size) they followed the general trend with the average groups. However, it is difficult to determine from these tables whether a high frequency of a certain debugging strategy is due to a tendency for this strategy to be used often (to be useful) or simply due to a tendency for a bug to occur more frequently.

The second analysis of this section will examine the relationship between bugs and debugging actions further in order to determine the usefulness of certain actions for specific bugs irrespective of the frequencies of the two variables. Mosteller and Tukey developed a method, the PLUS analysis, of interpreting the structure of outliers, that is, unusual permutations in descriptive data (Mosteller & Tukey, 1977; Chait, 1978). The PLUS analysis proved useful in the Chait study for isolating bug-specific debugging actions which were not mere "artifacts of bug frequency" (Chait, 1978, p. 80). In the present study, the analysis will similarly examine debugging actions for specific bugs in order to determine which were found useful by the gifted and average teams.

The model followed is explained as the two factor case-Two way PLUS analysis (Mosteller & Tukey, 1977). Here, the two factors are bugs (row effect) and debugging actions (column effect).

Table 19 provides a summary of the medians for the seven most frequently occurring bugs (described in detail above) and the more common debugging actions for the gifted groups. Table 20 provides the same summary for the average groups. In these tables, both rows (the bug frequency) and columns (the debugging action) are ordered according to their median values (Mosteller & Tukey, 1977).

TABLE 19

Medians of Common Bugs and Debugging Actions

Gifted Groups

Bug	Debugging Action											Md
	u,ia	rc	S	PT	p	u	E	PO	cr	rr	ia	
rlo	1	0	4	5	0	17	4	0	1	0	20	1
size	0	3	0	0	3	0	2	3	10	9	0	2
st.or	0	0	0	0	1	3	2	2	6	5	5	2
trace	0	14	1	1	1	4	11	6	4	19	0	4
len	0	0	0	0	4	29	4	13	6	11	102	4
ang	3	0	4	6	2	19	7	5	7	23	28	6
pos	0	1	8	5	12	0	28	11	14	19	12	11
Md	0	0	1	1	2	4	4	5	6	11	12	

TABLE 20

Medians of Common Bug Types and Debugging Actions

Average Groups

Bug	Debugging Action											Md
	u,ia	S	rc	P	PT	ia	PO	u	E	cr	rr	
st.or	0	0	0	0	2	0	0	4	0	15	3	0
rlo	17	0	1	0	12	9	0	5	1	1	11	1
size	0	1	7	3	2	0	5	0	4	14	30	3
pos	0	2	1	9	3	2	11	1	10	17	24	3
ang	1	2	1	1	18	61	4	19	12	3	6	4
len	0	0	0	1	0	167	10	18	7	4	22	4
trace	0	0	15	6	0	0	3	8	7	9	19	6
Md	0	0	1	1	2	2	4	5	7	9	19	

Using the data in Tables 19 for the gifted groups and 20 for the average, the value in each cell can be partitioned into the following contributions:

data = common PLUS row PLUS column PLUS residual

where "data" is the value of the observation in each cell, "common" stands for a value applied to every cell, "row" stands for a value which depends upon the row effect, "column" stands for a value which depends on the column's effect and "residual" stands for the value of the outlier. In this case, the medians of both bugs and debugging actions are each in turn subtracted from the actual observations. The sequence is as follows:

1) The row medians (bugs) are subtracted from each row observation resulting

in row residuals;

2) New column medians are computed from these row residuals;

3) Column residuals are computed by subtracting each row residual from the new column medians.

In fact, what this analysis does is remove two constants from the data and thus eliminates systematically each constant's effect. What is left then is a residual which should represent the debugging action for each bug. Tables 21 and 22 present a summary of the pattern of residuals thus computed. In order to make the tables more comprehensible at a glance, the residuals are rounded off at 10 point intervals in this way:

-20 to -30 = -30	+15 to +25 = +20
-10 to -20 = -20	+25 to +35 = +30
below -5 to -10 = -10	+35 to +45 = +40
-5 to +5 =0	+45 to +55 = +50
+5 to +15 = +10	+55 and up = +60

We can assume then that a debugging action of -20 would be extremely useless for a specific bug while on the other hand +60 would be extremely useful.

TABLE 21

Summary of Pattern of Residuals

Gifted Groups

Bug	Debugging Action										
	u,ia	rc	S	PT	p	u	e	PO	cr	rr	ia
rlo	0	0	+10	+10	0	+20	0	0	0	-10	+20
size	0	0	0	0	0	0	0	0	+10	0	0
st.or.	0	0	0	0	0	0	0	0	0	0	0
trace	0	+10	0	0	0	0	+10	0	0	+10	-10
len	0	0	0	0	0	+20	0	+10	0	0	+60
ang	0	0	0	0	0	+10	0	0	0	+10	+20
pos	-10	-10	0	0	0	-20	+20	0	0	0	0

TABLE 22

Summary of Pattern of Residuals

Average Groups

Bug	Debugging Action										
	u,ia	S	rc	p	PT	ia	PO	u	e	cr	rr
st.or.	0	0	0	0	0	0	0	0	0	+10	-20
rlo	+20	0	0	0	+10	+10	0	0	0	0	0
size	0	0	0	0	0	0	0	-10	0	+10	+10
pos	0	0	0	+10	0	0	+10	-10	+10	+10	+10
ang	0	0	0	0	+10	+60	0	+10	+10	0	-20
len	0	0	0	0	0	+60	+10	+10	0	0	+10
trace	0	0	+10	0	-10	-10	0	0	0	0	0

Interpreting these tables (21 and 22) further, and placing the debugging actions for the common bugs on a continuum from very useless (-20 and less), useless (-10), useful (+10) and very useful (+20 and up) for each group, Tables 23 and 24 represent a graphic representation of the gifted and average groups' actions for each bug.

TABLE 23

Summary of Very Useless, Useless, Useful and Very Useful Debugging Actions Gifted Groups

Bug	Debugging Action			
	Very Useless	Useless	Useful	Very Useful
rlo		rr	S, PT	u, ia
size			cr	
st.or.				
trace		ia	rc, e, rr	
len			PO	ia, u
ang			u, rr	ia
pos	u	u&ia rc		e

TABLE 24

Summary of Very Useless, Useless, Useful and Very Useful Debugging Actions Average groups

Bug	Debugging Action			
	Very Useless	Useless	Useful	Very Useful
st.or	rr		cr	
rlo			PT, ia	u&ia
size		u	cr, rr	
pos		u	p, PO, e,cr,rr	
ang	rr		PT, u, e	ia
len			PO, u, rr	ia
trace		PT, ia	rc	

The results in Table 23 show that the gifted found "ia" to be very useful to correct the "rlo", "len", and "ang" bugs but useless for the trace bug. "E" was found to be most useful to correct the "pos" bug and useful to correct the trace bug. "U" was very useful for the "rlo" and "len" bugs and very useless for the "pos" bug. These results are consistent with the descriptive section on errors and debugging strategies.

The following actions are isolated as useful in the PLUS analysis for the gifted groups, once the effect of bug frequency is removed from the data. "S" and "PT" were found to be useful to correct the "len" bug. No action was isolated for the "st.or." bug and inspection of Table 17 shows that the bug occurred infrequently to the gifted teams. This may be the reason why it did not show up in the PLUS analysis.

The results in Table 24 shed more light into the relationship between bugs and their debugging actions for the average teams. The average found "ia" to be very useful to correct the "len" and "ang" bugs and useless to correct the trace bug. The joint action of "u,ia" was found very useful for the correction of the "rlo" bug. Five actions were found useful by the average to correct the "pos" bug. "Cr" was found useful to correct the "st.or." and the size bug. These results are also consistent with the descriptive section on the relationship between bugs and their respective actions.

The PLUS analysis isolated several actions which proved useful for the average groups that the descriptive section did not. "PO" and "u" were found useful for the "len" bug and "p", "PO" , "e", "cr" and "rr" were useful for the "pos" bug.

In comparing Table 23 and Table 24, one can see several differences in the pattern of usage between the two groups. There seems to be greater variability in the actions that the average teams found useful in contrast to the choices made by the gifted teams. One example, is the "pos" bug. The average (Table 24) found five actions useful in this task: "p", "PO", "E", "cr" and "rr" and none very useful. The gifted groups (Table 23), on the other hand, found "e" very useful. Further examination of Table 24 shows that the average found only three debugging actions very useful: "ia" for the "len" and "ang" bugs and the dual action of "u,ia" for the "rlo" bug. The gifted teams as a whole were more consistent in their use of certain debugging actions. They found six actions very useful: "u", "ia" for the "rlo" and the "len" bugs, "ia" for the "ang" bugs and "e" for the "pos" bugs.

In summary, the PLUS analysis brought into focus the relationships between bugs and debugging actions for the gifted and the average groups. Similarities and differences in the use of certain actions described in an anecdotal fashion are further refined with the PLUS analysis.

Summary of Findings

1. In terms of project complexity, the gifted teams completed more complex projects than the average teams. They wrote more procedures, subsequently made more revisions, and their projects had a deeper hierarchical structure in terms of tree depth. Both the gifted and the average made the same number of errors. It could be speculated, then, that the average teams made more errors in the g&t activity than the gifted, while the latter had to correct more errors after the procedure was written.
2. No differences were observed between the gifted and the average teams in the kinds of errors made. The gifted groups made fewer errors in estimating the rotation of the Turtle and the direction of the rotation than the average groups. The gifted groups also made fewer errors related to the ability to estimate distance than the average groups. Errors related to procedure modularity, state transparency and project complexity were made equally by both groups and in some cases the gifted made more.
3. Both groups used the same kinds of debugging strategies to correct their errors. The gifted teams used more actions to correct their errors after the procedure was written, while the average made more errors in the g&t activity.
4. In terms of the relationship between bugs and debugging actions, the descriptive section revealed several differences. The gifted teams used

different strategies in some tasks, while in others they used similar strategies to the average groups. The average teams tended to use smaller increments in their inputs when they were correcting errors. Subsequently, they used overall more actions to correct a specific bug than the gifted. The PLUS analysis supported the majority of the observations made in the descriptive sections. Examination of the overall pattern of usage shows that the gifted tended to be more consistent as a group than the average teams in their use of strategies.

CHAPTER V

Discussion

This study has revised and used a classification system of protocol analysis in order to examine differences in problem-solving in a computer-based task environment. The study's purpose was two-fold. First, it was to demonstrate that the classification system is a viable and flexible tool that can describe meaningfully many of the learning experiences of students in Turtle geometry. Second, it was to explore differences in problem-solving between two varying ability groups and examine any trends. This chapter will address several issues. First, it will describe both the strengths and shortcomings of the classification system. Second, it will discuss the proposed problem-solving model in terms of its implications for education. Finally, the limitations of the study will be described and suggestions for further research given.

Evaluation of the Classification System

The classification system proved to be a powerful tool in analyzing meaningfully the protocols of the subjects. The analyses were performed at different levels. Appendix A represents the first level of the analysis; the students' work is summarized in a clear and detailed fashion. The method of transcribing the protocols developed by Chait (1978) was strengthened for the present study by the addition of tape recordings of each session, and the cross-checking of the division of the data into episodes by two experimenters. At this level, the study reinforced areas of strength in the previous study and revised some areas of weakness.

The addition of the "test" activity strengthened the sequential division of each team's protocols into episodes. In the previous study, the categories did not clearly indicate where one episode ended, although the starting point was more explicit (Chait, 1978). In the present study, the "running of the procedure to see if it worked" identified with more precision the end of one "goal" state and the subsequent beginning of another. This facilitated the development of the problem-solving model which stopped short of defining the "goal" states. Further work using the same data might involve a "goal" analysis of each team's work.

The next area that was revised and strengthened was the "evaluation" of each episode structure. The previous study based the evaluation of each episode on the inferred behavior of the students. In this study, the evaluation was based on observed behavior, making this category consistent with the others in the classification system. The activity was related to the results of the "test" classification. A positive evaluation meant that no errors were made by the students between the intended "goal" and the actual outcome of the graphic output. A negative sign meant that an error was made and the students had to revise their program to correct it.

The final major revision of the classification system was the extension of the procedural nets to give a more detailed representation of the actual program structures. In the previous study, horizontal and vertical differences in programming style were described. The revisions made to the procedural nets in the present study allowed for a description of the programming styles of the students in terms of top-down, bottom-up or intermediate, as well as the styles described by Chait. The observation of an intermediate style of programming which incorporates some aspects of top-down and bottom-up may represent a style which facilitated the programming of more complex projects for this group of students. Simple projects were easily programmed following a clearly bottom-up approach, which other studies have shown beginner programmers tend to favor (cf. Papert, Watt, diSessa & Weir, 1979). Further research between expert and novice approaches to programming may shed more light into this mixed style of programming.

The problem-solving model defined in the present study was developed to best describe the general problem-solving steps undertaken by the subjects within the constraints of the teaching environment adopted. It is powerful in several respects. First, it describes in general terms, the process that these students followed in problem-solving. It represents the experience as it was observed, "bottom-up". Other models proposed to prescribe or predict the performance of students (Polya, 1957; Miller, 1982) have been developed from the "top-down". Secondly, it replicates the existence of the generate and test (g&t) activity, first described by Chait (1978), and demonstrates its importance in the process of solving a problem. This activity resulting from the "direct command" feature of the Logo language seems to be an important area where a great deal of problem-solving takes place, before a procedure is defined, and afterwards in order to locate an error.

Papert (1978) describes an activity called "semi-programming", characterized by the students giving step by step direct commands and receiving immediate feedback from the graphic output. This may encourage the correction of errors by the students. Miller (1982) describes a "Localization Mode" whose function is to decide which part of the program is responsible for the error, once the "Tryout Mode" (in this study the "test" activity) discovered a bug. This mode could compare to the g&t activity but only after the procedure is written. Miller's model does not allow for generate and testing before the procedure is written. Semi-programming, Brown and Rubinstein 's (1978) bottom-up debugging, Polya's (1957) second or "planning" stage and Miller's (1982) "Localization Mode" may all be related to the generate and test activity.

It was suggested by Chait (1978) that this activity may be unique to beginning programmers and would decrease as one became more expert in the use of the language. The present study did not work with the students for a long enough period of time in order to see any development in that direction. However, towards the end of the study, one team of students started working in a slightly different manner, described in Appendix A p. 237, as using the "Editor-as-a-scratch-pad". They used the unique feature of this system's line editor of numbering each line to define a procedure or not (by not numbering the line) and were able to "simultaneously" work in "direct command" and procedure writing. Although this was not confirmed, this kind of activity could represent an intermediate stage of programming expertise, where the student may not use g&t before the definition of the procedure. It could be speculated then that the more expert student would use g&t more in order to correct an error that may be difficult to isolate than in the way it was used in the present study. Further research in a developmental fashion comparing expert and novice programmers is needed to examine this issue.

The "con" activity could be related to Phase I defined by Miller (1982) as understanding the problem. Although Miller's model of elementary programming in Turtle geometry does not address directly this phase, he explains it as "converting the pictorial representation into some internal symbolic representation" (Miller, 1982, p. 121). In this classification system, this category does not clearly define the activities that took place. These involved discussion among the peers and between the peers and the teacher, consensus reaching, verbalization and in general a great deal of problem-solving and systematic planning before the project was attempted. This area needs further clarification especially in this context when pairs of students collaborated in the project work. Many interesting interactions take place here of interest to educators and researchers alike. Burnett (1981) developed a system of diagramming the use of language in the analysis of problem-solving in a specific task environment (mathematics). These diagrams called covlagrams (chart of overt language), in conjunction with a classification system of functional language usage, give a detailed description of the dialogues between students and their teacher in terms of solving a problem. More detailed analysis of the tape transcriptions of the sessions using a system like Burnett's (1981) to examine the verbalization activity of the students may yield more insight into the differences in problem-solving between gifted and average in terms of pre-planning, cooperative learning and amount and type of verbalization activities.

The present classification system examined the activities of the students from a molar level of analysis (Chait, 1978). Another area of the classification, besides the "con" activity, may need a more refined analysis in order to provide more insight into the specific processes that take place.

This is the no-overt-goal category (nog). The students worked on open-ended explorations of various concepts (such as polygons), new concepts were introduced by the teacher, and they reviewed previous projects or concepts in this activity. A finer grained analysis is needed to describe these activities, perhaps related to Newell and Simon's method (1972) of a tighter framework of episodes of shorter duration.

Implications For Education

Howe (1978) describes three stages of learning to program. This study did not observe these students to progress through similar stages. The problem discussed in Chapter II is whether Howe saw the stages first or they reflect on his teaching methodology. Similarly, the same concerns may be raised about the problem-solving model. The activities it describes are in a way closely related to the teaching methodology adopted. The students were encouraged to plan their project work before they carried out the actual programming and this is reflected by the "con" activity. They were then encouraged to work on their projects in a specific manner: dividing them into smaller chunks, writing subprocedures and superprocedures so that debugging may be more manageable. The problem-solving model represents to a certain extent then the process of problem-solving within the constraints of the teaching approach adopted. It does, in a sense, mirror the amount of structure placed in the learning environment by the teacher and the two students.

The present approach may depart slightly from the free problem-solving model, described by Papert in Mindstorms. It is difficult to determine from Papert's description of the teaching approach what amount of structure may be optimal for different levels of development. Recent research (Bank Street, 1981-82; Krasnor & Mitterer, 1983) is posing critical restraints to the concept of the development of general heuristics in the process of learning Turtle geometry. The issues are complex: Are "powerful ideas" learned by students without any specific guidance? Do these ideas or heuristics transfer "automatically" to other areas of problem-solving? Is Papert's methodology of how Turtle geometry should be taught feasible and successful for all age and ability levels, and stages of development?

Although this study did not address directly these issues, the problem-solving model shows clearly that a specific amount of guidance was offered to the subjects. This guidance proved successful in allowing the students to first develop their own projects and then, to solve problems within them. In this sense, the students were working in a free problem-solving environment. However, the model also shows that the students were guided initially to plan and chunk their ideas so that success may be facilitated. In these phases, structure was placed in the environment by the teachers who ultimately had to use their best judgement as to when to interfere and when to allow the students to work independently. In conclusion, the problem-solving model gives specific guidance to the teacher of problem-solving in this specific task environment and for these groups of subjects. Whether the model also represents the activities of students in general areas of problem-solving, can only be answered by examining the activities of students in different task environments where procedural thinking is emphasized. More research is needed to answer critically these complex issues. Indirectly, the model shows that the role of the teacher is again of the greatest importance (Watt, 1982) and that teacher training is a critical factor in the success of Logo in the school setting.

Limitations of the Study

In terms of methodology and research design, the current study had several shortcomings. The most striking is the fact that the small number of teams per group did not allow for meaningful statistical interpretation of the described differences in a quantitative fashion. The identification procedures used were also not as complete as possible. Studies in the literature of giftedness (cf. Martinson, 1977), have emphasized the administration of individual IQ tests in the final stages of the identification process. However, due to time constraints the formal identification of the two populations in this study, was limited to the administration of a group IQ test and the results may have been influenced by this form of testing. Another possible influence may be the fact that the subjects were instructed in pairs. A replication of the study with single subjects may yield quite different results. Another variable that the study did not control for was the effect of the two different teachers in the learning environment. Although the two teachers conducted a pilot before the study began in order to develop a more consistent teaching approach, a random assignment of teacher to team might have controlled for any possible effect. The teaching schedule, however, did not allow for this randomization. Finally, the students were instructed the same times each day of the week. This was determined by a schedule that was organized with the help of the classroom teachers, so that the students would not miss any of the core subjects, such as mathematics, reading and french. Subsequently, some students would come to the "computer lab" at the latter part of the school day every week.

Sometimes, they were tired and restless. The study did not control for the effect of this variable - time of day - on the teaching as well as the learning environment. An optimal design would have randomized the hours the subjects spent using the computer in order to control for this influence.

The study did not address several areas of interest. Although sex differences were observed between boys and girls in terms of programming style and problem-solving strategies, due to the small number of teams, no meaningful comparisons were possible. Similarly, some students expressed difficulty working with a partner and a preference for working alone was observed by some of the boys who took part in the study. The girls, on the other hand, enjoyed the sharing of ideas involved in the learning experience. An examination of the learning preferences between boys and girls in this type of cooperative learning environment might offer insight into the process of learning in teams. Due to the relative lack of computers in the schools at the present time, more and more students are asked to work in pairs or even in larger groups to program in Logo or other languages (Cartwright, 1973). Further studies in this area may reveal more about the learning process in these kinds of contexts.

Possible analysis of the same data could investigate differences in debugging efficiency in terms of when (in what episode) errors were made and in what episode they were eliminated. Comparisons could be drawn within the teams to see if efficiency improved from the beginning to the end of the study. Differences between the gifted and non-gifted in the amount of teacher intervention could also lead to interesting results. Using the same data, an examination of what strategies were most often suggested by the teacher could lead to an evaluation of which actions and bugs were most difficult in this task environment and for these groups of subjects.

Summary

The main purpose of the study was to examine differences between gifted and non-gifted students in terms of errors and concomitant coping strategies. These two variables were examined in an exploratory fashion both qualitatively and quantitatively (PLUS analysis). The results are inconclusive, however, and a replication of this study is needed with a larger number of subjects before any conclusions can be drawn. As important as the findings across groups, this study showed that there were a large number of individual differences between the teams in each group. The fact that the Logo environment enhances individual differences in programming and cognitive styles has been demonstrated by previous research (Watt, 1979). This suggests that it may not be useful to obtain quantitative data that are averaged among subjects, since this would introduce error from the variations over subjects. This in turn may obscure wide individual differences in learning patterns and styles. This study has attempted to produce a balance between quantitative and qualitative statistics in the use of medians; furthermore, Mosteller and Tukey's PLUS analysis or a method analogous to it tends to isolate individual differences so important in examining the process of problem-solving rather than the actual product.

Since this study replicated to a certain extent the Chait research a comparison between the two sets of results is natural. In general terms, the Chait subjects made fewer errors, considerably fewer than the subjects in the present study. This could be related to the age of the subjects, since the Chait students were approximately two years older than the students in the present study. The project work undertaken by these students was considerably more complicated than the projects of the Chait study. The level of complexity, the fact that these subjects worked in teams, and the teaching method used may all have contributed to the differences between the two studies. Previous research has alluded to the fact that developmental level may have a significant influence on the ability to learn Logo (Statz, Folk, & Seidman, 1973). Recent reports (Krasnor & Mitterer, 1983) emphasize that only children of at least 12 years of age may be able to fully utilize all the benefits of Logo especially in terms of general problem-solving skills. Further research focusing in these issues would reveal if there are any discernible stages that children go through in learning how to program and if these stages are related to development.

In terms of programming style, the only differences that were observed were in the area of project complexity. The measure of complexity in terms of tree depth of procedural nets, number of procedures, number of revisions and number of bugs may not represent a reliable indicator of complexity. Further research with a larger number of subjects could examine in more detail this issue. A checklist of complexity developed by an outside rater might also evaluate the complex nature of projects in a more reliable fashion.

Based on this classification system, both the gifted and non-gifted subjects made similar kinds of errors and used the same concomitant coping strategies. They differed only in the frequency of occurrence of bugs and debugging actions. The differences in bug frequency between gifted and non-gifted subjects could be related to each group's ability to estimate distance and angle rotation in some tasks, while in others the bugs were more closely related to the task at hand, its complexity, its difficulty level, and the programming expertise of the students. In the former, the gifted were able to estimate better the state of the Turtle, while in the latter categories, both groups made similar numbers of errors and in some cases, the gifted made more. These trends are inconclusive, however, and more quantitative analyses, using perhaps a multivariate analysis of variance with larger numbers of subjects would clarify these qualitative statements.

Debugging actions were found to differ in frequency between the two groups. The descriptive analysis indicated several trends. The gifted made more use of actions that indicated that a procedure had been written while the average used more actions in the g&t activity before the procedure was defined. This could be related to the kinds of errors made and further research in this area could clarify the issue.

The relationship between bugs and debugging strategies showed qualitatively and quantitatively the differences in strategies between the two groups. The gifted were more consistent as a group in their choice of debugging actions than the average groups. Only more research examining different populations of subjects in similar task environments could give more information about the different cognitive styles exhibited here.

Most research on Logo has followed either a classical experimental or the anecdotal case study approach. This research examined the Logo experience from a different perspective. Using the technique of protocol analysis, it became possible to describe in a meaningful way the differences in problem-solving between different groups of subjects. It should be emphasized that no attempt has been made to evaluate this experience but only to understand it better. As computer cultures become a more common part of our educational environments, educators will be asked repeatedly to use them in a meaningful fashion with children of different ages and ability levels. More research is needed in this area to examine what are the interactions between levels of knowledge, different abilities and the kind of learning that takes place in these cultures. If computer microworlds are to enhance the educational process, understanding what happens in them becomes the first step toward using them appropriately. The research described herein is a tentative step in that direction.

REFERENCES

Abelson, H., Bamberger, J., Goldstein, I., & Papert, S. A. Logo progress report 1973-75, Logo Memo 22, Cambridge, MA., M.I.T., 1976.

Abelson, H. & diSessa, A. A. Turtle geometry: The computer as a medium for exploring mathematics. Cambridge: Massachusetts Institute of Technology, 1980.

Abelson, H., & Goldenberg, E. P. Teacher's guide to computational models of animal behavior. A..I. Memo 432, Cambridge, MA, M.I.T., April, 1977.

Ackerman, R. Theories of knowledge: A critical introduction. New York: McGraw-Hill, 1965.

Austin, H. Teaching teachers Logo. Logo Memo No. 23, Cambridge, MA., M.I.T., April 1976.

Baldwin, A. L. Theories of child development (2nd. ed.). Toronto: John Wiley, 1980.

Bamberger, J. Developing a musical ear: a new experiment. Logo Memo 6, Cambridge, MA., M.I.T., July, 1972.

Bank Street College of Education, Center for Children and Technology, New York: Annual Report, 1981-82.

Boden, M. A. Jean Piaget. New York: Viking Press, 1980.

Brown, J. S. & Burton, R. R. Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science, 1978, 2, 155-192.

Brown, J. S., & Rubinstein, R. Recursive functional programming for the student in the humanities and social sciences. Technical Report No. 27a, Department of Information and Computer Science, The University of California, 1974.

Bruner, J. S. On knowing: Essays for the left hand (Exp. ed.). Cambridge: Belknap Press of Harvard University Press, 1979.

Burnett, D., Educational psychology and mathematical knowledge: An analysis of two student protocols. Focus on Learning Problems in Mathematics, 1981, 3 (2 & 3).

Burrill, D. Personal Communication, March, 1982.

Cartwright, G. F. Social, personality, and attitudinal dimensions of individual learning with computer-assisted group instruction. Unpublished doctoral dissertation, Edmonton, Alberta, The University of Alberta, 1973.

Chait, S. An analysis of children's problem-solving in a graphics oriented computer programming environment. Unpublished M.A. thesis. Montreal: McGill University, August, 1978.

Clark, B. Growing up gifted. Columbus, Ohio: Charles E. Merrill, 1979.

Copeland, R. W. How children learn mathematics (3rd. ed.). New York: MacMillan, 1979.

Dewey, J. The child and the curriculum and the school and society. Chicago: University of Chicago Press, 1956.

diSessa, A. A. , & White, B. Y. Learning Physics from a Dynaturtle. Byte, 1982, 7 (8), 324.

Doctorow, G. Influence of computers on secondary school mathematics curricula. ECOO Newsletter, 1982, 3 (2), 27-34.

du Boulay, B. Learning teaching mathematics. Mathematical Teaching, March, 1977, 78, 52-57.

Dwyer, T. Heuristic strategies for using computers to enrich education. In R. P. Taylor (Ed.), The computer in the school: Tutor, tool, tutee. New York: Teachers College, 1980.

Elkind, D. Piaget and developmental psychology. In F. B. Murray (Ed.), The Impact of Piagetian theory: On Education, philosophy, psychiatry, and psychology. Baltimore: University Park Press, 1979.

Ellis, A. B. The Use and misuse of computers in education. New York: McGraw-Hill, 1974.

Evans, C. The Mighty micro: The impact of the micro-chip revolution. Coronet, Hodder & Stoughton, 1980.

Ferguson, G. A. Statistical analysis in psychology and education (4th ed.). New York: Mcgraw-Hill, 1976.

Feurzeig, W., & Lukas, G. The use of dribble files as instructional aids. Logo working paper, Bolt Beranek and Newman, August, 1975.

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. Programming language as a conceptual framework for teaching mathematics. Technical Report No. 1889, Bolt Beranek and Newman, November, 1969.

Fischer, G. Computational models of skill acquisition processes. In R. Lewis & D. Tagg (eds.), Computers in education. Amsterdam: North-Holland, 1981.

Goldenberg, E. P. Special technology for special children: Computers to serve communication and autonomy in the education of handicapped children. Baltimore: University Park Press, 1979.

Gorman, H., Jr. The Lamplighter Logo project. 99'er Magazine, 1981 (a), 1 (1), 60-62.

Gorman, H., Jr. Extending Logo: Applications for very young children. 99'er Magazine, 1981 (b), 1 (4), 68-69 and 77.

Gorman, H., Jr. The Lamplighter project. Byte, 1982, 7 (8), 331-332.

Gregg, L. W. Spatial concepts, spatial names, and the development of exocentric representations. Technical report, Carnegie-Mellon University, 1978.

Groen, G. J. The Theoretical ideas of Piaget and educational practice. In P. Suppes (Ed.), Impact of research on education: Some case studies. Washington, D.C.: National Academy of Education, 1978.

Harvey, B. Why Logo? Byte, 1982, 7 (8), 163-193.

Hawkins, J., Sheingold, K., Gearhart, M., & Berger, C. Microcomputers in schools: Impact on the social life of elementary classrooms. Center for Children and Technology, Bank Street College of Education, New York: Draft, May 1982.

Hilgard, E. R., & Bower, G. H. Theories of learning (4th ed.). Toronto: Prentice-Hall, 1975.

Hofstadter, D. R. Godel, Escher, Bach: An eternal golden braid. New York: Basic Books, 1979.

Howe, J. A. M. Developmental stages in learning to program. D.A.I. Research Paper No. 119, Edinburgh: The University of Edinburgh, 1978.

Howe, J. A. M. Learning mathematics through Logo programming. In 1981 Conference Proceedings of the Association for the Development of Computer-Based Instructional Systems. Computer-based instruction: Frontiers of thought. Bellingham, Washington: Western Washington University, 1981, 109-113.

Howe, J. A. M., & O'Shea, T. Computational metaphors for children. In F. Klix (ed.), Human and artificial intelligence. Berlin: Deutscher Verlag, 1978.

Howe, J. A. M., O'Shea, T., & Plane, F. Teaching mathematics through Logo programming: An evaluation study. D.A.I. Research Paper No. 115, Edinburgh: The University of Edinburgh, 1980.

Howe, J. A. M., Ross, P., Johnson, K., Plane, F., & Inglis, R. Learning mathematics through Logo programming: The transition from laboratory to classroom. D.A.I. Working Paper No. 118 (b), Edinburgh: The University of Edinburgh, August, 1982.

Hyman, A. Charles Babbage: Pioneer of the computer. Princeton, New Jersey: Princeton University Press, 1982.

Jewson, J., & Pea, R. D. Logo research at Bank Street College. Byte, 1982, 7 (8), 332-333.

Johnson-Laird, P. N. Book Review of the Grasp of Consciousness. Times Higher Education Supplement, Sept. 30, 1977.

Kamii, C., & DeVries, R. Piaget, children and number: Applying Piaget's theory to the teaching of elementary number. Washington, DC: National Association for the education of young children, 1976.

Kant, I. Critique of pure reason (N. K. Smith, trans.). London: MacMillan, 1964. (Originally published, 1787.)

Keppel, G. Design and analysis: a researcher's handbook. (2nd. ed.). Englewood Cliffs, N.J.: Prentice-Hall, 1982.

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

Lawler, R. W. Extending a powerful idea. A.I. Memo No. 590, Cambridge, MA., M.I.T., July, 1980.

Lawler, R. W. Designing computer-based microworlds. Byte, 1982, 7 (8), 138-160.

Lukas, G., & Feurzeig, W. Computers and learning to learn. Proceedings of the Second City University of New York Conference, Innovations in Educational Technology, May 1972.

Martinson, R. A. The Identification of the Gifted and Talented. Reston, Virginia: The Council of Exceptional Children, 1977.

Miller, M. L. A structured planning and debugging environment for elementary programming. In D. Sleeman and J.S. Brown (eds.), Intelligent tutoring systems. London, England: Academic Press, 1982.

Miller, M. L. & Goldstein, I. P. Structured planning and debugging. Paper presented at the International Joint Conferences on Artificial Intelligence, August, 1977.

Milner, S. The effects of computer programming on performance in mathematics. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, February, 1973.

Minsky, M. Computation: Finite and infinite machines. Englewood Cliffs, N.J.: Prentice-Hall, 1967.

Murray, B. F. (Ed.). The impact of Piagetian theory: On education, philosophy, psychiatry, and psychology. Baltimore: University Park Press, 1979.

Newell, A. & Simon, H. A. Human problem solving. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

Papert, S. A. Teaching children to be mathematicians vs. teaching about mathematics. Logo Memo No. 4, Cambridge, MA., M.I.T., July 1971.

Papert, S. A. Teaching children thinking. Programmed Learning and Educational Technology, 1972, 9 (5), 245-255.

Papert, S. A. Uses of technology to enhance education. Logo Memo 8, Cambridge, MA., M.I.T., June 1973.

Papert, S. A. Some poetic and social criteria for education design. Logo Memo 27, Cambridge, MA., M.I.T., June 1976.

Papert, S. A. Computer-based microworlds as incubators for powerful ideas. Unpublished report, Cambridge, MA., M.I.T., March 1978.

Papert, S. A. Computer-based microworlds. In R. P. Taylor (ed.), The Computer in the school: Tutor, tool, tutee. New York: Teachers College, 1980a.

Papert, S. A. Mindstorms: Children, computers and powerful ideas. New York: Basic Books, 1980b.

Papert, S. A., & Solomon, C. Twenty things to do with a computer. Educational Technology, 1972, 12 (4), 9-18

Papert, S. A., Watt, D., diSessa, A., & Weir, S. Final report of the Brookline Logo project. Part II: Project summary and data analysis. A.I. Memo No. 545, Cambridge, MA., M.I.T., 1979.

Perlman, R. TORTIS: Toddler's own recursive turtle interpreter system. Logo Memo No. 9, Cambridge, MA., M.I.T., December, 1974.

Perlman, R. Using computer technology to provide a creative learning environment for preschool children. Logo Memo No. 24, Cambridge, MA., M.I.T., May 1976.

Piaget, J. Genetic epistemology (E. Duckworth, trans.). New York: Columbia University, 1970.

Piaget, J. Biology and knowledge Chicago: University of Chicago Press, 1971.

Piaget, J. Developments in mathematical education, Proc. of the Second International Congress on Mathematical Education. In H. E. Gruber and J. J. Voneche (eds.), The Essential Piaget. New York: Basic Books, 1977. (Originally published, 1972.)

Piaget, J. To understand is to invent. New York: Grossman, 1973.

Polya, G. How to solve it (2nd ed.). New York: Anchor Books, 1957.

Renzulli, J. S. What makes giftedness? Phi Delta Kappan, 1978, 60 (3), 180-184; 261.

Renzulli, J. S. , Reis, S. M. & Smith, L. H. The Revolving door identification model. Mansfield Center, Conn: Creative Learning Press, 1981.

Renzulli, J. S., Smith, L. H., White, A. J., Callahan, C. M., & Hartman, R. K. Scales for rating the behavioral characteristics of superior students. Mansfield Center, Conn: Creative Learning Press, 1976.

Ridge, H. L., & Renzulli, J. S. Teaching mathematics to the talented and gifted. In V. J. Glennon (ed.), The Mathematical education of Exceptional children and youth. Reston, Virginia: National Council of Teachers of Mathematics, 1981.

Rousseau, J. F. & Smith, S. M. Whither goes the Turtle? Microcomputing, 1981, 5 (9), 52-55.

Sahakian, W. S. History and systems of psychology. New York: John Wiley, 1975.

Scheffler, I. Conditions of knowledge: An introduction to epistemology and education. Chicago: Scott Foreman, 1965.

Solomon, C. J. Problem-solving in an anthropomorphic computer culture. Unpublished M.A. thesis, Boston University, 1976.

Solomon, C. J. Introducing Logo to children: Teaching Logo requires an awareness of different learning styles. Byte, 1982, 7, (8), 196-208.

Sperry, R. W. Mind, machines and humanistic values. In J. R. Platt (ed.). New views on the nature of man. Chicago: University of Chicago Press, 1965.

Statz, J., Folk, M., & Seidman, R. Final report. Syracuse University Logo Project, 1973.

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

Tessier, B. Implantation du systeme Logo dans un internat de reeducation. IXe Congres de l' A.I.E.J.I., 27 avril, 1978.

Toffler, A. Future shock. New York: Random House, 1970.

Turing, A. On computational numbers, with an application to the entescheidungsproblem. Proceedings of the London Mathematical Society, 1937, 42, 230-265.

Vuyk, R. Overview and critique of Piaget's genetic epistemology, 1965-1980 (2 vol.). New York: Academic Press, 1981.

Watt, D. Final report of the Brookline Logo project. Part III: Profiles of individual student's work. A.I. Memo No. 546, Cambridge, MA., M.I.T., 1979.

Watt, D. Logo in the schools: Putting Logo in the classroom has led to some interesting results. Byte, 1982, 7 (8), 116-134.

Wavrik, J. J. Mathematics education for the gifted elementary school student. Gifted Child Quarterly, 1980, 24 (4), 169-173.

Weir, S., & Emanuel, R. Using Logo to catalyze communication in an autistic child. D.A.I. Research Report No. 15, Edinburgh: The University of Edinburgh, January 1976.

Weir, S., Russell, S. J., & Valente, J. A. Logo: An approach to educating disabled children. Byte, 1982, 7 (9), 342-360.

Winston, P. H. Artificial intelligence. Don Mills, Ontario: Addison-Wesley, 1977.

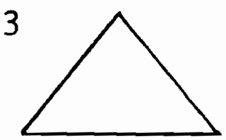
Zimbardo, P. The age of indifference. Psychology Today, 1980, 14 (3), 70-76.

APPENDIX A

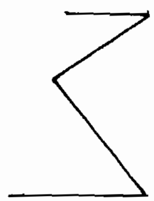
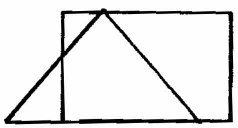
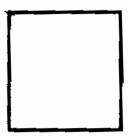
Complete Descriptive Summary Tables

Note: A key to the Descriptive Summary Tables
is found starting on p. 419

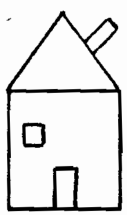
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 1: Introduction										
Mar 15 Ses.1	1.1			a)nog			dc		FD, RT, CS	ob ob	1.1 1.1	da da
	1.2			a)g&t			dc			len ang	1.2 1.2	ia *
			RECTANGLE	b)pw			t		Procedure writing			
				c)test			dc	+				
	1.3			a)g&t			dc		LT, BK, HT, ST	rlo ang rlo trace rlo len ang ang	1.3 1.3 1.3 1.3 1.3 1.3 1.3 1.3	u, ia u u ass u ia, ia, ia cr ia, u

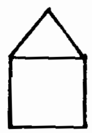
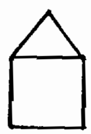
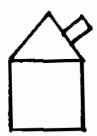
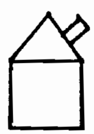
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ang	1.3	u, u
										ang	1.3	u, u, ia, ia
			TRIANGLE	b)pw			t			cop	–	na
				c)test			dc	–		cop	1.4	e
	1.4			a)rev	1.3	rev	e		EDIT, EDL, PO			
				b)test			dc	+	HOME			
	1.5			a)con			dc					
			SQUARE	b)pw			t					
				c)test			dc	+				

Project 2: House

	2.1			a)con			noc					

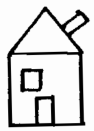
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.2			a)g&t			dc			ang len ang trace	2.2 2.2 2.2 2.2	u, u ia ia ass
				b)pw			noc					
Mar 22 Ses.2	2.3		HOUSE	a)pw			t					
				b)test			dc	+				
	2.4			a)g&t	2.3	sub	dc			rlo len	2.4 2.4	u ia
			CHIMNEY	b)pw			t					
				c)test			dc	+				
	2.5			a)add	2.3 2.4	rev sub	e					PO
				b)test			dc	+				

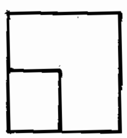
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.6			a)g&t	2.5	sub	dc		PU, PD	ang	2.6	ia
										rlo	2.6	ia
										rlo	2.6	ia
										len	2.6	u
			WINDOW	b)pw			t			cop	-	na
				c)test			dc	-		name	2.6	cr
										cop	2.7	e
	2.7			a)rev	2.6	rev	e					
				b)test			dc	+				
	2.8			a)add	2.5	rev	e					
					2.7	sub						
	2.9			a)g&t	2.8	sub	dc			len	2.9	u, ia, u
										trace	2.9	u
										trace	2.9	u
			DOOR	b)pw			t			cop	-	na

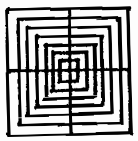
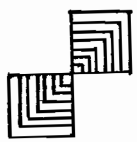
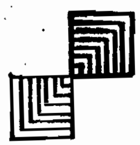
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-				
	2.10			a)g&t	2.9	rev	dc t		STEP ALL ctrl G ER STEP ALL			PO S*
				b)rev			e			cop	2.10	e
				c)test			dc	+				
	2.11			a)add	2.8 2.10	rev sub	e					PO
				b)test			dc	+				

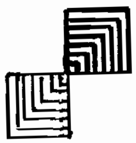
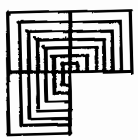
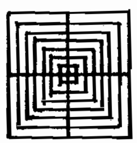
Project 3: Teaching Session REPEAT

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 23 Ses.3	3.1			a)nog	2.11 1.5	sub sub	dc t					
	3.2			a)nog			dc		REPEAT			

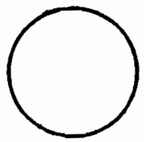
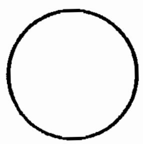
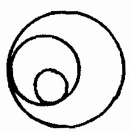
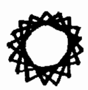
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	3.3			a)g&t			dc			re/syn	3.3	ass* (ctrl G)
										re/syn	3.3	ass* (ctrl G)
										rlo	3.3	u
										rlo	3.3	ia
										re/syn	3.3	ass (ctrl G)
										re/syn	3.3	ass (ctrl G)
			THE.21. SQUARES	b)pw			t			re/syn	3.3	ass (ctrl G)
										re/syn	3.3	ass (ctrl G)
				c)test			dc	+				
	3.4			a)g&t	3.3	sub	dc			st.or	3.4	ass
				b)add	3.3 3.3	rev sub	e			stop/r	-	na
				c)test			dc	-		stop/r	3.4	ass* (ctrl G)

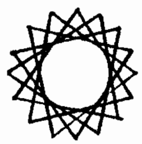
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	3.5			a)add	3.4	rev	e					PO
					3.3	sub				pos	3.5	*
				b)test			dc	-		stop/r	3.7	ass, p* (ctrl G)
	3.6		THE.SQUARES	a)pw	3.3	sub	t					PO*
	3.7			a)rev	3.3	rev	e		ERL	stop/r	-	e* PO
				c)test			dc	+				
	3.8			a)g&t			dc			ang	3.8	ia*
				b)add	3.6	rev	e					PO*
					3.7	sub			sub, super, POTS			PO*, PO*
				c)test			dc	+				

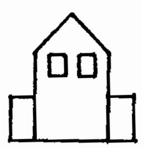
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 4: Circles										
Mar 29 Ses.4	4.1			a)con			noc		Circle			
	4.2			a)g&t			dc			re/syn re/syn size re/syn	4.2 4.2 4.2 4.2	ass (ctrl G) rc cr ass, rc (ctrl G)
	4.3			a)nog			dc					
	4.4		DESIGN. CIRCLE	a)pw			t		Input EDT			
				b)test			dc	+				
	4.5			a)nog	4.4	sub	dc					PO

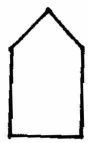
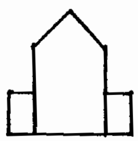
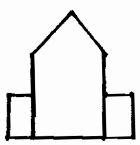
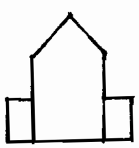
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ob(s)	4.5	cr*
										ob(s)	-	
										ob(s)	-	
										ob(s)	4.5	cr
										ob	4.5	cr
										ob	4.6	p
	4.6		CIRCLE	a)pw			t					sp*
	4.7		SQUARE	a)pw			t		ERASE			
				b)test			dc	+		ob(s)	4.7	rc
	4.8			a)nog	4.7	sub	dc			ob(s)	4.7	cr
							t			ob(s)	4.7	PO

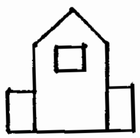
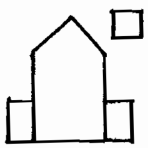
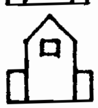
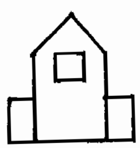
Project 5: Rocket

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.1			a)con			noc					

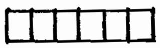
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.2			a)g&t			dc			ob len len trace	5.2 5.2 5.2 -	da cr ia, u na
			ROCKET	b)pw			t					
				c)test			dc	+				
	5.3			a)g&t	5.2	sub	dc			len trace	5.3 5.3	u ass, p*
			ENGINES	b)pw			t					
				c)test			dc	+				
Apr 19 Ses.5	5.4			a)g&t	5.2 5.3	sub sub	dc					
			BODY	b)rev	5.2	rev	e					PO, PO
	5.5		ROCKET	a)pw	5.4 5.3	sub sub	t					

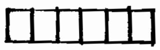
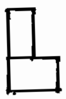
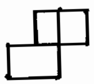
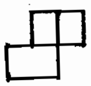
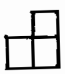
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	5.6			a)g&t			dc			st.or len len rlo	5.6 5.6 5.6 5.6	u, u ia ia ia
			ROCKET. WINDOW	b)pw			t			cop	–	na
				c)test			dc	-				S*
	5.7			a)g&t	5.6	rev	dc					S*
				b)rev			e			cop	5.7	e
				c)test			dc	+				
	5.8			a)add	5.5 5.7	rev sub	e					PO
				b)test			dc	+				

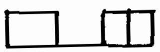
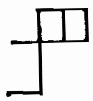
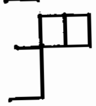
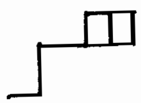
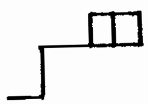
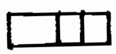
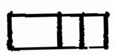
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 6: Review of Projects										
	6.1			a)nog	4.4	sub	dc					PO
					4.6	sub	t					PO
		Project 7: Sidewalk										
	7.1			a)con			noc					
	7.2		SIDEWALK	a)pw			t			pos	-	na
				b)test			dc	-				
	7.3			a)g&t	7.2	rev	dc			pos	7.3	PT*, cr*
										pos	7.3	cr*
										pos	7.3	cr
										size	7.3	cr
										re/syn	7.3	cr, ass (ctrl G)

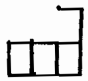
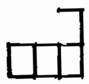
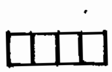
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										re/syn	7.3	ass (ctrl G) post
Apr 20 Ses.6	7.4			a)g&t	7.2	rev	dc			rlo	7.4	ia
			SIDEWALK	b)pw			t			cop	-	na
				c)test			dc	-				
	7.5			a)g&t	7.4	rev	dc			cop	7.5	S, e
				b)rev			e					
				c)test			dc	-				
	7.6			a)g&t	7.5	rev	dc			ang	7.6	S, e
				b)rev			e					
				c)test			dc	-				
	7.7			a)g&t	7.6	rev	dc			len	7.7	S, e

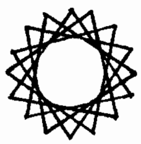
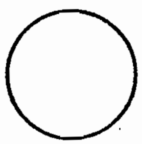
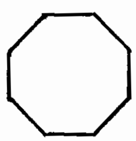
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e					
				c)test			dc	-				
	7.8			a)g&t	7.7	rev	dc			ang	7.8	S, e
				b)rev			e					
				c)test			dc	-				
	7.9			a)g&t	7.8	rev	dc			rlo	7.9	S, e
				b)rev			e			re/syn	7.9	ass (ctrl G) PO
				c)test			dc	-				
	7.10			a)g&t	7.9	rev	dc			pos	7.10	S, e
				b)rev			e					
				c)test			dc	-				
	7.11			a)g&t	7.10	rev	dc					

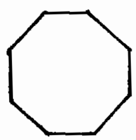
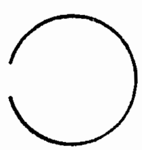
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e			ang rlo rlo	7.11 7.11 -	S, e PT* na
				c)test			dc	-				
	7.12			a)rev	7.11	rev	e			rlo	7.12	e
				b)test			dc	+				
	Project 8: Recursion											
	8.1			a)nog	3.6 5.8 5.12	sub sub sub	dc t					PO
	8.2			a)g&t			dc t		Recursion			PO
			CIRCLE	b)pw			t			stop/r	8.2	ass (ctrl G)

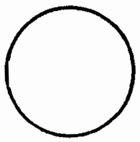
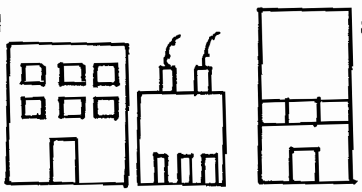
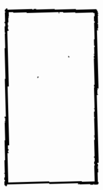
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+		stor	8.2	ass
	8.3		CIRCLE1	a)pw			t			stop/r	8.3	ass (ctrl G)
				b)test			dc	+		stor	8.3	ass
	8.4		CIRCLE4	a)pw			t			name stop/r	8.4 8.4	cr ass (ctrl G)
				b)test			dc	+		stor	8.4	ass
	8.5		CIRCLE3	a)pw			t			rec	8.5	e
	8.6			a)rev	8.5	rev	e					
				b)test			dc	+		stor	8.6	ass

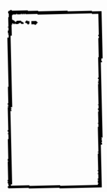
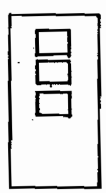
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 26 Ses.7	8.7			a)nog	8.2	sub	dc					PO*
							t					PO
										stop/r	8.7	ass (ctrl G)
										stor	8.7	ass
	8.8		CIRCLE.5	a)pw			t					
				b)test			dc	+		stop/r	8.8	ass (ctrl G)
	8.9		CIRCLE.9	a)pw			t			rec	–	na
				b)test			dc	–		rec	8.10	PO*, e
	8.10			a)rev	8.9	rev	e					
				b)test			dc	+		stop/r	8.10	ass (ctrl G)
										stor	8.10	ass

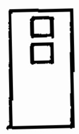
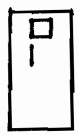
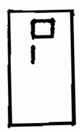
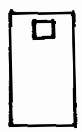
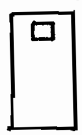
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	8.11		CIRCLE.7	a)pw			t					
				b)test			dc	+		stop/r	8.11	ass (ctrl G)
										stor	8.11	ass
		Project 9: City										
	9.1			a)con			noc					
	9.2			a)g&t			dc			rlo	9.2	ia
							t			len	9.2	ia, ia
										ob(p)	9.2	da
										ob(p)	9.2	da
										ob(p)	9.2	da
										re/syn	9.2	cr
										st.or	9.2	cr, PO, p*

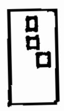
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			RECT	b)pw			t					
				c)test			dc	+				
	9.3			a)g&t	2.7 9.2	sub sub	dc		RUBDIS	size name size	9.3 9.3 9.3	PU, rr*, rr, rr, rr, rr PO p*
			APARTMENT	b)pw	9.2	sub	t					
				c)test			dc	+				
Apr 27 Ses.8	9.4			a)g&t	9.3	sub	dc t			name size re/syn	9.4 9.4 9.4	PO*, PO* PO* rp, rr, ns, p ass (ctrl G)
	9.5		SQU	a)pw			t					

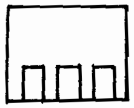
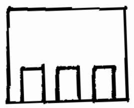
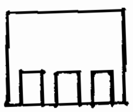
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	9.6			a)g&t	9.3	sub	dc			trace	9.6	rc
					9.5	sub				pos	9.6	rr
										trace	9.6	rr
			WI	b)pw	9.5	sub	t			cop	-	na
				c)test			dc	-		name	9.6	cr
										ob(p)	9.8	S
	9.7			a)g&t	9.6	rev	dc			cop	9.7	e
										trace	9.7	rc, PO*
				b)rev			e					
				c)test			dc	-		ob	9.8	S, PO*, e
	9.8			a)g&t	9.7	rev	dc t			rlo	9.8	PT*
				b)rev			e noc					

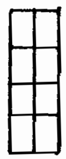
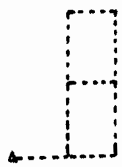
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL.	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-		trace	-	S
	9.9			a)g&t	9.8	rev	dc			trace	9.9	e
				b)rev			e					PO*
				c)test			dc	-		ob(p)	9.10	e
	9.10			a)rev	9.9	rev	e					PO*
				b)test			dc	-		trace	9.11	e, PO*
	9.11			a)rev	9.10	rev	e					
				b)test			dc	+				
	9.12			a)add	9.3 9.11	rev sub	e					
				b)test			dc	+				

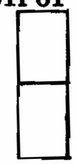
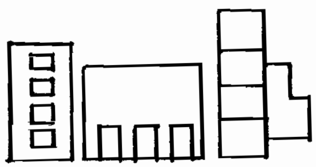
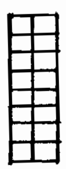
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 3 Ses.9	9.13			a)con			noc					
	9.14			a)g&t	9.12	sub	dc			st.or	9.14	rr
										size	9.14	rr
										len	9.14	ia
										rlo	9.14	ia
										len	9.14	ia
										re/syn	9.14	rr, rr, p*
			DR	b)pw			t					
	9.15		FACTORY	a)pw	9.14	sub	t					
				b)test			dc	+				
	9.16		CITY	b)pw	9.12 9.15	sub sub	t					

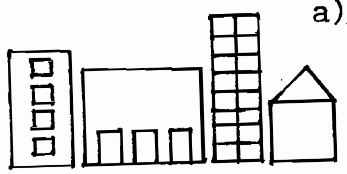
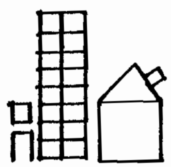
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	9.17			a)con			noc					
	9.18			a)g&t			dc			ob(s) rlo rlo rlo	9.18 9.18 9.18 9.18	rr ia ia ia
			PVM	b)pw			t					
				c)test			dc	+				
	9.19			a)add	9.18	rev	e			re/syn	9.19	ass (ctrl G)
				b)test			dc	-		trace	9.20	e
	9.20			a)rev	9.19	rev	e					

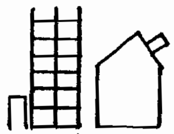
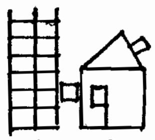
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	9.21			a)add	9.20	rev	e			cop	–	na
	9.22			a)add	9.16 9.21	rev sub	e					
				b)test			dc	–		cop cop	9.23 9.23	e* *
	9.23			a)rev	9.21	rev	e					
				b)test			dc	+				
	9.24			a)add	9.23	rev	e					

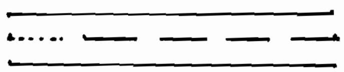
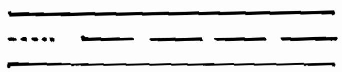
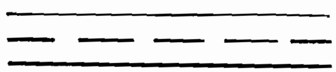
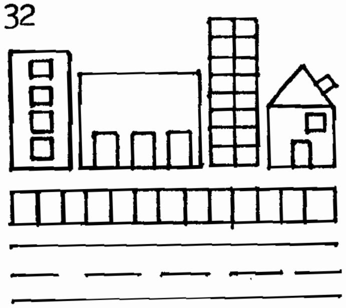
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
May 10	9.25			a)g&t	9.22	sub	dc			rlo	9.25	ia
Ses.10					2.11	sub	t			len	9.25	ia, ia
										ob(p)	9.25	da, ia, ia
										ob(p)	9.25	da, ia, ia
										ob(p)	9.25	da, ia, ia
										len	9.25	u, ia
										st.or	9.25	PO, PO*, rp
				b)test			dc	-				
	9.26			a)g&t	2.11	sub	dc			pos	9.26	S, PO, rr, ns*, p*
			HOU	b)pw	2.11	sub	t					

FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	9.27			a)g&t	9.22 9.26	sub sub	dc			pos	9.28	S
				b)rev	2.10	rev	e					
				c)test			dc	-				
	9.28			a)g&t	9.27	sub	dc			pos	-	S, *
				b)rev	9.27	rev	e					
				c)test			dc	+				
May 11 Ses.11	9.29			a)g&t	9.22 9.28	sub sub	dc			len ob(p) rlo rlo trace len trace ang ang pos	9.29 - 9.29 9.29 9.29 9.29 9.29 9.29 9.29 9.29	ia ia ia u ass u ass ia, u PT* p

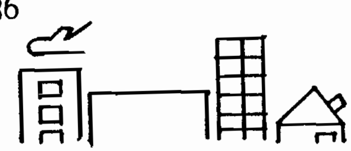
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			STREET	b)pw			t			cop	-	na
				c)test			dc	-		cop	9.30	PO, S
	9.30			a)g&t	9.28 9.29	sub sub	dc					
				b)rev	9.29	rev	e			cop	-	e
	9.31			a)add	9.22 9.28 9.30	rev sub sub	e					PO
				b)test			dc	+				
	9.32			a)g&t			dc t			pos rlo ang len rlo	9.32 9.32 9.32 9.32 9.32	PO, rp*, cr, ns, p u ia ia ia

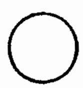
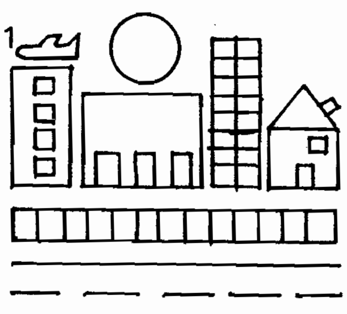
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										len	9.32	u
										rlo	9.32	ia
			SW	b)pw			t					
				c)test			dc	+				
May 17 Ses.12	9.33			a)con			noc					
	9.34			a)g&t			dc			len	9.34	ia
										size	9.34	rr
										re/inp	9.34	rr
										re/inp	9.34	ia, ia, ia, ia, ia
										len	9.34	u
										rlo	9.34	u
										len	9.34	ia, ia, ia
										ang	9.34	ia*

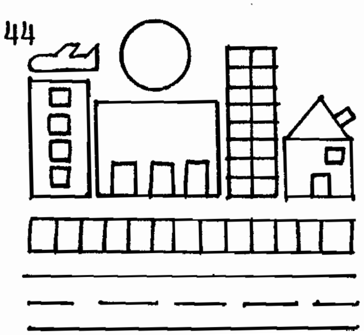
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										len	9.34	rr
										ang	9.34	ia
										len	9.34	rr, u, u, ia, ia
										trace	9.34	ass
										rlo	9.34	u
										len	9.34	ia, ia, ia
			PLANE	b)pw			t			cop	-	na
	9.35			a)add	9.34	rev	e		MOVET	stop/M	-	na, PO
	9.36			a)add	9.31	rev	e					PO
					9.35	sub	dc			ob	9.39	S, e
				b)test			dc	-				
	9.37			a)g&t	9.35	rev	dc					

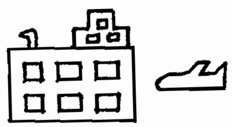
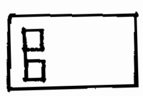
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e					
				c)test			dc	-		ob	-	S, e
	9.38			a)g&t	9.37	rev	dc					
				b)rev			e					PO
				c)test			dc	-		ob	-	S, e
	9.39			a)g&t	9.38	rev	dc					
				b)rev			e					
				c)test			dc	+				
May 18 Ses.13	9.40			a)con			noc					
	9.41			a)g&t	9.36	sub	dc			len ob(p)	9.41 9.41	u rr, cr

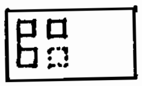
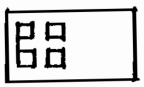
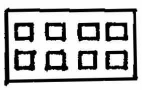
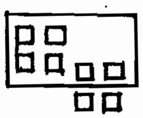
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										re/inp	9.41	ia, ia, ia, ia
										size	9.41	rr, rr
										pos	9.41	p
			SUN	b)pw			t			trace	-	na
				c)test			dc	-		trace	9.42	e
										pos	9.42	e
	9.42			a)rev	9.41	rev	e					PO
				b)test			dc	-		re/inp	9.43	e
	9.43			a)rev	9.42	rev	e					
	9.44			a)add	9.36	rev	e					PO
					9.43	sub						

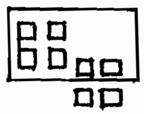
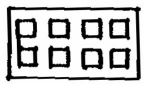
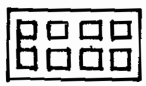
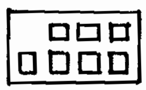
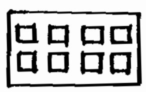
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
		Project 10: Airport										
	10.1			a)con			noc					
	10.2			a)g&t	9.5	sub				rlo	10.2	ia
										size	10.2	cr, cr*
										len	10.2	u, PO, PO, PO
										trace	10.2	*
										len	10.2	ia
										trace	10.2	rc
										rlo	10.2	u
										len	10.2	ia
										rlo	10.2	ia
										trace	10.2	rc
				b)pw			noc					

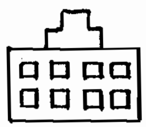
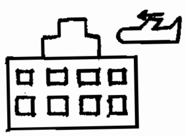
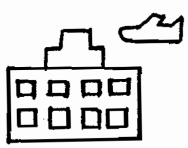
FIRST ANALYSIS – GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 25 Ses.14	10.3		HANGER	a)pw	9.5	sub	t			cop	–	na
				b)test			dc	–		cop	10.4	e
	10.4			a)rev	10.3	rev	e					
				b)test			dc	+		trace	10.4	ass
	10.5			a)g&t	10.4	sub	dc			len	10.5	u
					9.5	sub				pos	10.5	PO
										trace	10.5	rc
										len	10.5	u, ia
										rlo	10.5	ia
										len	10.5	u
										u	10.5	ia
										trace	10.5	rc, PO
				b)add	10.4	rev	e			cop	–	na
					9.5	sub						
				c)test			dc	–		cop	10.6	S

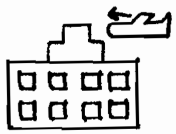
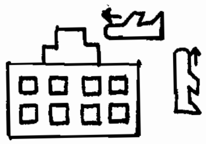
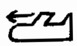
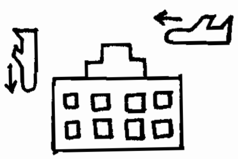
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	10.6			a)g&t	10.5	rev	dc			cop	–	e
				b)rev			e					
				c)test			dc	–				
	10.7			a)g&t	10.6	rev	dc			trace	10.7	PO, PO, e
				b)rev			e					
				c)test			dc	–				
	10.8			a)rev	10.7	rev	e			trace	10.8	e
				b)test			dc	+				
	10.9		AIRPORT	a)pw	10.8 10.10	sub super	t					
	10.10			a)g&t			dc			len rlo	10.10 10.10	ia, u ia

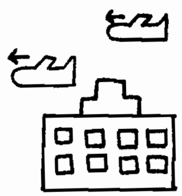
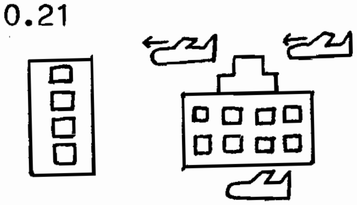
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										len rlo	10.10 10.10	rr, rr u
			TOWER	b)pw			t					
				c)test			dc	+				
	10.11			a)g&t	10.9	sub	dc					
				b)add	9.39 10.9	sub sub	e					
				c)test			dc	+				
	10.12			a)g&t	9.39 10.11	sub sub	dc			trace pos stop/M	10.12 - 10.12	PO na e*
				b)rev	9.39	rev	e		MOVET -			
				c)test			dc	+		trace pos	10.12 -	cr, PO post
May 31	10.13			a)g&t	10.12	sub	dc			pos	10.13	PO, p

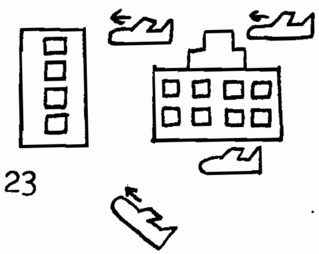
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL.	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Ses.15							t					
				b)pw			t					
	10.14			a)g&t	10.13	sub	dc					
					10.12	sub						
				b)test			dc	-				PO*
										pos	10.15 e	
	10.15			a)rev	10.13	rev	e					
				b)test			dc	+				
	10.16			a)add	10.11	rev	e					PO
					10.15	sub						
				b)test			dc	-		pos	10.17 e	
										trace	10.17 e	
	10.17			a)rev	10.16	rev	e					PO

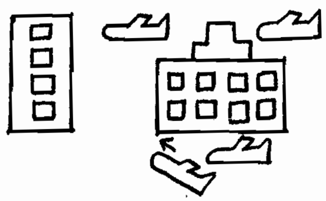
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										trace	-	e
				b)test			dc	-		pos	10.18	e
	10.18			a)rev	10.17	rev	e			pos	-	e
				b)test			dc	+				
	10.19		PLANE22	a)pw			t					PO
				b)test			dc	+				
	10.20			a)con	10.1	rev	noc					
Jun 1 Ses.16	10.21			a)g&t	10.19 10.18	sub sub	dc					
				b)add	10.18	rev	e					

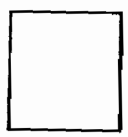
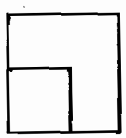
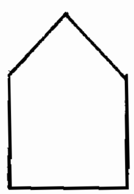
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					10.19 10.22	sub super						
	10.22		HOTEL	a)pw	9.12	sub	t					
				b)test			dc	+				
	10.23			a)g&t			dc t					PO
			PLANE.3	b)pw								
				c)test			dc	-		ob(p) ang	10.26 -	p* na
	10.24			a)add	10.21 10.23	rev sub	e					
				b)test			dc	-		ang	10.25	e
	10.25			a)rev	10.24	rev	e					

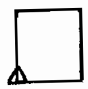
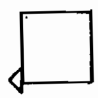
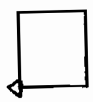
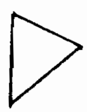
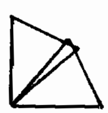
FIRST ANALYSIS - GROUP A

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	-		pos	10.26	e
	10.26			a)rev	10.25	rev	e					
				b)test			dc	+				

FIRST ANALYSIS – GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 1: Introduction										
Mar 16 Ses.1	1.1			a)nog			dc		FD, RT, CS	ob		
	1.2			a)g&t			dc		LT, Procedure writing	rlo	1.2	PT*,u
			SQUARE	b)pw			t					
				c)test			dc	+				
	1.3			a)nog	1.2	sub	dc		REPEAT			
	1.4			a)g&t			dc		RUBDIS, HT, ST, BK	ang len	– 1.4	na ia, ia, u, rr,

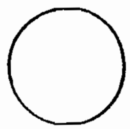
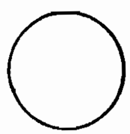
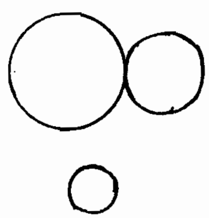
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
												ia, ia, ia
Mar 18 Ses.2	1.5			a)rev	1.2	rev	e		PO, EDIT, ERL, ctrl N			
				b)test			dc	+				
	1.6		SQ	a)rev	1.5	rev	e		EDT			
				b)test			dc	+				
	1.7			a)g&t			dc			ang	1.7	ia
			TRI	b)pw			t					
				c)test			dc	+				
	1.8			a)nog	1.7	sub	dc		SPIN			

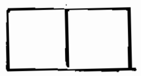
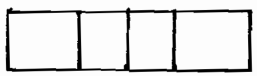
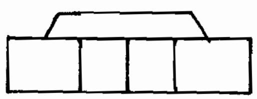
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 2: Car										
	2.1			a)con			noc		sub, super			
Mar 24 Ses.3	2.2			a)nog	1.6	sub	dc t		SPIN - PRINT HOME			PO* PO* PO
	2.3			a)con			noc		Circle			PT*
	2.4			a)g&t			dc		Recursion			
			CIRCLE	b)pw			t		ctrl G	stop/R	2.4	ass (ctrl G)
				c)test			dc	+				

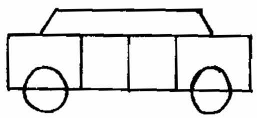
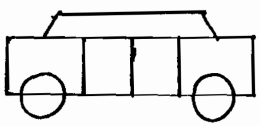
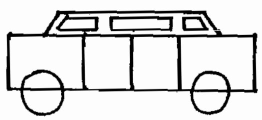
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.5			a)rev	2.7	rev	e		EDL	size	2.5	PO*
				b)test			dc	+		stop/R	2.6	p*, ass (ctrl G) PO
	2.6		CIRCLE1	a)pw			t					
				b)test			dc	+	PU, PD	ob(s)	2.6	ass
	2.7		CIRCLE2	a)pw			t					
				b)test			dc	+		stop/R	2.7	ass, (ctrl G) ns
	2.8			a)g&t			dc		CIRCLER CIRCLEL	size	2.8	rc, cr, ns

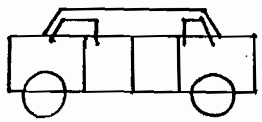
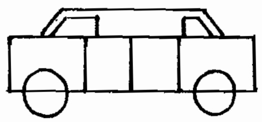
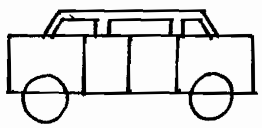
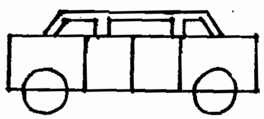
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.9			a)g&t			dc			len	2.9	cr*
										len	2.9	p*
										len	2.9	ia*
										ang	2.9	u, ia
			DOOR	b)pw			t					
				c)test			dc	+				
Mar 25 Ses.4	2.10			a)con			dc					
			BODY	b)pw	2.9	sub	t					
				c)test			dc	+				
	2.11			a)g&t			dc			len	2.11	ia
							t			ang	2.11	PO, u
										len	2.11	ia
										len	2.11	ass, p
			TOP	b)pw			t					
				c)test			dc	+		name	2.11	cr

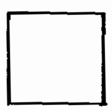
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.12			a)g&t			dc			len	2.12	rr
										size	2.12	rc*
										pos	2.12	rc*
										size	2.12	rc
										len	2.12	ia
										trace	2.12	rc
			WHEELS	b)pw			t					
	2.13		CAR	a)pw	2.10	sub	t					
					2.11	sub						
					2.12	sub						
				b)test			dc	+				
Apr 1 Ses.5	2.14			a)g&t			dc			ang	2.14	u
							t			len	2.14	PO, ia, ia*
										rlo	2.14	u
										trace	2.14	u
										rlo	2.14	u
			T.D.	b)pw			t			cop	-	na

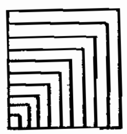
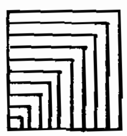
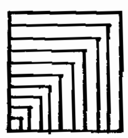
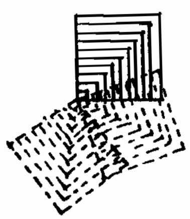
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	–				
	2.15			a)rev	2.14	rev	e					PO
										cop	2.15	e
				b)test			dc	–				
	2.16			a)rev	2.15	rev	e			cop	2.16	e
				b)test			dc	+				
	2.17			a)add	2.13	rev	e					PO*
				b)test			dc	+				

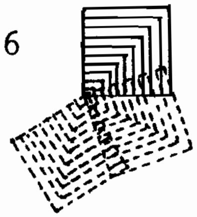
Project 3: Teaching Variable Input

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 2	3.1		SQUARE	a)pw			noc		POTS			PO*

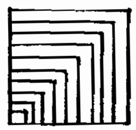
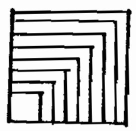
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Ses.6							t		ERASE Input			
				b)test			dc	+				
	3.2			a)nog	3.1	sub	dc			1/g	3.2	ass
	3.3		EMBEDSQ	a)pw	3.1	sub	t		Input +5			
				b)test			dc	-		stop/R	3.4	ass, ctrl G) e*
	3.4			a)rev	3.3	rev	e		Stop rule			PO
				b)test			dc	+				
	3.5			a)nog	3.4	sub	dc					

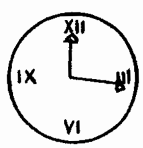
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	3.6		BOOK	a)pw	3.4	sub	t					

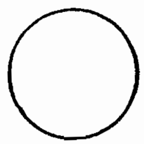
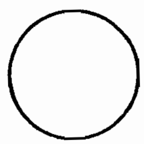
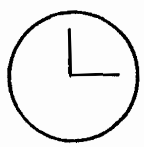
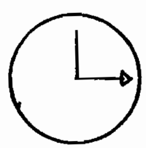
Project 4: Review of Previous Concepts

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 21 Ses.7	4.1			a)nog	3.4 3.6	sub sub	dc t		STEP ER STEP ALL			PO, PO, PO, PO
	4.2			a)nog	3.4 3.6	sub sub	dc t					S* PO

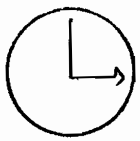
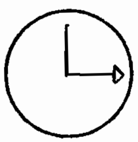
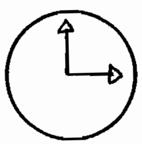
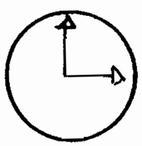
Project 5: Clock

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.1			a)con			noc					

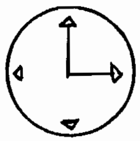
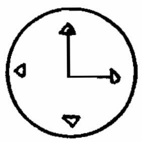
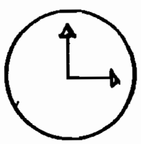
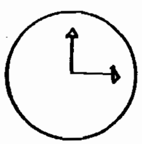
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.2			a)g&t			dc			len ob(p) trace	5.2 5.2 5.2	u cr cr
			FACE	b)pw			t					
				c)test			dc	+				
	5.3			a)g&t	5.2	sub	dc			trace len	5.3 5.3	rr* ia
Apr 22 Ses.8	5.4			a)g&t			dc t			len len len	5.4 5.4 5.4	rr rr, rr PU PO, PO
			HANDS	b)pw			t					
				c)test			dc	+				
	5.5			a)g&t			dc			len	5.5	rr

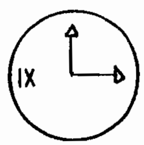
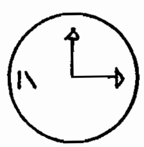
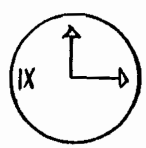
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ang	5.5	rr
			TRIQ	b)pw			t			typ	5.5	e
										cop	-	na
	5.6			a)rev	5.5	rev	e		ctrl C, W, G			
				b)test			dc	-				
	5.7			a)rev	5.6	rev	e			cop	5.7	PO, e
				b)test			dc	+				
	5.8			a)g&t			dc			pos	5.8	rr, rr, rr*
				b)add	5.7	sub	e			Rec	-	na
				b)test			dc	-		Rec	5.9	e
												

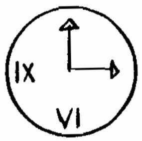
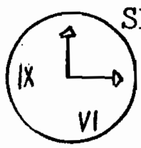
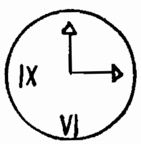
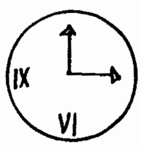
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.9			a)rev	5.8	rev	e					
				b)test			dc	-		stop/R	5.10	ass, (ctrl G) post
Apr 28 Ses.9	5.10			a)g&t	5.9	rev	dc t			stop/R	5.10	ass (ctrl G) PO
										name	5.10	cr
										stop/R	5.10	ass, (ctrl G) cr
										stop/R	5.10	ass, (ctrl G) PO, S, PO, p*
				b)rev			e					
				c)test			dc	+				
	5.11		WATCH	a)pw	5.2	sub	noc					

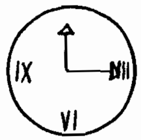
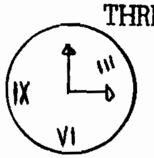
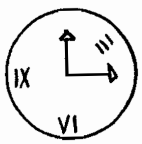
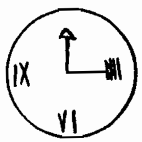
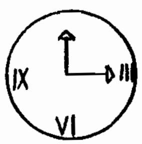
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					5.4	sub	t					
					5.7	sub						
					5.10	sub						
				b)test			dc	+				
	5.12			a)g&t	dc					name	5.12	PO*, ns cr
										ang	5.12	u
										len	5.12	u
										len	5.12	ia
										trace	5.12	rr
										len	5.12	ia, ia, ia
										ang	5.12	ia
			NINE	b)pw			t			cop	-	na
				c)test			dc	-	ctrl R	cop	5.13	e
	5.13			a)rev	5.12	rev	e					
				b)test			dc	+				

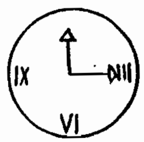
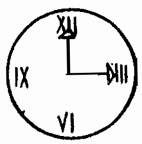
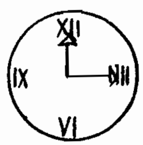
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 5 Ses.10	5.14			a)g&t			dc					PO
							t			len	5.14	u
										len	5.14	u
										len	5.14	ia
										ang	5.14	rr, ia*, ia
										len	5.14	ia, ia
										ang	5.14	ia
		 SIX		b)pw			t			cop	–	na
				c)test			dc	–		cop	5.15	e
	5.15			a)rev	5.14	rev	e					
				b)test			dc	+				
	5.16			a)add	5.11	rev	e					PO
					5.13	sub						
					5.15	sub						
				b)test			dc	+				

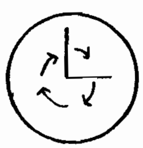
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.17			a)g&t	5.16	sub	dc			trace	5.17	rr
										len	5.17	u
										pos	5.17	rr, rr
										pos	5.17	p
			THREE	b)pw			t			cop	–	na
				c)test			dc	–		cop	5.20	post
May 6 Ses.11	5.18			a)g&t	5.17	rev	dc t		HEADING	cop	–	PO, PO, S, PO, e
				b)rev			e			ang	5.18	H*, e
										len	5.18	e
				c)test			dc	–		cop	–	e
	5.19			a)rev	5.18	rev	e					
				b)test			dc	–		cop	–	e

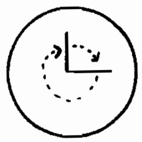
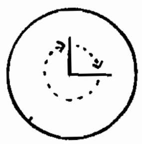
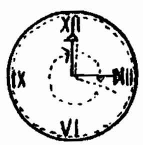
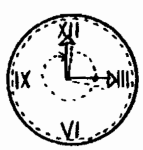
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.20			a)rev	5.19	rev	e					
				b)test			dc	+				
May 12 Ses.12	5.21		TWELVE	a)g&t	5.20	sub	dc		EDIT AS	rlo	5.20	u, ns*
				+			+		SCRATCH	len	5.20	PO, ia,
				pw			t		PAD			u
							+			len	5.20	ia, PO,
							e					PO
				b)test			dc	+				
	5.22			a)add	5.16 5.20 5.21	rev sub sub	e					PO
				b)test			dc	+				

Project 6: Phantom Clock

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	6.1			a)con	5.1	rev	noc					

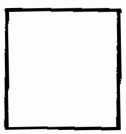
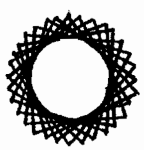
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	6.2			a)g&t			dc		WAIT	name		cr
							t			ob	6.2	cr, PO
										trace	6.2	rr, rr
										trace	6.2	rr
May 13	6.3		PHANTOM	a)g&t			dc			len	6.3	PO, PO
Ses.13				+			+					PO, PO
				pw			t			trace	6.3	rr, PO
							+					
							e					
				b)test			dc	+				
	6.4			a)g&t	6.3	sub	dc			ob	6.4	cr, PO
										st.or	6.4	e
				b)rev	6.3	rev	e					PO
				c)test			dc	+				
	6.5		SUPER	a)pw	5.22	sub	t					
			WATCH		6.4	sub						

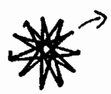
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	6.6			a)nog	6.5	sub	dc					PO
					2.17	sub	t					
					3.6	sub						

Project 7: Poly Exploration

May 19 Ses.14	7.1			a)nog			t					PO*
							dc					PO*
							noc					PO*
	7.2		SQU:SIDE :ANGLE	a)pw			t		2 Inputs			
				b)test			dc	+		stop/R		ass (ctrl G)
	7.3			a)nog	7.2	sub	dc					
	7.4		POLY	a)pw			t					
				b)test			dc	+				

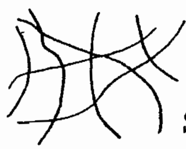
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.5			a)nog	7.4	sub	dc		TTTT			
May 20 Ses.15	7.6			a)nog	7.4	sub	dc			trace	7.6	rr
										pos	7.6	rr, cr*
										ob(p)	7.6	cr
	7.7			a)g&t			dc		WRAP NOWRAP	pos	7.7	p
			SPINSTAR	b)pw	7.4	sub	t					
				c)test			dc	-		stop/R	7.9	e
	7.8			a)g&t			dc		LEVEL			PO
				b)rev	7.4	rev	e					
				c)test			dc	-		stop/R	-	S*
	7.9			a)g&t			dc			stop/R	-	e
			POLYII	b)rev	7.8	rev	e					

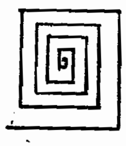
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.10			a)rev	7.7	rev	e					
				b)test			dc	-		stop/R	7.11	PO*
	7.11			a)rev	7.10	rev	e					
				b)test			dc	+		stop/S	7.11	e*
	7.12			a)rev	7.11	rev	e					PO
				b)test			dc	+				
	7.13		POLY	a)pw			t					PO
				b)test			dc	+		stop/R	7.13	ass (ctrl G)
	7.14			a)nog	7.13	sub	dc					

FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 8: Exploration With Large Numbers										
May 26 Ses.16	8.1			a)nog	7.9	sub	dc					
	8.2			a)g&t			dc					
			MOVEWEB	b)pw	7.9	sub	t			stop/S	8.2	*
				b)test			dc	+				
	8.3			a)g&t			dc			size size	8.3 8.3	cr cr
			SYMMETRY	b)pw	7.9	sub	t					
				c)test			dc	+				
	8.4			a)nog	7.9	sub	dc			ob(s)	8.4	cr
		Project 9: Demo										

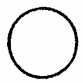
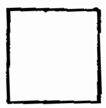
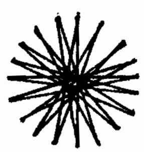
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 27 Ses.17	9.1		POLYSPI	a)pw			t		2 Inputs+			PO
				b)test			dc	-		ob	-	ass
	9.2	    		a)g&t	9.1	sub	dc			ob	-	ass

FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			BJDEMO	b)pw	9.1	sub	t					PO
				+			+					rp
				g&t	7.9	sub	dc			size	9.2	rp
				+								
				rev	7.12	sub				size	9.2	rp
												e
												
												
												
				c)test			dc	-		stop/S	9.3	e
	9.3			a)g&t			dc					PO*

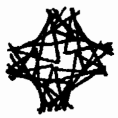
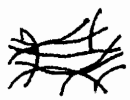
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
							t					PO
												
												
				b)rev	9.2	rev	e					e
												
				c)test			dc	+				

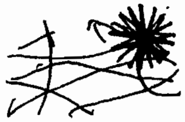
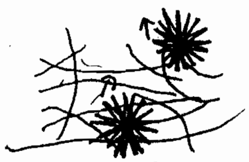
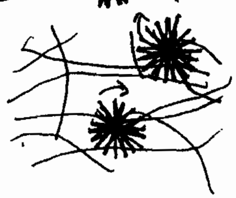
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Jun 3 Ses.18	9.4			a)g&t	7.9 8.2 8.3	sub sub sub	dc t			name	9.4	PO* cr, PO* PO*, PO* PO*
			EQUATOR	b)pw	7.9	sub	t					
				c)test			dc	-		stor	9.5	ns*, p
	9.5			a)g&t			t dc					PO*
			POLYIII	b)pw	7.9	rev	e			Rec	-	na
				c)test			dc	-		Rec	9.6	e
	9.6			a)rev	9.5	rev	e					
				b)test			dc	-		stop/R	9.7	e*
	9.7			a)rev	9.6	rev	e					PO*

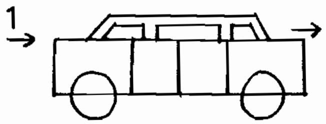
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	9.8			a)rev	9.4	rev	e					
				b)test			dc	+				
	9.9			a)g&t			t dc					PO
				b)add	9.3 8.2 8.3 9.8	rev sub sub sub	e					
				c)test			dc	-		stor	9.10	e
	9.10			a)g&t			dc					PO*
				b)rev	7.9	rev	e			stop/R	9.10	e*, PO*
				c)test			dc	+				
	9.11			a)g&t			dc					PO
				b)add	8.3	rev	e					

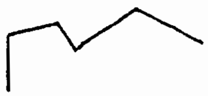
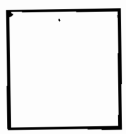
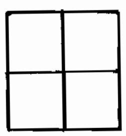
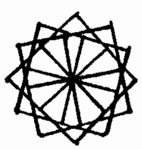
FIRST ANALYSIS - GROUP B

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-		pos	9.12	e
	9.12			a)rev	9.11	rev	e					PO*
				b)test			dc	+				
	9.13			a)test	9.9	sub	dc	+				PO*
				b)test			dc	+				

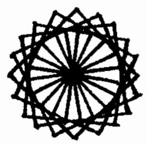
Project 10: Move Car

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	10.1			a)g&t			dc		MOVET			PO, PO*, *
				b)add	2.13	rev	e					PO
				c)test			dc	+				

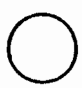
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK		MODE	EVAL	NEW MATERIAL	ERROR		DEBUG ACTION
					EPIS	TYPE				TYPE	ELIM	
		Project 1: Introduction										
Mar 18 Ses.1	1.1			a)nog			dc		FD, RT, LT, BK, CS	ob	1.1	u
	1.2			a)g&t			dc			len	1.2	ia
			SQUARE	b)pw			t		Procedure writing			
				c)test			dc	+				
	1.3			a)nog			dc					
			WINDOW	b)pw	1.2	sub	t					
				c)test			dc	+				
	1.4			a)nog			dc					
			FLOWER	b)pw	1.3	sub	t					
				c)test			dc	+				

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	1.5			a)con			dc					
			SUPER FLOWER	b)pw	1.4	sub	t		REPEAT			
				c)test			dc	+				
Mar 19 Ses.2	1.6			a)nog	1.4 1.5	sub sub	dc t		PO			PO, PO

Project 2: Logo

	2.1			a)con			noc		super, sub			
	2.2			a)g&t			dc			len len	2.2 2.2	cr cr
			ROUND	b)pw			t		Circle, Recursion			

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+	ctrl G	stop/r	2.2	ass (ctrl G)
	2.3	L		a)g&t			dc		PU, PD	len	2.3	ia
			ELL	b)pw			t					
				c)test			dc	+				
	2.4		SUPERLOGO	a)pw	2.3	sub	t			pos	-	na
				b)test			dc	-		ob(p)	2.6	e
	2.5			a)rev	2.3	rev	e		EDIT, ERL, EDL, ctrl N			PO
		L		b)test			dc	+				
Mar 24 Ses.3	2.6	O		a)g&t			dc		ARCRIGHT	ar.sh len	2.6 2.6	ia PO*, p*

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			EGG	b)pw			t					
		LO		c)test			dc	–		size	2.7	e
	2.7			a)rev	2.6	rev	e					
		LO		b)test			dc	+				
	2.8	LO		a)g&t	2.4	sub	dc			trace	2.8	*
					2.7	sub				rlo	2.8	ia
										ob(p)	2.8	cr
										trace	2.8	rc
				b)add	2.4	rev	e					PO*
					2.7	sub						
				c)test			dc	+				
	2.9	LOL		a)g&t	2.8	sub	dc		HT, ST	len	2.9	ia, cr
					2.3	sub				st.or	2.9	cr
				b)pw			noc					

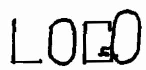
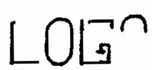
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 30	2.10			a)g&t	2.8	sub	dc		RUBDIS	trace	2.10	rr*, PO*,
Ses.4					2.3	sub						rr, rr
									HOME	len	2.10	ia
			GEE	b)pw	2.3	sub	t					
	2.11			a)add	2.8	rev	e					PO*
					2.10	sub						
												
				b)test			dc	+				
	2.12			a)g&t	2.11	sub	dc			trace	2.23	e
					2.7	sub				ob(p)	2.23	e*,PO
				b)rev	2.11	rev	e					
	2.13			a)add	2.12	rev	e					
					2.7	sub						
												
				b)test			dc	–		ob(p)	2.23	e

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK		MODE	EVAL	NEW MATERIAL	ERROR		DEBUG ACTION
					EPIS	TYPE				TYPE	ELIM	
	2.14			a)rev	2.13	rev	e					
				b)test			dc	-		ob(p)	-	post
Apr 2 Ses.5	2.15			a)g&t	2.14	rev	dc t		POTS	ob(p) ob(p)	2.15 2.15	cr e
				b)rev			e		ctrl N, C, W, R, G			
				c)test			dc	-		ob(p)	-	e
	2.16			a)rev	2.15	rev	e					
				b)test			dc	-		pos	-	e
	2.17			a)rev	2.16	rev	e					PO

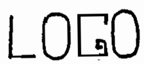
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	-				
	2.18			a)rev	2.17	rev	e noc					PO, PO* PT, PO
				b)test			dc	-				
	2.19			a)rev	2.18	rev	e					
				b)test			dc	-		ob(p)	-	e
	2.20			a)rev	2.19	rev	e		 STEP			PO, ns PO, PO
	2.21			a)g&t	2.20	rev	dc					S
				b)rev			e		ER STEP ALL			PO

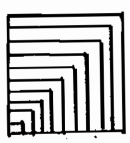
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		LOG		c)test			dc	-		name	2.21	cr
	2.22	LOG		a)g&t	2.21	rev	t			ob(p)	2.23	PO, PT
				b)rev			t					
	2.23			a)rev	2.22	rev	t					PO
		LOGO		b)test			dc	+		trace	2.23	e
	2.24			a)nog	2.23	sub	dc					PO, PO
					1.5	sub	t			ob		PO
	2.25			a)rev	2.23	rev	e					PO, PT
		LOGO		b)test			dc	-				post

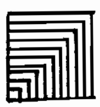
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 21 Ses.6	2.26			a)g&t	2.25	rev	dc t			 trace	 2.26	PO PT, e
				b)rev			e			typ	2.27	PO, e
				c)test			dc	–				PO
	2.27			a)rev	2.26	rev	e					
				b)test			dc	+				

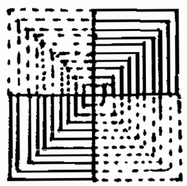
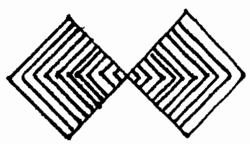
Project 3: Teaching of Variable Input

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	3.1		SQ	a)pw			noc t		TTTT Input			PO, PO
				b)test			dc	+				
	3.2			a)nog	3.1	sub	dc					

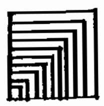
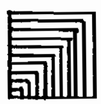
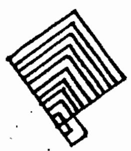
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	3.3		EMBEDSQ	a)pw	3.1	sub	noc t		Input +			
				b)tst			dc	-		stop/r	3.4	ass, e (ctrl G) e
	3.4			a)rev	3.3	rev	e		LEVEL Stop Rule			
				b)test			dc	+				
	3.5			a)nog	3.4	sub	dc					
Apr 23 Ses.7	3.6			a)nog	3.4	sub	dc		SPIN + SPIN -	1/g ob	3.6	PO, PO ass, cr* (ctrl G) S*

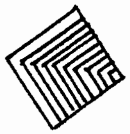
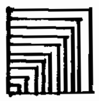
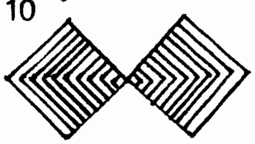
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	Project 4: Motion											
	4.1		MOTION	a)pw	3.4	sub	t			stop/r	4.1	*
				b)test			dc	+				
	Project 5: Bow											
	5.1			a)con			noc					
	5.2			a)g&t	3.4	sub	dc			st.or ob	5.2	ia, ia PO, ns, p
			DIAMOND	b)pw	3.1	sub	t			typ	-	na
				c)test			dc	-		Rec	-	na

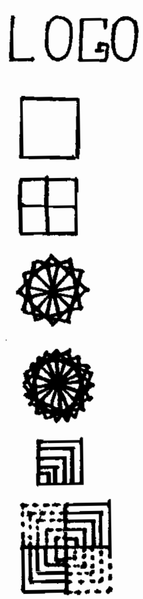
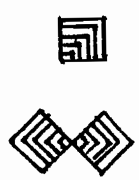
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.3			a)rev	5.2	rev	e		EDT	typ	5.3	e
				b)test			dc	-		Rec	5.4	e
	5.4			a)rev	5.3	rev	e					
				b)test			dc	+				
Apr 28 Ses.8	5.5			a)g&t	5.4	sub	dc t					PO
			BOW	b)pw	5.4	sub	t			st.or	-	na
				c)test			dc	-		ob(p) st.or	5.8 -	e na
	5.6			a)rev	5.5	rev	e			st.or	5.6	PO, PT, e

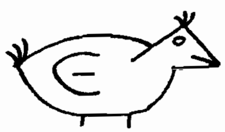
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	-		ob	-	rp
	5.7			a)g&t	5.4	sub	dc t					PO, PO
	5.8			a)g&t	5.4	sub	dc			ob(p)	5.8	S, cr, ns, e
				b)rev	5.4	rev	e					PO
				c)test			dc	+				
	5.9			a)g&t	5.6	rev	dc			pos	5.9	e
				b)rev	5.6	rev	e					
				c)test			dc	-				
	5.10			a)rev	5.9	rev	e			ang	5.10	e
				b)test			dc	+				

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 6: Demo Program										
Apr 30 Ses.9	6.1		IBMDEMO	a)pw	1.2	sub	t		WAIT			PO
					1.3	sub						
					1.4	sub						
					1.5	sub						
					2.27	sub						
					3.4	sub						
					4.1	sub						
					5.4	sub						
					5.10	sub						
				b)test			dc	+				PO, PO, PO

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 7: Decoy										
	7.1			a)nog			dc			ang ang	7.1 7.1	ia ia
	7.2			a)con			noc					
	7.3			a)g&t			dc			ang	7.3	rr, rr, rr, u*
			HEADY	b)pw			t					
				c)test			dc	+				
May 5 Ses.10	7.4			a)g&t	7.3	sub	noc dc t			size ar.d ang	7.4 7.4 7.4	PO*, PO* ia, ia rr, rr, ia

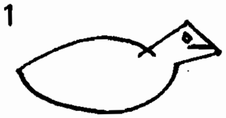
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ar.sh	7.4	ia
										ar.sh	7.4	ia*
			BODY	b)pw			t					
				c)test			dc	+				
May 7 Ses.11	7.5			a)nog	7.3	sub	dc		MOVET	1/g	7.5	*, PO
					7.4	sub	t					
	7.6			a)g&t	7.4	sub	dc			ang	7.6	PO, rr, rr, rr, rr, rr, rr, rr, rr, PO
					7.3	sub						
										u	7.6	*, PO*
										ar.sh	7.6	.rr
										len	7.6	ia
										ar.sh	7.6	ia, PO
				b)add	7.4	rev	e			cop	-	na

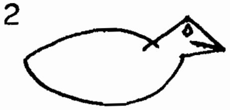
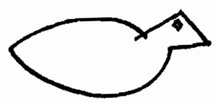
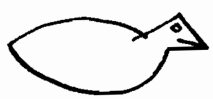
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-				
	7.7			a)g&t	7.6	rev	dc			cop	7.8	S, e
				b)rev			e					
				c)test			dc	-		cop	-	e
	7.8			a)rev	7.7	rev	e					
				b)test			dc	-		cop	-	e
	7.9			a)rev	7.8	rev	e					
				b)test			dc	+				
May 12 Ses.12	7.10			a)con	7.2	rev	noc					

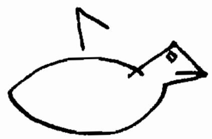
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.11			a)g&t	7.9	sub	dc		HEADING	ang	7.11	u*
					3.1	sub	noc			len	7.11	u
										name	7.11	rr, rr
										trace	7.11	rc
										ang	7.11	H*, ia
										len	7.11	ia
										ang	7.11	u
										len	7.11	u
										trace	7.11	rc
										u	7.11	ia
										pos	7.11	rr, rr, rr, cr*
										name	7.11	cr
										ang	7.11	u
										ang	7.11	ia, ia
										rlo	7.11	PT*
										len	7.11	ia
										ang	7.11	ia
										len	7.11	p
										pos	7.11	p
										ang	7.11	p
			FACEY	b)pw	3.1	sub	noc					
							t			cop	-	na

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-		cop	-	post
May 14 Ses.13	7.12			a)g&t	7.11	rev	dc			ang	7.12	PO, u, u
										len	7.12	ia, ia
										pos	7.12	rr, rr, H
										len	7.12	ia, H*
										ang	7.12	rr, ia
										len	7.12	ia, PO*
				b)rev			e			cop	-	na
				c)test			dc	-		cop	7.13	e
	7.13			a)rev	7.12	rev	e					
				b)test			dc	+				
	7.14			a)g&t	7.13	sub	dc			trace	7.14	cr
										ang		PT
										len	7.14	ia

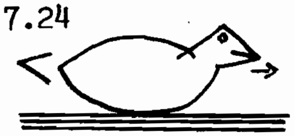
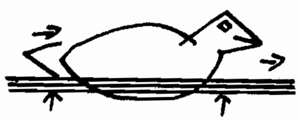
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			FLAPPY	b)pw			t			cop	-	na
	7.15		DECOY	a)pw	7.3 7.9 7.13 7.14	sub sub sub sub	t					
	7.16			a)g&t	7.15	sub	dc			1/g	7.16	*, cr, ns
				b)test			dc	-		cop	7.16	e
	7.17			a)rev	7.14	rev	e					
				b)test			dc	-		ang	-	post
May 21 Ses.14	7.18			a)g&t	7.17	rev	dc t			 ang	 7.18	PO e

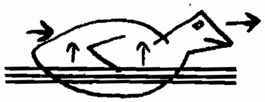
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e					
				c)test			dc	+				
	7.19			a)g&t	7.15	sub	dc			ang trace ang	7.19 7.19 7.19	H rr PT, ns
			TAILLY	b)pw			t		ERASE			
	7.20			a)con	7.10	rev	noc					
	7.21			a)g&t	7.15	sub	dc		 WRAP NOWRAP	len ang len	7.21 7.21 7.21	ia, ia ia ia, ia, ia, ia
				b)pw			noc					
May 26	7.22			a)g&t	7.15	sub	dc			ang	7.22	H

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Ses.15										trace	7.22	rc
										len	7.22	ia
			F.WATER	b)pw			t					
	7.23			a)rev	7.22	rev	e					
				b)test			dc	+				
	7.24			a)g&t	7.23	sub	dc			l/g	7.24	cr, PO
					7.15	sub						PO, PO, p
										pos	7.24	ass
			M.DECOY	b)pw	7.15	sub	t					PO
										stop/M	–	na
										pos	–	ass
				c)test			dc	–		stop/M	7.25	e*
	7.25			a)rev	7.24	rev	e		MOVET –			PO*

FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ELIM	DEBUG ACTION
	7.26			a)rev	7.18	rev	e			pos	7.26	e, PO
				b)test			dc	-		stop/M	-	na
	7.27			a)rev	7.26	rev	e			stop/M	7.27	e, PO
				b)test			dc	-				
	7.28			a)rev	7.25	rev	e					PO
	7.29			a)g&t	7.23 7.28	rev sub	dc					ns*, rp*, p*
			F.WATER	b)pw			t			trace	7.29	*
				c)test			dc	+				
	7.30			a)add	7.28 7.29	rev sub	e					PO

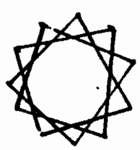
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				

Project 8: Extension of Demonstration

Jun 4 Ses.16	8.1			a)g&t	7.30	sub	dc t					PO
				b)add	6.1	rev	e					
				c)rev	7.30	sub						
				d)test			dc	-				
	8.2			a)rev	8.1	rev	e					PO

Project 9: Poly Exploration

	9.1		POLY	a)pw			t		2 Inputs			
				b)test			dc	+		stop/r	-	ass (ctrl G)

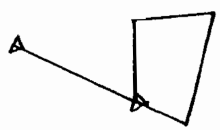
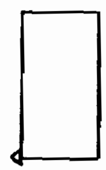
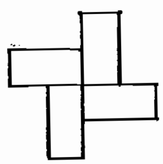
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM.	DEBUG ACTION
	9.2			a)nog	9.1	sub	dc					
	9.3			a)rev	9.1	rev	e					PO
				b)test			dc	+				
	9.4			a)nog	9.3	sub	dc					
	9.5			a)rev	9.3	rev	e		2 Inputs +			
				b)test			dc	+				
	9.6			a)nog	9.5	sub	dc					
	9.7			a)rev	9.5	rev	e					
				b)test			dc	+				

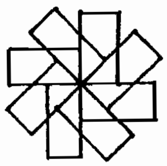
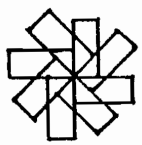
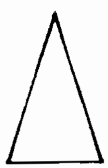
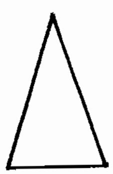
FIRST ANALYSIS - GROUP C

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	9.8			a)nog	9.7	sub	dc					

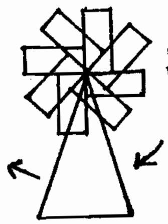
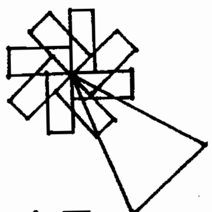
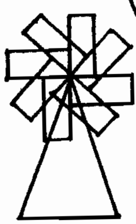
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 1: Introduction										
Mar 18 Ses.1	1.1			a)nog			dc		FD, RT, LT, BK, CS			
	1.2			a)g&t			dc		RUBDIS	ang	1.2	ia
										ang	1.2	u, PT*
										len	1.2	rr*
			RECT	b)pw			t		Procedure writing			
				c)test			dc	+				
	1.3			a)nog	1.2	sub	dc		REPEAT			PO*
			WIND	b)pw	1.2	sub	t		PO			
				c)rev			e		ERL			
				d)test			dc	+				

FIRST ANALYSIS - GROUP D

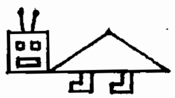
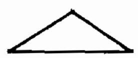
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 2: Windmill										
	2.1			a)g&t	1.3	sub	dc		SPIN + SPIN -			
Mar 19 Ses.2	2.2			a)g&t	1.3	sub				1/g stop/s	2.2 2.2	cr *
			WHEEL	b)pw	1.3	sub						
				c)test			dc	+				
	2.3			a)con			noc					
	2.4			a)g&t			dc		HOME	st.or st.or	2.4 2.4	ia, ia, cr ia, rr,

FIRST ANALYSIS - GROUP D

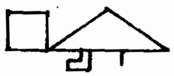
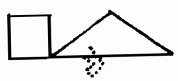
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
												u
										ang	2.4	rr, ia
										u	2.4	rr, ia
			BASE	b)pw			t		EDL	st.or	2.4	e
	2.5		SPIN WHEEL	a)pw	2.4 2.2	sub sub	t		super sub			
				b)test			dc	-		stop/s	2.6	e*, PO*
	2.6			a)rev	2.5	rev	e		EDIT			
				b)test			dc	-		pos	2.7	e, PO*
	2.7			a)rev	2.6	rev	e					
				b)test			dc	+				

Project 3: Superturtle

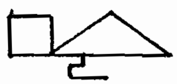
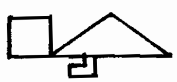
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 23 Ses.3	3.1			a)con			noc					
	3.2			a)g&t			dc			st.or	3.2	cr*
										ang	3.2	rr*, ia
										ang	3.2	ia, u, ia, ia, rr, ia
										len	3.2	ia, ia, ia, cr*
										ang	3.2	PT*, rr
									HT, ST	ang	3.2	rr, rr, rr
										len	3.2	ia, ia, ia
			SHELL	b)pw			t					
				c)test			dc	+				
	3.3			a)g&t	3.2	sub	dc		PU, PD	len	3.3	ia
										ang	3.3	ia

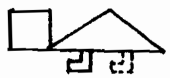
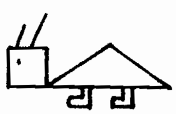
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			HEAD	b)pw			t					
				c)test			dc	+				
	3.4			a)g&t	3.2	sub	dc			len	3.4	ia
					3.3	sub				rlo	3.4	u
										trace	3.4	rr
										trace	3.4	rr, ns*
			LEGS	b)pw			t			cop	3.5	e
										cop	–	na
				c)test			dc	–				
	3.5			a)rev	3.4	rev	e			trace	3.5	PO*
				b)test			dc	–		name	3.5	cr
	3.6			a)g&t	3.5	rev	dc		STEP	rlo	3.7	S*, post

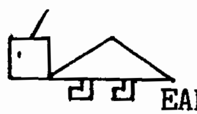
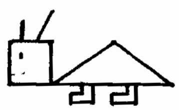
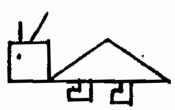
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 26 Ses.4	3.7			a)g&t	3.6	rev	dc noc			rlo		S, e
				b)rev			e					
				c)test			dc	-				
	3.8			a)g&t	3.7	rev	dc			rlo len	3.8 3.8	S, PT* ia, e, PO
				b)rev			noc e					
				c)test			dc	+	ER STEP			
	3.9			a)g&t	3.2 3.3 3.8	sub sub sub	dc			len pos	3.9 3.10	u ns*, p, e*
				b)rev	3.8	rev	e					
	3.10		MOVE	a)pw			t			trace	-	na
	3.11		SUPER	a)pw	3.2	sub	t					

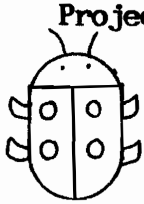
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	-		TURTLE		3.3 3.9 3.10	sub sub sub	noc					
				b)test			dc	-		trace	3.12	e
	3.12			a)rev	3.11	rev	e					
				b)test			dc	+				
	3.13			a)g&t	3.12	sub	dc			pos	3.13	*
			EYE	b)pw			t					
				c)test			dc	+		name	3.13	cr
	3.14			a)add	3.12 3.13	rev sub	e					
	3.15			a)nog	3.14	sub			MOVET	1/g	-	post
Apr 1	3.16			a)con	3.1	rev	noc					

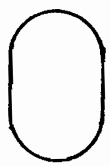
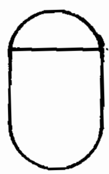
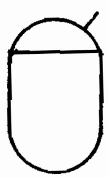
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Ses.5											
	3.17		EAR1	a)g&t	3.14 sub	dc			ang	3.17	rr, ia
				b)pw		t					
				c)test		dc	+				
	3.18		EAR2	a)g&t	3.14 sub 3.17 sub	dc			trace	3.18	rr
				b)pw		t					
				c)test		dc	+				
Apr 2 Ses.6	3.19			a)add	3.14 rev 3.17 sub 3.18 sub	e					
				b)test		dc	+				

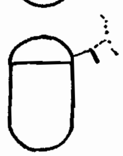
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 4: Review and Fooling Around										
				a)nog	3.19	sub	dc t		POTS MOVET -			PO
		Project 5: Ladybug										
	5.1			a)con			noc					
	5.2			a)g&t			dc		ARCRIGHT ARCLEFT	ar.sh ar.d ob(s) st.or trace ob(p)	5.2 5.2 5.2 5.2 - 5.3	ia cr cr cr, ns* ass* cr, p*
	5.3		PLACE	a)pw			t					
				b)test			dc	+				

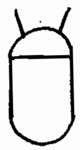
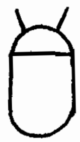
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.4			a)g&t	5.3	sub	dc			len rlo	5.4 5.4	ia, ia PT*
			BODY	b)pw			t					
				c)test			dc	+				
	5.5			a)g&t			dc			len	5.5	ia, ia, ia, ia, ia
			NECK	b)pw			t					
				c)test			dc	+				
	5.6			a)g&t			dc			ob ob trace len	5.6 5.6 5.6 5.6	da da u ia
	5.7			a)rev	5.3	rev	e			ob	5.8	ns*, e*, PO*

FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	-		ob(p)	5.8	e
	5.8			a)rev	5.7	rev	e					
				b)test			dc	+				
Apr 21 Ses.7	5.9			a)nog			dc t					PO
	5.10			a)g&t			dc			size trace len	5.10 5.10 5.10	PO* rr ia
			FEELER	b)pw			t			cop	-	na
				c)test			dc	-		ob	-	e
	5.11			a)rev	5.10	rev	e			cop	5.11	e

FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	5.12		SUPERBUG	a)pw	5.4	sub	e					
					5.5	sub						
					5.11	sub						
				b)test			dc	+				
Apr 23 Ses.8	5.13			a)g&t			dc		Circle	ang	5.13	cr
										ang	5.13	PT*, cr
										ang	5.13	cr
										ang	5.13	cr
										ang	5.13	cr
										ang	5.13	cr, cr
										re/inp	5.13	rc*
										re/inp	5.13	rc*, cr
			CIRCLE	b)pw			t					
	5.14		CIRCLE1	a)pw			t					
				b)test			dc	+				

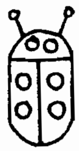
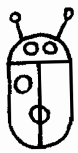
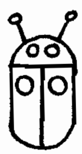
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK		MODE	EVAL	NEW MATERIAL	ERROR		DEBUG ACTION
					EPIS	TYPE				TYPE	ELIM	
	5.15			a)g&t			dc			size	5.15	rc*
			AGEDOT	b)pw			t					
	5.16			a)g&t			dc			size	5.16	rc
			DOT	b)pw			t					
	5.17		EYES	a)pw			t			name	5.17	PO
	5.18			a)g&t	5.12	sub	dc t			pos len len rlo trace	5.18 5.18 5.18 5.18 5.18	rr PO u u rc
				b)add	5.11 5.16	rev sub	e					PO*
				c)test			dc	+				

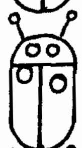
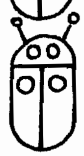
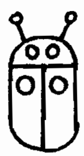
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 27 Ses.9	5.19			a)g&t	5.12 5.17	sub sub	dc t			len len len	5.19 5.19 5.19	PO*, PO ia*, ia ia
			BUGEYES	b)pw	5.17	sub	t			cop	–	na
				c)test			dc	–				
	5.20			a)g&t	5.19	rev	dc t			cop	5.20	PO, PT, e
				b)rev			e					
				c)test			dc	+				
	5.21			a)g&t	5.12 5.20	sub sub	dc			len	5.21	ia, ia, ia
			SPINE	b)pw			t					
				c)test			dc	+				
	5.22			a)add	5.12 5.20 5.21	rev sub sub	e					PO

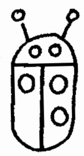
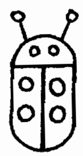
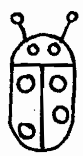
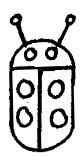
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+				
	5.23			a)con	5.1	rev	noc					
	5.24			a)g&t	5.15	sub	dc			name	5.24	PO
					5.22	sub	noc			ob(p)	5.24	cr
										len	5.24	u, c(noc)
										trace	5.24	rc
										pos	-	post
Apr 30	5.25			a)g&t	5.15	sub	dc			len	5.25	ia, ia
Ses.10					5.22	sub	t			pos	5.25	rr, rr*
										pos	5.25	rr
										trace	5.25	rr
										rlo	5.25	PT*
										len	5.25	u, ia
										pos	5.25	rr, PO
										pos	5.25	rr
										trace	5.26	ass
										pos	5.25	rr

FIRST ANALYSIS - GROUP D

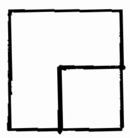
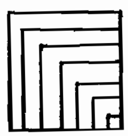
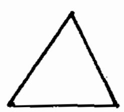
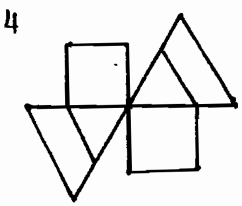
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			AGEDOTS	b)pw	5.15	sub	t			cop	-	na
				c)test			dc	-				
	5.26			a)g&t	5.25	rev	dc			cop	5.27	S, e
				b)rev			e					
				c)test			dc	-				
	5.27			a)g&t	5.26	rev	dc t					PO*, PT*
				b)rev			e			cop	-	e
				c)test			dc	+				
	5.28			a)add	5.22 5.27	rev sub	e					PO
				b)test			dc	+				

FIRST ANALYSIS - GROUP D

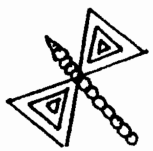
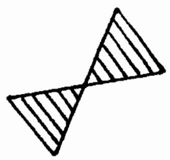
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 4 Ses.11	5.29			a)g&t	5.28	sub	dc t			len name trace	5.29 5.29 5.29	ia, ia rc rc
				b)add	5.27 5.15	rev sub	e			pos	5.31	PO, e
	5.30			a)g&t			dc			trace	5.30	rr
				b)add	5.29 5.15	rev sub	e			pos	–	e*
				c)rev	5.29	rev	e					
				d)test			dc	–		pos	–	e*
	5.31			a)rev	5.30	rev	e					
				b)test			dc	+				

Project 6: Teaching Variable Input

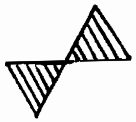
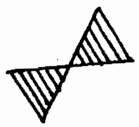
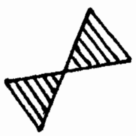
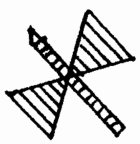
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	6.1			a)g&t			dc t					PO*
			SQUARE	b)pw			noc t		Input			
				c)test			dc	+				
	6.2			a)nog			dc					
	6.3			a)g&t			dc t			ang	6.3	PO, p
			TRIANGLE	b)pw			t					
				c)test			dc	+				
	6.4			a)nog	6.1 6.3	sub sub	dc			ang	6.4	ia

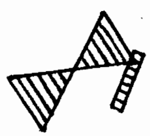
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 7: Dragonfly										
May 7 Ses.12	7.1			a)con			noc					
	7.2			a)g&t			dc			ang	7.2	ia
			EMBEDTRI	b)pw	6.3	sub	t		Recursion Input -			
				c)test			dc	-		ob	7.4	cr
	7.3			a)g&t	7.2	rev	dc		ctrl G	stop(r)	7.4	e*
				b)rev			e		Stop Rule			
				c)test			dc	-				
	7.4			a)g&t	7.3	rev	dc			ob	7.4	S*, e

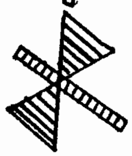
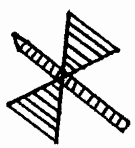
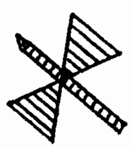
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e					
				c)test			dc	+				
	7.5			a)rev	7.4	rev	e					
				b)test			dc	+				
	7.6		WINGS	a)pw	7.5	sub	t					
				b)test			dc	+				
May 11 Ses.13	7.7			a)g&t	7.6	sub	dc t noc			st.or	7.7	PO, PO, rr, rr, rr, cr, ns
										size	7.7	cr
										trace	7.7	rc
										pos	7.7	ass
										pos	7.7	rr
										rlo	7.7	u

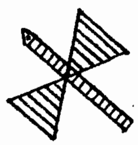
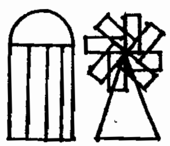
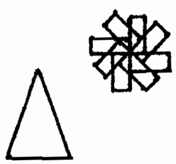
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										trace	7.7	*
										ang	7.7	ia
										u	7.7	rr
										len	7.7	ia
									HEADING	ang	7.7	ia, rr
									TTT	ang	7.7	H*, c(noc)
										ang	7.7	u, rr, rr, u
										pos	7.7	PT*, p
										trace	7.7	ass
										ang	7.7	ia
										len	7.7	ia
										pos	7.7	p*
			FBODY	b)pw	6.1	sub	noc t					
				c)test			dc	-		ob	7.9	PO, post
May 14 Ses. 14	7.8			a)g&t	7.7	rev	dc t			ob(p)	7.9	PO, S
				b)rev			e		ctrl W	rlo	7.8	S, e

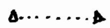
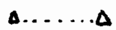
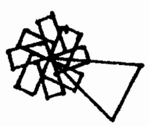
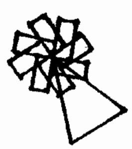
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-		pos	7.9	rp
	7.9			a)g&t	7.8	rev	dc noc			pos	-	PT*, e
				b)rev			e					PO
				c)test			dc	+		name	7.9	cr
	7.10			a)g&t	7.6 7.9	sub sub	dc			rlo	7.10	u, ia
			FHEAD	b)pw			t					
				c)test			dc	+				
	7.11			a)g&t	7.6 7.9 7.10	sub sub sub	dc			len len	7.11 7.11	u ia
			FEYES	b)pw			t					
				c)test			dc	+				

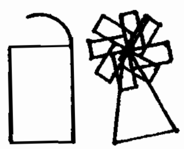
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.12		FFLY	a)pw	7.6	sub	t					
					7.9	sub						
					7.10	sub						
					7.11	sub						
				b)test			dc	+				
		Project 8: Farm Scene										
May 18 Ses.15	8.1			a)con			noc					
	8.2			a)g&t	2.7	sub	dc			rlo	8.2	cr
							t			trace	8.2	rr
										trace	8.2	rc
										ob(p)	8.2	cr
										trace	8.2	rr
										pos	8.2	PO, PO, ns, p*
			GO1	b)pw			t					

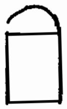
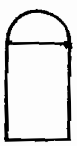
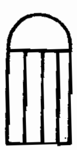
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-				
	8.3			a)rev	8.2	rev	e					
				b)test			dc	+				
	8.4			a)g&t	2.2 8.3	sub sub	dc t			stop/s stop/s	8.4 8.4	cr, PO e*
				b)rev	2.2	rev	e					
				c)test			dc	+				
	8.5			a)g&t	2.4 8.3 8.4	sub sub sub	dc			ang pos	8.5 8.6	ns, S*, rr, ia, rr, u e
				b)rev	2.4	rev	e					PO
				c)test			dc	-				
	8.6			a)rev	8.5	rev	e			pos	-	e

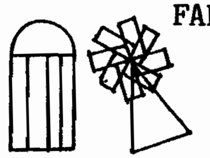
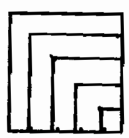
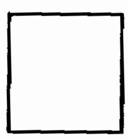
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	-				
	8.7			a)rev	8.6	rev	e			ang	8.7	e
				b)test			dc	+				
	8.8			a)g&t			dc			pos	-	ass
			WELL	b)pw	8.3	sub	t					
					8.4	sub						
					8.7	sub						
				c)test			dc	+				
May 21	8.9			a)g&t			dc			len	8.9	ia
Ses.16							t			ar.d	8.9	PT, cr
							noc			ar.d	8.9	rr, PO,
												rr, rr,
												ns*
										ar.d	8.9	p*
			SILO	b)pw			t					

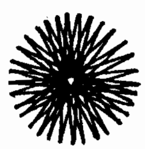
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-		ob	8.10	PO, e
	8.10			a)g&t	8.9	rev	dc t					
				b)rev			e			ob(s) pos	- -	post post
				c)test			dc	-				
May 25 Ses.17	8.11			a)g&t	8.10	rev	dc			ob(s)	8.11	e
				b)rev			e					
				c)test			dc	-		ob(p)	8.12	e
	8.12			a)rev	8.11	rev	e					
				b)test			dc	+				
	8.13	 		a)g&t	8.8	sub	dc			len	8.13	ia

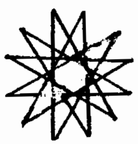
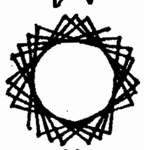
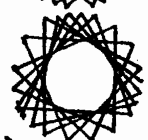
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					8.12	sub						
				b)add	8.12	rev	e					PO
				c)test			dc	+				
	8.14		FARM	a)pw	8.8 8.13	sub sub	t					
				b)test			dc	+				
		Project 9: Poly Exploration										
	9.1			a)nog	6.1 6.3	sub sub	dc t					PO PO
	9.2		POLY	a)pw			noc t		2 Inputs POLY EDT			

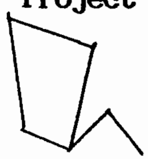
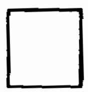
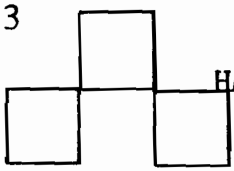
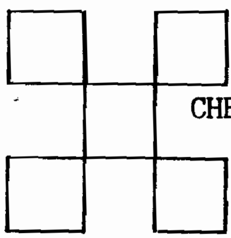
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	9.3			a)nog	9.2	sub	dc			stor	-	ass, cr
							t			ang	9.3	PO*
										stor	-	ass, cr
										stor	-	ass, cr
										stor	-	ass, cr
May 28	9.4			a)nog	9.2	sub	dc		WRAP			PO
Ses.18							noc			stor	-	ass
Jun 1	9.5			a)nog	9.2	sub	dc					PO
Ses.19							t		LEVEL			
		Project 10: Stars										
	10.1			a)g&t	9.2	sub	dc					
				b)rev	9.2	rev	e			stop/r	-	na

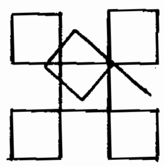
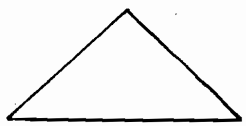
FIRST ANALYSIS - GROUP D

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK		MODE	EVAL	NEW MATERIAL	ERROR		DEBUG ACTION
					EPIS	TYPE				TYPE	ELIM	
				c)test			dc	-		stop/r	10.2	rp
	10.2			a)g&t	10.1	rev	dc			stop/r	-	S, e,
				b)rev			e					PO
				c)test			dc	+				
	10.3			a)g&t	10.2	sub	dc noc		WAIT			
												
												
			STARS	b)pw	10.2	sub	e					
				c)test			dc	+				
												
												

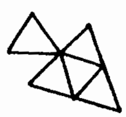
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 1:		**Introduction**								
Mar 18 Ses.1	1.1			a)nog			dc		FD, LT, RT, BK, HT, ST, CS	ob ob	1.1 1.1	da da
	1.2			a)g&t			dc			ang	1.2	*
			SQUARE	b)pw			t		Procedure writing			
				c)test			dc	+				
	1.3			a)con	1.2	sub	dc		PU, PD			
			HAT	b)pw			t					
				c)test			dc	+				
	1.4			a)con	1.3	sub	dc					
			CHECKER	b)pw			t					
				c)test			dc	+				

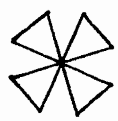
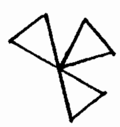
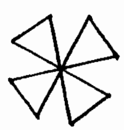
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	1.5			a)nog	1.4	sub	dc			ob(p)	1.5	cr
	1.6			a)g&t			dc			ang	1.6	*
		Project 2: Triangles										
Mar 19 Ses.2	2.1			a)nog	1.2 1.3 1.4	sub sub sub	dc					
	2.2		TRIANGLE	a)pw			t			a/s	-	na
				b)test			dc	-				

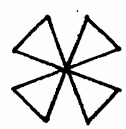
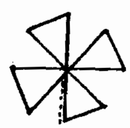
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.3			a)nog	2.2	sub	dc		SPIN			
	2.4			a)g&t	2.2	rev	dc		STEP ER STEP	 a/s	 2.4	S* e*
				b)rev			e		EDIT, EDL, ctrl N			
				c)test			dc	+				
	2.5			a)nog	2.4	sub	dc					
			TRI	b)pw	2.4	rev	e		EDT			
				c)test			dc	+	RUBDIS	ob(p)	2.5	rr*
	2.6			a)nog	2.5	sub	dc					

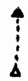
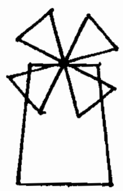
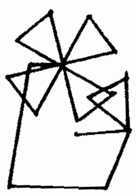
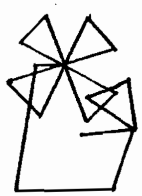
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.7			a)g&t			dc			st.or	2.7	cr
										st.or	2.7	cr
			TRI1	b)pw			t		REPEAT	st.or	2.7	cr
				c)test			dc	+				
	2.8			a)nog	2.7	sub	dc			ob(p)	2.8	rr
										ang	2.8	ia
	2.9		BARON	a)pw	2.7	sub	t			cop	–	na
				b)test			dc	–		typ	2.10	e
										cop	2.10	e
	2.10			a)rev	2.9	rev	e					
				b)test			dc	+				

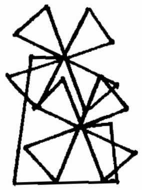
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	2.11			a)nog	2.10	sub	dc		SPIN -			
	Project 3: Windmill											
Mar 24 Ses.3	3.1			a)con			noc		Sub, Super			
	3.2			a)g&t	2.10	sub	dc t		PO	trace	3.2	PO*, cr, p*
			UP	b)pw			t			trace	-	na
				c)test			dc	-		trace	3.3	e
	3.3			a)rev	3.2	rev	e					ns*

FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	3.4			a)g&t			dc			len	3.4	ia
										ang	3.4	ia, PT*, rr, ia
										len	3.4	ia, ia, ia, ia, ia
										rlo	3.4	u
										ang	3.4	rr, ia
										len	3.4	ia
										ang	3.4	rr, ia
										rlo	3.4	u
										len	3.4	ia, ia
			SUPERWM	b)pw	3.3 2.10	sub sub	t			cop	-	na
				c)test			dc	-		ob	-	rp
	3.5			a)g&t	3.4	rev	dc					S
										cop	3.5	e
										cop	3.5	e

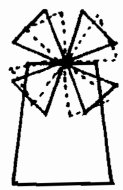
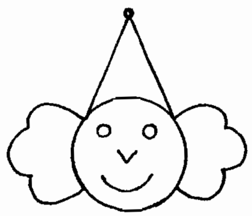
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e		ERL			
				c)test			dc	-		cop	-	post
Mar 25 Ses.4	3.6			a)g&t	3.5	rev	dc					S
										ang	3.6	e
				b)rev			e					
				c)test			dc	+				
	3.7			a)g&t			dc			pos	-	na
										l/g	-	na
										stop/r	-	na
				b)add	3.6	rev	e					
	3.8			a)rev	3.6	rev	e			pos	3.9	e
										l/g	3.8	e
										stop/r	3.9	e
				b)test			dc	-				
	3.9			a)g&t	3.8	rev	dc					S

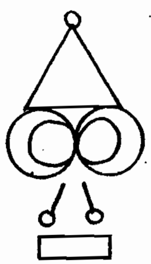
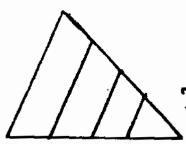
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e			pos	-	e
										stop/r	-	e
				c)test			dc	-				
	3.10			a)rev	3.9	rev	e		HOME			
				b)test			dc	-		ob	3.13	e
	3.11			a)rev	3.10	rev	e					
				b)test			dc	-		ob	-	ns*
	3.12			a)rev	3.11	rev	e					PO*,
				b)test			dc	-		ob	-	e
	3.13			a)rev	3.12	rev	e					

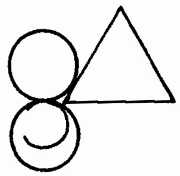
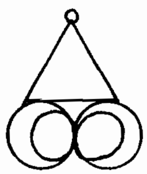
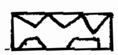
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
		Project 4: Clown										
Apr 1 Ses.5	4.1			a)con			noc					
	4.2			a)g&t			dc		POTS Circle	trace re/inp ob(s)	4.2 4.2 4.2	rc rc* rc
			CIRCLE	b)pw			t					
	4.3		CIRCLE1	a)pw			t					
	4.4		CIRCLE2	a)pw			t		TTTT			
				b)test			dc	+				PO

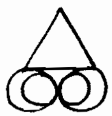
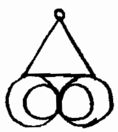
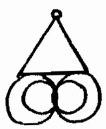
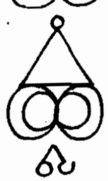
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	4.5		CIRCLE3	a)pw			t					
				b)test			dc	+				
	4.6		EYES	a)pw	4.2	sub	t					
					4.3	sub						
					4.4	sub						
					4.5	sub						
				b)test			dc	+				
	4.7			a)con	4.1	rev	noc					
	4.8			a)g&t			t					
							noc		Input			
			TRI2	b)pw			t					
				c)test			dc	+				

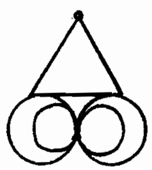
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	4.9			a)g&t	4.8	sub	dc		ctrl G	st.or	4.9	cr
										st.or	4.9	cr
										st.or	4.9	cr
										name	4.9	ass (ctrl G)
										pos	4.9	cr
										pos	-	post
Apr 2 Ses.6	4.10			a)g&t	4.8	sub	dc t		CIRCLER	ang	4.10	ia, ia
										u	4.10	ia
										st.or	4.10	PO, u, u, u
										ob(s)	4.10	rr
										ob(s)	4.10	rr
			TRI.HAT	b)pw	4.8	sub	t					
				c)test			dc	-		ob(s)	-	post
	4.11			a)con	4.7	rev	noc					

FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK		MODE	EVAL	NEW MATERIAL	ERROR		DEBUG ACTION
					EPIS	TYPE				TYPE	ELIM	
	4.12			a)g&t			dc		CIRCLEL			
			MADEYES	b)pw			noc					
Apr 21 Ses.7	4.13			a)g&t	4.10	rev	dc			ob(s)	4.14	e
				b)rev			e					PO
				c)test			dc	–		ob(s)	–	e
	4.14			a)rev	4.13	rev	e					PO,PO
				b)test			dc	+		name	4.14	cr
	4.15		SUPER CLOWN	a)pw	4.6 4.14	sub sub	t					
				b)test			dc	+				
	4.16			a)g&t	4.15	sub	dc			trace	4.16	rr

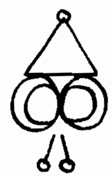
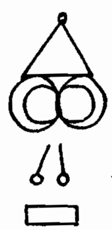
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ang	4.16	PT*
										pos	4.16	rr
										len	4.16	ia
										ang	4.16	ia
										trace	4.16	u
										u	4.16	rr, cr
	4.17		SUPERC	a)rev	4.15	rev	e					
	4.18			a)g&t	4.15	sub	dc			trace	4.18	u
										ang	4.18	ia
										rlo	4.18	u
										pos	4.18	p*
										size	4.18	p*
			NOSE	b)pw			t					
				c)test			dc	-		pos	-	post

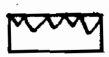
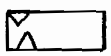
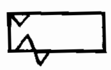
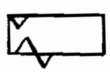
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 22 Ses.8	4.19			a)g&t	4.18	rev	dc			pos	4.19	e
				b)rev			e					
				c)test			dc	+				
	4.20			a)con			dc					
				b)add	4.19	rev	e					PO*
										cop	-	na
				c)test			dc	-		cop	4.21	e
										trace	-	na
	4.21			a)rev	4.20	rev	e					
				b)test			dc	-		trace	4.22	e
	4.22			a)rev	4.21	rev	e					
				b)test			dc	+				

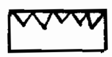
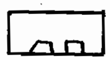
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	4.23			a)add	4.17 4.22	rev sub	e					PO
	4.24			a)rev	4.14	rev	e			ob(s)	4.24	ns, PO
				b)test			dc	+				
	4.25			a)g&t	4.23	sub	dc			pos len size	4.25 4.25 4.25	* u, PO PO, rp
			MOUTH	b)pw			t			cop	-	na
				c)test			dc	-		cop	4.26	e
	4.26			a)rev	4.25	rev	e					PO
				c)test			dc	+				

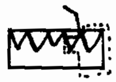
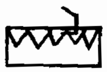
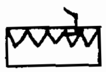
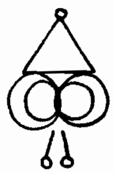
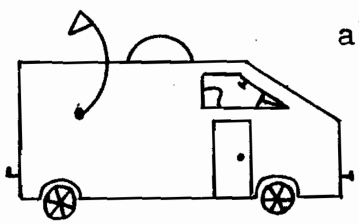
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 28 Ses.9	4.27			a)con	4.11	rev	noc					
	4.28			a)g&t	4.23	sub	dc			ang	4.28	rr
					4.8	sub				trace	4.28	rc
										ang	4.28	rr, ia
				b)add	4.26	rev	e			trace	4.28	rc
				c)test			dc	-		cop	-	na, PO*
										trace	4.28	e
										typ	-	na
	4.29			a)rev	4.28	rev	e			typ	4.29	e
				b)test			dc	-				
Apr 29	4.30			a)g&t	4.29	rev	dc			cop	4.30	e

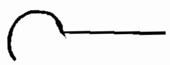
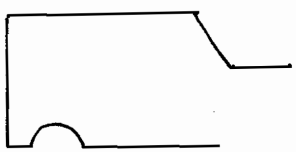
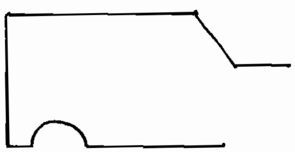
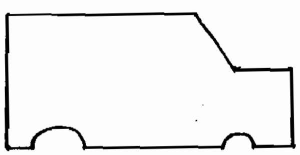
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Ses.10										ob(p)	4.30	e
				b)rev			e					
				c)test			dc	+				
	4.31			a)add	4.23	rev	e					PO
					4.30	sub						
	4.32			a)con	4.27	rev	noc					
	4.33			a)g&t	4.30	sub	dc			ang	4.33	ia
										rlo	4.33	u, ia
										len	4.33	ia
										ang	4.33	PT*
										ang	4.33	ia
										len	4.33	ia
				b)add	4.31	rev	e					PO
										cop	-	na

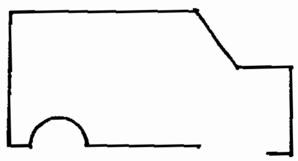
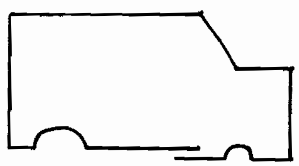
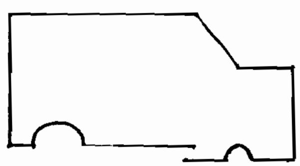
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	-		name	4.33	cr
	4.34			a)g&t	4.33	rev	noc			trace	4.35	PO, PT,
							t					S
							dc			cop	-	post
May 5	4.35			a)g&t	4.34	rev	dc					PO
Ses.11							t					S
										cop	4.35	e
										trace	4.35	e
				b)rev			e					
				c)test			dc	+		name	4.35	cr
		Project 5: Van										
	5.1			a)con			noc					

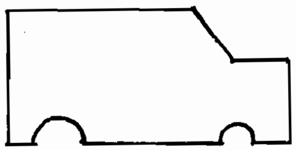
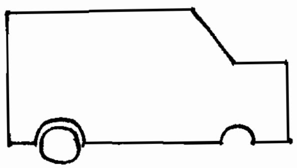
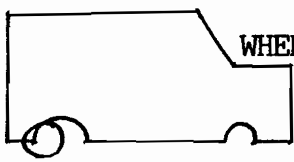
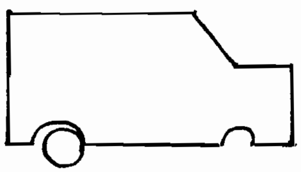
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.2			a)g&t			dc		ARCLEFT	len	5.2	ia
									ARCRIGHT	ar.d	5.2	ia
										ob(s)	5.2	rr*, rr
										pos	5.2	rr, rr, rr
				b)pw			noc					
May 6 Ses.12	5.3			a)g&t			dc			ang	5.3	PT*
										rlo	5.3	u
										len	5.3	ia, ia
										ob(p)	5.3	p*
			BODY	b)pw			t			typ	-	na
				c)test			dc	-				
	5.4			a)rev	5.3	rev	e					PO*
										typ	5.4	e
												PO
				b)test			dc	+				
	5.5			a)g&t	5.4	sub	dc			len	5.5	rr, rr,

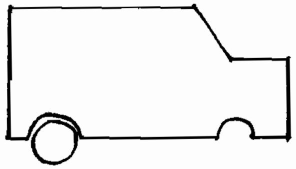
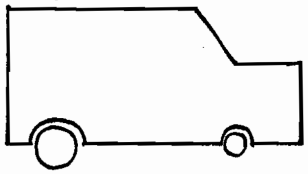
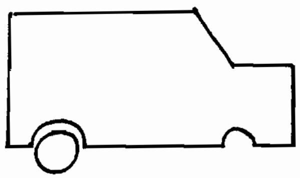
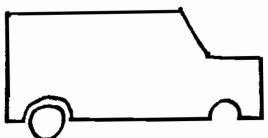
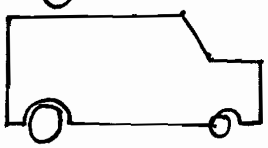
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
												rr, rr
										ar.sh	5.5	ia
										len	5.5	ia, ia, ia, ia
				b)add	5.5	rev	e					PO
										cop	–	na
				c)test			dc	–				
	5.6			a)rev	5.5	rev	e			cop	5.6	e PO, PO
				b)test			dc	–		len	–	post
May 12 Ses.13	5.7			a)g&t	5.6	rev	dc			len	5.7	rr, rr, rr, rr, rr, rr, rr, rr, PO*
							t					ia, ia, e
				b)rev			e					

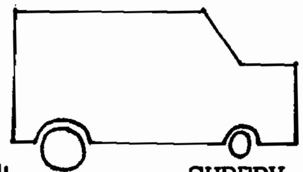
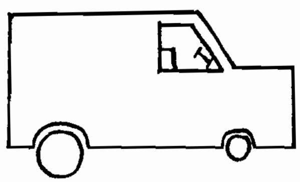
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+				
	5.8			a)g&t	5.7	sub	dc			len	5.8	ia, ia, ia
										ang	5.8	u
										rlo	5.8	ia, u
										pos	5.8	p*
			WHEELS	b)pw			t			cop	-	na
				c)test			dc	-		cop	5.9	e
	5.9			a)rev	5.8	rev	e		ctrl R ctrl C ctrl W			
				b)test			dc	-		pos	5.10	e
	5.10			a)rev	5.9	rev	e					

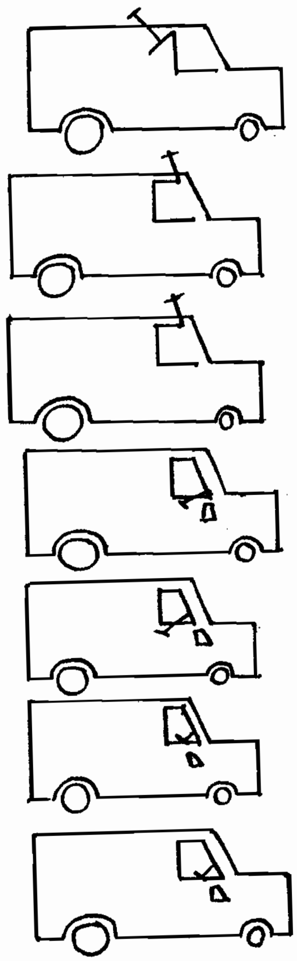
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	5.11			a)g&t	5.7 5.10	sub sub	dc			ang	5.11	PT*
										len	5.11	ia, ia, ia
										trace	5.11	rc
										pos	5.11	rr, rr, rr, rr, rr
				b)add	5.10	rev	e					PO
										cop	-	na
				c)test			dc	-		cop	-	post
May 13 Ses.14	5.12			a)g&t	5.11	rev	dc					PO, S
										name	5.12	cr
				b)rev			e			cop	5.12	e
				c)test			dc	-		pos	5.13	e
	5.13			a)rev	5.12	rev	e					

FIRST ANALYSIS - GROUP a

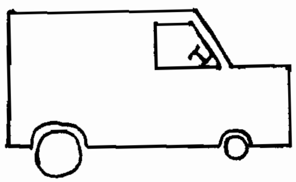
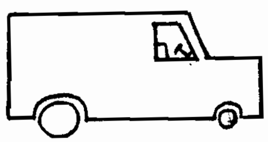
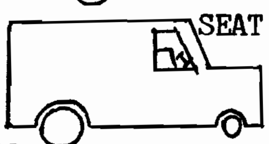
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	5.14		SUPERV	b)pw	5.7	sub	t					
					5.13	sub						
	5.15			a)g&t	5.14	sub	dc			len	5.15	ia, u, ia, ia, ia
										len	5.15	ia, ia
										len	5.15	ia
										len	5.15	ia
										len	5.15	ia, ia
										ang	5.15	ia
										len	5.15	ia
										len	5.15	ia, ia, ia
										len	5.15	ia
										len	5.15	rr
										rlo	5.15	rr
			WINDOW	b)pw			t			cop	-	na

FIRST ANALYSIS - GROUP a

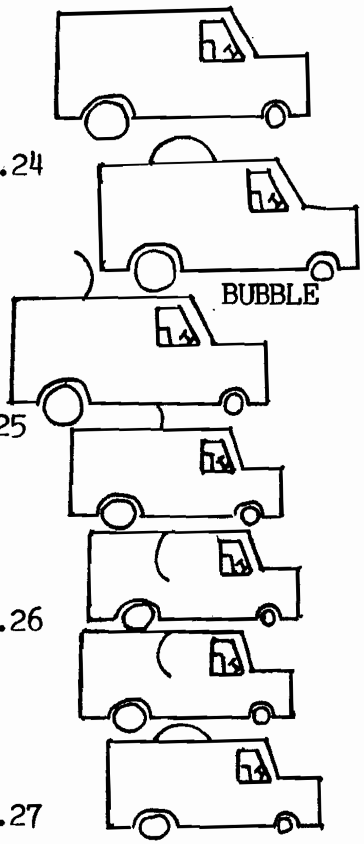


DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 19 Ses.15	5.16			a)g&t	5.15	rev	dc					PO
							t					S
							noc			cop	5.16	e
				b)rev			e					
				c)test			dc	-		name	5.16	cr
	5.17			a)g&t	5.16	rev	dc					S
										cop	5.17	e
				b)rev			e					PO*, PO
				c)test			dc	-				
	5.18			a)g&t	5.17	rev	dc					S
										cop	5.18	e, PO*
				b)rev			e			len	5.18	e
				c)test			dc	-				
	5.19			a)g&t	5.18	rev	dc					S, ns*

FIRST ANALYSIS - GROUP a

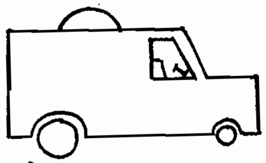
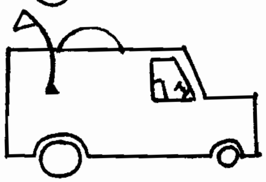
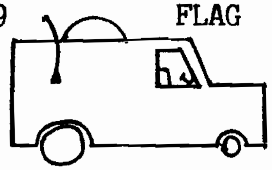
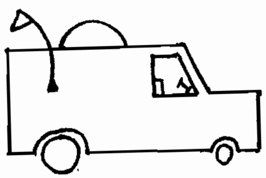
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.20			a)con	5.1	rev	noc					
	5.21			a)g&t	5.19	rev	dc			pos	5.21	*
							t			len	5.21	ia
										len	5.21	u, PO*
				b)rev			e					e*
				c)test			dc	+				
May 20 Ses.16	5.22			a)g&t	5.14	sub	dc			len	5.22	ia
					5.21	sub				len	5.22	ia
										len	5.22.	ia
				b)pw			t					
				c)test			dc	-		size	5.23	e
	5.23			a)rev	5.22	rev	e					PO

FIRST ANALYSIS - GROUP a

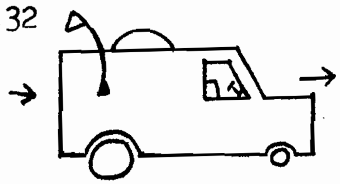
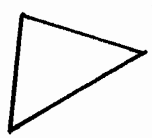


DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	5.24			a)g&t	5.14	sub	dc					
					5.21	sub						
					5.23	sub						
			BUBBLE	b)pw			t			cop	-	na
				c)test			dc	-				
	5.25			a)g&t	5.24	rev	dc					PO
				b)rev			e			cop	5.26	e
				c)test			dc	-				
	5.26			a)g&t	5.25	rev	dc			cop	-	S, e
				b)rev			e					
				c)test			dc	+				
	5.27			a)add	5.14	rev	e					
					5.21	sub						

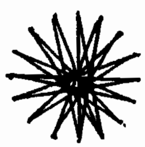
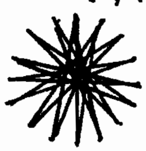
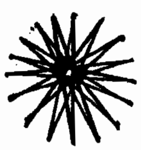
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					5.23	sub						
					5.26	sub						
				b)test			dc	+				
	5.28			a)g&t	5.27	sub	dc			ar.d	5.28	*
										st.or	5.28	rr
										ar.d	5.28	p*
				b)pw			noc					
May 26 Ses.17	5.29		FLAG	a)pw			t			cop	-	na
				b)test			dc	-		ob(p)	5.30	e*
	5.30			a)rev	5.29	rev	e					
				b)test			dc	+				
	5.31			a)add	5.27	rev	e					PO

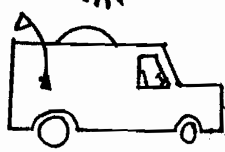
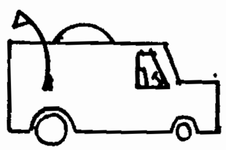
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					5.30	sub						
	5.32			a)g&t	5.31	rev	dc		MOVET +			PO
									MOVET -	1/g	5.32	cr, PO
				b)add	5.31	rev	e			pos	5.32	*
										stop/m	5.32	PO, *
				c)test			dc	+				
		Project 6: Polygons										
May 27	6.1			a)g&t			dc					PO, PO*
Ses.18							t		Recursion 2 Inputs			PO
			POLY	b)pw			noc					
							t					
				c)test			dc	+		stor	6.1	ass

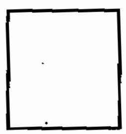
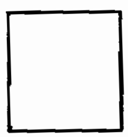
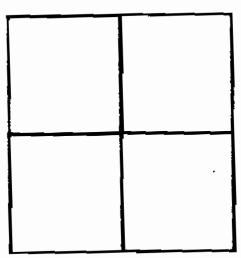
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	6.2			a)nog	6.1	sub	dc		WRAP			
Jun 3 Ses.19	6.3			a)nog	6.1	sub	dc					PO, PO
		Project 7: Sun										
	7.1			a)g&t	6.1	sub	dc			stop/r	7.1	ass (ctrl G)
			SUN	b)pw			t					
				c)test			dc	-		stop/r	7.3	PO*
	7.2			a)g&t	7.1	rev	dc		LEVEL Stop Rule			e
				b)rev	6.1	rev	e					
				c)test			dc	-		stop/r	-	e
	7.3			a)rev	7.2	rev	e					
				b)test			dc	+				

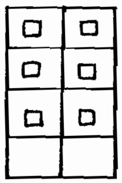
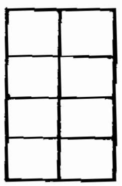
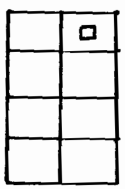
FIRST ANALYSIS - GROUP a

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.4			a)nog			dc					
	7.5			a)g&t	5.32		dc			pos	7.5	cr
					7.3	sub	t			stor	7.5	PO, ass
										pos		p*
			PLACE	b)pw								
	7.6			a)add	5.32	rev	e					PO
					7.3	sub						
					7.5	sub						
				b)test			dc	+				

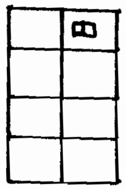
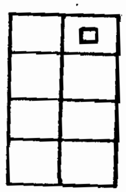
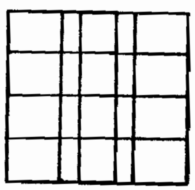
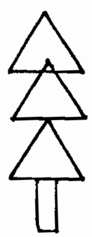
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				Project 1: Introduction								
Mar 16 Ses.1	1.1			a)nog			dc		FD, RT, BK, LT, CS			
	1.2			a)g&t			dc					
			SQUARE	b)pw			t		Procedure writing			
Mar 19 Ses.2	1.3			a)nog			dc					
	1.4			a)nog	1.2	sub	dc					

FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 2: Building										
	2.1			a)con			noc					
	2.2			a)g&t			dc		PU,PD, RUBDIS	trace len	2.2 2.2	rc rr*, ia
			BUILDING	b)pw	1.2	sub	t		REPEAT			
				c)test			dc	+	HT, ST			
	2.3			a)g&t	2.2	sub	dc			len rlo	2.3 2.3	ia, ia u, ia, ia, rr, rr, ia
			WINDOW	b)pw			t		PO, ERL			
				c)edit			e		EDIT, EDL			PO*

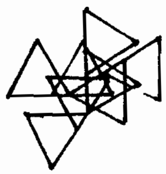
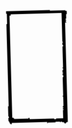
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				d)test			dc	-		pos pos	2.3 -	rc post
Mar 23 Ses.3	2.4			a)g&t	2.3	rev	dc			len ang trace	2.4 2.4 2.4	ia*, ia ia, u rr*, SP
		Project 3: CITY										
	3.1			a)g&t	2.2	sub	dc		HOME	len rlo ob(s)	3.1 3.1	ia PT* SP
		Project 4: Forest										
	4.1			a)con			noc					

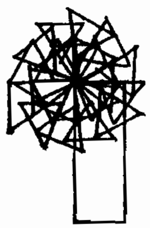
FIRST ANALYSIS – GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	4.2			a)g&t			dc			st.or	4.2	cr
										st.or	4.2	cr
										st.or	4.2	cr
										st.or	4.2	cr
										rlo	4.2	rr
										ang	4.2	ia, ia*
			SUPERTREE	b)pw			t		EDT			
							e		ctrl N			
				c)test			dc	+				
	4.3			a)g&t			dc			rlo	4.3	u*, ia
										ang	4.3	ia, ia*
										len		*
			LTREE	b)rev	4.2	rev	e					
	4.4		SUPERTREE	a)pw	4.3	sub	t		Recursion			
				b)test			dc	+				
	4.5			a)add	4.4	rev	e					PO*

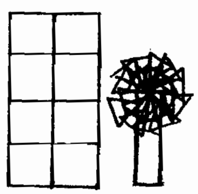
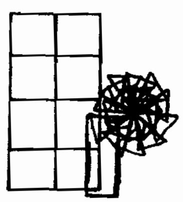
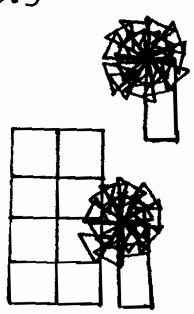
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					4.4	sub			ctrl G	stop/r	-	ass (ctrl G)
				b)test			dc	-				
	4.6			a)nog	4.5	sub	dc t					PO*
Mar 26 Ses.4	4.7			a)con			noc					
	4.8			a)g&t	4.5	rev	dc			stop/r	4.8	ass (ctrl G)
										len	4.8	ia, ns*, cr*, PO*
				b)rev			e		Stop Rule LEVEL			
				c)test			dc	+				

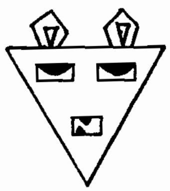
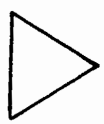
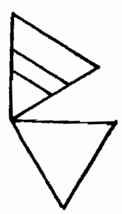
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	4.9			a)g&t	4.8	sub	dc			st.or len len len	4.9 4.9 4.9 4.9	* rr ia, ia, ia ia
			BASE	b)pw			t					
				c)test			dc	+				
	4.10			a)add	4.8 4.9	rev sub	e		Super Sub			PO*
				b)test			dc	-		stop/r	4.11	ass, (ctrl G) p*, e*
	4.11			a)rev	4.10	rev	e					
	4.12		SUPERTREE1	a)pw	4.9 4.11	sub sub	t		POTS			

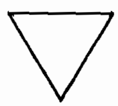
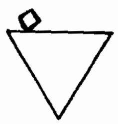
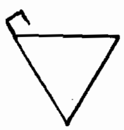
FIRST ANALYSIS – GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
		Project 5: City										
Mar 30 Ses.5	5.1			a)con			noc					
	5.2			a)g&t	2.2	sub	dc			len	5.2	ia
					4.12	sub	t			pos	5.2	cr
										name	5.2	cr*
										pos	5.3	PO*, p*
			CITY	b)pw	2.2	sub	noc					
					4.12	sub	t			pos	–	ass*
	5.3			a)g&t	4.12	rev	dc		STEP			S*
										pos	–	e*
				b)rev	4.9	rev	e					PO*
				c)test			dc	+	ER STEP			

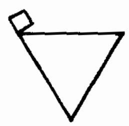
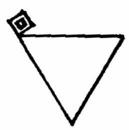
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
									ALL			
	5.4			a)nog	4.12 5.3	sub sub	dc					
		Project 6: Superpup										
Apr 1 Ses.6	6.1			a)con			noc					
	6.2			a)g&t			t dc			 size	 6.2	PO* p*
			TRIANGLE	b)pw			t		Input			
				c)test			dc	+				
	6.3			a)g&t	6.2	sub	dc			size ob(s) ob(s) ob(s)	6.3 6.3 6.3 6.3	rc* rc rc rc

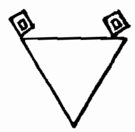
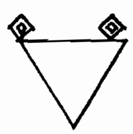
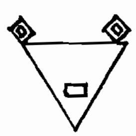
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
									PRINT	st.or	6.4	*
	6.4			a)g&t	6.2	sub	dc			st.or	6.4	*
										ob(p)	6.4	cr
										ob(p)	6.4	*
			PUPS	b)pw			t					
				c)test			dc	+				
	6.5			a)g&t	6.4	sub	dc			len	6.5	ia
							noc			ang	6.5	rr, ia, c(noc)*
										trace	6.5	u*
										len	6.5	ia
			EAR111	b)pw			t			cop	-	na
				c)test			dc	-		name	6.5	cr
										cop	6.8	post
	6.6			a)nog			dc					
							t					

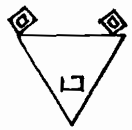
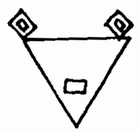
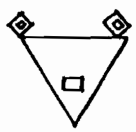
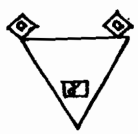
FIRST ANALYSIS – GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 20 Ses.7	6.7			a)nog			dc t					
	6.8			a)g&t	6.5	rev	dc			len	6.8	ia, ia,
				b)rev			e			cop	–	e, PO*
				c)test			dc	+				
	6.9			a)g&t			dc			ang	6.9	ia, u, ia
										pos	6.9	cr*
										ang	6.9	ia, ia*, PT*
										ang	6.9	PT*
										trace	6.9	u
										len	6.9	ia
										ob	6.9	da
										len	6.9	p*
				b)add	6.8	rev	e					PO*
				c)test			dc	+				

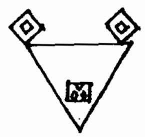
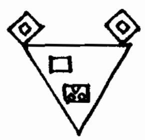
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 23	6.10			a)g&t	6.9	sub	dc			rlo	6.10	u
Ses.8					6.4	sub	noc			ang	6.10	ia, u
										len	6.10	u*
										rlo	6.10	u, ia
										ang	6.10	ia, u, ia*
										len	6.10	u
										ang	6.10	ia
										len	6.10	ia, ia, u
										pos	6.10	rr, rr
										len	6.10	u
										ang	6.10	c(t)*
			EAR222	b)pw	6.9	sub	t					
	6.11		SUPERPUP	a)pw	6.4	sub	t					
					6.9	sub						
					6.10	sub						
				b)test			dc	+				
	6.12			a)g&t			dc			st.or	6.12	*
										len	6.12	ia, PO
										size	6.12	rc*

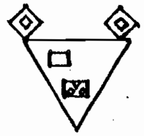
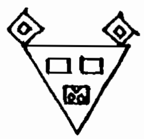
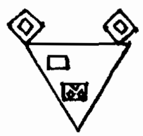
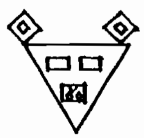
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			MOUTH	b)pw			t			typ	-	na
				c)test			dc	-		typ	6.13	e
	6.13			a)rev	6.12	rev	e					
				b)test			dc	+				
	6.14			a)add	6.11	rev	e					PO
					6.2	sub						
					6.13	sub						
				b)test			dc	+				
	6.15			a)g&t	6.2	sub	dc			len	6.15	ia
					6.14	sub				ang	6.15	*
										pos	6.15	rc*
										pos	-	PO, ass
				b)add	6.14	rev	e					
					6.2	sub						
				c)test			dc	+				

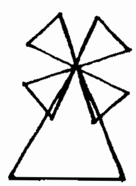
FIRST ANALYSIS – GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 27 Ses.9	6.16			a)g&t	6.15	sub	dc			trace	6.16	rc
					6.2	sub	t			len	6.16	ia
										ang	6.16	ia
												PO, PO
												PO*
										trace	6.16	rc
				b)add	6.15	rev	e			trace	6.16	e, PO*
					6.2	sub						
				c)test			dc	+				
	6.17			a)g&t	6.16	sub	dc			len	6.17	ia, u
										len	6.17	ia, rr,
												rr, ia
										ang	6.17	PT
			EYE	b)pw			t					
				c)test			dc	+				
	6.18			a)add	6.16	rev	e					PO*
					6.17	sub						

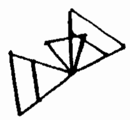
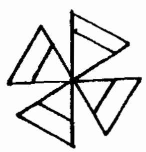
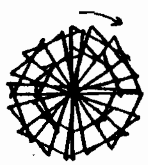
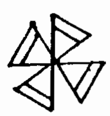
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
Apr 30 Ses.10	6.19			a)g&t	6.18	sub	dc	 t		pos	6.19	ns* ob(p) 6.19rr* PO, PO
				b)add	6.17	rev	e			cop	–	na
				c)test			dc	–		cop	6.20	e
	6.20			a)rev	6.19	rev	e					PO
				b)test			dc	+				

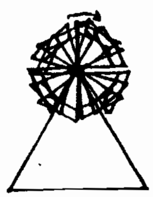
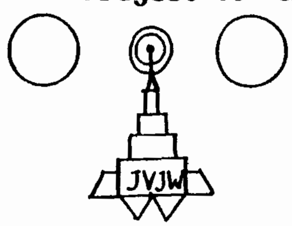
Project 7: Windmill

	7.1			a)con			noc					

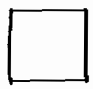
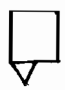
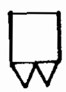
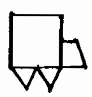
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL.	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.2			a)g&t	6.2	sub	dc			ob(s) size	7.2 7.2	rc rr, p*
			WIND	b)pw	6.2	sub	t					
				c)test			dc	+				
	7.3			a)con	7.2	sub	dc					
			WINDMILL	b)pw			t					
				c)test			dc	+				
	7.4			a)g&t	7.3	sub	dc		SPIN + SPIN -			
May 4 Ses.11	7.5			a)g&t			dc			l/g	7.5	cr
			SUPERJJ	b)pw	7.3	sub	t			typ	-	na
				c)test			dc	-				

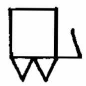
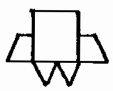
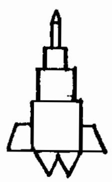
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.6			a)rev	7.5	rev	e			typ	7.6	e
				b)test			dc	+				
	7.7			a)g&t	7.6	sub	dc			stop/s	7.7	*
										ang	7.7	ia
										len	7.7	u, cr*, ns*
										pos	7.7	cr*
										ang	7.7	ia SP
		Project 8: Rocket										
	8.1			a)con			noc					
	8.2		SQU:SIDE	a)pw			t					

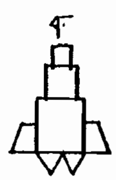
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	8.3			a)g&t	8.2 6.2	sub sub	dc			size len trace pos	8.3 8.3 8.3 8.3	cr ia rc p*
			JVJW	b)pw			t					
				c)test			dc	+				
May 7 Ses.12	8.4			a)nog	7.6	sub	dc	 t				PO
	8.5			a)g&t	8.3	sub	dc			len len ang len ang	8.5 8.5 8.5 8.5 8.5	u ia, ia ia ia u, rr*, u, rr, u
			ENGINES	b)pw			t			typ	–	na

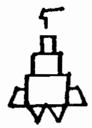
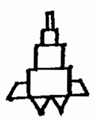
FIRST ANALYSIS – GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	–				
	8.6			a)rev	8.5	rev	e			typ	8.6	e
				b)test			dc	+				
	8.7			a)g&t	8.6	sub	dc			pos	8.7	*
										rlo	8.7	u
										rlo	8.7	PT, u, ia
										ang	8.7	ia
				b)add	8.6	sub	e					PO*
				c)test			dc	+				
	8.8			a)g&t	6.2	sub	dc			len	8.8	ia
					8.2	sub	t			name	8.8	PO*
					8.7	sub				trace	8.8	rc
										pos	8.8	ass
										trace	8.8	rc
										trace	8.8	rr*
										len	8.8	ia

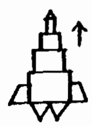
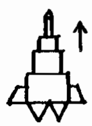
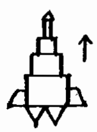
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										len	8.8	*
										pos	8.8	p*
										trace	8.8	rc
										ang	8.8	PO
										pos	8.8	rr, rr, rr, rr, rr
										pos	8.8	rr
										ang	8.8	ia, rp
										pos	8.8	rr, ia, rp, rr, ia, rp
				b)pw			noc					
May 11 Ses.13	8.9		ROCKET	a)pw	8.2	sub	t			cop	-	na
					6.2	sub						
	8.10			a)add	8.3	rev	e					PO
					8.7	sub						
					8.9	sub						
				b)test			dc	-				

FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	8.11			a)g&t	8.10	rev	dc					PO
							t			cop	8.11	S, e
				b)rev			e					
				c)test			dc	–				
	8.12			a)g&t	8.11	rev	dc					S
							noc					S
							t					S
												S, PO
				b)rev			noc					
							e					
				c)test			dc	–		pos	8.13	e
	8.13			a)rev	8.12	rev	e					
				b)test	dc		+					
	8.14			a)con	6.2	sub	dc					
					8.13	sub						

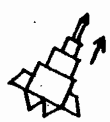
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)add	8.13 6.2	rev sub	e					
				c)test			dc	+				
	8.15			a)g&t	8.14	sub	dc		MOVET	1/g	8.16	cr
May 18 Ses.14	8.16			a)g&t	8.14	sub	dc t					PO
				b)add	8.14	rev	e					PO
				c)test			dc	+				
	8.17			a)add	8.16	rev	e					
				b)test			dc	+				
	8.18			a)g&t			dc		Circle	ang ang ang re/inp	8.18 8.18 8.18 8.18	cr cr * ia, ia,

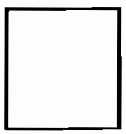
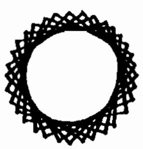
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										re/inp	8.18	ia, cr ia, ia
			ZORBOR	b)pw			t					
				c)test			dc	+				
	8.19			a)con			noc					
	8.20			a)g&t	8.18	sub				ang re/inp	8.20 8.20	u ia
				b)pw			noc					
May 25 Ses.15	8.21		ZING	a)pw			t					
	8.22			a)add	8.18 8.21	rev sub	e					PO

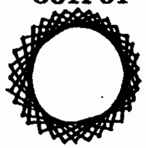
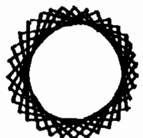
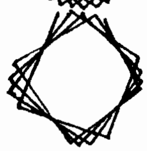
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	8.23			a)g&t			dc			len	8.23	ia
										ang	8.23	ia, ia
										ang	8.23	ia, ia, ia, u
			ZONG	b)pw			t					
	8.24			a)add	8.22	rev	e					PO
					8.23	sub						
				b)test			dc	+				
	8.25	 		a)g&t	8.17	sub	dc			pos	8.25	cr
					8.24	sub				ang	8.25	u
										len	8.25	ia
										ob(p)	8.25	ass
										ob(p)	8.25	u*
										trace	8.25	rc, cr
										len	8.25	u
										trace	8.25	cr
									WRAP	ob(p)	8.25	cr, ns*
										pos	8.25	cr

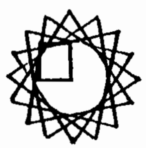
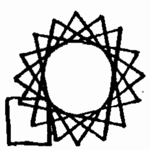
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			WV	b)pw	8.24 8.17	sub sub	t					
				c)test			dc	+				
		Project 9: Poly Exploration										
Jun 1 Ses.16	9.1		POLY	a)pw			noc	t		2 Inputs		
				b)test			dc	+		stop/r		ass (ctrl G)
	9.2			a)nog	9.1	sub	dc			stop/r		ass (ctrl G)
Jun 4 Ses.17	9.3			a)nog	9.1	sub	dc			 stop/r stop/r	 9.4	PO ass (ctrl G) e
	9.4			a)rev	9.1	rev	e					PO*

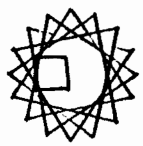
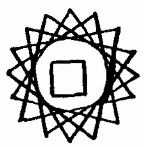
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	-				
	9.5			a)g&t	9.4	rev	dc			stop/r stop/r	- 9.5	S* e
				b)rev			e					PO
				c)test			dc	+				
	9.6			a)nog	9.5	sub	dc					

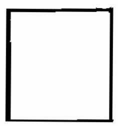
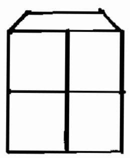
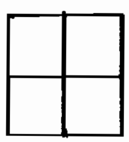
Project 10: Game of Challenge

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	10.1			a)g&t	9.5	sub	dc			len len pos	10.1 10.1 10.4	ia ia cr
			SUN	b)pw	9.5	sub	t					
	10.2			a)g&t	10.1 9.5	sub sub	dc			trace pos	10.2 10.4	cr* cr

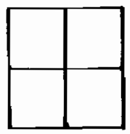
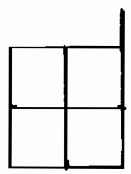
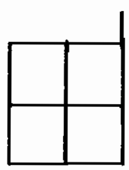
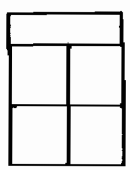
FIRST ANALYSIS - GROUP b

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	10.3			a)g&t	10.1 9.5	sub sub	dc			pos	10.4	cr
	10.4			a)g&t	10.1 9.5	sub sub	dc					

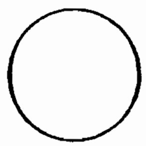
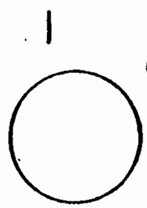
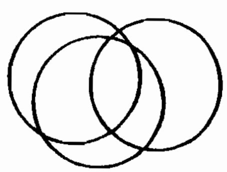
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 1: Introduction										
Mar 18 Ses.1	1.1			a)nog			dc		FD, RT, LT, CS	ob	1.1	da
	1.2			a)g&t			dc			len	1.2	ia
										ang	1.2	ia
			SHAPE	b)pw			t		Procedure writing			
				c)test			dc	+				
	1.3			a)nog	1.2	sub	dc		PU, PD	ang	1.3	u
									RUBDIS	trace	1.3	rr*
									HT, ST	ang	1.3	ia*, u*
										len	1.3	ia
										ang	1.3	rr, ia
	1.4		BOX	b)pw	1.2	sub	t			cop	-	na
				c)test			dc	-		ob	1.7	post

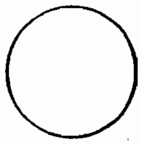
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 19 Ses.2	1.5			a)g&t	1.4	rev	dc			ob	1.5	cr
							t		PO	ob(s)	1.7	PO*, PO*, S*
									STEP	rlo	1.5	e
				b)rev			e		EDIT EDL ctrl N ER STEP ALL			
				c)test			dc	–		ob(s)	–	e
	1.6			a)g&t	1.5	rev	dc t			len	1.5	PO, e
				b)rev			e					
				c)test			dc	–		ob(s)	–	e
	1.7			a)g&t	1.6	rev	dc			len	1.7	ia, ia ia, PO*
				b)rev			e		ERL			PO, PO

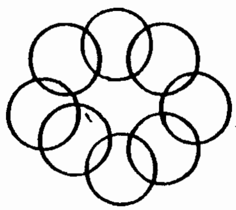
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+				
		Project 2: Circles										
	2.1			a)con			noc					
	2.2			a)g&t			dc		Circle	ang	2.2	u, ia, PT*
			CIRCLE	b)pw			t		Recursion			
				c)test			dc	+		stop/r	2.2	ass (ctrl G)
	2.3			a)nog	2.2	sub	dc			ang	2.3	ia, ia
										len	2.3	ia
										stop/r	2.3	ass (ctrl G)
										ang	2.3	ia
										stop/r	2.3	ass (ctrl G)

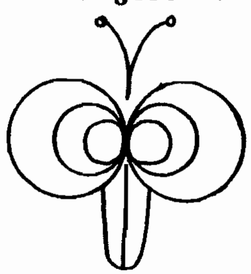
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 24 Ses.3	2.4			a)nog	2.2	sub	dc			stop/r	-	ass (ctrl G)

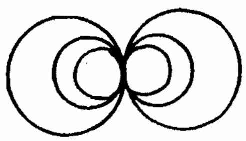
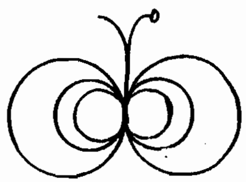
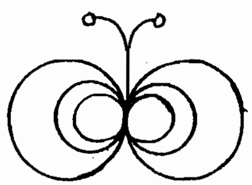
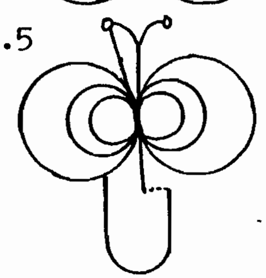
Project 3: Tubes With Circles

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	3.1			a)con			noc		CIRCLER CIRCLEL			
	3.2			a)g&t			dc			size	3.2	rc SP

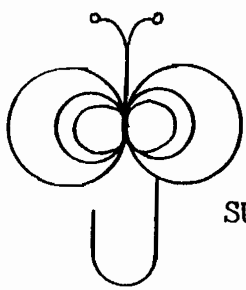
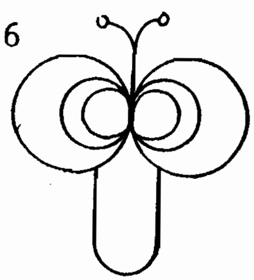
Project 4: Bug

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	4.1			a)con			noc					

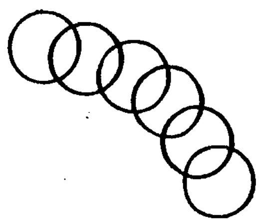
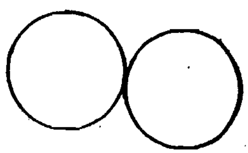
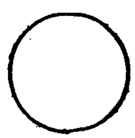
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	4.2			a)g&t			dc			ob(s)	4.2	rr*, rr, rr, rr
			EYE	b)pw			t					
				c)test			dc	+				
	4.3			a)g&t	4.2	sub	dc		ARCRIGHT	pos	4.3	rr, rr
										pos	4.3	rr, rr
										trace	4.3	cr*
									ARCLEFT	len	4.3	ia
										trace	4.3	p*
			GOOGOO	b)pw			t					
	4.4		BUGGY	a)pw	4.2 4.3	sub sub	t		sub, super			
				b)test			dc	+				
Mar 25 Ses.4	4.5			a)g&t	4.4	sub	dc noc			ang pos len	4.5 4.5	ia, PT* cr*, PT ia

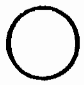
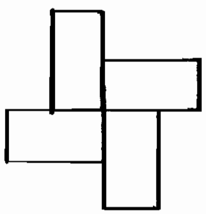
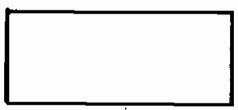
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										pos	4.5	cr
										trace	4.5	ass, p*
										len	4.5	ia
										rlo	4.5	PT*
										ang	4.5	ia
										len	4.5	ia, ia*
										ang	4.5	*
										ar.sh	4.5	ia
										len	4.5	ia
										len	4.5	rr, rr, rr
										size	4.5	rr, rr
										rlo	4.5	u, ia
										trace	4.5	rr
										ar.sh	4.5	ia
										pos	4.5	p*
			SUPEBUGGY	b)pw	4.4	sub	t			cop	–	na
				c)test			dc	–		cop	4.6	e
	4.6			a)rev	4.5	rev	e					
				b)test			dc	+				

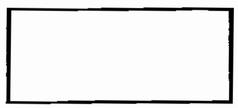
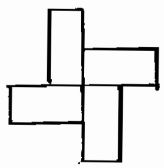
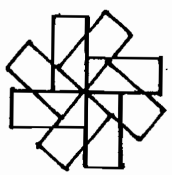
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 5: Review of Projects - Commands										
Apr 28	5.1			nog	2.2	sub	dc		POTS			PO*
Ses.5					4.4	sub	t					PO
					4.6	sub						
		Project 6: Teaching of REPEAT										
	6.1			a)con			dc noc					
	6.2			a)nog			dc		REPEAT	size	6.2	cr
										re/syn	6.2	ass (ctrl G), rc*
										re/inp	6.2	*
			CHAIN	b)pw			t			re/inp	6.2	rc
				c)test			dc	+				

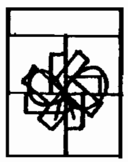
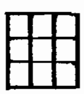
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	6.3			a)g&t			dc			pos	6.3	ass (ctrl G), PT SP
		Project 7: Windmill										
Apr 29 Ses.6	7.1			a)con			noc					
	7.2			a)g&t			dc noc			len	7.2	ia, ia, ia*
										rlo	7.2	u, ia, c(noc)
										ang	7.2	PT
			REX	b)pw			t					
				c)test			dc	-		typ	7.3	e

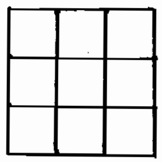
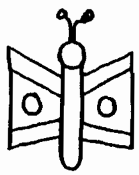
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	7.3			a)rev	7.2	rev	e					
				b)test			dc	+				
	7.4		STAR	a)g&t	7.3	sub	dc			rlo	7.4	u, ia, cr*
				b)pw	7.3	sub	t					
				c)test			dc	+				
	7.5		STARSPIN	a)g&t	7.4	sub	dc		SPIN			
				b)pw	7.4	sub	t					
				c)test			dc	+				
May 4 Ses.7	7.6			a)nog	1.6	sub	dc					
					7.2	sub	t					
					7.5	sub						

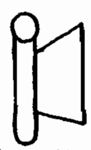
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 8: Fooling Around - Serendipity										
	8.1			a)g&t	1.7 7.5	sub sub	dc			pos	8.1	*
			SUPERSPIN	b)pw	1.7 7.5	sub sub	t					
				c)test			dc	+				
	8.2			a)nog	1.2 1.7	sub sub	dc t					PO PO
		Project 9: Teaching - Variable Input										
	9.1		SQU	a)pw			noc t		Input			PO*
				b)test			dc	+				
	9.2			a)nog	9.1	sub	dc					
	9.3			a)g&t	9.1	sub	dc			ang	9.3	PT

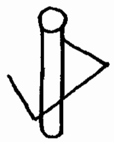
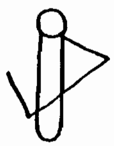
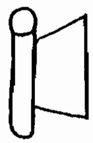
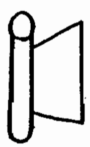
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										rlo	9.3	ia*
										trace	9.3	p
			CHEX	b)pw	9.1	sub	t					
				c)test			dc	+				
		Project 10: Butterfly										
May 5 Ses.8	10.1			a)con			noc					
	10.2			a)g&t			dc			len	10.2	ia
										size	10.2	cr
										trace	10.2	p
			BHEAD	b)pw			t					
				c)test			dc	+				

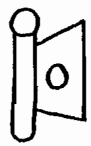
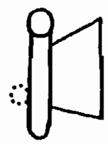
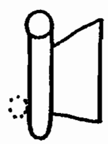
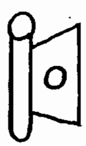
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	10.3			a)g&t			dc			rlo	10.3	rr, rr
										pos	10.3	*
										len	10.3	ia, ia, ia
										trace	10.3	u
										len	10.3	ia
										rlo	10.3	rr, rr
										ar.sh	10.3	ia
										len	10.3	ia ia
			BBODY	b)pw			t					PO*
				c)test			dc	+		name	10.3	cr
May 6 Ses.9	10.4			a)g&t			dc			len	10.4	ia
										ang	10.4	PT, u, u, ia
										ang	10.4	rr, ia, rr, u, PT*
										len	10.4	ia, ia, ia
										ang	10.4	ia, rr,

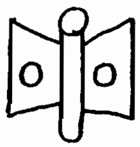
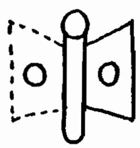
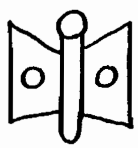
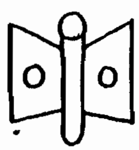
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										len	10.4	ia* ia, ia, ia
			BWING	b)pw			t			cop	-	na
				c)test			dc	-				
	10.5			a)g&t	10.4	rev	dc			cop	10.5	PO*, e
				b)rev			e					
	10.6		BBUTTER	a)pw	10.2 10.3 10.5	sub sub sub	t					
				b)test			dc	+				
	10.7			a)g&t	4.3	sub	dc			len len len ob(p) trace	10.7 10.7 10.7 - 10.7	ia ia ia, ia rp rp

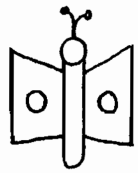
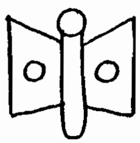
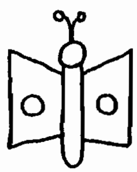
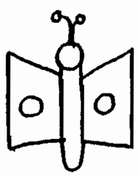
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ob(p)	–	PO*, rr, ns
										u	–	post
				b)pw			noc					
May 12 Ses.10	10.8			a)g&t			dc			name	10.8	rr*, rr, cr
							t			ang	10.8	ia, u, u
										trace	10.8	PO*
				b)add	10.5	rev	e			cop	–	na
				c)test			dc	–				
	10.9			a)g&t	10.8	rev	dc t			name	10.9	rr, cr PO
										trace	10.9	e
										ang	10.9	e
				b)rev			e					PO
				c)test			dc	+				

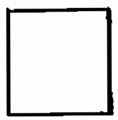
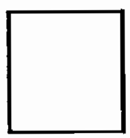
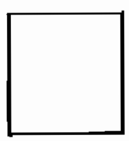
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	10.10			a)g&t			dc t			len	10.10	PO, u*, c(t), PO
										trace	10.10	u
										ang	10.10	PT*, ia, u
										len	10.10	ia
			BWING	b)pw			t			cop	10.10	e*, PO*
										cop	–	na
				c)test			dc	–				
	10.11			a)rev	10.10	rev	e			cop	10.11	e
				b)test			dc	+				
	10.12		SUPEBUTTER	a)pw	10.11	sub	t					
				b)test			dc	+				

FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 13	10.13			a)g&t	4.3	sub	dc					PO*
Ses.11							noc					
							t			trace	10.13	rr
										len	10.13	ia
										len	10.13	ia, u*, ia
										ang	10.13	ia*
										ang	10.13	u*
										len	10.13	*
										trace	10.13	rr, rr, rr
			FEELIE	b)pw	4.3	sub	t			typ	–	na
				c)test			dc	–		typ	10.14	e
	10.14			a)rev	10.13	rev	e					
				b)test			dc	+				
	10.15			a)add	10.12	rev	e					PO
					10.14	sub						

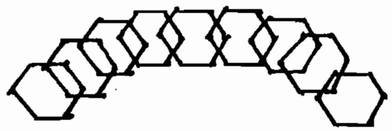
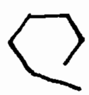
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
		Project 11: Poly Exploration										
May 19 Ses.12	11.1		SQU1	a)pw			noc t					PO*
				b)test			dc	+		stop/r	11.1	ass (ctrl G), S*
	11.2			a)g&t			dc t					PO*
	11.3		POLY	a)pw			noc t		2 inputs			
				b)test			dc	+		stor	11.3	ass
	11.4			a)nog	11.3	sub	dc			ob(p)	11.4	cr

FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										stop/r	11.4	ass (ctrl G)
	11.5			a)g&t	11.3	rev	dc		HEADING			
				b)rev			e		Stop Rule			
				c)test			dc	+				
	11.6			a)g&t	11.5	sub	dc			stop/r		e*
	11.7			a)rev	11.5	rev	e		LEVEL			PO*, PO
	11.8			a)g&t	11.7	sub	dc					

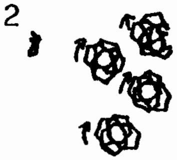
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 12: Chain										
May 20	12.1			a)g&t	11.5	sub	dc					PO
Ses.13							t			ob(p)	12.1	cr
										len	12.1	ia
										trace	12.1	rr*, cr
										ob(p)		rr, cr, ns*
			SLINKY	b)pw	11.5	sub	t			ob(p)	12.1	p*
										cop	-	na
				c)test			dc	-				
	12.2			a)g&t	12.1	rev	dc			ob(p)	12.6	PO*, e
							t					
				b)rev			e					
				c)test			dc	-				
	12.3			a)g&t	12.2	rev	dc					
				b)rev			e					

FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	–				
	12.4			a)g&t	12.3	rev	dc					
												
				b)rev			dc					
				c)test			dc	–		ob(p)	–	post
May 26 Ses.14	12.5			a)g&t	12.4	rev	dc t			 ob(p)	 –	PO cr PO, ns*, e
				b)rev			e		WRAP			e
												
				c)test			dc	+				
	12.6			a)rev	12.5	rev	e			ob(p)	–	*
				b)test			dc	+				

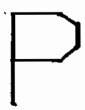
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 13: Flowers										
	13.1			a)g&t	12.6	sub	dc			l/g trace len	13.1 13.1 13.1	* rc ia
				b)pw	12.6	sub	t			typ	13.1	e, PO*
	13.2			a)rev	13.1	rev	e					
				b)test			dc	–				
	13.3			a)g&t	13.2	sub	dc t		NOWRAP	pos	–	PO post
May 27 Ses.15	13.4			a)g&t			dc t					PO SP

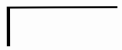
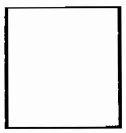
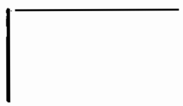
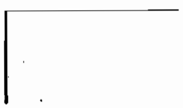
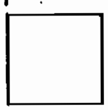
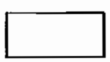
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 14: Umbie (no spin)										
	14.1			a)g&t	12.6	sub	dc			trace	14.1	rc
										pos	14.1	cr
										len	14.1	u
										trace	14.1	u, cr
										ang	14.1	ia
										trace	14.1	rr*
										len	14.1	ia
			UMBIE	b)pw	12.6	sub	t					
				c)test			dc	-				
	14.2			a)g&t			dc					PO*
							t					SP
		Project 15: Names										
	15.1			a)con			noc					

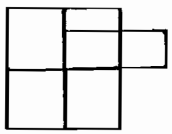
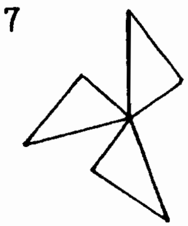
FIRST ANALYSIS - GROUP c

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	15.2			a)g&t			dc			pos	15.2	cr
										len	15.2	ia
										rlo	15.2	ia
										len	15.2	ia
			MOVE	b)pw			t					
				c)test			dc	+				
	15.3			a)g&t	15.2	sub	dc			len	15.3	ia
										rlo	15.3	u, ia,
			P	b)pw			t					
				c)test			d	+				

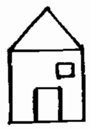
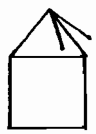
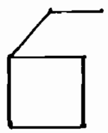
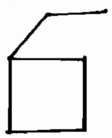
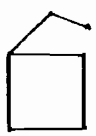
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 1:		**Introduction**								
Mar 15 Ses.1	1.1			a)nog			dc		FD, RT, CS			
	1.2			a)con			noc dc					
			SQU	b)pw			t		Procedure writing	ang a/s	1.2 1.2	* *
				c)test			dc	-		a/s	-	na
	1.3			a)g&t	1.2	rev	dc		PO	a/s	1.3	PO*, e*
				b)rev			e		EDIT			
				c)test			dc	+				
	1.4			a)g&t			dc			len ang	1.4 1.4	ia ia
			RECT	b)pw			t			ang	1.4	*

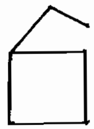
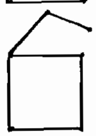
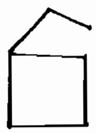
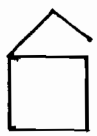
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+				
	1.5			a)nog	1.4	sub	dc					
					1.2	sub						
	1.6			a)g&t			dc			ang	1.6	ia
										len	1.6	ia
										ang	1.6	cr*
										len	1.6	ia
										len	1.6	*
			TRI	b)pw			t					
				c)test			dc	+				
	1.7			a)nog	1.6	sub	dc					

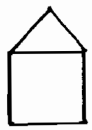
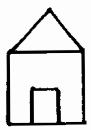
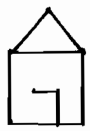
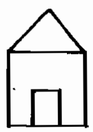
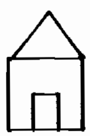
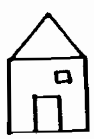
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 2: House										
	2.1			a)con			noc					
	2.2			a)g&t	1.3	sub	dc		LT, RUBDIS, BK	ang rlo len ang	2.2 2.2 2.2 2.2	u, ia u, ia rr* ia, u
			HOUSE	b)pw	1.3	sub	t			trace	2.2	e
				c)test			dc	-		cop	-	na
	2.3			a)g&t	2.2	rev	dc			cop len	2.3 -	PO, e na
				b)rev			e					
				c)test			dc	-		len	-	post

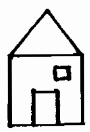
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 22 Ses.2	2.4			a)g&t	2.3	rev	dc		POTS	ang	2.7	PO, e
				b)rev			e					
				c)test			dc	–		ang	–	e
	2.5			a)rev	2.4	rev	e					
				b)test			dc	–		ang	–	e
	2.6			a)rev	2.5	rev	e					
				b)test			dc	–		ang	–	e
	2.7			a)rev	2.6	rev	e					
				b)test			dc	–		len	2.8	e
	2.8			a)rev	2.7	rev	e					

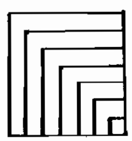
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL.	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	2.9			a)g&t	2.7	sub	dc		HOME, PU, PD	trace ang ang rlo	2.9 2.9 2.9 2.9	cr* ia ia* PT*
			DOOR	b)pw			t			cop	–	na
				c)test			dc	–		cop	2.10	e
	2.10			a)rev	2.9	rev	e					PO*
				b)test			dc	+				
	2.11			a)add	2.8 2.10	rev sub	e					PO*, PO
				b)test			dc	+				
	2.12			a)g&t	2.11	sub	dc			len	2.12	ia

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			WINDOW	b)pw			t					
				c)test			dc	+				
	2.13			a)add	2.11 2.12	rev sub	e		HT, ST			PO*, PO
				b)test			dc	+				

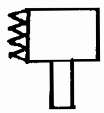
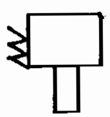
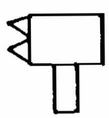
Project 3: Teaching of REPEAT

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	3.1			a)nog			dc		REPEAT, ctrl G	re/syn re/syn re/syn	3.1 3.1 3.1	ass (ctrl G) ass (ctrl G) ass (ctrl G)

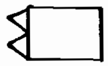
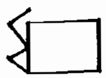
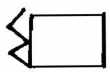
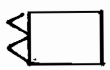
Project 4: Review

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Mar 23 Ses.3	4.1			a)nog	1.3 1.4	sub sub	dc t		sub, super			PO, PO, PO

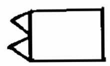
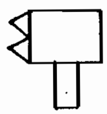
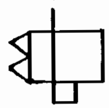
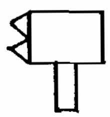
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 5: Hammer									
	5.1			a)con			noc				
	5.2			a)g&t			dc		size	5.2	PO*, cr*
									st.or	5.2	cr*
									ang	5.2	u*
									ang	5.2	*
									len	5.2	u*
									rlo	5.2	PT*
									size	5.2	cr*, ns*
				b)pw			noc				
	5.3			a)g&t	1.3	sub	dc		rlo	5.3	u, ia, PT*
									a/s	5.3	*
									ang	5.3	ia
									size	5.3	cr*, ns*
	5.4			a)con	5.1	rev	noc				

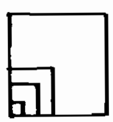
FIRST ANALYSIS – GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	5.5			a)g&t			dc			st.or	5.5	*
										len	5.5	ia
										size	5.5	cr
										len	5.5	ia*
										ang	5.5	p*
			HAMMER	b)pw	1.3	sub	t			ang	–	na
				c)test			dc	–				
	5.6			a)rev	5.5	rev	e			ang	5.7	e
				b)test			dc	–		ang	–	e
	5.7			a)rev	5.6	rev	e					
				b)test			dc	–		len	5.8	e
	5.8			a)rev	5.7	rev	e			len	5.8	ia, e

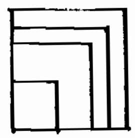
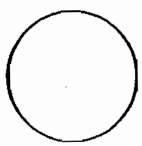
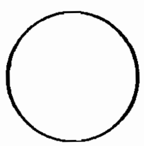
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
	5.9			a)g&t	5.8	sub	dc			len	5.9	ia
				b)add	5.8	rev	e					PO*
										cop	–	na
				c)test			dc	–		cop	5.10	e
	5.10			a)rev	5.9	rev	e					PO*
				b)test			dc	+				

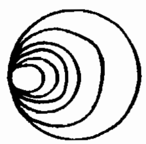
Project 6: Teaching of Variable Input

Mar 29 Ses.4	6.1			a)nog	5.10 2.13	sub sub	dc					
	6.2			a)nog			dc					

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			SQUARE	b)pw			t		input			
	6.3			a)g&t	6.2	sub	dc			ob(s)	6.3	da
		Project 7: Circles										
	7.1			a)con			noc		Circle			
	7.2			a)g&t			dc			len re/syn re/inp re/inp	7.2 7.2 7.2 7.2	cr ass (ctrl G) ia, ia, ia, ia, cr ia

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
			CIRCLE	b)pw			t					
				c)test			dc	+				
	7.3			a)g&t			dc			size	7.3	cr*
										ob(s)	7.3	da
										re/syn	7.3	ass (ctrl G)
										re/inp	7.3	rr, sp
	7.4			a)nog	6.2	sub	dc					

Project 8: Review of Previous Concepts

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 19	8.1			a)nog	7.2	sub	dc					PO
Ses.5					1.6	sub	t					PO*
					6.2	sub						

Project 9: Eyes

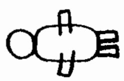
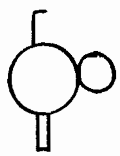
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	9.1			a)g&t			dc			re/syn re/syn (ctrl G)	9.1 9.1	ass (ctrl G) ass
			CIRC	b)pw			t			ang	-	na
	9.2		EYE	a)pw	7.2	sub	t					
				b)test			dc	-	ERASE	name	9.3	p
	9.3		EYE	a)pw	9.1	sub	t					
				b)test			dc	-				
	9.4			a)rev	9.1	rev	e		EDL, ctrl N	ang	9.4	PO*, e*
				b)test			dc	+				

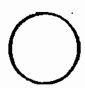
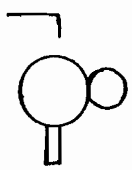
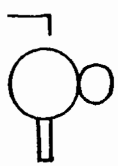
FIRST ANALYSIS – GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 10: Ice Cream Cone										
	10.1			a)con			noc					
	10.2			a)g&t	9.4	sub	dc			size	10.2	rc, cr, PT
										ang	10.2	ia*
										len	10.2	ia
										u	10.2	u
										ang	10.2	u
										len	10.2	ia
										rlo	10.2	u, ia
										ang	10.2	ia, ia, ia
										len	10.2	u
										ang	10.2	ia
			ICE	b)pw	9.4	sub	t			cop	–	na
				c)test			dc	–		cop	10.3	post

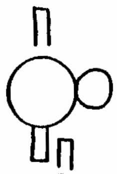
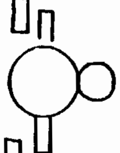
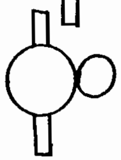
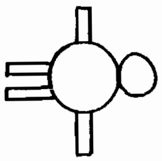
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Apr 20 Ses.6	10.3			a)g&t	10.2	rev	dc		STEP	cop	10.3	S*, cr, e
				b)rev			e					
				c)test			dc	+				
		Project 11: Person										
	11.1			a)con			noc					
	11.2			a)g&t	9.4	sub	dc		ER STEP ALL			PO*
										size	11.2	cr
										st.or	11.2	cr*
										size	11.2	cr
										size	11.2	cr
										len	11.2	ia
										size	11.2	cr
										ang	11.2	ia, PO*
										pos	11.2	*

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ang	11.2	u, u
										len	11.2	ia, ia
										rlo	11.2	u, ia
										rlo	11.2	PT*, u, ia*
										rlo	11.2	*
										len	11.2	ia
										ang	11.2	u
										trace	-	na
										ang	11.2	*
			MAN	b)pw	9.4	sub	noc t					
Apr 26 Ses.7	11.3			a)g&t	11.2	sub	dc t			trace	11.3	ass, PO*
				b)add	11.2 9.4	rev sub	e			cop	-	na
				c)test			dc	-				
	11.4			a)g&t	11.3	rev	dc t			pos cop	11.5 11.4	PO, S* e

FIRST ANALYSIS - GROUP d

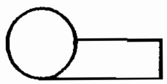
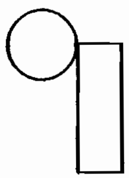
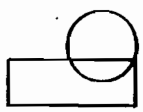
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)rev			e					
				c)test			dc	-				
	11.5			a)g&t	11.4	rev	dc					PO*
							t			pos	11.5	e, rp
				b)rev			e			pos	11.5	e, rp
				c)test			dc	+				
	11.6			a)g&t			dc			st.or	11.6	rr, rr, PT*
										rlo	11.6	rr
										rlo	11.6	PT*
										len	11.6	ia
										rlo	11.6	rr
										trace	11.6	rc
				b)add	11.5	rev	e					PO
				c)test			dc	+				PO

Project 12: Apple

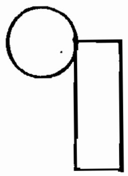
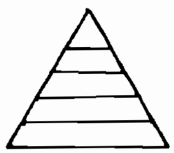
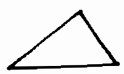
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK		MODE	EVAL	NEW MATERIAL	ERROR		DEBUG ACTION
					EPIS	TYPE				TYPE	ELIM	
May 3 Ses.8	12.1			a)con			noc					
	12.2			a)g&t	9.4	sub	dc					PO*, PO*
							t			pos	12.2	rr, cr
										st.or	12.2	cr, ns*
										trace	12.2	cr
										st.or	12.2	*
										pos	12.2	*
										re/inp	12.2	*
										len	12.2	cr*
										st.or	12.2	cr
										st.or	12.2	cr
			APPLE	b)pw	9.4	sub	t					
				c)test			dc	+				

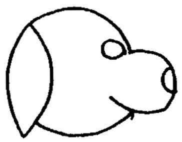
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 13: Pipe										
	13.1			a)con			noc					
	13.2			a)g&t	9.4	sub	dc					PO*, PT
							noc					c(noc)
										trace		ass
										size	13.2	cr*
										rlo	13.2	u, ia, PO
										pos	13.2	PO*, rp, p
			BALL	b)pw	9.4	sub	t					PO
										cop	-	na
				c)test			dc	-		cop	13.3	PO, e
	13.3			a)rev			e		ERL			PO

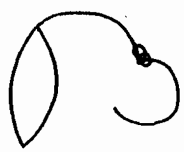
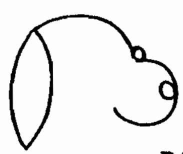
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test			dc	+				
		Project 14: Pyramid										
	14.1			a)con			noc					
	14.2			a)g&t			dc					PO*, ns*, cr
										len	14.2	ia
										ang	14.2	ia, ia
										rlo	14.2	rr
										ang	14.2	rr, ia
										ang	14.2	rr, ia
										ang	14.2	rr, ia
			TOP	b)pw			t					
				c)test			dc	+				

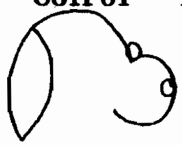
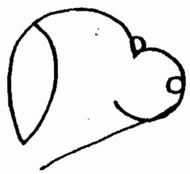
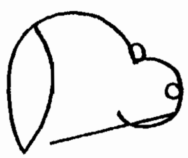
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		Project 15: Snoopy										
May 4 Ses.9	15.1			a)con			noc					
	15.2			a)g&t			dc			st.or	15.2	PT*
										ang	15.2	u
										size	15.2	cr*
										re/syn	15.2	cr
										re/inp	15.2	ia, ia, ia, ia
										ang	15.2	u
										trace	15.2	rr, rr, rr
										rlo	15.2	u
										ang	15.2	u
										re/inp	15.2	rr*
										ang	15.2	ia, ia
										size	15.2	rr, rr, rr, p*
			SNOOPY	b)pw			t					

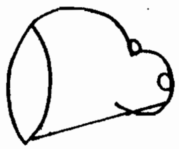
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+				
	15.3			a)g&t	9.4	sub	dc			trace	15.3	rr
					15.2	sub				st.or	15.3	u
										size	15.3	rr, rr rr, rr, rr, rr, rr, rr, rr, rr, rr
										ang	15.3	u
										pos	15.3	rr
										rlo	15.3	rr, ia
										ang	15.3	rc
										trace	15.3	ass
May 10 Ses.10	15.4			a)g&t	9.4	sub	dc			trace	15.4	cr*
					15.2	sub						
			PARTS	b)pw			t					
	15.5			a)add	15.2	rev	e					PO*
					15.4	sub						

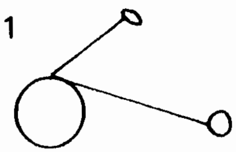
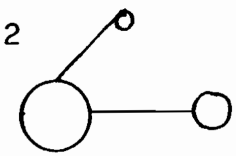
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS TYPE	MODE	EVAL.	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test		dc	+				
	15.6			a)g&t	15.5 sub	dc			rlo	15.6	PT*, u, ia, ia
									a/s	15.6	*
									trace	15.6	rr, rr*
									len	15.6	ia, ia
									ang	15.6	PT*
									ang	15.6	rr, ia
									len	15.6	ia, ia
			NECK	b)pw		t			cop	-	na
	15.7			a)add	15.5 rev 15.6 sub	e					
				b)test		dc	-				
	15.8			a)g&t	15.6 rev	dc					PO
						t			len	15.8	ia, ia, e
				b)rev		e					

FIRST ANALYSIS – GROUP d

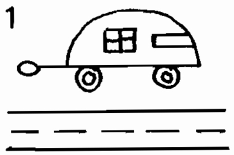
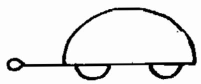
DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	+				

Project 16: Scooter

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
	16.1			a)con			noc					
	16.2			a)g&t	9.4	sub	dc			len	16.2	ia
							t			pos	16.2	PT*, cr*
										re/inp	16.2	PO*
										ang	16.2	u
										trace	16.2	rr
										len	16.2	ia
			BIKE	b)pw	9.4	sub	t					
				c)test			dc	+				

Project 17: Trailer

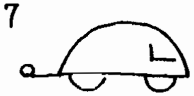
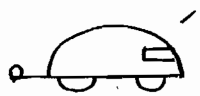
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
May 11 Ses.11	17.1			a)con			noc					
	17.2			a)g&t	9.4	sub	dc			re/inp	17.2	ia
										trace	17.2	rc
										re/syn	17.2	ass (ctrl G)
										re/inp	17.2	ia
										ang	17.2	rr, ia
										len	17.2	ia, ia, ia, ia, ia
										ang	17.2	ia
										len	17.2	ia
										rlo	17.2	u, ia
										trace	17.2	rc
										size	17.2	rr
										len	17.2	ia
			TP	b)pw	9.4	sub	t					
	17.3		WHEELS	b)pw			t					

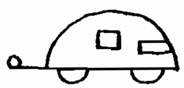
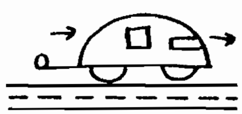
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test		dc	+				
	17.4		TRAILER	a)pw	17.2 sub 17.3 sub	t					
				b)test		dc	+				
	17.5			a)g&t		dc			len	17.5	ia
									len	17.5	ia
									len	17.5	ia, u
									len	17.5	ia
									ang	17.5	ia
									len	17.5	ia
									len	17.5	u
									len	17.5	ia
									len	17.5	ia
				b)pw		noc					
May 17 Ses.12	17.6		WIND	a)pw		t			cop	-	na

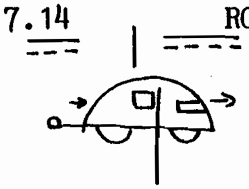
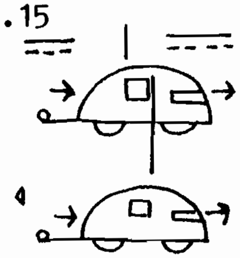
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				b)test		dc	-				
	17.7			a)g&t	17.6 rev	dc			cop	17.7	S* e
				b)rev		e					
	17.8			a)add	17.6 rev	e					
				b)test		dc	-				
	17.9			a)g&t	17.8 rev	dc			ob ang	17.9 17.9	S*, cr S, PT*, e
				b)rev		e					
				c)test		dc	+				
	17.10			a)add	17.4 rev	e					PO

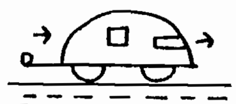
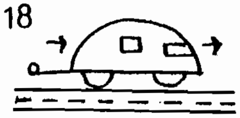
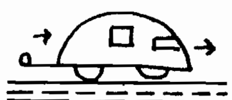
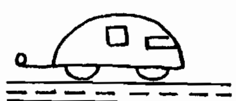
FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					17.9	sub						
				b)test			dc	+				
	17.11			a)add	17.2	rev	e		MOVET +			PO*
										stop/M		ass
												(ctrl G)
				b)test			dc	+				
	17.12			a)add	17.10	rev	e		MOVET -			
	17.13			a)g&t	17.12	sub	dc			len	17.13	ia, ia
										ob		ass
										rlo	17.13	ia, ia*
										len	17.13	u
										ang	17.13	PT*
										len	17.13	ia
										ob		ass
										rlo	17.13	u, ia

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										trace	17.13	u*
										len	17.13	rr*
				b)pw			noc					
May 18 Ses.13	17.14		ROAD	a)pw			t			cop	–	na
				b)test			dc	–				
	17.15			a)g&t	17.14	rev	dc					S
										cop	17.15	S, e
				b)rev			e					
				c)test			dc	–				
	17.16			a)g&t	17.15	rev	dc			ob	17.17	e, S
				b)rev			e			cop	17.16	e

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	–				PO
	17.17			a)g&t	17.16	rev	dc					S
							t			ob	–	e
				b)rev			e					
				c)test			dc	–				
	17.18			a)g&t	17.17	rev	dc					PO, S
							t			cop	17.18	e
				b)rev			e					
				c)test			dc	+				
	17.19			a)add	17.12	rev	e					PO
					17.18	sub						
				b)test			dc	+				
	17.20			a)nog	17.19	sub	dc					PO

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
					15.8	sub	t					PO
		Project 18: Letters										
May 25 Ses.14	18.1	CHERYL SANDRA		a)con			noc					
	18.2	C		a)g&t			dc			len rlo len len ob(p) trace size	18.2 18.2 18.2 18.2 18.2 18.2 18.2	ia, ia PT, ia ia, ia ia rr* rr* rr, rr*
			C	b)pw			t					
		C		c)test			dc	+				
	18.3	CH		a)g&t	18.2	sub	dc			len	18.3	ia

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										len	18.3	u
			"H	b)pw			t			cop	–	na
				c)test			dc	–		name	18.4	p
	18.4		H1	a)pw	18.3	rev	t			cop	–	na, post
May 31 Ses.15	18.5			a)g&t	18.2 18.4	sub sub	dc			ob	18.6	S, cr*, S
				b)rev	18.4	rev	e			cop	18.5	e
				c)test			dc	–		ob	–	rp
	18.6			a)g&t	18.5	rev	dc t					PO* S
				b)rev			e			cop	18.6	e

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
		CH		c)test			dc	-				
	18.7	CH		a)g&t	18.6	rev	dc					S
										trace	18.7	e
				b)rev			e			cop	18.7	e
		CH		c)test			dc	+				
	18.8	CHE		a)g&t	18.2	sub	dc			len	18.8	ia
					18.7	sub	t			ang	18.8	ia
										len	18.8	PO*. ia
			E	b)pw			t					
				c)test			dc	+				
	18.9	CHER		a)g&t	18.2	sub	dc			rlo	18.9	PT
					18.7	sub				trace	18.9	p
					18.8	sub				re/syn	18.9	ass (ctrl G)
										size	18.9	rr
										size	18.9	rr
										size	18.9	rr*

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										re/inp	18.9	rr*
										rlo	18.9	PT*
										ang	18.9	ia, ia, rr, ia
			R	b)pw			t			trace	-	na
										cop	-	na
	18.10		LETTERS	a)pw	18.2	sub	t					
					18.7	sub						
					18.8	sub						
					18.9	sub						
		CHEP		b)test			dc	-				
	18.11			a)rev	18.9	rev	e			trace	18.11	PO*
										cop	18.11	e
		CHER		b)test			dc	+				
Jun 1	18.12	CHERS		a)g&t	18.10	sub	dc			rlo	18.12	u, ns,

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
Ses.16							t					
										len	18.12	PO, u, u
										trace	18.12	rr
										size	18.12	p
			S	b)pw			t					
				c)test			dc	+				
	18.13	CHER SA		a)g&t	18.10	sub	dc			rlo	18.13	rr
					18.12	sub				ang	18.13	ia
										len	18.13	ia
										ang	18.13	ia, ia, rr, ia, rr, ia
										ang	18.13	ia, rr, ia
										ang	18.13	ia
										ang	18.13	rr, u, rr, ia
										trace	18.13	rr
			A	b)pw			t			cop	-	na

FIRST ANALYSIS - GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
				c)test			dc	–				
		S A										
	18.14	S A		a)g&t	18.13	rev	dc			cop	18.14	S, e
				b)rev			e					
		SA		c)test			dc	–				
	18.15	SA		a)g&t	18.14	rev	dc			cop	18.15	e
				b)rev			e					
		SA		c)test			dc	+				
	18.16	CHER		a)g&t	18.10	sub	dc			ang	18.16	u
					18.12	sub				len	18.16	ia
		SAN			18.15	sub				ang	18.16	ia, u
										trace	18.16	rr
										len	18.16	ia, u
										ang	18.16	ia, rr, ia
										len	18.16	ia

FIRST ANALYSIS – GROUP d

DATE	EPISODE	VISUAL OUTPUT	NAME OF PROC.	TYPE OF ACTIVITY	LINK EPIS	LINK TYPE	MODE	EVAL	NEW MATERIAL	ERROR TYPE	ERROR ELIM	DEBUG ACTION
										ang	18.16	ia, rr, ia, rr, ia
			N	b)pw			t			cop	–	na
	18.17	CHER		a)add	18.10 18.12 18.15 18.16	rev sub sub sub	e					PO
		SAN		b)test			dc	–				
	18.18	CHER SAN		a)g&t	18.16	rev	dc			cop	18.18	PO e
				b)rev			e					
		CHER SAN		c)test			dc	+				

Key

The following are abbreviations found in the descriptive summary tables. They are grouped according to category column headings. The categories "Error Type" and "Debugging Action" include a brief explanation of each type.

Type of Activity

add	adding
con	constructing
g&t	generate and testing
nog	no overt goal
pw	procedure writing
rev	revising
test	testing

Link Type

rev	revision
sub	subprocedure
super	superprocedure

Evaluation (eval)

+	positive
-	negative

Mode

dc	direct command
e	edit
noc	not on computer
t	text screen

Error Type		Description
ang	angle	error in input to turn commands
a/s	angle-step confusion	confusion in step and turn command
ar.d.	arc diameter	error in input to arc command
ar.sh	arc shape	error in input to arc command
cop	copying	error in copying instructions
len	length	error in input to FD command
l/g	local/global	error in perspective
name	name	confusion of procedure names
ob	out of bounds	error when T. passes screen boundary
ob(p)	out of bounds (position)	error message due to position error
ob(s)	out of bounds(size)	error message due to size error
pos	position	error in T.'s starting or ending position
rec	recursion	error in recursion
R/inp	REPEAT/input	error in input to REPEAT command
R/syn	REPEAT/syntax	error in typing of REPEAT statement
rlo	right-left orientation	error in direction
size	size	error in size of figure in proportion to rest of drawing
st.or	starting orientation	error in T.'s starting heading
stop/R	stop/Recursion	error in stopping recursive procedure
stop/S	stop/SPIN	error in stopping SPIN command
stop/M	stop/MOVET	error in stopping MOVET command
stor	storage	no storage left in workspace
typ	typing	error in typing
trace	trace	error in T.'s trace state
u	undo	error in reversing commands

Debugging Action		Description
ass	assimilate	being aware of bug but incorporating into instructions
c(noc) or c(t)	computation	action of using a calculation to correct bug
cr	clearscreen-restart	action of clearing screen and starting again
da	decremental adjustment	action of decreasing input to a command to correct error
e	edit	action of editing a procedure to correct a bug
h	heading	check heading of T. to debug
ia	incremental adjustment	action of using small increments to debug
na	not aware	not aware of a bug
ns	new strategy	using new strategy to debug
p	procedure	correct error by writing procedure
PO	print-out	printing out a procedure or titles
post	postpone	debugging is postponed
PT	play turtle	playing turtle to find error
rc	retry without clearscreen	trying another action without clearing the screen
rp	run procedure	executing procedure to locate error
rr	rubout-retry	rubbing out last command to eliminate bug
S	step	using STEP command to locate bug
SP	stop project	stopping work in a project
u	undo	reversing an action to eliminate bug
*	asterisk	action generated by experimenter
PU	PENUP	putting T.'s trace up to debug

APPENDIX B

Copies of Identification Instruments

Informal Measures

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________________ Date __________

School ______________________________ Grade ______ Age ______

Teacher or person completing this form ______________________________

How long have you known the child? ______________________________ Months.

Part I: Learning Characteristics	Seldom or never	Occasionally	Considerably	Almost always
1. Has unusually advanced vocabulary for age or grade level; uses terms in a meaningful way; has verbal behavior characterized by "richness" of expression, elaboration, and fluency.	☐	☐	☐	☐
2. Possesses a large storehouse of information about a variety of topics (beyond the usual interests of youngsters his age).	☐	☐	☐	☐
3. Has quick mastery and recall of factual information.	☐	☐	☐	☐
4 Has rapid insight into cause-effect relationships; tries to discover the how and why of things; asks many provocative questions (as distinct from informational or factual questions); wants to know what makes things (or people) "tick."	☐	☐	☐	☐
5. Has a ready grasp of underlying principles and can quickly make valid generalizations about events, people, or things; looks for similarities and differences in events, people, and things.	☐	☐	☐	☐
6. Is a keen and alert observer; usually "sees more" or "gets more" out of a story, film, etc. than others.	☐	☐	☐	☐
7. Reads a great deal on his own; usually prefers adult level books; does not avoid difficult material; may show a preference for biography, autobiography, encyclopedias, and atlases.	☐	☐	☐	☐
8. Tries to understand complicated material by separating it into its respective parts; reasons things out for himself; sees logical and common sense answers.	☐	☐	☐	☐
Add Column Total	☐	☐	☐	☐
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	☐ >	☐ >	☐ >	☐
Total				☐

CLP

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________ Date __________

School ______________________ Grade ______ Age ______

Teacher or person completing this form ______________________

How long have you known the child? ______________ Months.

Part II: Motivational Characteristics	Seldom or never	Occasionally	Considerably	Almost always
1. Becomes absorbed and truly involved in certain topics or problems, is persistent in seeking task completion. (It is sometimes difficult to get him to move on to another topic.)	☐	☐	☐	☐
2. Is easily bored with routine tasks.	☐	☐	☐	☐
3. Needs little external motivation to follow through in work that initially excites him.	☐	☐	☐	☐
4. Strives toward perfection; is self critical; is not easily satisfied with his own speed or products.	☐	☐	☐	☐
5 Prefers to work independently; requires little direction from teachers.	☐	☐	☐	☐
6. Is interested in many "adult" problems such as religion, politics, sex, race — more than usual for age level.	☐	☐	☐	☐
7. Often is self assertive (sometimes even aggressive); stubborn in his beliefs.	☐	☐	☐	☐
8. Likes to organize and bring structure to things, people, and situations.	☐	☐	☐	☐
9. Is quite concerned with right and wrong, good and bad; often evaluates and passes judgment on events, people, and things.	☐	☐	☐	☐
Add Column Total	☐	☐	☐	☐
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	☐ >	☐ >	☐ >	☐
Total				☐

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________ Date ______________

School ______________________ Grade ________ Age ________

Teacher or person completing this form ______________________

How long have you known the child? ______________ Months.

Part III: Creativity Characteristics

	Seldom or never	Occasionally	Considerably	Almost alwa
1. Displays a great deal of curiosity about many things; is constantly asking questions about anything and everything.				
2. Generates a large number of ideas or solutions to problems and questions; often offers unusual ("way out"), unique, clever responses.				
3. Is uninhibited in expressions of opinion; is sometimes radical and spirited in disagreement; is tenacious.				
4 Is a high risk taker; is adventurous and speculative.				
5. Displays a good deal of intellectual playfulness; fantasizes; imagines ("I wonder what would happen if . . ."); manipulates ideas (i.e., changes, elaborates upon them); is often concerned with adapting, improving and modifying institutions, objects, and systems.				
6. Displays a keep sense of humor and sees humor in situations that may not appear to be humorous to others.				
7. Is unusually aware of his impulses and more open to the irrational in himself (freer expression of feminine interest for boys, greater than usual amount of independence for girls); shows emotional sensitivity.				
8. Is sensitive to beauty; attends to aesthetic characteristics of things.				
9. Nonconforming; accepts disorder; is not interested in details; is individualistic; does not fear being different.				
10. Criticizes constructively; is unwilling to accept authoritarian pronouncements without critical examination.				
Add Column Total				
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	>	>	>	
Total				

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith/ Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________ Date ________

School ______________________ Grade ________ Age ________

Teacher or person completing this form ______________________

How long have you known the child? ______________________ Months.

Part IV: Leadership Characteristics

	Seldom or never	Occasionally	Considerably	Almost always
1. Carries responsibility well; can be counted on to do what he has promised and usually does it well.	☐	☐	☐	☐
2. Is self confident with children his own age as well as adults; seems comfortable when asked to show his work to the class.	☐	☐	☐	☐
3. Seems to be well liked by his classmates.	☐	☐	☐	☐
4. Is cooperative with teacher and classmates; tends to avoid bickering and is generally easy to get along with.	☐	☐	☐	☐
5. Can express himself well; has good verbal facility and is usually well understood.	☐	☐	☐	☐
6. Adapts readily to new situations; is flexible in thought and action and does not seem disturbed when the normal routine is changed.	☐	☐	☐	☐
7. Seems to enjoy being around other people; is sociable and prefers not to be alone.	☐	☐	☐	☐
8. Tends to dominate others when they are around; generally directs the activity in which he is involved.	☐	☐	☐	☐
9. Participates in most social activities connected with the school; can be counted on to be there if anyone is.	☐	☐	☐	☐
10. Excels in athletic activities; is well coordinated and enjoys all sorts of athletic games.	☐	☐	☐	☐
Add Column Total	☐	☐	☐	☐
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	☐ >	☐ >	☐ >	☐

Total ☐

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ________________________________ Date ____________

School ______________________ Grade ________ Age ________

Teacher or person completing this form ________________________________

How long have you known the child? ________________ Months.

Part V: Artistic Characteristics

	Seldom or never	Occasionally	Considerably	Almost always
1. Likes to participate in art activities; is eager to visually express ideas.	☐	☐	☐	☐
2. Incorporates a large number of elements into art work; varies the subject and content of art work.	☐	☐	☐	☐
3. Arrives at unique, unconventional solutions to artistic problems as opposed to traditional, conventional ones.	☐	☐	☐	☐
4. Concentrates for long periods of time on art projects.	☐	☐	☐	☐
5. Willingly tries out different media; experiments with a variety of materials and techniques.	☐	☐	☐	☐
6. Tends to select art media for free activity or classroom projects.	☐	☐	☐	☐
7. Is particularly sensitive to the environment; is a keen observer—sees the unusual, what may be overlooked by others.	☐	☐	☐	☐
8. Produces balance and order in art work.	☐	☐	☐	☐
9. Is critical of own work; sets high standards of quality; often reworks creation in order to refine it.	☐	☐	☐	☐
10. Shows an interest in other student's work — spends time studying and discussing their work.	☐	☐	☐	☐
11. Elaborates on ideas from other people — uses them as a "jumping off point" as opposed to copying them.	☐	☐	☐	☐
Add Column Total	☐	☐	☐	☐
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	☐ >	☐ >	☐ >	☐
Total				☐

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ____________________ Date ____________

School ____________________ Grade ______ Age ______

Teacher or person completing this form ____________________

How long have you known the child? ____________ Months.

Part VI. Musical Characteristics

	Seldom or never	Occasionally	Considerably	Almost always
1. Shows a sustained interest in music — seeks out opportunities to hear and create music.	☐	☐	☐	☐
2. Perceives fine differences in musical tone (pitch, loudness, timbre, duration.)	☐	☐	☐	☐
3. Easily remembers melodies and can produce them accurately.	☐	☐	☐	☐
4. Eagerly participates in musical activities.	☐	☐	☐	☐
5. Plays a musical instrument (or indicates a strong desire to).	☐	☐	☐	☐
6. Is sensitive to the rhythm of the music; responds through body movements to changes in the tempo of the music.	☐	☐	☐	☐
7. Is aware of and can identify a variety of sounds heard at a given moment — is sensitive to "background" noises, to chords that accompany a melody, to the different sounds of singers or instrumentalists in a performance.	☐	☐	☐	☐
Add Column Total	☐	☐	☐	☐
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	☐ >	☐ >	☐ >	☐
Total				☐

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ____________________ Date ________

School ____________________ Grade ________ Age ________

Teacher or person completing this form ____________________

How long have you known the child? ________ Months.

Part VII: Dramatics Characteristics	Seldom or never	Occasionally	Considerably	Almost alw
1. Volunteers to participate in classroom plays or skits.				
2. Easily tells a story or gives an account of some experience.				
3. Effectively uses gestures and facial expressions to communicate feelings.				
4. Is adept at role-playing, improvising, acting out situations "on the spot."				
5. Can readily identify himself with the moods and motivations of characters.				
6. Handles body with ease and poise for his particular age.				
7. Creates original plays or makes up plays from stories.				
8. Commands and holds the attention of a group when speaking.				
9. Is able to evoke emotional responses from listeners — can get people to laugh, to frown, to feel tense, etc.				
10. Can imitate others — is able to mimic the way people speak, walk, gesture.				
Add Column Total				
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	>	>	>	
Total				

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________________ Date ______________

School ______________________________ Grade ________ Age ________

Teacher or person completing this form ______________________________

How long have you known the child? ______________ Months.

Part VIII: Communication Characteristics — Precision	Seldom or never	Occasionally	Considerably	Almost alwa
1. Speaks and writes directly and to the point.				
2. Modifies and adjusts expression of ideas for maximum reception.				
3. Is able to revise and edit in a way which is concise, yet retains essential ideas.				
4. Explains things precisely and clearly.				
5. Uses descriptive words to add color, emotion, and beauty.				
6. Expresses thoughts and needs clearly and concisely.				
7. Can find various ways of expressing ideas so others will understand.				
8. Can describe things in a few very appropriate words.				
9. Is able to express fine shades of meaning by use of a large stock of synonyms.				
10. Is able to express ideas in a variety of alternate ways.				
11. Knows and can use many words closely related in meaning.				
Add Column Total				
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	>	>	>	
Total				

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________________ Date __________

School ______________________ Grade ______ Age ______

Teacher or person completing this form ______________________________

How long have you known the child? ______________ Months.

Part IX. Communication Characteristics — Expressiveness	Seldom or never	Occasionally	Considerably	Almost always
1. Uses voice expressively to convey or enhance meaning.	☐	☐	☐	☐
2. Conveys information non-verbally through gestures, facial expressions, and "body language.".	☐	☐	☐	☐
3. Is an interesting storyteller.	☐	☐	☐	☐
4. Uses colorful and imaginative figures of speech such as puns and analogies.	☐	☐	☐	☐
Add Column Total	☐	☐	☐	☐
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	☐ >	☐ >	☐ >	☐
Total				☐

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________ Date ______________

School ______________________ Grade ________ Age ________

Teacher or person completing this form ______________________

How long have you known the child? ______________ Months.

Part X: Planning Characteristics

	Seldom or never	Occasionally	Considerably	Almost alwa.
1. Determines what information or resources are necessary for accomplishing a task.				
2. Grasps the relationship of individual steps to the whole process.				
3. Allows time to execute all steps involved in a process.				
4. Foresees consequences or effects of actions.				
5. Organizes his or her work well.				
6. Takes into account the details necessary to accomplish a goal.				
7. Is good at games of strategy where it is necessary to anticipate several moves ahead.				
8. Recognizes the various alternative methods for accomplishing a goal.				
9. Can pinpoint where areas of difficulty might arise in a procedure or activity.				
10. Arranges steps of a project in a sensible order or time sequence.				
11. Is good at breaking down an activity into step by step procedures.				
12. Establishes priorities when organizing activities.				
13. Shows awareness of limitations relating to time, space, materials, and abilities when working on group or individual projects.				
14. Can provide details that contribute to the development of a plan or procedure.				
15. Sees alternative ways to distribute work or assign people to accomplish a task.				
Add Column Total				
Multiply by Weight	1	2	3	4
Add Weighted Column Totals	>	>	>	

Summary Sheet

Scales for the Rating Behavioral Characteristics of Superior Students

Joseph S. Renzulli / Linda H. Smith / Alan J. White / Carolyn M. Callahan / Robert K. Hartman

Name ______________________________ Date ______________

School ______________________ Grade ________ Age ________

Teacher or person completing this form ______________________________

How long have you known the child? ______________________ Months.

Directions. These scales are designed to obtain teacher estimates of a student's characteristics in the areas of learning, motivation, creativity, leadership, art, music, drama, communication and planning. The items are derived from the research literature dealing with characteristics of gifted and creative persons. It should be pointed out that a considerable amount of individual differences can be found within this population; and therefore, the profiles are likely to vary a great deal. Each item in the scales should be considered separately and should reflect the degree to which you have observed the presence or absence of each characteristic. Since the 10 dimensions of the instrument represent relatively different sets of behaviors, the scores obtained from the separate scales should not be summed to yield a total score. Please read the statements carefully and place an X in the appropriate place according to the following scale of values:

1. If you have **seldom or never** observed this characteristic.
2. If you have observed this characteristic **occasionally**.
3. If you have observed this characteristic to a **considerable** degree.
4. If you have observed this characteristic **almost all of the time**.

Space has been provided following each item for your comments.

Scoring. Separate scores for each of the ten dimensions may be obtained as follows:

— Add the total number of X's in each column to obtain the "Column Total."
— Multiply the Column Total by the "Weight" for each column to obtain the "Weighted Column Total."
— Sum the Weighted Column Totals across to obtain the "Score" for each dimension of the scale.
— Enter the Scores below.

I Learning Characteristics .... ________
II Motivational Characteristics .... ________
III Creativity Characteristics .... ________
IV Leadership Characteristics .... ________
V Artistic Characteristics .... ________
VI Musical Characteristics .... ________
VII Dramatics Characteristics .... ________
VIII Communication Characteristics — Precision .... ________
IX Communication Characteristics — Expressiveness .... ________
X Planning Characteristics .... ________

PARENT INVENTORY

NAME________________________________ DATE________________
SCHOOL______________________________ GRADE_______________
BIRTHDATE_________________

A. What special talents or skills does your child have?_______________

____

Give examples of behavior that illustrates this.________________

____

B. Check the following items that best describe your child as you see him or her.

	Little	Some	A Great Deal
1. Is alert beyond his years			
2. Likes school			
3. Has interests of older children or of adults in games and reading material			
4. Sticks to a project once it is started			
5. Is observant			
6. Has lots of ideas to share			
7. Has many different ways of solving problems			
8. Is aware of problems others often do not see			
9. Uses unique and unusual ways of solving problems			
10. Wants to know how and why			
11. Likes to pretend			
12. Other children call him/her to initiate play activities			
13. Asks a lot of questions about a variety of subjects			
14. Is not concerned with detail			
15. Enjoys and responds to beauty			
16. Is able to plan and organize activities			
17. Has above average coordination, agility and ability in organized games.			
18. Often finds and corrects mistakes			
19. Others seem to enjoy his/her company			
20. Makes up stories and has ideas that are unique			

PARENT INVENTORY (cont'd)

21. Has a wide range of interests
22. Gets other children to do what he/she wants
23. Likes to play organized games and is good at them
24. Enjoys other people and seeks them out
25. Is able and willing to work with others
26. Sets high standards for self
27. Chooses difficult problems over simple ones
28. Is able to laugh at himself (if necessary)
29. Likes to do many things and participates whole-heartedly
30. Likes to have his/her ideas known

C. Reading interests (favorite type of book and/or titles of favorite books)

D. Favorite school subject

E. General attitude toward school

F. Favorite playtime, leisure time activity

G. Hobbies and special interests (collections, dancing, making models, swimming, singing, painting, cooking, sewing, drama, etc.)

H. What special lessons, training or learning opportunities does your child have outside of school

I. What are some of the influences at home or at school that may negatively influence your child's performance in school

J. What other things would you like us to know that would assist us in planning a program for your child

PEER INVENTORY

1. Pretend our class found a puppy in the playground.

a) Which two students would be most likely to think up the most unusual names for the puppy?

_________________ _________________

b) Which two would be most likely to write the most interesting story about the puppy?

_________________ _________________

c) Which two would probably think up different ways to teach the puppy a trick?

_________________ _________________

d) If we design a collar for our puppy which two students would probably come up with the best design?

_________________ _________________

e) Which two students would give the teacher the best reasons for allowing the dog to come in the classroom?

_________________ _________________

f) Which two students would make the best suggestions of what could be done with the puppy?

_________________ _________________

2. You are learning to play a new game in the gym.

a) Which two students would probably learn the quickest?

_________________ _________________

b) Which two students would be most able to help others who were having difficulties?

_________________ _________________

c) Which two students would make the best captains?

_________________ _________________

d) Which two students would probably suggest a way to improve the game?

_________________ _________________

e) Which two students would be most able to teach the new game to another class?

_________________ _________________

f) Which two students would make the best referee and be able to settle arguments?