Transactions of Illinois Academy of Sciences
(1982), Volume 75, 3 and 4, pp. 247-252

COMPUTER PROJECTS FOR ATTITUDINAL IMPROVEMENT

Patrick Lamont
QIS Department
Western Illinois University
Macomb, Illinois 61455

INTRODUCTION

This paper discusses the positive effects of software development projects on beginning and intermediate students of computing. Projects include computer assisted instruction (CAI), graphics, business, and mathematical applications. The taxonomic level of internalization (receiving, responding, valuing, or organizing) of students before and after the completion of each project was studied, and an analysis was made of affective factors in the projects conducive to attitudinal improvement. The factors discovered are useful to teachers in planning interesting and stimulating programming projects. Here attitudinal improvement means making the state of mind and feelings of students more productive and valuable with respect to computing.

The student using a computer has a cultural advantage over his peers. Whether using a personal microcomputer, batch processing facilities, or time-sharing terminals, the student is confronted with an intellectual environment that challenges and reinforces his capabilities. The use of computer graphics as a lecture aid in undergraduate mathematics, as described by Porter [5], provides such a rich environment. In our study it was observed that the student able to complete projects creating a stimulating environment for himself showed attitudinal improvement of a high order.

An element of discovery is one ingredient for attitudinal improvement. A recent report [2] states that there is a strong relationship between discovery and attitudinal improvement. Teachers were asked to develop courseware modules based on modeling and the "closed simulation" (guided discovery) or the "open simulation" (investigative) mode rather than produce software for mere programmed learning. Programmed learning tends to inhibit attitudinal improvement because of its impersonal aspects. In our study advances were sought in the area of humanizing and personalizing the CAI interface of man and machine by involving some of our students in the development of software to aid in scientific rediscovery and discovery.

The investigation was conducted recently with the help of three groups of students: (1) a class of thirty students taking an Introduction to Computers course,

*This material is based upon work supported by the National Science Foundation under Grant Number SPI-7901345.

(2) a group of twenty students attending an External File Structures course, and (3) a group of twenty high school teachers of mathematics attending an NSF funded summer workshop at Monmouth College, Illinois.*

OBJECTIVES

Attitudinal improvement corresponds to the affective, as opposed to cognitive, domain of Krathwohl, Bloom, and Masia [4]. For any course the following objectives hold for students.

1. Willingness to accept responsibility for assignments.
2. Ability to discuss and ask questions and express a point of view.
3. Ability to develop a common purpose with fellow students and teachers.
4. Development of habits of accuracy, persistence, conscientiousness, and thoroughness.
5. Development of personal attributes of self-understanding, self-discipline, self-respect, initiative, and independence.

For computing courses the following additional objectives may be stated.

(i) Awareness of the value and power of the computer as an end in itself, and as a tool for other disciplines and for society at large.
(ii) Involvement in computing for satisfaction and discovery.
(iii) Development of interest in computing as a leisure time activity and as a vocational pursuit.
(iv) Ability to read technical literature.
(v) Ability to integrate computing into one's own personal *Weltanschauung*.

Programming assignments should promote all of these objectives.

The project specifications for the first group could require only moderate programming maturity. The following projects produced positive results in the affective domain. A simple algorithm producing a sequence of positive integers that terminates for choice of first element between 1 and 100 was programmed. The students discovered that the sequence terminates for initial integers between 1 and 200. A CAI package was devised to contain review and testing facilities using randomly selected questions. Graphics exercises were also completed to produce computer art. The programs were written in BASIC and operate on microcomputers in the QIS laboratory at Western Illnois University. Interests of the second group were partly vocational in nature and centered on business applications of COBOL. A package of thirty programs was written to accomplish tasks of business file processing. A strong element of achievement in the business computing field was shown by the group. Projects for the third group were chosen from topics covered in mathematics lectures at the workshop. They included projects in finite mathematics, calculus, linear algebra, and number theory. An example project on smooth curve approximations will be described.,

Among our wider objectives in devising and approving the projects were the following:

A. To stimulate lasting student interest in computing;
B. To assess attitudinal improvements in students involved in the projects;
C. To provide students with experience in designing and implementing software;
D. To use successful software produced to improve instruction and demonstration facilities;

E. To provide help in planning departmental offerings by providing a framework for assessing the quality of programming assignments.

ULAM'S CONJECTURE

The teachers at the NSF funded workshop were partly concerned with "back to basics" and suggested many programs that are fun to implement on the computer and suitable for the elementary student. For example, start with any positive integer. If it is even, divide by 2. If it is odd, multiply by 3 and add 1. Repeat the process. It is Ulam's conjecture that this sequence terminates with 1 for any starting positive integer n, and some partial results are kown [6]. This example is suitable as a discovery stimulant for introductory computer students. Intricate tree diagrams of the paths traced out by odd elements in the sequences can be drawn. Nested loops enable more ambitious students to run the program for ranges of initial element. This project demonstrates to beginning students the use of the computer as a research tool.

COMPUTER ASSISTED INSTRUCTION

The instructor's aims in devising a CAI package for introductory computer courses were (1) that the system be simple to learn and easy to use, (2) that the system be flexible enough to allow students to progress towards success when using it, (3) that the system be readily adaptable for use on larger computers with time-sharing facilities and on the microcomputers available in our laboratory, and (4) that students be able to participate in programming the package.

To aid with the first objective the package was written in BASIC. The first course in computers at this university is taught in BASIC and large numbers (over 2,000 per year) of the students become acquainted with the language. Within the framework of the package, instructors were able to create additional material of any desired degree of sophistication. Students of modest computing maturity were able to help with programming the package.

The second objective was implemented by allowing students using the package more than one attempt at each of the multiple choice questions randomly selected. For tutorial work the questions were arranged within subject headings and students were encouraged to study areas with which they were unfamiliar, learning from the correct answers provided by the computer. An examination situation could also be created. Questions on computer development, hardware, number systems, software, programming, personal computing, computer networks, and applications to education, art, music, medicine, government, transportation, criminal justice, automation, business and finance were programmed by the students.

The third aim of portability was emphasized to the introductory computer students. Again the choice of BASIC was fortuitous. The package could be implemented on the PDP 11-40 at Monmouth. Transfer of the package from the microcomputers in the QIS laboratory to the PDP system involved minimal changes. The objectives set down for the CAI package were also applied to the design and development of the other packages.

The fourth objectives of having students participate in the project was accomplished by letting beginning students program individual questions with the correct answers flagged. A more advanced student classified the questions and, with the instructor's help, put them together in the final package.

EXTERNAL FILE STRUCTURES

The second group of students designed and developed structured COBOL programs for the IBM 370-155 at Western Illinois University utilizing data files on tape and disk storage devices. Objectives were to introduce concepts and techniques for data organization and manipulation on tape and disk and to develop the advanced skills needed in typical applications programming projects. Sequential, direct, and indexed sequential file projects were specified. Because of the length and difficulty of some of the tasks, a feeling of considerable achievement was experienced by students completing the tasks successfully.

Bank data processing was the subject chosen for the following projects. (1) A sorted transaction file containing new accounts, closed accounts, and updates was used to create a new master disk file from a parent sequential checking account file. (2) An unsorted transaction file containing bad data was verified and sorted on checking account number and used to update the master file. Bank statements for customers were produced. (3) A back-up copy of the new master file was created on tape. (4) An indexed sequential master file was created and updated against a transaction file using random access. (5) A project similar to (4) using direct file organization was also completed.

Attitudinal improvement is not easily achieved without motivation and fulfillment. Ausubel [1] states that achievement motivation involves affiliative, ego-enhancing, and cognitive drives. The ego-enhancing drive was found to be strongest in the second group. In this group forty-five percent felt that the ego-enhancing, thirty-five percent the cognitive drive, and twenty percent the affiliative drive to be the most important as a factor contributing to motivation and attitudinal improvements in this course. Ninety percent had the immediate fulfillment of passing the course. Students who have completed this course successfully are employable as COBOL programmers at relatively high starting salaries.

SMOOTH CURVE APPROXIMATIONS

A sample project by a participant in the NSF funded summer workshop is described. Let (a_i, b_i) be *m* experimentally determined points. Let

$$A = [1, a_i, \ldots, a_i] \quad \text{and } B = [b_i]$$

by matrices of type *m* by *n + 1* and *m* by *1* respectively. Then

$$X = (A^t A)^{-1} A^t B$$

gives the least squares fit of the experimental data of the form (y, t) to the smooth curve

$$y = x_0 + x_1 t + x_2 t^2 + \ldots + x_n t^n$$

where $X = [x_i]$. Using these formulas, the necessary software for a classroom demonstration of Hooke's constant for a spring or the value of the gravitational constant *g* is easily devised. Color graphics showing the actual smooth curve approximating results enhance the presentation.

This package was particularly successful in producing attitudinal improvements. The package provides facilities for discovering new functional relationships and constants from empirical evidence. The uses of the package are therefore many and varied. For example a medical study where a variable quantity of drug administered is compared to the time length of induced state could be made using smooth curve approximations of small integer degree. The package is a useful research tool, and may be used in a science laboratory as an aid to discovering and understanding formulas.

Conclusions. On the basis of evaluation of written responses to questionnaires and oral statements, data was collected on the level of internalization of students. Student responses of the type "I often try to convince my friends to study computing" or "I like to read articles in *Datamation*" were judged to put the student at the valuing level. A statement like, "Just as elegance is possible in mathematical proof, it is possible through structure in computer programming" was judged to put the student at the organization level. At the beginning (Figure 1) and end (Figure 2) of the courses the following data expressed as percentages was obtained.

Group	1	2	3
1. Receiving	97	100	100
2. Responding	50	70	85
3. Valuing	30	35	80
4. Organization	7	10	75

Figure 1.

Group	1	2	3
1. Receiving	83	90	100
2. Responding	77	85	100
3. Valuing	40	45	90
4. Organization	20	25	85

Figure 2.

From a survey it was learned that attitudinal improvements are engendered by computer projects that:

1. increase the feeling of usefulness of the student;
2. provide opportunities for discovery of ideas;
3. increase the students' ability to communicate with his or her peers;
4. enable the student to use the knowledge he has gained in a productive way;
5. provide opportunities for self-discovery of talents;
6. aid in developing self-confidence;
7. provide a unity of purpose with instructors;
8. facilitate participation in the stewardship of the new science of computing;
9. provide opportunities for creativity;
10. make the student aware of adding to his or her store of knowledge;
11. are fun to do;
12. enable the student to understand why new concepts are introduced.

In order to compare the relative importance of the various factors with the level of internalization of the students, a questionnaire was devised. Each student voted for three factors. The following results were obtained expressed as percentages.

Receiving, responding, valuing, and organizing are recorded as levels 1, 2, 3, and 4 respectively.

		1	2	3	4
	1	9	6	4	6
	2	12	5	10	10
	3	2	5	4	3
	4	25	24	27	25
	5	9	2	4	3
FACTOR	6	7	8	6	9
	7	3	1	2	1
	8	12	11	10	9
	9	6	7	9	6
	10	6	13	10	11
	11	0	6	5	3
	12	9	12	9	14

Figure 3.

Our results show that the vocational factor of participation in the stewardship of computer science may be conjectured to decrease in importance with increase of level of internalization. Enabling students to use knowledge gained in a productive way is chosen as important most frequently at all levels.

The instructor may check his programming assignments against the list of factors to determine the assignments' possibilities for producing attitudinal improvement. Willingness to learn, to reason, to accept change, to cope with decision making, to do research, in short, to be valuable and productive, should be encouraged as objectives for all students through the choice of appropriate programming projects.

REFERENCES

Ausubel, David P. *Education Psychology: A Cognitive View*, New York (1968), Chapter 10.

Hebenstreit, Jacques, *10,000 microcomputers for French schools*, Computer 13, 7 (July, 1980), 17-21.

King, Ronald S. and Lamont, Patrick. Microcomputer Laboratories for Applied Computing, Proceedings of the Fourteenth Hawaii International Conference on System Sciences (1981), 178-191.

Krathwohl, David R., Bloom, Benjamin S., and Masia, Bertram B. *Taxonomy of Educational Objectives, Handbook II—Affective Domain*, New York—London (1964).

Porter, Gerald J. *Computer graphics as a lecture aid in undergraduate mathematics*, Proceedings of the National Educational Computing Conference (1979), 237-249.

Dunn, Richard. On Ulam's Problem. Report. Department of Computer Science, University of Colorado (1973), 1-10.