Computer-based instruction can add interest to your classes by presenting material to students in a different format. It lets students learn at their own paces and is available when you’re not; in addition, it integrates computer utilization into different courses, helping familiarize students with the machine. Computer programs commercially available for science instruction range from question banks to complex simulations which allow students to design and perform experiments. Using tutorials has the same inconveniences as using any other media. They have to be previewed for content level, accuracy, ease of operation, and compatibility with other course material, and often they don’t emphasize what you’d like them to. For many subjects, computer tutorials are not yet available. Two years of searching for tutorials for a nursing pathophysiology course led me to begin writing my own. With a Macintosh computer and Hypercard, a tutorial, once designed, can be made into a computer program in a few days, providing material tailored to the class and the students.

Hypercard allows the construction of branching frame-based tutorials (see Figure 1) - those in which the student sees one computer screen (frame) of information at a time, responds to it, and is routed to another frame depending on the response given.

Making your first Hypercard tutorial is not difficult; at a presentation in Milwaukee, teachers who had never used Hypercard before created three-frame tutorials in fifteen to twenty minutes.

DESIGNING AN EFFECTIVE TUTORIAL

The easiest format to base a frame-based tutorial on is a teaching situation. Imagine yourself tutoring a student about a given concept. You explain it briefly, and then ask a question which requires the student to use the concept. If the answer is correct, you compliment the student and go on to a more complex question, perhaps one which requires some other concepts. If the answer is incorrect, you
might review the information, explaining why the answer was wrong, or you might give more information or another way to attack the problem. This exchange can be duplicated on the computer. The major difference - and the major drawback - is that in a computerized version you cannot simply ask a question and have the student respond freely, but instead usually limit the student to a multiple-choice question and have the computer respond to each choice. It is important, therefore, to design questions in which the wrong answers indicate a particular kind of mistake on the student's part, so that you can have the computer respond appropriately.

Several authors have outlined useful principles for designing computer tutorials (Hazen, 1985; Yang, 1987; Poppen and Poppen, 1988; Weller, 1988), putting greatest emphasis on learner control, interaction, and feedback.

• Learner control

Students take learner control for granted, in my experience, and notice quickly if they can't page forwards and backwards or return to the main menu. These features help make students with any computer tutorial experience feel at home, and reassure those who are unfamiliar with the computer.

• Interaction

However, the single most important feature in students' reactions to tutorials I have used has been interaction. I very rarely present more than two frames without a question, except in review tutorials. Students enjoy having a dialog with the computer; this seems especially helpful with students who are afraid of the machine. Caught up in answering questions and reading the computer's immediate responses, they soon overcome their initial discomfort.

• Feedback

My students prefer to have positive feedback for every correct answer. This sort of continuous reinforcement is recommended for students mastering new skills (Poppen and Poppen, 1988), and would be most appropriate in beginning tutorials or reviews. In tutorials where students must apply information to solve problems, continuous reinforcement can be counterproductive - students can "race" through the program, looking only for right answers and not reading the responses to wrong ones. I therefore sometimes bury positive reinforcement in additional information, so that students must read the whole screen to find out whether or not they were correct.

Incorrect answers require instructional feedback. Many commercially available programs give this feedback at the end of the program, but in problem-solving tutorials I prefer immediate feedback which will make the student deal with an incorrect answer before reasoning further from it. Ideally, the computer should address the mistake the student made and explain how to correct it. Instructional feedback often reviews terms and concepts or offers the student a chance to go through a review program. If a student has given several incorrect answers, the feedback may suggest a review with more urgency. However, I would urge caution in using this strategy; since individually-paced tutorials are perhaps most useful for slow learners, it is important that the computer feedback be supportive and not judgmental.
DECIDING ON CONTENT

There are few limitations on the type of material which can be presented in this way, but I find tutorials most useful for leading students through complex problems in which I want them to reason step-by-step rather than by leaping to conclusions. In my tutorials they apply concepts like osmosis, sympathetic system activity, control of blood pressure, and pH to mapping the sequence of events by which a disease leads to various symptoms and complications (see Figure 2). The computer gives choices for each step in the sequence. If the answer given indicates confusion about the concept, a review is offered; if the student chooses an answer which doesn't directly follow from the previous step, the computer points it out and asks for a different choice. To liven up the tutorials, each is based on a patient's case and the patient sometimes develops complications during the tutorial which require the student to identify the probable problem and make nursing care decisions. To really make it lively, patients can get steadily sicker until the proper decision is made. Another way to introduce variety is with illustrations. Hypercard has several tools which allow you to draw on the screen, and anatomical illustration banks for the program are commercially available.

The best way I have found to plan a tutorial is to make a branching diagram, writing in the contents of each frame and the pathway the student will follow between them (see Figure 1).

PUTTING IT ON THE COMPUTER

1. Getting started

This article assumes that the reader has used a Macintosh and is familiar with the terms click, double-click, open, and drag, with folders, and with selecting commands from a menu. If you're not, please refer to the Macintosh System Software User's Guide (Apple Computer Inc., 1988) section on Basic Macintosh

![hypokalemia
less than 3.5 mEq/L
↓
less K⁺ will enter the muscle and nerve cells
↓
The cell charge will be more negative (hypermolelized cells)
↓
the cells will be harder to fire

Which problem is most likely for the client?
- cramps
- hypoesthesia
- hyperesthesia
- tetany

Figure 2. A sample card from a tutorial on hypokalemia
Techniques. You will need a Macintosh with Hypercard installed on the hard drive.

The first thing to master before beginning to program is a little Hypercard terminology. Hypercard is patterned after a stack of index cards, so each tutorial you create will be called a stack. Within the stack, each of the frames you want the student to see is called a card; the cards are connected to one another by buttons, which can be pushed by clicking on them.

When you open Hypercard you will see the Home card, from which you can reach all of Hypercard’s functions. You should also see a menu list at the top of the screen, displaying the words File, Edit, Go, Tools, and Objects. If you don’t see Tools and Objects, you won’t be able to make a tutorial until you’ve reset the Hypercard User level. To do this, select Last from the Go menu, and click on the Authoring box of the card which appears. This card is only accessible from the Go menu. To get back to the Home card, click on the forward arrow at the bottom of the card.

Take a look at the different menus to orient yourself. The File menu concerns itself with large things - stacks - and allows you to print cards or quit the program. The Edit menu controls cards and the buttons on them. Go deals with moving around in the stack. The Tools menu offers a wide assortment of graphics tools, as well as tools for making fancy printing and for selecting and moving pictures; for information on them, see the Hypercard User’s Guide (Apple Computer, Inc., 1988). For a first session you will only use the top row of tools - the hand icon, button tool, and field tool. The Objects menu, finally, lets you create objects like buttons and fields.

2. Creating the cards in your tutorial stack

When you choose New Stack from the File menu a dialog box will appear on the screen and ask you to name the new stack. Type in the name and click on New; you will create a stack of cards with a background similar to that in the Home card (see Figure 3a). Each Hypercard stack has a background, which appears on every card and is accessible by choosing Background from the Edit menu. Hatch marks will appear on the menu bar to indicate you are working with the background. Since whatever you put on the background appears on every card, I usually do not clutter it up with more than a title, text field, and forward and backward buttons (see Figure 3b). If you have created your new stack from the Home card, its background already contains these buttons and one text field; to see the field, select the field tool (the top right-hand tool) from the Tools menu. Click on the field to select it, and you can drag its corners to wherever you want them. A large text field is best, and I find it simplest to have only one.

When you have finished moving the text field to where you want it, click twice on it to see its properties. You will probably want to change the text size and alignment, and this can be done by clicking on the Font command in the properties box (or by using the Text Style command from the Edit menu).

Now that you have the basic pattern of the cards, you can begin entering the information you want on them (see Figure 4a, page 25). To type in information, choose the hand icon from the Tools menu, click on the card, and type. When you have finished the question card, choose New Card from the Edit menu (or
Figure 3. Steps in constructing your tutorial.

a. The card you will see when you have created a New Stack based on the Home card. To see the text field, select the field tool from the Tools menu.

b. After choosing Background from the Edit menu, a title can be typed in (in this example, I used the A tool from the tool menu to make the large letters). The field tool has been used to stretch the text field across the screen. To control the text in the field, use the Text command from the Edit menu or from the field properties box.

Press open apple - N). A new card with the same background will appear. Type the feedback to each answer on a separate card.

3. Connecting a question card to feedback cards

For students to see a feedback card when they choose an answer from the question card, each answer needs to have a button which is linked to the appropriate feedback card. The quickest way to do this is to use the arrows at the bottom of the screen to go back to the question card you created and choose New Button from the Objects menu. The button which appears in the middle of the screen will be large (see Figure 4b), but you can change its shape by dragging its corners, just as you changed the shape of the text field. Once you have the button shape you desire, choose Copy Button from the Edit menu (or press open apple - C) to save it in the computer's memory for the rest of the session.

When you have dragged the button to the vicinity of an answer (see Figure 4c), click twice on it and you will see a box containing information about button properties. This allows you to change the appearance or response of the button (to make it a plain outline, erase its name from the Name box), but the important property for now is Link to. Click on this box, and a new dialog box will appear on the screen. Your button is now open for linking to another card. Use the forward arrows (either the buttons on the screen or the arrow keys) to page forward to the card containing the feedback you wish the student to see after choosing this answer. Then click on the This Card button in the dialog box. The box will disappear, returning you to the original card. Choose the hand icon from the Tools menu and click on your new button; you should find yourself at the feedback card. Notice that when you created a button, the computer automatically switched you into the button
a. I left the background by choosing Background again from the Edit menu, used the Font command to choose Helvetica 14 pt. text aligned at the Left, and typed in the question and answers.

Now that the text for the text field is chosen, any cards made with the New Card command will use the same text style, so this is the time to make the cards that will contain feedback and type in the feedback.

b. To connect the answers with the feedback, I chose New Button from the Objects menu. Using the button tool, this button can be made. You can choose how many choices of feedback you want. For the example given here, two choices were made.

c. ... and size. I clicked on it twice to see its properties, and used the Link to command to connect it to the feedback card tool - whenever you want to move buttons you must choose this tool, and whenever you want to go back to typing you must choose the hand icon again from the Tools menu. For this reason, it will probably be quickest in future to type in all your cards first and then shift to the button tool and add all your buttons.

You have now made the first segment of a tutorial! Simply repeat these steps, entering more questions and linking their answers to feedback cards. It will be easier now, since when you need a button you can choose Paste Button from the Edit menu (or press open apple-V) to apply the button you've already designed. You will, however, still have to link each button individually to the card to which you want it to take the student.

When finished with the session, simply Quit Hypercard and it will automatically save your stack. One problem with the Macintosh is that your stack may be saved inside a folder, so it will not appear on the directory the next time you want to work on it. If this happens to you, choose Find File from the apple menu at the top of the screen, type in the name of your lost stack, and click on the walking man; it will list the location of your stack.
WHAT'S NEXT?

As you become more expert with Hypercard, you will find many options this article has not mentioned: sounds you can attach to buttons, the possibility of linking between stacks, and creative ways to use graphics. If you really get interested in tutorials, though, you'll probably find yourself wanting to do things like keep score of the numbers of right and wrong answers, generate numerical problems, or even incorporate some simulations or animations into the tutorials. These can be done with HyperTalk, a simple language which lets you write commands which the buttons carry out. To explore this language, I recommend the HyperCard Script Language Guide (Apple Computer, Inc., 1988) or The Waite Group's HyperTalk Bible (Waite et al., 1989).

LITERATURE CITED


BioQUEST will present a series of workshops at the meeting of the annual Association of Biology Laboratory Educators from June 4-8, 1990, at Southwest Missouri State University, Springfield, Missouri. If interested, please contact Professor Barbara Newman there.