Creating Computer-based Instructional Animations

Donald W. Hall

Abstract

Computer-based animations can enhance learning in lecture courses by maintaining student attention, by making material more interesting, and by adding dynamic visual reinforcement to verbal commentary. Simple frame and path animations may be created without the requirement of computer programming skills. More sophisticated animations may be produced with multimedia authoring software packages that have relatively simple programming (scripting) languages.

Introduction

Animation is the phenomenon wherein two or more static objects are presented in a manner that results in the perception of real motion. Animation also could include flashing or cycling of colors of text or graphic objects

Rieber (1994) listed the following five applications for instructional graphics: cosmetic, motivation, attention gaining, presentation, and practice. Animations have the same potential applications with possible increased effectiveness due to motion. Research on the pedagogical effectiveness of animations is fraught with difficulties, and results have been mixed. Rieber (1994) provides an excellent review of the research literature on animation. At the least, cosmetic animations can add a professional appearance to lecture presentations and send the message that the instructor cares enough about the course to take the time to create animations. Also, motion is known to be a powerful force in maintaining attention (Strauss, 1991). However, inappropriate use of animations also may distract the learner by attracting attention away from the desired focal point of the screen.

I have been using a multimedia approach to teaching a large enrollment lecture course in entomology to undergraduate liberal arts students since the spring semester of 1994. One component of the multimedia is a series of animations I have created to illustrate various biological concepts and insect behaviors. My students love the animations and ask me to include more of them in the course. They frequently comment on their course evaluations that the animations help them learn.

This paper describes techniques for creating simple computer-based animations.

Time Required

Considerable time is required for developing ideas for animations and trying to visualize how they should work to illustrate the desired lesson. This part of the process often occurs throughout the day during other activities that do not require active thought. I often take several weeks of intermittent thought before I figure out how to do a particular animation.

The time required for actual creation of animations depends on the complexity of the graphics and whether programming is required. The greatest commitment of time is required for creation of the graphics to be animated. Commercially available clip-art may be edited for some animations. Otherwise, the graphics must be drawn by an artist or the animator. It may be possible to recruit artistically talented students to draw the graphic images for credit. For my most complex animation, graphics production required approximately two weeks. After the images are drawn, they must be scanned and exported in a format that can be imported into the appropriate animation software.

Typically, 4-8 hours of programming time per animation is required. Much of this time is spent in "fine-tuning" the animation.

Hardware and Software Requirements

Animations may vary considerably in their sophistication and in the complexity of the graphics used. Complex animations may run very slowly on a computer with a slow processor. For this reason, at least a 486 processor (or equivalent Macintosh) is recommended.

Software ranges from simple presentation packages to more sophisticated animation software or multimedia authoring packages. Software packages are periodically reviewed in computer magazines and trade journals. It may be helpful to select software for which technical support is provided by one's institutional teaching resource center. A specialized graphics software package will also be helpful for

Hall is a professor in the Department of Entomology & Nematology, P.O. Box 110620, University of Florida, Gainesville, FL 32611-0620, (352) 392-1901, ext. 113; FAX (352) 1901; e-mail dwh@gnv.ifas.ufl.edu.

modifying graphics and for importing and exporting various graphic file formats.

Animation Techniques

Frame Animation

Frame animations work like the old "flip book" animations. Images and/or their positions are changed gradually on successive screens. The animations are played by rapidly advancing through the screens either manually or with a timer. This type of animation may be created even with simple presentation software packages. A disadvantage is that it may be impossible to run the animation at a sufficient rate of frames per second for the motion to appear acceptably smooth (depending on processor speed and the number and complexity of graphics per screen). The simplest way to create this type of animation is to put all the images to be used on a single screen and make additional copies of that screen equal to the number of steps in the animation. Then beginning at the final screen and working backward through the screens, successively select and delete the unneeded images on each screen. The first screen to have images deleted will then be the final screen in the actual animation. When the animation is completed, the composite slide can be deleted. This method has the advantage that the images may be accurately positioned in relation to each other. Screens five, eight, eighteen and the composite screen of an eighteen frame firefly animation created in Microsoft PowerPoint® is illustrated in Figure 1.

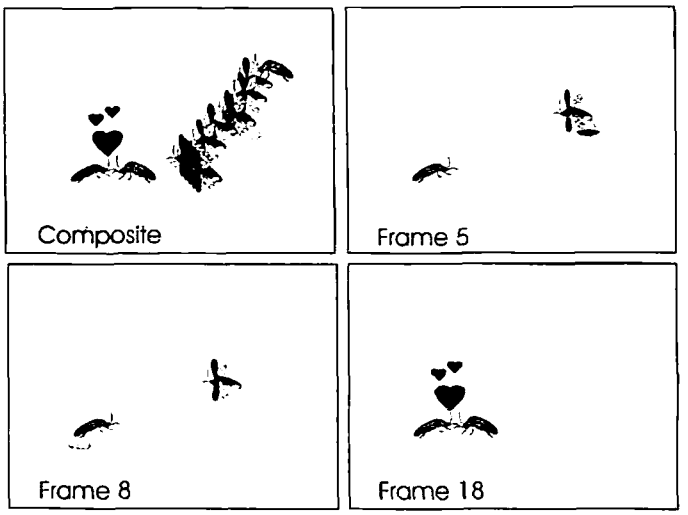

Figure 1. Composite and frames 5, 8, and 18 from firefly frame animation created in Microsoft PowerPoint.

Animation Recorders and Path Animation

Some software packages have animation recorders and/or path animation features. Animation recorders record certain user actions (e.g., clicking and dragging a graphic object with the mouse). The same sequence of actions is then repeated after an appropriate triggering event — typically when the graphic is clicked with the mouse. Software with path animation capability allows one to draw a path with the mouse and store this information as an animation file associated with a specified graphic (Fig. 2). The graphic will then follow the path when the animation is "played." It also may be possible to specify the speed of the path animation and the number of times it runs with each triggering event.

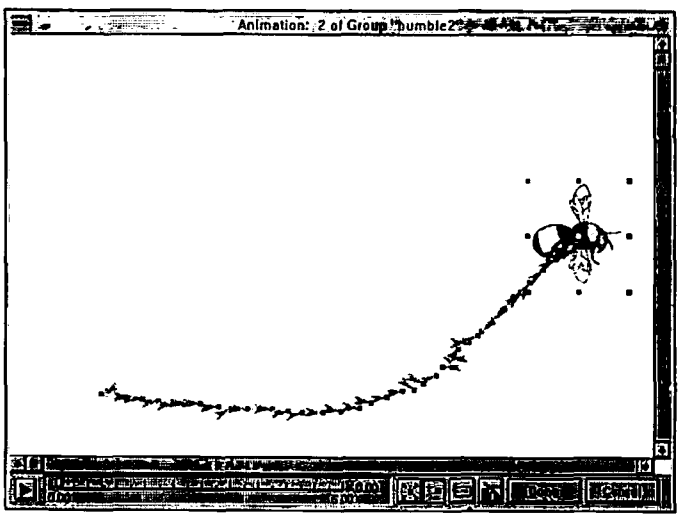

Figure 2. Path animation window with points on animation path created by clicking the mouse.

Programming-based Object Animations

Possibly the most useful animation technique for biology faculty is programming-based object animation. In object animations, the graphic images (objects) themselves are animated by hiding, showing, moving, resizing, reshaping or recoloring them.

This author currently uses a high end multimedia authoring package called Asymetrix™ Multimedia Toolbook™ (MMTBK) for IBM compatibles, that has its own high level programming language (OpenScript®). MMTBK uses books and pages as metaphors for files and screens. In MMTBK, every object placed on the screen may have a programming script associated with it. Upon the specified triggering event (typically a mouse click), the script is executed. Each object in MMTBK has "position" (screen address [x,y coordinates] of upper left hand corner of object) and "bounds" (screen address of upper left and lower right hand corners of object) properties (Fig. 3). These properties are specified in MMTBK page units. One MMTBK page unit equals 1/1440 inch. The position property may be manipulated with the "move" or "set" commands to change the location of an object on the screen. The bounds property may be changed with the set command to modify the location, size or shape of an object during an animation. Scanned or clip-art graphics also may

be animated with MMTBK's "hide", "show", and "move" commands.

Other features that are useful for creating animations are programming control structures. OpenScript control structures are similar to those of traditional lower level programming languages (e.g., Basic, Pascal and C); only the syntax differs. The MMTBK control structures as described by Pierce (1990) are:

If — Executes one statement or series of statements if a condition is true. Executes another statement or series of statements if a condition is false.

Conditions — Executes a statement or series of statements when a specific condition is met.

Step — Executes a statement or series of statements a predetermined number of times.

Do — Executes a statement or series of statements until a condition is true.

While — Executes a statement or series of statements repeatedly while a condition is true.

Other programming-based multimedia software packages (e.g., Apple HyperCard™) have similar capabilities to those of MMTBK but have slightly different programming syntax.

The following example of an object animation to demonstrate ladybird beetle defensive allomone (a defensive chemical) was created with MMTBK. In this animation, an ant walks across the screen and bites the ladybird on the leg. The ladybird reacts by releasing a repugnant defensive chemical allomone from the leg joint. The ant retreats and grooms the allomone from its body. To make the ant walk in two directions requires two ant bodies (one facing right and one left). Each body has two sets of legs in slightly different positions "grouped" onto it. Each set of legs is named and alternately

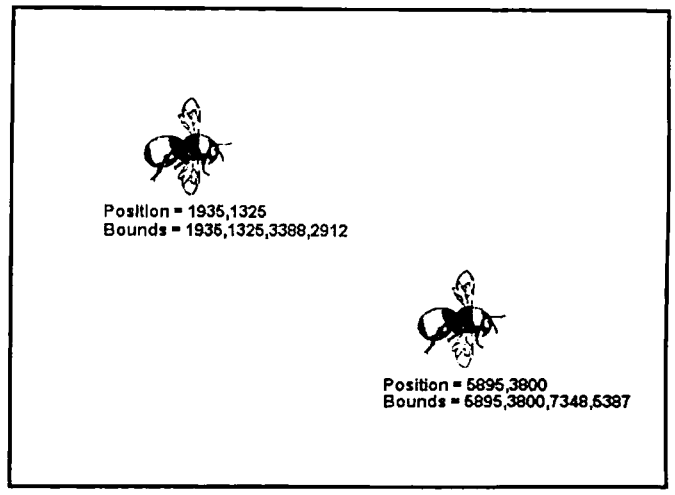

Figure 3. Position and bounds properties of two bumble bee graphics on a Multimedia Toolbook page.

shown and hidden as the whole group is moved across the screen with move commands by changing a variable representing the x coordinate of the group's position property in a do/until loop. The allomone droplet is made to increase in size by incrementing variables in the bounds property of an ellipse that was drawn with the MMTBK drawing tools. Frozen images from the animation are shown in Figure 4.

The script that runs the ladybird beetle animation is associated with a button beneath the ladybird labeled "Reflex Bleeding." The parts of the script that cause the ant to walk

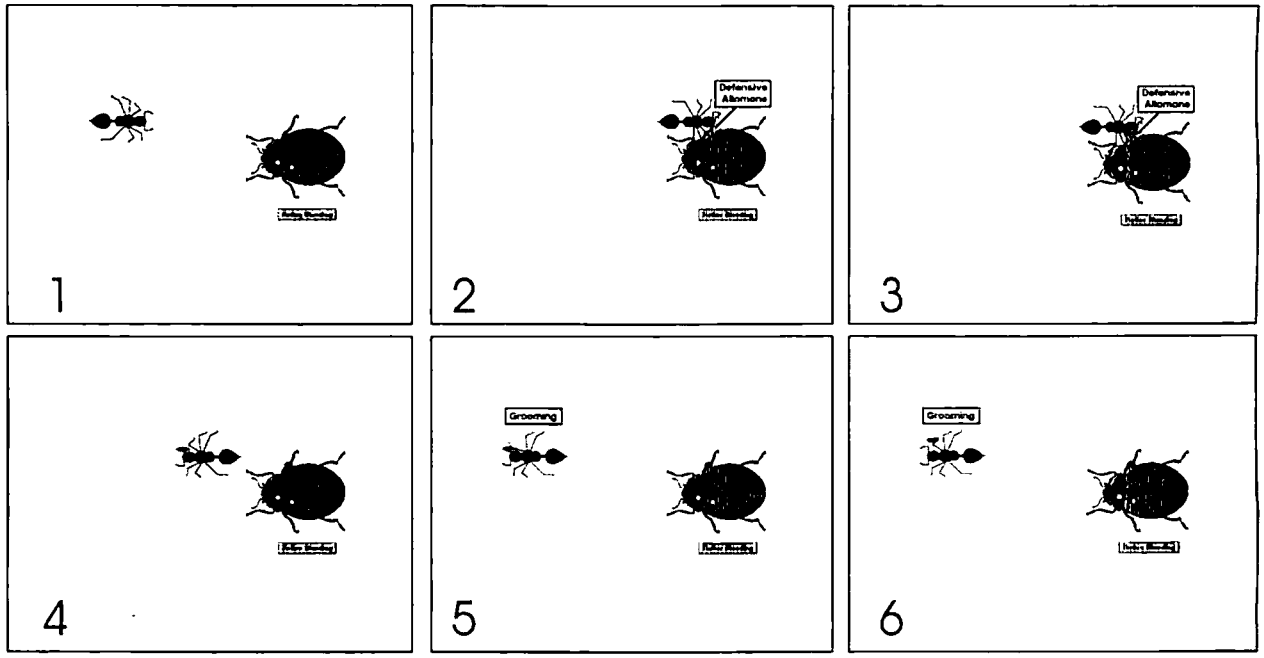

Figure 4. Frozen images from object animation of the action of ladybird beetle defensive allomone.

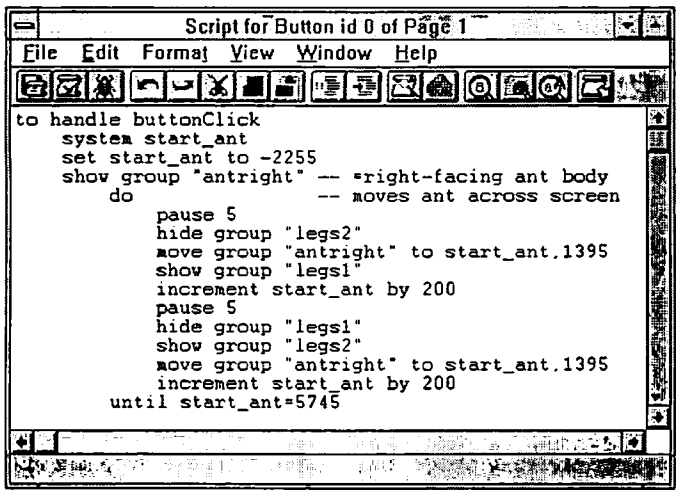

Figure 5. Script window with portion of programming script causing ant to walk from right to left in ladybird beetle defensive allomone animation.

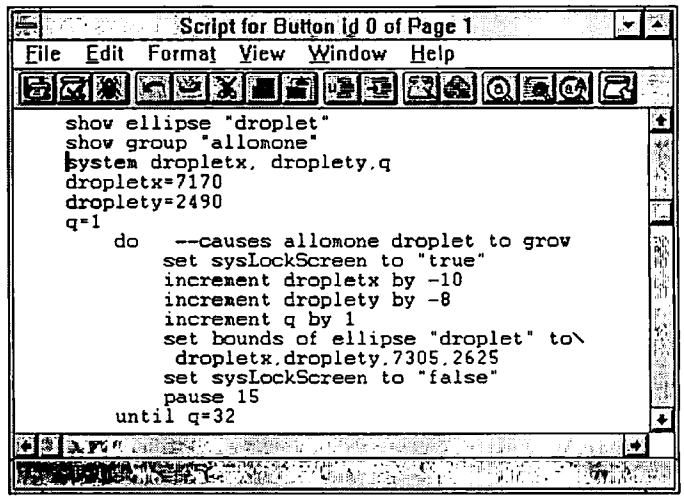

Figure 6. Script window with portion of programming script causing allomone droplet to grow in ladybird beetle defensive allomone animation.

from left to right and the allomone droplet to grow in size are shown in Figures 5 and 6, respectively.

Programming-based object animation is useful for a wide range of animation applications and typically results in a sufficiently rapid display of objects with present processor speeds to give relatively smooth motion. The major disadvantage is the requirement for learning a scripting language.

Animations have potential value in a wide range of agriculture and natural resources courses. It is hoped that this paper will encourage others to create animations for their courses.

Acknowledgments

Fireflies with their wings folded in Fig. 1 and the ladybird beetle in Fig. 4 are CorelDraw™ clip-art images that were edited by the author. All other graphics images were drawn by Margo Duncan. Ms. Duncan was hired on funds from a multimedia grant to the author from the University of Florida's Office of Instructional Resources and a grant from the Office of the Dean for Academic Programs of the College of Agriculture. The time commitment for creation of the animations would not have been possible without the support of my Department Chairman, John Capinera, and Dean for Academic Programs of the College of Agriculture, Larry Connor.

References

Anonymous. (1994). *The Concise Guide to Multimedia.* Belleview, Washington. Asymetrix Corporation.

Pierce, J.R. (1990). *Toolbook Companion.* Redmond, Washington. Microsoft Press.

Rieber, L.P. (1994). *Computers, Graphics, and Learning.* Dubuque, Iowa: William C. Brown Communications, Inc.

Strauss, R. (1991). Design blueprint: some basics of screen design for television-based multimedia. *Multimedia & Videodisc Monitor* (November 1991). p. 24-28.