Controlling the Display of Animation for Better Understanding, Part I

Shu-Ling Lai
Ling Tung College

Abstract

Computer-Based Learning (CBL) courseware was developed with three types of animation-control strategies: (a) program control, (b) linear control, and (c) learner control. A total of 186 college freshmen participated in this research. Students with different mathematical ability (higher or lower) were randomly assigned to one of three learner conditions. Participants who read the CBL lesson with program control on the animation scored significantly higher on the posttest and spent significantly longer time on the CBL task. Participants with higher mathematical ability were shown to perform better than participants with lower mathematical ability in the programming concept learning but worse when given linear control. Participants with lower ability performed worse when given learner control. Lower-ability participants indicated positive attitudes toward linear control strategies, while higher-ability students indicated positive attitudes toward learner control strategies. The results of this study suggest that accommodating learners' individual differences in the design of CBL lessons is an important concern.

When using animation in the teaching process, the CBL designer should consider the best learner-control strategies to improve the display of animation, students' internal cognitive involvement, and time on task in order to enhance students' performance and attitude toward learning. Mathematical ability is highly correlated with programming learning, but previous research cannot answer if ability levels influence learners' perceptions of a mental model with different interactions. Although research has been conducted to understand the role of animation used as part of explanatory presentations to understand abstract concepts (Lai, 1998; Mayer & Anderson, 1992; Mayer & Gallini, 1990; Rieber, 1990, 1995), little is known about the optimal strategy to display animation. The purpose of this study was to investigate the effects of method of control (program control, linear control, and learner control) and mathematical ability (low and high) on learning abstract concepts with animation. The dependent variables included students' achievement, time on CBL, and attitudes toward the instructional software and controlling the animation.

Many benefits have been claimed for interactivity in computer-based learning (CBL). It has been assumed that learning and cognitive processing are facilitated through the manipulation of variables such as pacing, sequencing, and displaying strategies but determining how to control the instruction for learners to obtain new knowledge is important and difficult to decide (Freitag & Sullivan, 1995). Though animation has become an important element in CBL lessons, little is known about the cognitive value of any controlling factors that allow users to pace or sequence the display of animation to stimulate the understanding of concept learning.

Learner control and screen design displayed with text, graphics, and animation are the main factors influencing cognitive load (Clark & Taylor, 1994; Stoney & Wild, 1998).

Mayer and Gallini (1990) proposed that good animation should be manipulated and controlled in the manner that helps the learner keep his or her attention on the relevant information and at the same time helps the learner build connections between the abstract and concrete domains (Dicheva & Close, 1996; Lai, 1998; Mayer, 1989; Mayer & Anderson, 1992). Animation has been used a lot in the CBL, but no systematic research has been able to state the best way to display or control animation.

Perspectives and Theoretical Framework

Animation and Mental Models

A *mental model* is a person's understanding of the environment. With adequate mental models, learners can mentally simulate and predict the result (Norman, 1983). The mental model theory proposes that the formation of a mental model is enhanced by instruction that provides an appropriate representation of the states and relationships of the learning procedure (Borgman, 1986; Gentner & Stevens, 1983; Payne, 1988). Computer-generated animation offers a potentially powerful medium that helps learners build mental representations for comprehension (Mayer & Gallini, 1990; Shih & Alessi, 1994). Because animated pictures can present different states of a subject matter, they provide more information to a learner and require more processing than static pictures (Rieber, 1990, 1995; Schnotz & Grzondziel, 1996).

Clark and Taylor (1994) suggested breaking training into chunks (chunking it) to reduce learners' cognitive overload. However, the continuous sequence provides learners with a systematic and completed conceptual model that supports mental simulations and helps learners assimilate learning (Lai, in press; Schnotz & Grzondziel, 1996). As stated by Ausubel's (1968) theory of meaningful learning, the prior presence of a clear conceptual model may act as an advance organizer, providing needed anchors to incorporate new material into the learner's cognitive structure (Mayer, 1976). It is suggested that animation with too many student-controlled interactions may curtail the effectiveness and efficiency of assimilated learning (Spotts & Dwyer, 1996).

Program Control and Learner Engagement

There has been a great deal of research in the area of learner control, much of it dealing with comparing the effects of program control with the effect of learner control (Morrison, 1992; Pollock & Sullivan, 1990; Ross & Rakow, 1981; Steinberg, 1977). In program-controlled CBL, learners follow an instructional path that has been predetermined by the designer. In learner-controlled CBL, learners are typically allowed to control instructional variables such as context, content, sequence, and pacing.

Building an interactive CBL environment that is truly engaging is a difficult task. Csikszentmihalyi's flow theory states that engagement can be defined as a nexus of "external" stimuli that promote the continued use of a computer-based learning environment and "internal" cognitive involvement (Csikszentmihalyi, 1997). The quality of multimedia assets such as images, sounds, and animations is a key factor (stimuli) in getting users interested in the design and development of educational software. Being able to exert control over actions within this environment is ultimately a pleasing experience for the learner. However, it should be noted that controlled processes may direct learners' attention toward the operation of the program rather than the content itself (Chung & Reigeluth, 1992). Because of the limited capacity of working memory, students cannot simultaneously focus on the content area and control the learning process (Park, 1992; Stoney & Wild, 1998; Tsai, 1989).

The CBL designer must decide how much to allow learners to control the program design. In program control, the designer decides the sequence for learners. Learners do not need to know how to control the program and can instead concentrate on the task at hand. They need only follow the predetermined instructional path to complete the task. Cho (1995), therefore, found that learners in a program-control group spend more time understanding the material. Many researchers (Chung & Reigeluth, 1992; Clark & Taylor, 1994) suggest that, if a learner is a novice and if a given task requires more effort, a program control is suggested. On the other hand, other researchers have proposed that allowing learners to control their instruction has intuitive appeal (Csikszentmihalyi, 1997) because it is assumed that individual learners know their own needs best and are qualified to control their own learning (Freitag & Sullivan, 1995; Mager, 1964; Merrill, 1975, 1980).

Program Control and Time on Task

The design of the controlling process in the CBL can influence the amount of time learners spend on task. Time is clearly an important consideration in instructional design (Spotts & Dwyer, 1996). Block (1971) proposed a mastery learning theory that indicated that learners required adequate time to learn for competency. Understanding involves the integration of new information with prior knowledge. Without sufficient time, learners cannot develop new or adapt to existing schema effectively (Garhart & Hannafin, 1986). The more time the learner spends interacting with the elements of instruction or questions, the better the learner will be able to move information into long-term memory for storage and retrieval (Craik & Lockhart, 1972; Slater & Dwyer, 1996). Additionally, Dwyer (1978) suggested that providing sufficient processing time for visuals with realistic details is important. Richly detailed visuals require learners to search for essential learning cues. If insufficient time is given, students may actually choose to ignore the animation.

Mathematical Ability and Learner Control

Accommodating learners' individual differences remains a concern for teachers at all levels. A common but incorrect assumption is that self-paced instructional methods are the best style for all types of microcomputer instruction (Belland, Taylor, Canelos, Dwyer, & Baker, 1985). Other researchers have reported that individual differences influence success in programming and that quantitative mathematical skills lead to success in computer science (Campbell & McCabe, 1984; Lai & Repman, 1996).

Pea and Kurland (1984) doubt the relationship between mathematics and programming abilities. They state that general intelligence instead of mathematical ability influence programming learning. Chee (1993) also supported the idea that mathematical ability is not correlated to programming performance. One of the pervasive findings related to abstract concept learning is that there are distinct differences between low-ability and high-ability learners in perceiving a mental model. Bayman and Mayer (1988) explain that students with high mathematical ability tend to use existing models to interpret learning and that a new mental model may actually distract their learning. However, students with low mathematical ability who presumably lack self-developed models would benefit from a relevant conceptual model provided in the instruction. Therefore, weaker students would be more likely to benefit from program control than would students with strong quantitative backgrounds who would be able to generate their own mental models.

Methodology

Subjects

A total of 186 freshmen in one commercial college participated in this research. The sample was built on the basis of randomly selecting 4 out of 12 classes. All students in these classes enrolled in the computer literacy course that is a common requirement for all college freshmen. Most were novices or had little experience with programming. Students were given extra credit for participating in this research.

Instructional Program

The content for the instructional program was designed to teach the abstract concepts of computer programming language. The CBL program consisted of several programming sequences. Each sequence had about six to eight statements. For example, the following programming sequence was displayed on the screen.

```
TOTAL = 0
DIM A(5) FOR J = 1 TO 4
READ A(J)
TOTAL = TOTAL + A(J)
NEXT J
PRINT "TOTAL IN THE ARRAY"; TOTAL
DATA 10, 20, 30, 40, 50
```

For each sequence, at the top of the screen, the statements were listed and highlighted line by line. At the bottom of the screen, a representative graphic was displayed in an animation format with narrative explanation as shown in Figure 1. For example, when the “Total = 0” statement was highlighted, a number 0 flew into the warehouse “Total.” It was explained with a narration stating that a variable “Total” is like a “warehouse,” and it was stored with a number “0.” When the next statement “DIM A(5)” was highlighted, it was associated with the display of five consequent rooms in the warehouse. When the FOR_NEXT loop was explained, the worker ran into the warehouse as many times as the loop executed. The CBL lesson was developed into three levels of learner control that allowed users to control and elaborate on the animation.

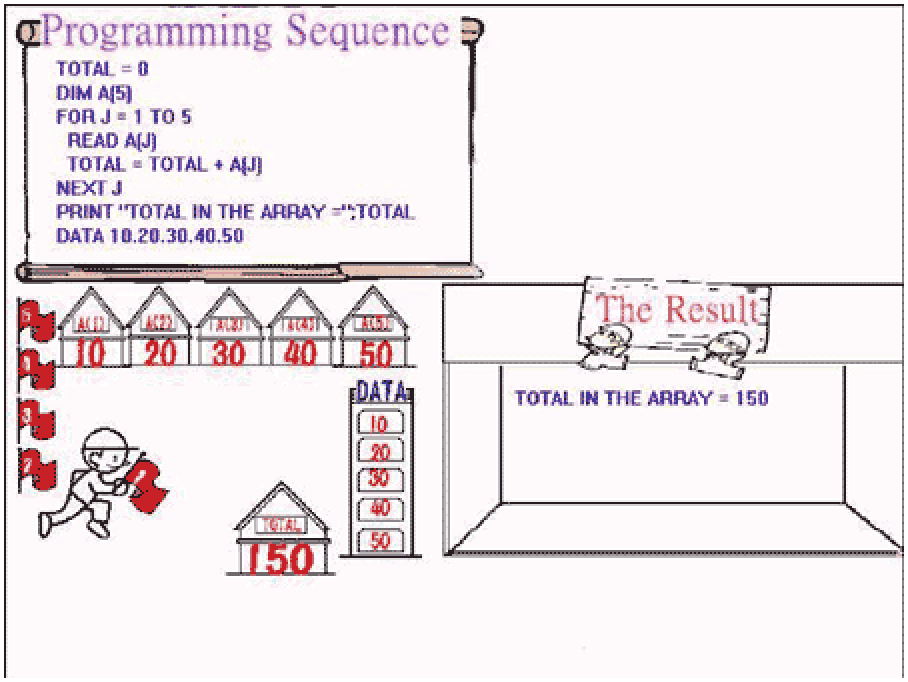

Figure 1. Visual display of the programming sequence excerpted from the animation.

1. *Program control version.* The computer used to display the animation for the whole programming sequence paced the instructional program. The animation for each statement was shown one by one without interruption. At the end of each programming sequence, learners could review the whole sequence by pressing the button as many times as they wished, or they could move to the next programming sequence.
2. *Linear control version.* Students could watch the animation as long as they wished and decide when to click on the button to proceed to the next statement. The sequence of the statement was linear. Rather than forcing subjects to go through the whole program at a pre-determined pace, subjects were able to go through each statement at their own pace. At the end of the program, they could go back to the first statement and review the animation again step-by-step.
3. *Learner control version.* Students could randomly click on any statement in the program and process the animated illustration. The sequence of clicking was nonlinear and totally controlled by the learner. Students could review any animation by clicking on the respective statement as many times as they wished.

Criterion Measures

Pretest and Achievement Posttest

The same test was used for the pretest and achievement posttest. The test consisted of 20 multiple choice questions that contained four response choices per item. A pretest score was obtained before the study to measure students' prior knowledge. The posttest was designed to measure students' understanding of the programming concept. The following test item shows a typical sequence. Upon seeing such an item, subjects would be asked to decide among a choice of possible execution results.

```
FOR A = 1 TO 3
READ M(A)
PRINT M(A) + 1
NEXT A
DATA 7,3,5,2,9,1
```

Test items were validated by inviting three faculty members and instructional designers to review the measurement instrument. These experts were either familiar with the programming language or CBL. Only items receiving agreement by each of the reviewers were included on the test. The result of pilot testing also demonstrated that students' programming performances in the course were highly correlated to the test result ($r = 0.84$). A Kuder-Richardson Formula 20 test was administered to measure the internal-consistency reliability for the test (a: 0.81).

Attitude Questionnaire

Student attitudes were measured using a 25-item Likert-scale survey. Among them, 15 items were related to attitudes toward the CBL program, and 10 items were related to attitudes toward the controlling CBL program. Responses for each item ranged from 1 (strongly disagree) to 5 (strongly agree). The attitude questionnaire was constructed on the basis of the review of related literature. Additional inputs were derived from the recommendations of faculty members and instructional designers who are familiar with measurement instruments. The modified KR-20 was used to measure the internal consistency of the attitude (0.8). The following are samples of attitude questions regarding CBL and controlling.

- I think this CBL software can help me understand the subject area.
- I like the way animation was presented through the controlling button.

Time on CBL

Students recorded the total time it took to review the CBL lesson from beginning to end.

Facilities and Environment

The experiment was conducted in a computer lab equipped with 60 Pentium 586 microcomputers. The software used in this study included Macromedia Director (1984–2001) and Adobe Photoshop (1989–2000).

Procedures

At the beginning of the semester, a pretest and a survey questionnaire were administered to four classes in their respective classrooms. In the survey questionnaire, information regarding student gender, age, and knowledge about the programming language was obtained. All subjects who indicated experience with programming language were eliminated from the study. Only students who were novices or had little experience were retained. Their mathematical abilities were determined by their mathematical scores on an entrance examination, a nationalized test administered to incoming college freshmen. Students were divided into two groups (higher- and lower-ability groups) according to their mathematical abilities. A total of 186 raw scores were included in the statistical analysis. Students in each ability group were randomly assigned to one of the experimental groups. The population of each of the six treatment groups was also stratified by sex so that the proportion of male to female subjects would be the same in each group.

One week later, when subjects arrived for the experiment, they were instructed on how to operate the CBL program properly before the treatment. Subjects were asked to complete their respective CBL lesson at their own pace in the computer lab. Students recorded the time they spent on the CBL courseware and raised their hands to signal completion of the lesson. Following the lesson on the computer, students completed the measurement instruments. These tests were administered using pencil and paper.

Design and Data Analysis

A 3 x 2 (Instructional Control x Ability) pretest and posttest experimental design was used. Posttest scores, time on CBL task, and attitude data were first analyzed using multivariate analysis of variance followed by analysis of variance when a significant multivariate effect was obtained. Significant mean differences were analyzed through separate follow-up procedures using the Least Significant Difference. The level of significance, in all cases throughout the analyses, was set at 0.05.

Read more...