Learning with Personal Computers

ALFRED BORK

A leader in computer-assisted learning presents his new research

LEARNING WITH PERSONAL COMPUTERS

Alfred Bork
University of California, Irvine

1817
HARPER & ROW, PUBLISHERS, New York
Cambridge, Philadelphia, San Francisco, Washington,
London, Mexico City, São Paulo, Singapore, Sydney

Sponsoring Editor: John Willig
Project Editor: Susan Goldfarb
Text Design Adaptation: Barbara Bert
Cover Design: Wanda Lubelska Design
Cover Photo: Paul Conklin/Monkmeyer Press
Text Art: Fineline Illustrations, Inc.
Production: Willie Lane
Compositor: Donnelley/Rocappi, Inc.
Printer and Binder: R. R. Donnelley & Sons Company

Learning with Personal Computers

Credits begin on page 237.

Library of Congress Cataloging-in-Publication Data

Bork, Alfred M.
Learning with personal computers.

1. Computer-assisted instruction—Addresses, essays, lectures. 2. Education—Data processing—Addresses, essays, lectures. I. Title.
LB1028.5.B616 1987 371.3'9445 85-27293
ISBN 0-06-040868-5

86 87 88 89 9 8 7 6 5 4 3 2 1

LEARNING WITH PERSONAL COMPUTERS

Harper & Row Computer Science and Technology Series

Barclay: *Using Graphics to Learn BASIC Programming*
Bartee: *BASIC Computer Programming,* Second Edition
Bellin: *The Complete Computer Maintenance Handbook*
Bork: *Personal Computers for Education*
Learning with Personal Computers
Chou: *Microcomputer Programming in BASIC with Business Applications,* Second Edition
Gallier: *Logic for Computer Science: Foundations of Automatic Theorem Proving*
Garavaglia: *PROLOG: Programming Techniques and Applications*
Greenwood and Brodinski: *Enjoying BASIC: A Comprehensive Guide to Programming*
Halpern: *Microcomputer Graphics Using PASCAL*
Huelsman: *Engineering and Scientific Computations in PASCAL*
Jackson and Fischer: *Learning Assembly Language: A Guide for BASIC Programmers*
Jones: *PASCAL: Problem Solving and Programming with Style*
Modula-2: Problem Solving and Programming with Style
Keller: *When Machines Teach: Designing Computer Courseware*
Lamprey, Macdonald, and Roberts: *Programming Principles Using Pascal*
Lin: *Computer Organization and Assembly Language Programming for the PDP-11 and VAX-11*
Mason: *Learning APL: An Array Processing Language*
Newell: *Introduction to Microcomputing*
O'Shea and Eisenstadt: *Artificial Intelligence: Tools, Techniques, and Applications*
Passafiume and Douglas: *Digital Logic Design: Tutorials and Laboratory Exercises*
Rafiquzzaman: *Microprocessors and Microcomputer Development Systems: Designing Microprocessor-Based Systems*
Shumate: *Understanding ADA*
Touretzky: *LISP: A Gentle Introduction to Symbolic Computation*
Ural and Ural: *Introduction to Programming with Modula-2*
Weir: *Cultivating Minds: A Logo Casebook*
Wood: *Theory of Computation*

PREFACE

This book is a collection of papers, most previously published, about the work of the Educational Technology Center at the University of California, Irvine. The selections cover the period from 1980 through 1984.

These papers were written after those in my earlier collection *Learning with Computers* (Digital Press, 1981). Except for minor editorial changes, the papers are in their original forms.

For convenience, this collection is divided into seven parts: "Overview," "Computers and Schools," "Scientific Reasoning," "Physics," "Production and Design," "Videodiscs," and "Computers and the Future of Education." The papers can be read independently, in any order the reader desires. Often, further detail on a given topic may be found in another of my recent books, *Personal Computers for Education* (Harper & Row, 1985). I will now review briefly the papers in each part, with the intent of helping the reader make decisions.

part one: *Overview*

The two papers in this first part provide a general introduction. The first paper, originally prepared for a conference at the University of Oregon, looks at a wide variety of uses for computers in education, with particular attention to the school system. It argues, as does a later paper, that BASIC is an unsuitable first programming language. It ends with a set of suggestions aimed primarily at teachers beginning to work in this area.

The second paper has a much narrower focus. It concerns the use of computer-based learning material as a learning aid in a variety of subject areas. This paper emphasizes the quality of learning units and points out common errors in developing such units.

part two: *Computers and Schools*

The first two papers in Part Two are concerned with how teachers confront this new learning technology. These teachers are often poorly prepared. Further, I regard most of the programs for training teachers about computers as worse than useless.

The third paper concerns the evils of BASIC, the junk food of computer languages. The problem is a major one, given the widespread teaching of unstructured BASIC in our schools. Again, teacher training programs bear a large part of the blame.

The last paper in this section was originally prepared for the meeting of an IFIP (International Federation of Information Processing) subgroup in Working Committee 3. It outlines a k–16 computer curriculum for science and engineering majors. But we are a long way from such a curriculum; we do not have more than a small fraction of the learning materials necessary to make this curriculum practical. The discussion at the end of the paper addresses the issue of how such a curriculum could be implemented in the future.

part three: *Scientific Reasoning*

Part Three moves from general discussion to specific material. The modules described in the papers in this section were all developed at the Educational Technology Center at the University of California, Irvine. Two grants supported this work, one from the Fund for the Improvement of Postsecondary Education, the other from the National Science Foundation. About twenty hours of computer dialogues were developed.

The purpose of these modules is to help students begin to think and reason the way a scientist thinks and reasons. Students are placed in "environments" in which they must think and act like scientists. These environments are friendly and supportive, giving help when appropriate. As their titles suggest, some of the papers concentrate on individual programs, while others look at groups of programs.

part four: *Physics*

The two selections in Part Four are very different, but both reflect my long-term interest in the problems of learning physics.

The article from *Physics Today* might be described as a survey, a review of various ways of using the computer to teach physics. It gives examples of various modes of usage and should apply equally to high school and college courses.

Newton, the subject of the second paper, has had a long history in our group. Our initial proposal to the National Science Foundation, in 1967, described such a program. The version we describe in this paper is the second one developed. The notion is to provide a rich collection of experiences about motion controlled entirely by the user. By this procedure we hope to improve students' understanding of physical laws.

part five: *Production and Design*

For many years I have argued that the key to successful use of the computer in a wide variety of learning modes is a thoughtful, effective production system. A major emphasis of our work at Irvine has been the development of such a system. All the papers in this section are about our production system.

The first paper is an extensive overview of the system, looking carefully at each of the stages involved. Three aspects are discussed: pedagogical design, technical implementation, and evaluation. The SADT charts summarize the process.

The rest of the papers in Part Five look at more specific issues in developing an effective system. Several are concerned with screen design.

part six: *Videodiscs*

When I first saw prototype videodisc players about a dozen years ago, I was struck with their potential for educational use when coupled with powerful personal computers. But I am frustrated with the lack of progress in this area and with the low quality of most existing materials. The papers in Part Six address this situation. The underlying theme is how to develop more effective learning material using the computer-videodisc combination.

part seven: *Computers and the Future of Education*

We are at an early stage in the educational involvement of computers, and the equipment is still evolving rapidly. So it is critical, if we are to use computers wisely, to think carefully about future directions for education.

Although the computer is being used increasingly in our learning environments, at all levels, it is not clear whether this use is improving education. The computer *can* lead to an improved educational system, but only if we work with care toward a better system. The papers in this final section discuss future pitfalls and possibilities.

ALFRED BORK

LEARNING WITH PERSONAL COMPUTERS

CONTENTS

part five PRODUCTION AND DESIGN 107

part six VIDEODISCS 173

part seven COMPUTERS AND THE FUTURE OF EDUCATION 195

NEWTON

The controllable world discussed is based on Newton's laws of motion. The computer "knows" that $F = ma$, and furthermore knows the mathematical tools necessary to turn this into visual information about how bodies move. The student is given plotting capability. After the force is chosen, the learner can plot various physical variables against each other, alter initial conditions and constants in equations, and move freely through the program. Newton is self-explanatory, not dependent on print material. But certain types of print material will typically be used by students with the program.

HISTORY

A program of this type was described in the initial proposal from the University of California, Irvine (to the National Science Foundation) for developing graphic computer-based learning material. Shortly after the grant, Richard Ballard joined Alfred Bork on the project. They developed the initial version of a controllable world called Motion. This program, in a timesharing environment, is still used with beginning physics students at Irvine. Motion went through a number of variations, as we experimented with how it could be used most effectively.[4]

Several years ago the focus of development at Irvine began to move from timesharing to the newer personal computer environment. At that time we also abandoned earlier software approaches, as they were no longer in keeping with what was known about the art of complex programming. Our new developmental language is Pascal, under the UCSD Pascal system.

Martin Katz, then an undergraduate student working with the Educational Technology Center, developed a Pascal version of Motion soon after we began to use Pascal. This version was not completely equivalent to Motion; it omitted some facilities but had some additional ones. This program eventually evolved into Newton.

The current version of the program was developed by Alfred Bork, Stephen Franklin, and John McNelly. It does not follow all the details of Motion. Rather, we tried to use what we had learned in the many years of using Motion with sizable numbers of students to guide the development of the new program. Motion ran on Tektronix displays. Newton was developed on the Terak 8510a. By and large, we found that the advantages of the personal computer far outweighed the disadvantages; that is, the switch from timesharing to the personal computer was primarily a gain. We gained selective erase capabilities and better control over timing issues at the expense of poorer resolution.

As of this writing, the latest version of Newton runs only on the Terak. However, earlier versions were successfully moved to the Apple, and we expect to eventually run on a variety of other personal computers. As with other recent developments at the Educational Technology Center, we find it convenient to develop materials on a more powerful machine than the eventual delivery machines.

CAPABILITIES OF NEWTON

As already suggested, Newton is primarily a plotting program. After the mechanical system has been picked, the student can ask to plot any two or three mechanical variables. Time in each case is the independent variable, as is generally the case with mechanical systems, but time does not need to be one of the variables plotted. The user can change the force, change the constants in the force law, change the initial conditions, choose what to plot (including functions of the variables), and query the system for various information. Control over scaling is also available. These capabilities will now be described in more detail.

1. Choosing the Force. When the program is initially entered, the learner must choose a force. At any time during the program, a NEW FORCE can be requested, and the choice will be offered again.

Two basic options in choosing a force are available. First, built-in forces can be picked. Currently the built-in forces are gravitational motion with one force center, gravitational motion with two force centers, simple harmonic motion, and force-driven harmonic motion. New built-in forces are being added. Built-in forces can be selected from a menu.

The user can also choose to enter almost any force whatsoever. These are accepted in a typical linear computer algebra form with some flexibility. The computer queries for each component of the force. In specifying the force, if constants are used that are not known to the system, the system will query the user as to what initial value should be assigned to these constants. The program can handle almost any force within the limitations of typing.

2. Plotting Capability. After a force has been chosen, either initially or at some later time in the program, the machine is prepared to plot something. That is, if the user simply types PLOT, a curve will appear. The curve is dependent on the force law chosen. We have chosen in advance an interesting case with all the initial conditions already chosen.

Many computer simulations query students for everything necessary to plot. Beginners seldom understand what things are necessary or what values to assign to them, so such querying should be delayed until the learner has attained better understanding. Our notion is to provide an interesting case to begin with and allow the student complete control over changing each of the variables involved.

If the student wants to plot two different variables, then the command is

```
PLOT X,T
```

or

```
GRAPH X VS T
```

Other forms are also possible. Newton is flexible about what terminology is needed, often providing alternates. One function of the variables can be plotted against another.

The typical way to stop plotting is by pressing the space bar. One can continue plotting by typing CONTINUE PLOT after such a stop. There are certain circumstances where plotting may stop on its own, such as when the body crashes into a sun, or the values calculated become abnormally large.

A number of other capabilities are associated with plotting. One that is frequently useful is OVERPLOT, which allows learners to keep a previous curve while seeing a new one. If parameters change, it is convenient to see the curves before and after the change.

The current values of variables plotted can be determined at any time by typing a question mark. This is often convenient when you need numerical values in addition to the curve.

3. Querying for Information. The user can ask for information about current values of variables, either before or after a plot. Thus, you can ask for INITIAL CONDITIONS and Newton will give you these values. Or if you want to see all the variables associated with a particular case, you can simply type ALL to see them.

Such information is often more detailed than necessary. Usually learners need to know only certain variables, either the initial conditions or some of the constants in the equations representing the force law. The student can ask directly for these in one of the following fashions:

a. WHAT IS X?
b. X = ?
c. X ?

Newton will regard any of these commands as equivalent.

4. Changing Variables. A user can not only query variables; variables can be altered. This is done by entering small assignment statements:

a. X = 3
b. FRED = 2
c. DT = DT/2

In the second case it is assumed that a user-specified force was used and that the variable FRED was picked by whoever entered the force. When a variable is changed, Newton verifies what the user has done by showing the value of the variable as changed. This is often not necessary, but it is a reasonable precaution to overcome typing errors.

5. Changing the Scale. Often it is necessary to modify the scale of the plotting to see a convenient picture on the screen. This must be done by the user, as Newton cannot know what details the user expects to see.

The overall changes of scaling are indicated by the commands MOVE BACK and MOVE CLOSER. Both of these produce a scale factor change of 2 on both axes.

Direct scaling is possible by plotting variables in the following way:

```
PLOT .5 * X*X, 3 * VX
```

The net effect will be that the scales of both variables change independently. Axes will be labeled appropriately, reminding the user of this change.

USES OF NEWTON

We have implied that the development of materials and their effective use in class or learning environments are two quite separate issues. In this section we discuss this situation and clarify the use of Newton and similar programs.

The primary use of such a program is to build intuition, to allow learners to gain a range of experiences that are not present in everyday life, and so have a feel for mechanical systems that goes beyond the ability to manipulate mathematical details to obtain solutions. Simulations, such as the present one, often have an immediate appeal to scientists. They are closest to the directions scientists follow in their professional activities using the computer. Most scientists are stimulated by running such simulations. Indeed, in our early days with Motion, scientists could hardly keep themselves away once they became exposed. We would have visitors spend large amounts of time running it. Motion would also draw very large crowds when presented at professional meetings.

We began to understand the distinction between the program itself and the program operating in a classroom when we began running Motion with sizable groups of students. Here the excitement of the scientists was often not present. Only a small percent of the students would become excited while using the program. The rest would use it a brief time and then stop unless forced to continue. While we thought the program exciting, the bulk of students in any large class did not appear excited.

This situation puzzled us until we began thinking about incorporating the material in the classroom. Our first step was to develop computer exercises that assured each student would see at least the most important experiences. These computer exercises are still in use in the timesharing environment, about six years after their initial development. Here is a sample of one exercise concerned with gravitational motion:

Now you are to see what would happen if gravitational force were *not quite* inverse square. Ask for the EQUATION again; the power is N. Set

N = −1.9

Return to plotting the X-Y space, investigating a range of values around −2. You may want to continue plotting each orbit. What can you say about the results? What happens for values less than 2? Greater than 2?

What happens if we examine behavior in velocity space?

Now consider the case of *two* gravitational force centers, as if you had two *fixed* suns. Request

2 FORCE CENTERS

at any input. The initial conditions will be reset. Determine them by typing

X,Y,VX,VY = ?

PLOT the orbit. Discuss the possibility of life on a planet with such an orbit.

See if you can find velocities that give closed (repeated) orbits. What velocities do this? Sketch the orbits.

We want to make certain that the student has some structured experiences that aid *learning* about how mechanical systems work.

This second round, using Motion with computer exercises, was not entirely successful either. In the evaluation of the course made by Michael Scriven and his colleagues, this was one of the most criticized activities. At this time we made it a required part of the course. We found that students did not see its connection to other material within the course. Now the material is explained better and used as an option with much greater success; it is used by a sizable number of students.

But we do not regard this approach as entirely satisfactory either, because we believe the experiences should be for *all* students. In another controllable world dealing with field lines, we have greater success with a different tactic. We built an on-line quiz around the simulation. The quiz notes if students have developed the insights we expect about the way field lines behave. As yet, we have not followed this same tactic with mechanics but probably will do so.

The computer experiences for Motion only cover some of the areas of beginning mechanics. We could increase the viability of this program by making it a constant component of the beginning course, making every unit depend on it to some entent. We are working with CONDUIT (specifically Arthur Luehrmann, Herbert Peckham, Harold Peters, and Alfred Bork) to develop a more extensive set of computer exercises for Newton. These are intended for use in high school and beginning college physics courses and will cover areas not [covered] by the present exercises. For example, we consider motion with no forces acting and motion with constant forces. We plan to have these new exercises available at the conference.

This project is supported by the National Science Foundation through a CAUSE grant. The project manager for the grant is Stephen Franklin. Other members of the Educational Technology Center, Barry Kurtz and David Trowbridge, have offered helpful suggestions for developing Newton and the associated exercises.

June 17–19, 1981

REFERENCES

1. Karplus, Robert, "The Psychology of Teaching and Thinking for Creativity," Anton Lawson, ed., 1980.
2. Bork, A., "Computers as an Aid to Increasing Physical Intuition," *American Journal of Physics,* 46 (8), August 1978.
3. Bork, A., "Learning with Computer Simulations," *Computer,* October 1979.
4. Bork, A., "The Physics Computer Development Project," *EDUCOM,* 10, 4, Winter 1975.

five

PRODUCTION AND DESIGN