Behavior Research Methods & Instrumentation
1982, Vol. 14 (2), 103-108

An experimental psychology laboratory system for the Apple II microcomputer

STEVEN E. POLTROCK and GREGORY S. FOLTZ
University of Denver, Denver, Colorado 80210

A system for on-line control of experiments is described that uses the Apple II microcomputer and requires no knowledge of programming. This system, APT, is used to construct test files, randomize stimulus order, and conduct experiments. Construction of an experiment is very easy through use of an interactive program. APT provides powerful options for controlling stimulus duration, feedback, and response type. Programs are available to conduct either a sequence of tests without any experimenter present or a single experiment. Over 200 subjects, ranging from 18 to 85 years of age, have completed a battery of cognitive and abilities tests using the system.

In the 1979 Presidential Address to these meetings, Bailey (1980) described a "basic black" ensemble of software tools suitable for an experimental psychologist. At the 1980 conference, participants in the Session on Microprocessors in the Undergraduate Teaching Laboratory spoke of the urgent need for one element of Bailey's Basic Black: an applications module library to facilitate the development of experiments. The implementation of new experiments in teaching laboratories has required sophisticated programming skills, thus discouraging students from designing their own experiments. (See Foltz & Poltrock, 1981, for further discussion of these problems.) An applications library should allow students to easily design and conduct their own experiments, adding considerably to their laboratory experience.

Although most acute in the teaching laboratory, the programming sophistication required to develop an experiment can be a problem for the research laboratory also. Those psychologists who have resisted the urge to become adept programmers may be discouraged from investigating some research problems because of the perceived complexity of a program. And even those psychologists who are skilled programmers may prefer not to devote their time to solving the same simple problems repeatedly for each experiment.

Recently, some systems have been developed for Apple II microprocessors that eliminate much of this programming drudgery. At the 1980 conference, Digitry Incorporated demonstrated a solution that requires purchasing an interface board and a library of PASCAL functions and procedures. The researcher still must write programs, but the programs are written in PASCAL rather than BASIC, and many functions are performed by the library. Other systems require no programming and no extra hardware. At this conference, Eamon (1982) described the CEDATS system, which includes programs written in BASIC that set up, conduct, and analyze categorization, same-different, and study-test experiments. Our system, the Apple PASCAL Tester (APT), requires programming only for data analysis.

APT was developed to conduct individualized psychometric and cognitive testing. It was essential to our purposes that (1) tests and experiments could be quickly developed and modified, (2) no knowledge of programming would be required to develop experiments, (3) no experimenter need be present to operate the computer, and (4) subjects need no prior experience with the computer. These goals were achieved by developing a set of PASCAL programs that construct and execute experiments and an applications library that contains special functions. The UCSD PASCAL operating system provided file management and editing capabilities that made these goals attainable.

Typically, a major problem with fixed systems such as APT is the loss of flexibility in stimulus control, timing, feedback, or responses. Often, an experimental design will not fit within the constraints of such a system. Admittedly, this problem exists with APT; there are experiments that cannot be conducted by this system. However, the system was designed to minimize this problem by including numerous options for the stimulus parameters. For example, there are seven alternative ways to control stimulus duration. Furthermore, the programs are easily modified if a new option is needed. The extraordinary flexibility of the APT system argues for its widespread use in both research and teaching laboratories.

The software system described in this paper is available at cost from the authors. Upon receipt of $25, we will send documentation and two diskettes containing the system, a program library, and sample tests. Please cite this article when reporting results obtained with the use of this system, so that others may also learn of its availability. Development of the APT system was supported by Grant R01 HD 13251 from the National Institute of Child Health and Human Development. Requests for reprints or the system should be sent to Steven Poltrock, Psychology Department, University of Denver, Denver, Colorado 80210.

Copyright 1982 Psychonomic Society, Inc.

0005-7878/82/020103-06$00.85/0

The remainder of this article describes the APT system and how it is used.

HARDWARE

The hardware required for the APT system is the minimal hardware required to run PASCAL, control stimulus timing, and measure response times. The UCSD PASCAL operating system requires an Apple II Plus computer with 48-KB RAM, a PASCAL language card (or an equivalent 16-KB RAM card), a monitor, and at least one minifloppy disk drive. Two or three disk drives are much easier to use than one for most PASCAL applications, including the construction and analysis of experiments. However, one disk drive is adequate to conduct an experiment.

A Mountain Hardware clock is required for timing. The clock provides millisecond precision, although the effective accuracy of stimulus and response timing is much longer, on the order of 10-20 msec. A major influence on the accuracy is the video refresh rate. A stimulus can be presented only when the video is refreshed, which occurs every 16 msec. Because the software is not synchronized with the refresh cycle, the onset of a stimulus display may be delayed by as much as 16 msec. A simple modification of the vertical synch monitoring circuit described by Reed (1979) and an introduction of a display function to monitor the synch pulse would eliminate this error.

The hardware is sufficient to construct and conduct experiments with stimuli composed of letters or numbers. We are now developing the capability to present graphic and auditory stimuli also. To construct graphics stimuli, an Apple graphics tablet is required. Auditory stimuli will be prerecorded and presented by a tape recorder. The tape recorder will be controlled by a Heuristics relay controller connected to the tape-recorder remote controls.

Currently, APT accepts only responses entered at the keyboard. Of course, it would be a simple matter to allow responses with the game paddles that are provided with the Apple computers, but we are satisfied with the flexibility provided by a full keyboard. When only a few keys are required for responses, we often use a 10-key number pad. The keys on the number pad are wired in parallel with the number keys on the keyboard, so a subject can respond with either.

TEST CONSTRUCTION

The flexibility and convenience of APT are due largely to the interactive test construction program TESTCON. The flowchart of TESTCON in Figure 1 provides an overview of the test construction procedure. As Figure 1 shows, the researcher explicitly enters every stimulus item as well as a set of parameters that affects that stimulus. The interactive nature of TESTCON facilitates the specification of these stimulus parameters by listing all options and by asking whether successive stimuli share the same parameter values. When all stimuli have been entered, TESTCON creates a disk file of the stimuli and parameters. Later, by reference to this file, the stimuli or parameters can be edited, the stimulus order can be randomized, and the experiment can be conducted.

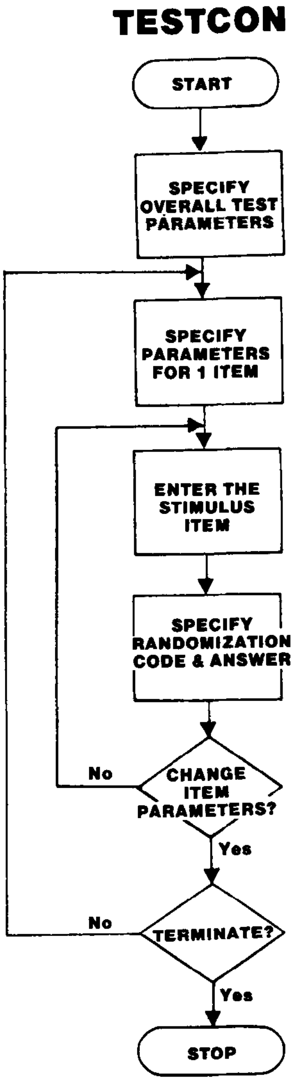
TESTCON
START
SPECIFY OVERALL TEST PARAMETERS
SPECIFY PARAMETERS FOR 1 ITEM
ENTER THE STIMULUS ITEM
SPECIFY RANDOMIZATION CODE & ANSWER
No
CHANGE ITEM PARAMETERS?
Yes
No
TERMINATE?
Yes
STOP

Figure 1. Flowchart of TESTCON, the program for constructing tests and experiments.

The major elements of the test construction procedure shown in Figure 1 are described in more detail below.

Test Parameters

Proper responses. Psychologists may want to restrict the possible responses for some stimulus items to a subset of the keyboard. For example, in a binary choice experiment, only two keys are appropriate responses. At this point, the psychologist identifies this subset of the keyboard by typing all proper responses or by pressing the "return" key if all responses are proper. Only one such subset can be defined for an experiment, and the parameters for each item specify whether the responses are restricted to this proper set. For those stimulus items requiring a proper response, any other response causes presentation of a warning message and a prompt for another response.

Feedback. TESTCON asks whether the subject's accuracy should be presented at the end of the experiment. If this option is selected, then an answer list is used to score each response. This option should not be selected when subjects are providing string responses, because string responses are not scored. If this option is selected, then the experiment will end with a message such as, "Your performance was good. 24 correct out of 30 problems."

Item Parameters

A test or experiment is composed of a sequence of items. Each item is a text display, a graphics display, or a command to control a tape recorder. The parameters specify the stimulus duration, possible responses, feedback options, data recording options, and randomization instructions.

Stimulus type. First, TESTCON asks whether the item is a graphics display, text display, tape-recorder control, or time interval. If the graphics display option is specified, then a graphics tablet will be used to construct the display. If the tape-recorder control is specified, then TESTCON asks whether to rewind, play, or stop the recorder. Software for both of these functions is currently incomplete.

The time interval provides a coarse control of interstimulus interval. If this function is specified, TESTCON asks for the interval duration. When no interval between text displays is specified, the interstimulus interval is at least 300 msec. During this interval, the test program switches to the graphics page, clears the text page, writes the text, and switches back to the text page. The time to perform these operations varies with the amount of text to be displayed. If an interval is specified, it is added to the time required for these operations.

If a text display is specified, TESTCON requests the associated parameter values, a randomization code, and the answer. The remainder of this section describes these variables.

Display duration. This option specifies when the text display will end. There are seven options for terminating a text display. First, the display may be presented for a fixed time interval, such as 500 msec. Three options cause a display to be terminated when a response is given. The display may be terminated by any keypress, by any proper keypress (defined above), or by a sequence of keypresses ending with a return (string response). Two other options combine the use of responses and time intervals for terminating a display. For these options, the display is terminated by a response if it occurs before a specified interval has elapsed. If no response occurs, then the display terminates at the end of the interval. One option accepts any keypress and the other requires a proper response.

The seventh option is particularly useful for experiments conducted without an experimenter present. Subjects are provided the opportunity to repeatedly view the instructions and practice problems. At the end of the practice problems, the subject is instructed to press the Y key if he or she wishes to start over at the beginning of the experiment. Thus, this option causes the display to end following any keypress. If the subject presses the Y key, all the preceding items are repeated. If any other key is pressed, the next item is presented.

Feedback. The researcher may choose to provide feedback regarding accuracy, response time, both, or neither. Accuracy is determined by comparing the subject's answer with the answer given by the researcher. Accuracy feedback cannot be provided for string responses because these answers are not checked. The feedback consists of the word CORRECT or WRONG and/or the value of the response time.

Response recording. The response, the response time, both, or neither may be recorded. The response is recorded instead of accuracy because researchers often are interested in the type of error committed. All recorded information is stored on a disk file.

Text. Finally, the stimulus item is typed just as it is to appear during the experiment. The only restriction on the stimulus is that it must not contain the # symbol. The # symbol is typed at the end of the item to notify TESTCON that the item is completed.

Randomization code and answer. The randomization code and answer are entered after the item is defined. TESTCON requests an answer for every stimulus item. These answers are used to provide item or test feedback. The researcher may enter any value for items that require no feedback or have no correct response.

A randomization code is also requested for every item. The code permits structured permutations of the items. Through use of the randomization codes, the researcher may group items into item sets and group item sets into blocks. Later, the order of blocks, item sets, or items may be randomized. The use of these codes is discussed further below.

Additional queries. After each item, TESTCON asks whether the parameters should be changed for the next item. If the next item has the same parameters, the researcher specifies only the item, the answer, and the randomization code. Otherwise, TESTCON asks whether there are more items to be specified. Following the last item, the experiment is stored in a specified disk file.

ITEM RANDOMIZATION

The test file written by TESTCON contains a four-line header of general test information: (1) the proper responses, (2) the answers, (3) the randomization codes, and (4) the test feedback flag. Before the experiment can be conducted, program RANDOMIZE must be executed to randomize the item order as desired and remove the randomization codes. RANDOMIZE reads the file created by TESTCON, randomizes the order as instructed by the researcher, writes a file of the test items in the randomized order, and writes a file of the item order for use in data analysis.

The use of randomization codes to form blocks and item sets is exemplified in Figure 2. All items with a code of zero serve as block markers and remain in a fixed position.

For example, items that contain instructions typically appear at the beginning of an experiment and should be assigned code zero.

Between two block markers, all items that have the same code define an item set. Item sets identify stimuli that must be grouped together. For example, pairs of items would be grouped together in an experiment requiring same-different judgments of successive stimuli. As another example, if subjects were required to read a paragraph and answer a series of questions, the paragraph would constitute an item set containing only one item and the questions would form another item set. The paragraph and questions together could be defined as a block by specifying preceding and succeeding items with code zero.

The researcher controls whether the order of items within item sets is randomized, whether the order of item sets within blocks is randomized, and whether the order of blocks is randomized. Thus, in a same-different judgment experiment with successive stimuli, the researcher could choose whether to randomize the order of the stimuli in each pair and independently decide whether to randomize the order of item pairs. In the experiment with paragraphs and questions, the order of questions for any paragraph could be randomized and the order of paragraph-question blocks could be randomized while not disturbing the necessary order of a paragraph before its associated questions.

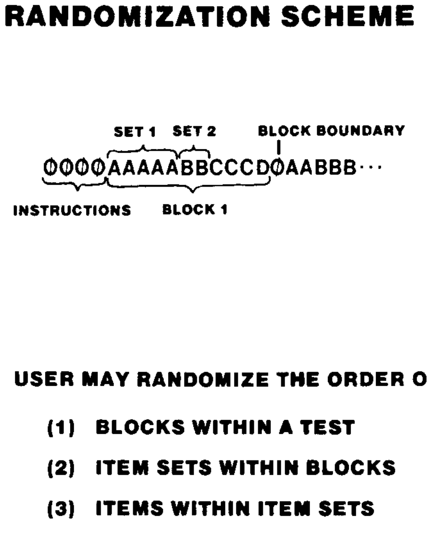

Figure 2. An example of randomization codes and their grouping into item sets and blocks.

When program RANDOMIZE is executed, it first reads the file created by TESTCON. Then RANDOMIZE asks whether the block order should be randomized. If so instructed, RANDOMIZE presents the new block order on the monitor. Next, RANDOMIZE asks whether item set order should be randomized. Again, the new item set order for each block is presented. Finally, RANDOMIZE asks whether to randomize the order of items within item sets. The entire test is written to a file in the new item order, and the item order is written to another file.

CONDUCTING AN EXPERIMENT

Two programs are available within APT for conducting experiments. One program conducts a single experiment, and the other conducts a series of experiments. A flowchart is presented of the single-experiment program, titled ONETEST, in Figure 3. First, ONETEST asks for the name of the test file and an output file to record the results. No further interactions with the experimenter are required. ONETEST reads the entire test into memory and presents the items in the order of their occurrence in the file. The parameters associated with each item control its timing, the permissible responses, feedback, and the data that are stored.

Our research on individual differences requires presentation of many tests with a fixed test order and a fixed item order. For this purpose, a turnkey program called TEST was developed. TEST requires no interactions with the experimenter. TEST executes when the power is turned on and obtains all information about the tests and the output file from a master file prepared by the researcher.

A flowchart of TEST is presented in Figure 4. TEST greets the subject with a graphics display and an invitation to press a key. If a subject accepts this invitation, TEST presents the information required for informed consent. The subject indicates consent by pressing the Y key. Then demographic information is requested, such as the subject's name, age, education, and sex.

Before every test, the program draws a graphic design and presents an opportunity to quite the experiment. A subject may quit by pressing the ESC key or continue the experiment by pressing any other key. Each test is conducted by TEST in exactly the same manner as by ONETEST. When all tests are completed, or when the subject quits the experiment, TEST thanks the subject for participation and presents its greeting for the next subject.

SYSTEM LIMITATIONS

APT has several limitations. Some of these can be readily overcome; others are inherent in the system approach. Examples of the latter are the inability of the system to permit contingent presentations and to control the tape recorder while a display is in progress.

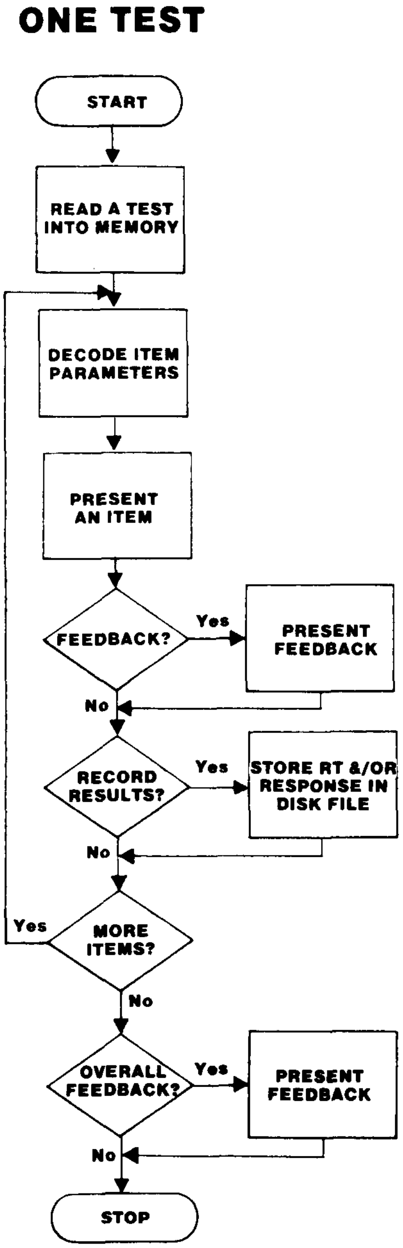

Figure 3. Flowchart of ONETEST, the program to conduct a single test or experiment.

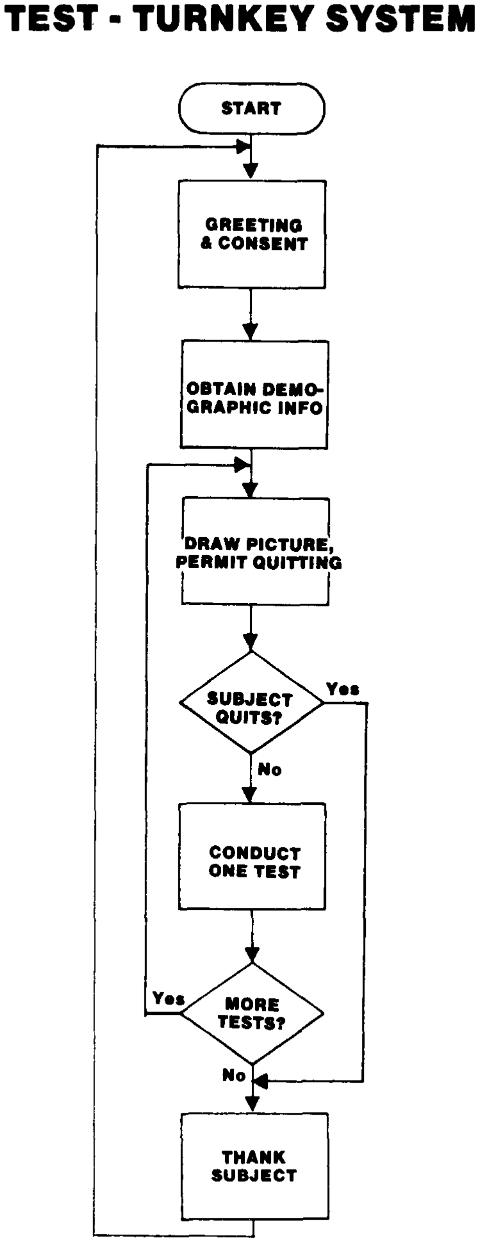

Figure 4. Flowchart of TEST, the turnkey program that conducts multiple tests or experiments.

Perhaps the most serious problems are the inaccuracies in timing noted earlier. The interstimulus interval includes at least 300 msec of processing time, and the timing of responses and display durations are influenced by asynchrony with the refresh cycle. The inaccuracy of display durations effectively prohibits tachistoscopic presentations, although the modifications suggested by Reed (1979) could solve this problem. Similarly, the innaccuracy of interstimulus intervals prohibits masking studies.

Another potentially serious problem is a limitation on the size of a test. All the APT programs read the entire test into memory. Thus, a test file may not exceed 26,624 test characters. Furthermore, a test may not include more than 255 items. We have skirted these problems by segmenting long experiments into two or more shorter files.

CONCLUSION

The APT system has successfully met all of the design criteria specified in the introduction. Tests and experiments have proved remarkably easy to construct. Thus far, more than 30 psychometric and cognitive tasks have been constructed. In one instance, an undergraduate student with no computer experience successfully constructed an experiment with less than 15 min of instruction.

Modification of existing experiments requires greater expertise with computers. Modifications are performed through use of the UCSD PASCAL editor. Fortunately, this editor is well documented, easy to use, and provides powerful editing capabilities.

Finally, the system has proved capable of testing subjects who have no computer experience, and without the presence of an experimenter. Over 200 subjects participated in a 2-h battery of cognitive and psychometric tasks presented by TEST. Subjects reported to our laboratory, in which a sign indicated that no experimenter would be present. All instructions and tests were presented by the computer. A majority of the subjects reported that they preferred the computer-administered tests to standard paper-and-pencil tests. Subsequently, the APT system has tested 40 elderly subjects (from 40 to 85 years old) under similar circumstances.

The APT system is a powerful tool appropriate for several settings. In the teaching laboratory, APT permits students to construct and conduct their own experiments without learning to program. In the research laboratory, APT provides the power and flexibility required for a wide range of research problems. And APT provides a means of conducting individualized tests without the presence of an experimenter.

REFERENCES

Bailey, D. E. "Basic black" renascent: A new wardrobe. *Behavior Research Methods & Instrumentation*, 1980, **12**, 96-102.

Eamon, D. B. CEDATS: A cognitive experimental design and testing system. *Behavior Research Methods & Instrumentation*, 1982, **14**, 142-145.

Foltz, G. S., & Poltrock, S. E. A microcomputer system for the human psychology laboratory. *Proceedings of the Fifth Western Educational Computer Conference*, November 1981.

Reed, A. V. Microcomputer display timing: Problems and solutions. *Behavior Research Methods & Instrumentation*, 1979, **11**, 572-576.