DOCUMENT RESUME

ED 463 817 JC 020 333

AUTHOR	Barton, Dick
TITLE	SoSHOWme: Does the Audiovisual Reinforcement of Instructional Content Really Work?
INSTITUTION	El Camino Coll., Torrance, CA.
SPONS AGENCY	California Community Colleges, Sacramento. Office of the Chancellor.
PUB DATE	2002-02-26
NOTE	23p.; Presented at the Annual Technology in Education (TechEd) Conference (Long Beach, CA, February 25-27, 2002). Figures and photographs may not reproduce well.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Animation; *Cognitive Processes; Color; Community Colleges; Computers; Grants; *Interactive Video; Learner Controlled Instruction; Learning; *Learning Processes; Memory; Methods; *Retention (Psychology); Screens (Displays); Technology; Two Year Colleges; *Visual Aids; Visual Perception
IDENTIFIERS	*El Camino College CA

ABSTRACT

This presentation describes the student retention program designed at El Camino Community College (ECC) (California) with a soSHOWme grant. SoSHOWme is the title of a $20,000 Funds for Instructional Improvement grant awarded by the California State Chancellor's Office to measure the improvement of student retention of lecture information that has been augmented by short audio-visual bursts--"bytes" or "clips." The idea was inspired by a presentation made by Doreen Shell and Carl Kozlosky of Lakeland Community College (Ohio). At ECC, student scores on identical questions were recorded for three years prior to the soSHOWme grant. For some courses, core issues--or critical definitions and concepts--were also defined. Only 10% of the critical concepts could be addressed with audiovisual support, while some non-core issues could be addressed visually. It was proposed to measure changes in students' responses to core issues and changes in overall performance. The primary objective of soSHOWme was to increase understanding of core concepts in ECC's Computer Information Systems Introduction to Microcomputers class by 15%, and of overall concepts by 10%. Results included an overall performance improvement of 8% and core issue retention improvement ranging from -9% to +39%. (NB)

SoSHOWme

Does the Audiovisual Reinforcement of Instructional Content Really Work?

Concurrent Session #320

Presented by
Dick Barton
Professor, Business
El Camino Community College
DBarton@elcamino.edu

February 26, 2002
Technology in Education Conference, Long Beach, CA

Table of Contents

Summary

Description

SoSHOWme is the title of a $20,000 Funds for Instructional Improvement (FII) grant awarded to El Camino College by the California State Chancellor's Office to measure the improvement in student retention of lecture information that has been augmented by short audiovisual bursts - "bytes" or "clips".

The idea was inspired by a presentation at October 1997 conference of the League for Innovation in the Community Colleges by Doreen Shell and Carl Kozlosky of Lakeland (Ohio) Community College. Shell and Kozlosky had started in 1995 with videotaped lectures for post-lecture use in the library, and by 1997 were in the process of amending the problems inherent with tape by converting to CD-ROM. In the two years since 1997, the technology had again evolved, and we could identify three distinctions to separate **soSHOWme** bytes from prior audiovisual aids:

1. the clips would be computer-based to facilitate rapid, on-demand presentation by an <u>instructor during</u> lecture instead of by <u>students after</u> lecture;
2. when possible, the clips would be <u>interactive</u>;
3. the clips would be used in <u>media-capable classrooms</u> with large screens and high-lumen projectors to enable visibility without interrupting the "flow" of lecture by pausing and dimming lights. Audio support is minimal to medium; unlike video, sound is always shared with adjacent classrooms.

While not technologically significant, it should be remembered that a key goal of the project was to study the effect of short – ideally 60 seconds or less – clips.

Results

Prior to the grant we had recorded students scores on identical questions for three years, and for some of our courses we defined "core issues" – critical definitions and concepts. Only 10% of the core issues could be addressed with audiovisual support; conversely, some non-core issues readily lent themselves to visualization. Accordingly, we proposed to measure changes in students' responses to core issue questions and changes in overall performance. Test results indicate that:

- Overall performance improved by 8% (short of our target of 10%)
- Retention of information in individual core issue questions ranged from –9% to +39% (ranging widely around our target of 15%).
- In classes where before-and-after metrics are not available, faculty members report a positive, but undocumentable, "better."
- In either case, our testing personnel described class sizes (24 to 54 students) as too small to be totally reliable.

On that basis, the virtue of developing **soSHOWme** clips is definitely slanted towards the positive but still arguable. The purpose of this conference is to share what we got out of it; to show some "products", to share some how-to-do processes, and to let you make your own evaluations.

Grant Basics

Preliminary interest in the project was gauged by student surveys and discussions with faculty already attempting to produce their own audiovisual support. When the grant was awarded, the project was presented to the faculty as a whole at the beginning of the school year, and all were invited to participate. General enthusiasm was high, but individual proposals were not forthcoming because syllabi had been set for the semester. Follow-up recruitment was done in each division; the ideas that emerged from these meetings were developed into what became the sum of **soSHOWme**.

The grant provided a digital video camera, an editing station and software, and a modicum of staff/student support; the college countered with projectors, supplies, and staff support (see Appendix A, "Grant Proposal"). We entered the project knowing that some clips would be "educational" in the traditional sense, and that others would be triggers or pneumonics - items such as already-seen movie clips and ads used in this case to promote retention.

A number of **soSHOWme** bytes were readily created by creating simple "videos" or by re-using available materials. In almost all cases of existing material we edited down. We sometimes divided longer pieces into several smaller sections. We sometimes excised material – the visual equivalent to a text ellipsis. We sometimes added emphasis with video or audio – the visual equivalent to italicizing cited text.

We found it advisable to prepare some clips with no (or optional) audio tracks so instructors could use their own narration and/or not have to worry about speaker systems. We produced some clips with soft background music, but did not find an appreciable benefit that warranted the added file size (see Appendix B, "Does Music Have a Place in the Training Room?").

Copyrighted materials were used under fair-use criteria cited by our PBS representative (PBS being our main source of such material): clips were small excerpted sections of previously published materials; they were presented in instructor-led situations; and they were not distributed. We did include a credit tag at the end of each clip that included copyrighted material.

We found that examples requiring high color accuracy were beyond the ability of even high-quality projectors; a clip designed to show the results of mixing the additive primary colors red, green, and blue did not project well (the same project did work well on individual computer screens).

We use Apple's QuickTime to play the clips; it works on all platforms and is already installed on classroom PCs. QuickTime is the most versatile of media architectures, equally capable of playing anything from low bitrate streaming video to High Definition TV. QuickTime's open architecture and track-based structure means that it can play MPEG, AVI, Flash, and more than 160 other file formats. More importantly, and unlike Microsoft's MediaPlayer, it is the only architecture that adds wired sprite interactivity so that that users can control the presentation. The QuickTime player is a free but fairly large download from Apple; for $30 additional the player can be upgraded to a "Pro" version that allows editing.

Player programs usually present movie, video, and sound files from local drives; browser Netscape-style *plug-ins* typically play the same files from web sites. Microsoft places obstacles in the path of users in both modes.

- In "local" mode, the Microsoft MediaPlayer arrogantly thinks that it can play all files, and tries to hijack QuickTime as well as other files. It fails, and presents the user with a broken-picture icon or a black window. The solution simply is to install the QuickTime player and assign ".mov" files to it.
- For web users of the newest version of Internet Explorer for Windows (not Macintosh), Microsoft's decision to not support plug-ins means that **all** plug-ins are ignored. The solution is to use ActiveX controls, equivalents to the earlier plug-ins. Apple has created such a control for QuickTime (see Appendix C, "QuickTime and Internet Explorer...for Windows").
-

Grant Details

The primary objective of **soSHOWme** is to increase understanding of core concepts in our Computer Information Systems *Introduction to Microcomputers* (CIS 13) class by 15% and understanding of overall concepts by 10%. To broaden the base for the grant, we also attempted to raise test results in courses taught in other Divisions by 12% (core) and 8% (general).

"Core concepts" were defined by compiling a list of questions already used by staff members in tests, and ranking them to determine a consensus. We then produced a list of keywords and concepts that could be reworded and still be within limits of being the same concept. This method worked well for classes in which the entire section used the same test. We could not track classes in which automated programs such as Course Test Manager™ generate a random sequence of questions for each student.

The ideal audience for **soSHOWme** presentations is the student who is in need of an extra sensory "push" to elevate information up to (and beyond) the threshold of perception/retention, but there is no way to identify those individual students. The clips evolved into one more tool at the disposal of an instructor. Parenthetically, the project was NOT targeted at a stereotypical young, media-dulled student. To prepare the grant, we conducted student surveys to assess interest in and definition of the project, and received a strongly positive response in evening classes, which are more heavily attended by returning, adult students.

The grant was for the 2000-2001 academic year. Budget schedules and staff availability forced a late start, and the project was extended to the fall semester of the 2001-2002 school year. The project is now continuing on a self-sustaining basis with no financial impact to the college.

Clip Examples

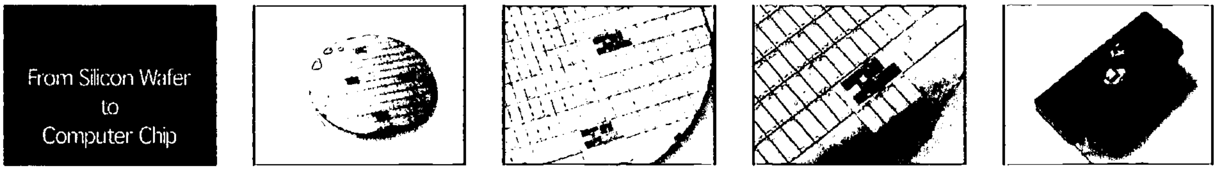
Bytes, or clips, for purposes of this project, were defined as:

- close-ups – close-up views of small items or experiments
- animations - commercials, excerpts, or self-produced illustrations
- videos – commercials, excerpts, or self-produced audio and/or image sequences

1. Close-ups

Show-and-tell demonstrations are commonly used in the classroom, and instructors immediately appreciated the electronic close-ups afforded by **soSHOWme**. We anticipated that such demonstrations would be videoed "live" by student assistants, but were soon convinced us to record close-ups prior to lecture by the awkwardness of having an extra person involved in the demo, keeping optical focus on small objects, and scheduling assistants and equipment. Playback then became an option that could be exercised (or not) at any time during any lecture. Not surprisingly, we found that opening titles, if any, worked best when kept to a less than three seconds.

It should be noted that a variation of our requirement for close-ups had been indicated by Pamela Gore of DeKalb (Georgia) College at the League conference noted above. Gore described the Georgia Geosciences Online distance education program in which close-ups were "live" and distributed simultaneously to multiple campuses.



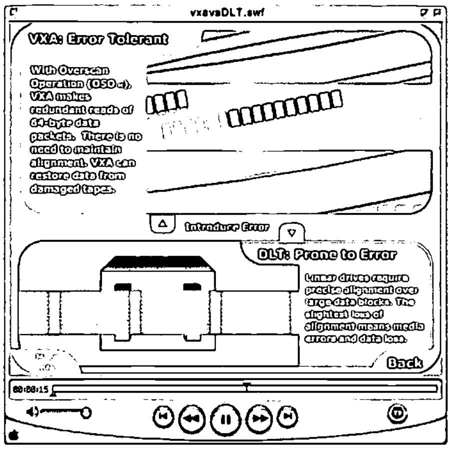
"Wafer to Chip"

2. Animations

We included animations within the scope of the project to allow for the demonstration of complex ideas and concepts that simply could not be filmed "real". Two broad sources were assumed: professionally-produced and self-produced. For the latter we planned to team instructors with student animators from the Fine Arts Division.

Professional animations were relatively easy to acquire; in every case the copyright holder granted permission to use them. Other animations were already a part of the "Instructor's Resource" materials that accompany many textbooks. The Internet proved to be an excellent source of free animations; the "Helical Scan" animation, is from the web page of Exabyte Corporation, and is used by permission.

Animations for the classroom, as opposed to those we see in a movie theater, have evolved from passive to interactive. All of the animations we use are interactive to some degree; the user controls them instead of just watching them. This degree of sophistication undid our plan to selfproduce many animations; most of our student-produced work is static illustration, or dynamic but non-interactive *Flash* graphics, that work better as elements in larger files such as web pages.

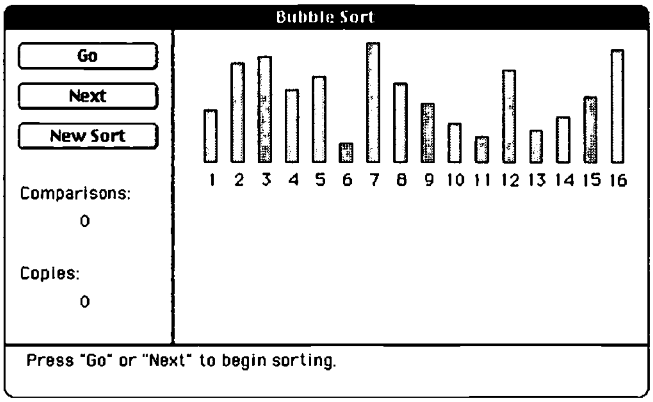


VXA: Error Tolerant

With Overscan Operation (OSO™), VXA makes redundant reads of 64-byte data packets. There is no need to maintain alignment. VXA can restore data from damaged tapes.

DLT: Prone to Error

Linear drives require precise alignment over large data blocks. The slightest loss of alignment means media errors and data loss.



Comparisons: 0

Copies: 0

Press "Go" or "Next" to begin sorting.

Helical scanning technology animation, courtesy of Exabyte Corporation
Sort visualization, courtesy of David Eck, Hobart and William Smith Colleges

Illustration Notes:
The helical scan animation is a Flash .swf file embedded in the Ecrix web page, but is here a standalone file in the QuickTime player with user control buttons at the bottom of the frame, or "skin".
The "Sort" demo included on the workshop CD is a Macintosh-only application; the cross-platform Java applet version downloadable from the author's website requires an Internet connection.

3. Videos

"Videos", in the standard sense of scripted, acted, and photographed combinations of video and audio, were also derived from several sources: commercials, excerpts from our existing library of educational films and videotapes, and videos that we produced for this project.

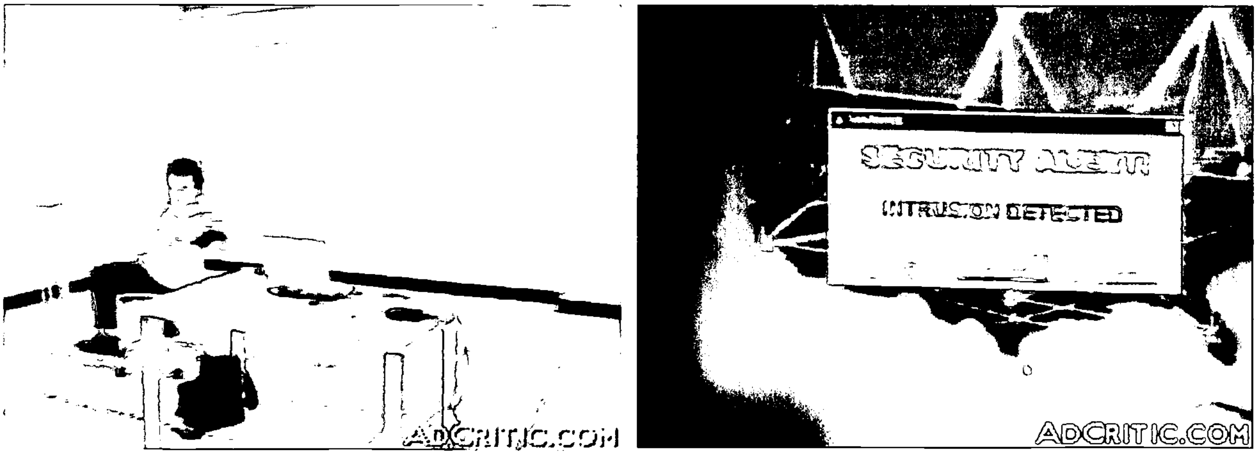
Commercials worked well to reinforce lecture materials. We selected commercials that students had already seen to use as "hooks." In a Skinner sense, we piggybacked our responses onto existing stimuli, with the hope that when the students saw the commercial again – outside of class – they would be reminded of classroom discussions. In many cases the commercials added to class discussions with humor and even with negative examples; the Budget "Bad Plan" and "Aromatherapy" ads, for example, pungently illustrate the results of bad systems analysis, and serve as excellent springboards for "What's wrong with this?" discussion.

Budget Car Rentals "Bad Plan"

Budget Car Rentals "Aromatherapy"

In some cases the reverse occurred - the class aided the commercial; after viewing the PacBell "DSL" ad, students often commented that they had not really understood all of the dialog in the commercial until after we had compared various ways to connect to the Internet. The "Hacker" commercial adroitly illustrates security challenges but does not show any detail; these are completed in classroom discussions.

Pacific Bell "DSL"



Southwestern Bell/Pacific Bell "Hacker"

Successful producers of commercials have learned how to deliver a message in 60 seconds or less, so it is understandable that commercials fit so well into the **soSHOWme** format of short presentations.

Both the Budget Rental ads and the PacBell ads were first used in introductory computer classes during the corresponding sections on systems analysis and telecommunications. The same clips have since also been used in advanced classes in Systems Analysis and E-Commerce.

Until recently, many popular ads were available online from the *AdCritic* website - already captured, digitized, and compressed. This rapid availability is important; ads used as reinforcement tools have an extremely short shelf life. What "works" this semester will be outdated and ineffective next semester. Unfortunately the AdCritic site was closed down at the end of 2001 in the wake of the dot.com meltdown. We have returned to recording commercials off-the-air.

Educational videotapes have been purchased by the college for classroom use, but are used less and less frequently because of the inconvenience of cueing up a tape and the obsolescence of the material. The monitors on the typical video carts are considered "small" when compared to projections; the tape players can be connected to the big screen projectors, but rewiring audiovisual equipment daunts some instructors. Yet parts of those tapes are still valuable, and should be used.

We dubbed small portions of some tapes into a digital format to solve this problem. This gave us the opportunity to perform small edits – the visual equivalents of ellipses and italics – in order to fit the appropriate and current material into a lecture. We made only one copy, in effect the purchased original becoming the copyright-allowable "archive" copy and the dubbed version the "working" copy.

In all cases we were extremely careful to annotate the source of the material, regardless of the length of the excerpt. The technology serves us well here; towards the end of the project we realized that movie files stop on the last frame and leave that image on the screen. We used this feature whenever possible to place the copyright credits on a single frame at the end of the file. *Cleaner*, a program that compresses movie files, even has a High Quality First Frame/Last Frame option to enhance the clarity of text in these frames.

"Into the Future"
© 1997
Commission on
Preservation and Access

"3-4 Years is OLD" courtesy Commission on Preservations and Access

"They've Lost ECC!" Courtesy Commission on Preservations and Access

"ARPA to IMP" Courtesy Oregon Public Broadcasting

Self-produced videos turned out to be the most challenging projects, perhaps because each is a unique combination of instructor personality, teaching style, subject matter, and pedagogical purpose. Even a one-minute video is a complete movie. We wanted to avoid the simple recording of lectures, yet capture something of the instructor teaching. We wanted to keep the bytes short, yet make them substantive.

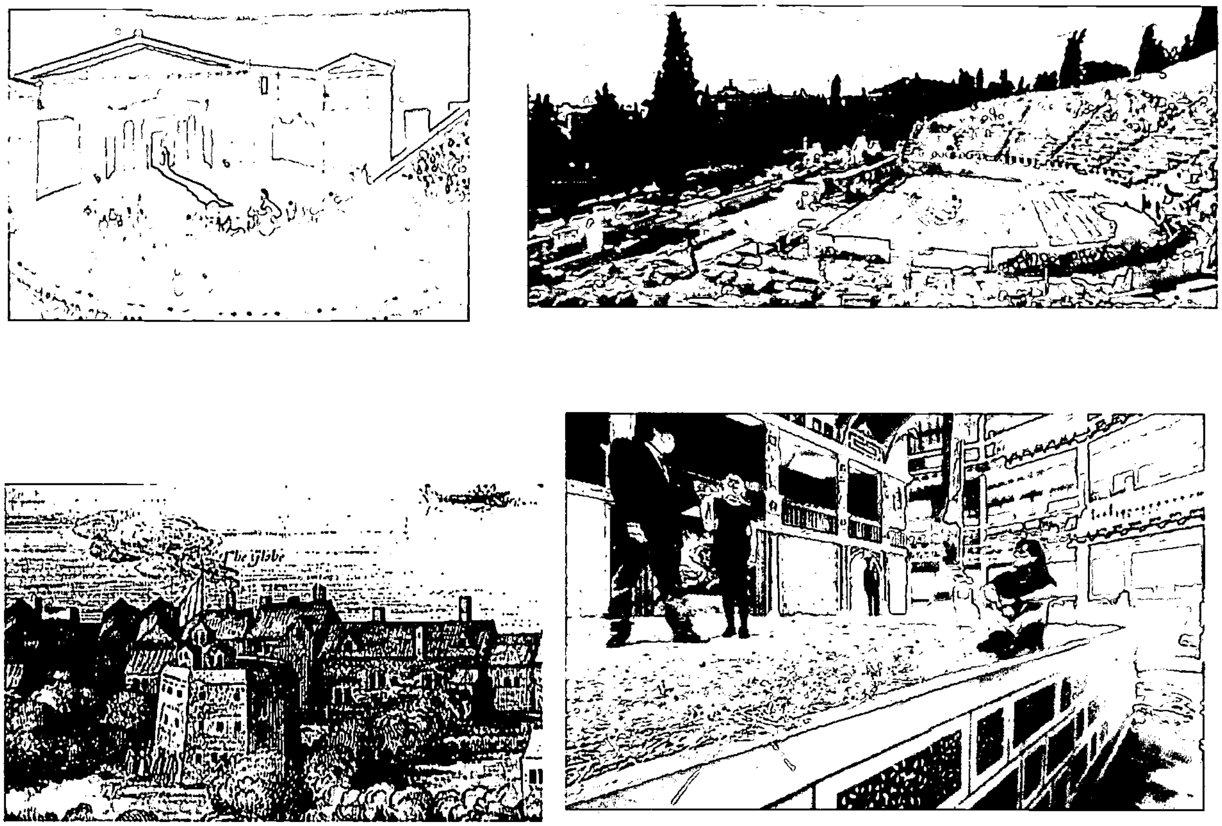
Most projects quickly outgrew their original scope, developing from a simple "Let's do this" into unique and diverse forms. Five typical projects are discussed in detail (below) under "Beyond the Basics".

Beyond the Basics – Self-Produced Clips

Each self-produced, **soSHOWme** clip took a unique form it met the requirements of instructors and students:

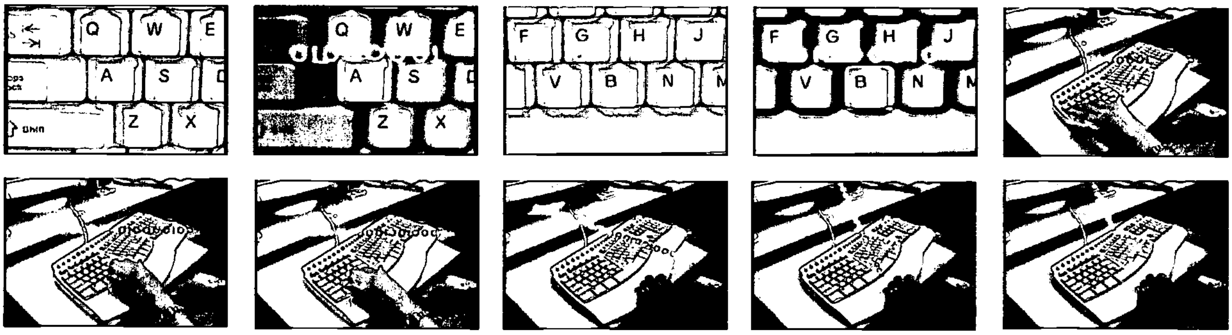
1. For theater history courses we converted faded, decaying filmstrips into still JPEG files, enhanced them with *PhotoShop*, and put them into libraries for use in *PowerPoint* or any other presentation format. Along the way we added immersive 360-degree panoramas so students can explore various theaters. A secondary goal was to explore the virtue of developing similar libraries of digital files for any department.

   Technologically, the span is from individually restoring faded images to the incorporation of interactive virtual reality images into presentations. Both of the panoramas seen here are *multi-node* virtual reality files that allow the viewer to "jump" to different viewpoints. Both were discovered on the Internet, and can be viewed online or downloaded for offline use.



Theater History: .JPEG "slides" and matching QuickTime interactive panoramas.

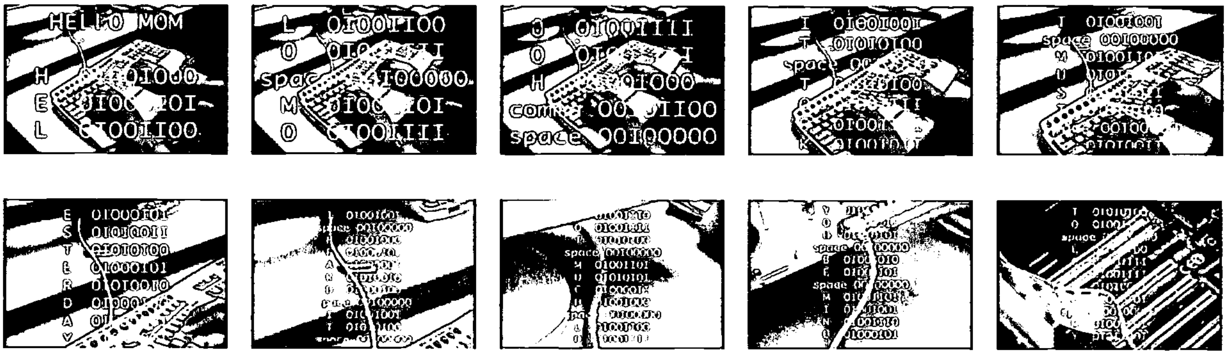
2. For introductory computer classes we produced short videos to show how keyboard characters get changed into ASCII bytes that get stored in the computer's memory. We found that we had to add animated overlays to properly convey the idea of keyboard actions being converted to ASCII bytes, and those bytes traveling into RAM.



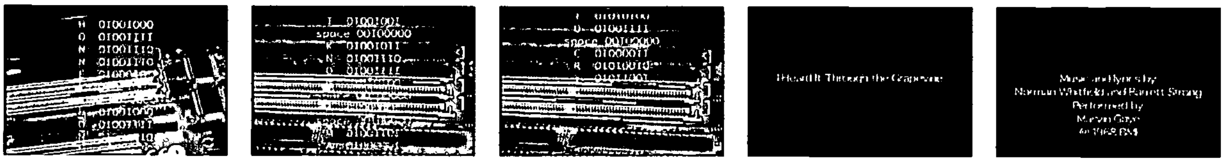
Computer Classes: "ABC to ASCII".

In order to repeat this lesson at a later point in the semester, we shot a longer version that included more keyboarding and more extensive animated text graphics. This version included larger text in a lighter color, and also added a significant audio component: 15 seconds of Marvin Gaye's version of "I Heard It Through the Grapevine". The technology starts with basic video, adds editing and titling with *Final Cut Pro*, and touches on copyright issues involved in the use of commercial materials.

The pedagogy of both long and short versions is to show that each character typed at the keyboard is processed as an ASCII byte, and that a long string of characters is really a long string of bytes. Visually we follow the ASCII bytes as they leave the keyboard and travel down the cable into the RAM chips. The copyright tags at the end of the sequence give credit to the author, performer, and copyright holder. Because of copyright issues involved in distribution (as opposed to presentation), this file is loaded on the server in a restricted area, not accessible to students. To follow copyright guidelines (if not actual law as of this writing), the brevity of the piece ensures that not only must an instructor be present to start the playback, but also to be present for the entire showing.

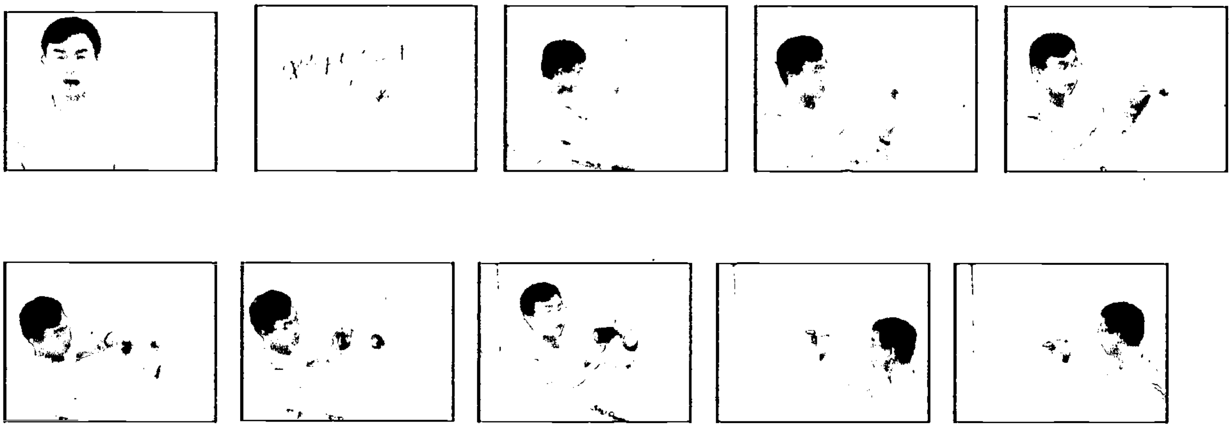


(continued...)



Computer Classes: "Longer Text Strings to ASCII"

3. Calculus students are confronted with two-dimensional textbook illustrations of three-dimensional situations. The convention used in most textbooks is to draw one axis horizontally, another vertically, and to then "project" the third axis by drawing it at a 45° angle to the other two. Depth is not accurately represented because the illustration is still on a two-dimensional plane, and the third axis often looks like nothing more than a line drawn at a 45° angle to the other two axes. For these classes we produced videos that began as "talking head" lectures of an instructor at the whiteboard, moved on to medium shots of two-dimensional and then three-dimensional illustrations, and then to three-dimensional plastic models.



Calculus Classes: "$X^2 + Y^2 = 1$"

The calculus class **soSHOWme** clips were hybrids; unlike in-class clips, they were designed for pre-lecture and post-lecture use, and referenced lectures which already included a great deal of manual visual support such as live board drawings and plastic models. A high-bandwidth version was prepared for in-class use, but was seldom shown; a low-bandwidth version loaded onto the instructor's web page received several hundred "hits".

The primary technological issue involved perceptual phenomena associated with two- and three-dimensional images, which we resolved by camera location. An interactive model that could be manipulated by students would be a far superior solution, and is in our work queue.

The technology exists to generate movies from *Mathematica*, but we recognize that we might be spending considerable effort to assist a relatively small set of students who are not likely to be in need of added audiovisual support in the first place.

4. For general computer classes we produced animations to illustrate various aspects of 'how computers work'. Six such animations were encased in a user/instructor-driven shell to allow the viewer to play and replay any of the videos at will. Technically, each of the animations is a 'child' movie-in-a-movie driven by a 'parent' movie of six command buttons, created with authoring software *WebPainter* and *LiveStage Pro*. Pedagogically, each animation was simple enough to be usable at the beginning of a semester, yet accurate and complete enough to be used at the end of the semester. The animations proved to be too simplistic for our audience, and are being redrawn.

5. For classes in computer operations we produced a series of 10 video tutorials to show the progressive disassembly of a computer. We included close-ups, added graphics - and layered in six user-selectable language soundtracks (English, Chinese, Farsi, Japanese, Spanish, Tagalog, and Vietnamese). In the first pass for these clips, the technology hurdles were how to how to make on-screen 'buttons' that would allow the user to select a language but not be intrusive, and to synchronize the differing lengths of the various sound tracks . In the second pass the challenge was to explore various audio compression schemes – CODECS – to keep the file size to a reasonable level. The third pass started as a casual plan to introduce a text track for the hearing-impaired but grew to address issues introduced by Section 508 of the Rehabilitation Act (29 U.S.C §794d). "508" is not specific to states at this time, but it does provide guidelines that are appropriate and that should be considered when preparing media for electronic presentation and distribution.

   Technologically, the addition of the text track involved perception choices, e.g., scrolling vs. stationary 'subtitle' text, font size, phrasing and punctuation conventions, and synchronizing the text with the video image. In the background, we attempted to develop a process which could be adapted as necessary to retrofit existing media in order to bring it into compliance with Section 508 guidelines.

   The production of this series embraced almost every component of the project. Accordingly, the project is used in the tutorials that follow.

Tutorial: How to Create a Clip

Basic clips are made by capturing existing media or filming scenes, and adding any desired titles and/or credit tags. Anything beyond that level requires a varying amount of preparation. This document does not attempt to deal with filmmaking or visual communication, but does define the basic process for the non-film-making instructor.

Pre-Production

Most experienced filmmakers recommend four steps before shooting an inch of film:

1. The IDEA
   An individual or group has an idea and brainstorms it, visualizing it from beginning to end. Most of us have models in mind – other images, movies, or commercials that have captured our attention. We can study those models for camera angles, for storyline or instruction sequence, and for visual style. Most educators know this as "Borrow from one source, it's stealing. Borrow from two, it's research".
2. The OUTLINE
   After brainstorming, all the sections or parts of the movie are listed, from beginning to end.
3. The SCRIPT
   Every word that will be spoken by actors or a voice-over narration track is written. Projects such as close-ups and still photographs do not require scripting.
4. The STORYBOARD
   The last step in pre-production is storyboarding. Many non-cinema teachers consider it the pictorial equivalent of a *PowerPoint* "light table" view. The storyboard is a comic book version of the movie that helps everyone visualize all of the scenes in the project. The storyboard is used during filming as a checklist, and when editing as a guideline.

Production

Scenes can be shot in any sequence. When filming is complete (or almost complete), editing can begin.

Log and Capture with an Editor program

Scenes recorded on tape must be entered into a computer for editing. At its heart, this process is the same as entering data into a word processor or a spreadsheet. On the other hand, the data has already been entered when the camera was running, so we now have a step that is more like using the Delete key when removing erroneous data; we have to prune out the "bad" scenes and enter only the "good" scenes. The linear nature of tape requires that we watch everything ("log"), before copying the selected scenes to the hard disk ("capture").

We use Apple's *Final Cut Pro* as our editor program because of its power and versatility. On most product comparison charts it stands above Apple's *iMovie* and Adobe's *Premier*, and slightly below high-end programs. We record on miniDV tape, so the data is already digitized (existing analog VHS tapes must be copied onto digital tape). Whatever software is used, the human editor plays back the digital tape via the editing software, logging all scenes and noting the scenes to capture. When the entire tape has been logged, the editing program rewinds the tape and captures the selected scenes to the hard drive.

Prepare Titles and Credit Tags

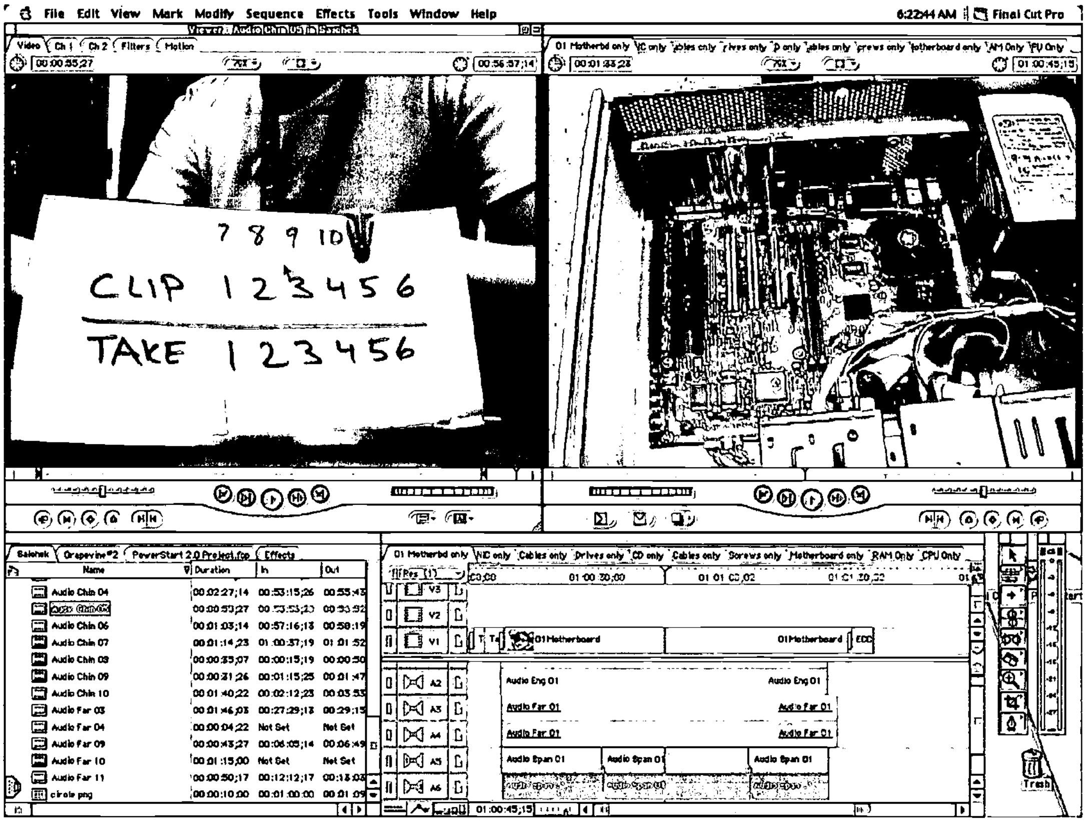
Most editing software includes tools for creating text screens such as titles and credit tags, and providing animations such as scrolls, text-on-a-path, fly-throughs, etc. More complex tags with graphic elements such as logos can also be generated within the editor.

Assemble the Scenes

The captured scenes are then assembled in the desired order – hopefully that which was defined on the storyboard. The term "non-linear editing" refers to the ability to go back and insert a scene into an existing sequence; computer users take this cut-copy-paste algorithm for granted, but it was not possible in video editing until the computer was introduced into the editing process.

Shown below is the *Final Cut Pro* screen. The workflow process is to

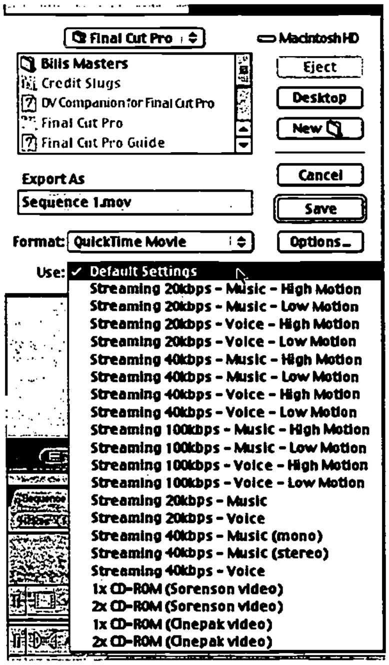
- log-and-capture scenes which puts them into the *bin* at the lower left,
- drag the scenes to the *viewer* window at the upper left to modify them for a particular usage in the total production (e.g., color adjustments, start and end points within the scenes)
- drag the adjusted scenes to the *canvas* window at the upper right
- monitor the overall *timeline* at the lower right.



Name	Duration	In	Out
Audio Chin 04	00:02:27;14	00:53:15;26	00:55:43
Audio Chin 05	00:00:53;27	00:53:53;23	00:53:52
Audio Chin 06	00:01:03;14	00:57:16;13	00:58:19
Audio Chin 07	00:01:14;23	01:00:37;19	01:01:52
Audio Chin 08	00:00:35;07	00:00:15;19	00:00:50
Audio Chin 09	00:00:31;26	00:01:15;25	00:01:47
Audio Chin 10	00:01:40;22	00:02:12;23	00:03:53
Audio Far 03	00:01:46;03	00:27:29;13	00:29:15
Audio Far 04	00:00:04;22	Not Set	Not Set
Audio Far 09	00:00:43;27	00:06:05;14	00:06:49
Audio Far 10	00:01:15;00	Not Set	Not Set
Audio Far 11	00:00:50;17	00:12:12;17	00:13:03
circle png	00:00:10;00	00:01:00;00	00:01:09

Joins between scenes in the timeline may be styled with any number of transition effects. Audio and video tracks are linked to ensure synchronization, but may be separated and edited independently; a descriptive narration of an object can have a visual close-up inserted in the middle without deleting the existing voice narration.

The project may be previewed and adjusted as necessary. New scenes can be added; existing scenes can be shortened or lengthened, or entirely new scenes can be filmed, log-and-captured, and added to the project's bin.



Export As: Sequence 1.mov

Format: QuickTime Movie

Use: ✓ Default Settings

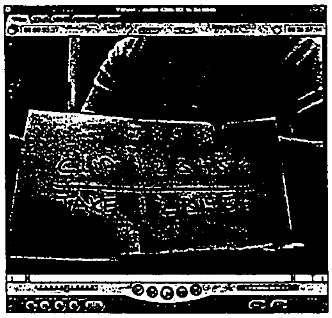
Final Cut Pro
Macintosh HD
Bills Masters
Credit Slugs
DV Companion for Final Cut Pro
Final Cut Pro
Final Cut Pro Guide
Eject
Desktop
New
Cancel
Save
Options…
Streaming 20kbps - Music - High Motion
Streaming 20kbps - Music - Low Motion
Streaming 20kbps - Voice - High Motion
Streaming 20kbps - Voice - Low Motion
Streaming 40kbps - Music - High Motion
Streaming 40kbps - Music - Low Motion
Streaming 40kbps - Voice - High Motion
Streaming 40kbps - Voice - Low Motion
Streaming 100kbps - Music - High Motion
Streaming 100kbps - Music - Low Motion
Streaming 100kbps - Voice - High Motion
Streaming 100kbps - Voice - Low Motion
Streaming 20kbps - Music
Streaming 20kbps - Voice
Streaming 40kbps - Music (mono)
Streaming 40kbps - Music (stereo)
Streaming 40kbps - Voice
1x CD-ROM (Sorenson video)
2x CD-ROM (Sorenson video)
1x CD-ROM (Cinepak video)
2x CD-ROM (Cinepak video)

The file is then rendered, and finally exported in the desired file format. Like the "Save as…" command, the Export command provides the ability to direct the finished file to any storage location. More importantly, the Export command offers options for various compression schemes such as streaming video or CD-based video so that the user can select the most optimal scheme. The same project can be exported in any number of schemes.

"Default" is shown in this example to NOT compress the movie because we found it advisable to process the movie files further with Terran *Cleaner*, a program designed for that specific purpose. Graphic images look darker on PCs than on Macintoshes, for example, and *Cleaner* allowed us to boost the gamma of videos to a compromise level. Audio filters allowed us to remove sibilant "S" from dialog and ventilator fan noises.

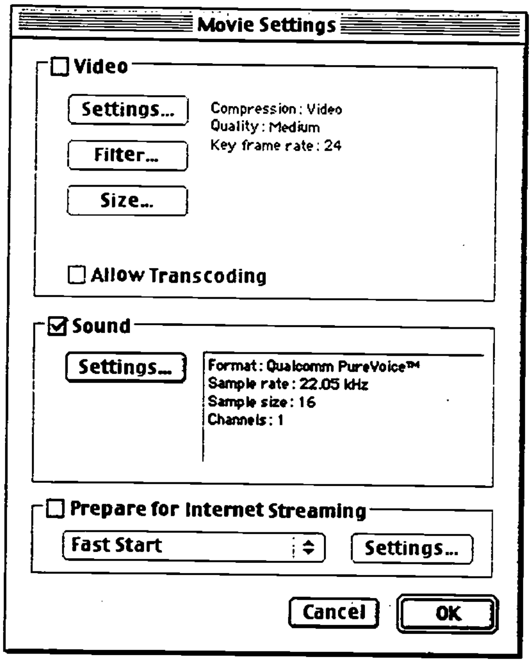
Tutorial: Adding Alternate Audio Tracks

We added these tracks after the clips were (we thought) finished. We gave the scripts from Step 3 of Pre-Production to our readers to translate, and then had them read from their translations while watching the existing movies - in effect creating new movies. Impromptu changes were made to fit the widely varying lengths of audio in different languages; "CPU" may "CPU" in most languages, but "motherboard" often needs a more literal translation. Our readers were computer-savvy, and conversant with both the street and formal computer terms for the components of our demonstration computer.



Although destined to be audio-only tracks, the scenes were shot with accompanying video of a "clapboard" defining the clip and take. This allowed editors with no knowledge of the alternate language to distinguish one take from another. The reader – in the case the only person familiar with the content of the audio – selected the best take to use.

We evaluated the on-camera microphone and a high-quality external microphone, and found them to be fairly similar; the on-camera microphone picked up some camera motor noise, but the high-quality microphone picked up more ambient office sounds.



Movie Settings

□ Video

Compression: Video
Quality: Medium
Key frame rate: 24

□ Allow Transcoding

☑ Sound

Format: Qualcomm PureVoice™
Sample rate: 22.05 kHz
Sample size: 16
Channels: 1

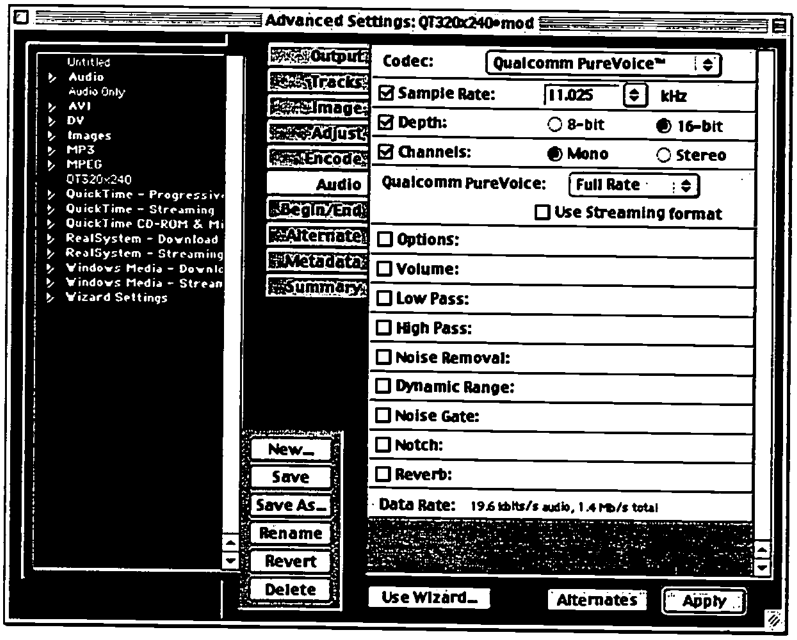
□ Prepare for Internet Streaming

Fast Start

After the alternate language scenes for all ten clips were shot, the tapes were logged and captured as described above. Each of the ten was then stripped of its video, and temporarily layered into the existing movie to ensure that the spoken text matched the video.

The sound tracks were then exported as audio-only movies ("Video" unchecked). They could not simply be left in the original movie because the function of an editor program is to merge all of the sound tracks into a single stereo pair. In the case of multiple languages the result would be a babble of all the tracks playing simultaneously. The interactivity to control which track should "play" is resolved in the next step.

The addition of multiple audio tracks presented the challenge of file size; the movies remained the same length, but exploded in size with the addition of additional languages. We experimented with various audio compression schemes (CODECs) to reduce the size of the audio files, and found that the Qualcomm *PureVoice*™ worked the best. We batch processed the 10 audio clips of each language through Terran *Cleaner* software to achieve a reduction factor of almost 75%; a 1 MB file would be reduced to between 220 KB and 260 KB with no loss of sound quality.



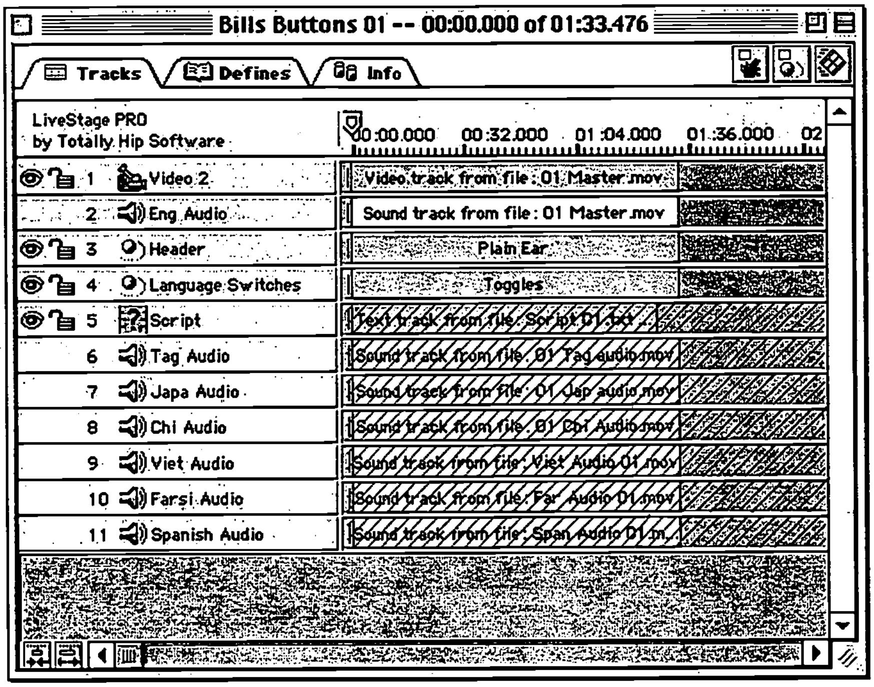
Advanced Settings: QT320x240•mod

Codec: Qualcomm PureVoice™
☑ Sample Rate: 11.025 kHz
☑ Depth: ○ 8-bit ● 16-bit
☑ Channels: ● Mono ○ Stereo
Qualcomm PureVoice: Full Rate
□ Use Streaming format
□ Options:
□ Volume:
□ Low Pass:
□ High Pass:
□ Noise Removal:
□ Dynamic Range:
□ Noise Gate:
□ Notch:
□ Reverb:
Data Rate: 19.6 kbits/s audio, 1.4 Mb/s total

Tutorial: Adding User Control Buttons for Interactivity

We use *LiveStage Pro* from Totally Hip Software as the authoring program to add user control to the *QuickTime* movies. Similar to the more familiar *Director*, *LiveStage Pro* allows the user to add and combine video, images, sounds, and, yes, *Flash* – and script any of those objects with a scripting language ("QScript" in this case). *LiveStage Pro* has an element called a wired sprite that enables interactivity. A wired sprite can be as static as a button or as dynamic as user control over the timeline.

The program necessarily has different screens for the various functions. Shown below is the Timeline window, where elements are added by dragging them into the grey area at the bottom of the screen. Item may be renamed if desired. In this example:



1	Video 2	Video track from file : 01 Master.mov
2	Eng Audio	Sound track from file : 01 Master.mov
3	Header	Plain Ear
4	Language Switches	Toggles
5	Script	Text track from file: Script01.txt
6	Tag Audio	Sound track from file: 01 Tag audio.mov
7	Japa Audio	Sound track from file: 01 Jap audio.mov
8	Chi Audio	Sound track from file: 01 Chi Audio.mov
9	Viet Audio	Sound track from file: Viet Audio 01.mov
10	Farsi Audio	Sound track from file: Far Audio 01.mov
11	Spanish Audio	Sound track from file: Span Audio 01.m..

Video 2 and **Eng Audio** are the original movie from *Final Cut Pro*,
Header is the graphic of a human ear; mouse entry toggles the display of language switches,
Language Switches is the area where all the language buttons are gathered; mouse exit toggles hiding all buttons except the ear,
Script is the subtitles (added in the next tutorial),
Tag Audio, Japa Audio, etc. are the alternate language movies.

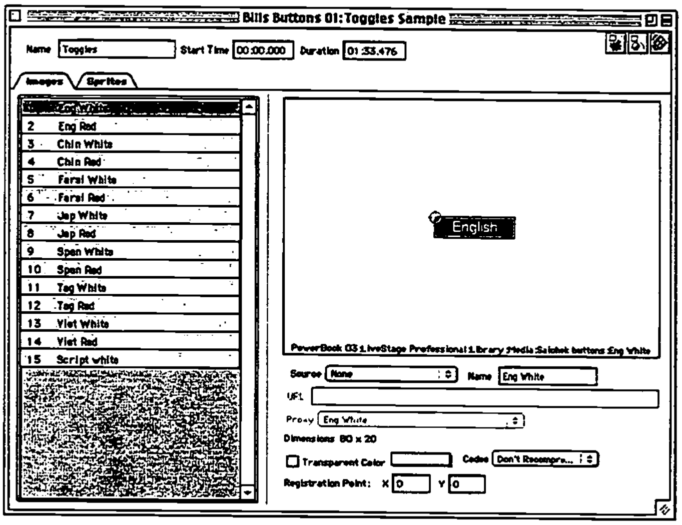
The diagonal lines through tracks indicates that they are disabled (not active). The scale at the top of the timeline indicates time. In this case all of the items except the Script are the same length.

Track items are automatically appended to the bottom of the list as they are added to the project; they may then be moved up or down as desired.

Graphics for the "Up" and "Down" state of each button were created in Totally Hip's *Web Painter* program, which is included in the box with *LiveStage Pro*. It is not as sophisticated as *Flash*, but met our needs adequately. Totally Hip does not exhibit a "not invented here" phobia about *Flash*; they promote the combination of *Flash* and *LiveStage Pro* as companion programs to enhance each other.

Once placed in the common library, the graphics were available for all ten of the clips.

Note that the Start Time and Duration are denoted at the top of the window. In our case, we need the buttons to be available – but not visible - for the entire duration of the clip. We achieved this effect by keeping the buttons enabled, but moving them "behind" the movie by changing their layer. This is where the wired sprite concept came into play.



Bills Buttons 01:Toggles Sample

Name: Toggles Start Time: 00:00.000 Duration: 01:33.476

Images / Sprites

	Eng White
2	Eng Red
3	Chin White
4	Chin Red
5	Farsi White
6	Farsi Red
7	Jap White
8	Jap Red
9	Span White
10	Span Red
11	Tag White
12	Tag Red
13	Viet White
14	Viet Red
15	Script white

PowerBook G3:LiveStage Professional:Library Media:Salobek buttons:Eng White

Source: None Name: Eng White

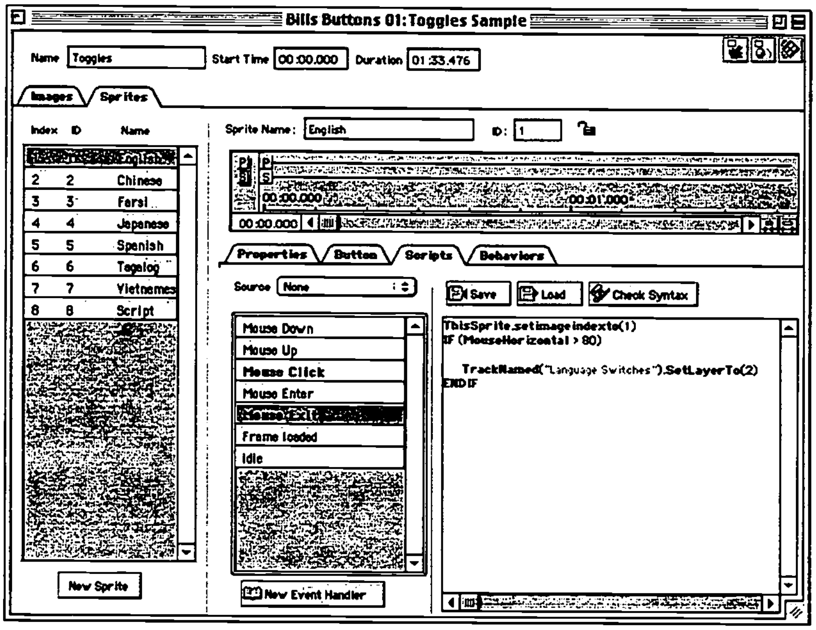
URL:

Proxy: Eng White

Dimensions: 80 x 20

□ Transparent Color Codec: Don't Recompre...

Registration Point: X 0 Y 0



Bills Buttons 01:Toggles Sample

Name: Toggles Start Time: 00:00.000 Duration: 01:33.476

Images / Sprites

Index	ID	Name
		English
2	2	Chinese
3	3	Farsi
4	4	Japanese
5	5	Spanish
6	6	Tagalog
7	7	Vietnames
8	8	Script

Sprite Name: English ID: 1

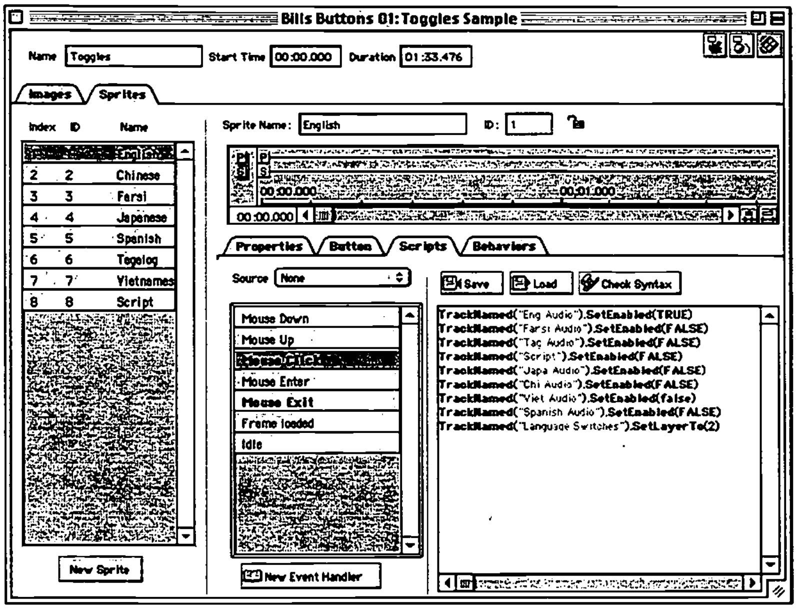
Source: None

```
ThisSprite.setimageindexto(1)
IF (MouseHorizontal > 80)

   TrackNamed("Language Switches").SetLayerTo(2)
ENDIF
```

In the Sprites tab of the same window we can create the Sprites (listed in the left pane), and the scripts (wires) to be associated with them in the right pane. Syntax for QScript is fairly straightforward; most command structures can be dragged from the script library into the white area at the bottom right. The syntax is denoted by blue text; the user enters variables by filling in the blanks. Clicking the "Check Syntax" button confirms proper syntax before the entire project is compiled.

This sample shows hiding the area of buttons when the mouse exits it. The same script is used for all eight sprites.



Name: Toggles Start Time: 00:00.000 Duration: 01:33.476

Sprite Name: English ID: 1

Index	ID	Name
		English
2	2	Chinese
3	3	Farsi
4	4	Japanese
5	5	Spanish
6	6	Tagalog
7	7	Vietnames
8	8	Script

Source: None

- Mouse Down
- Mouse Up
- Mouse Click
- Mouse Enter
- Mouse Exit
- Frame loaded
- Idle

```
TrackNamed("Eng Audio").SetEnabled(TRUE)
TrackNamed("Farsi Audio").SetEnabled(FALSE)
TrackNamed("Tag Audio").SetEnabled(FALSE)
TrackNamed("Script").SetEnabled(FALSE)
TrackNamed("Japa Audio").SetEnabled(FALSE)
TrackNamed("Chi Audio").SetEnabled(FALSE)
TrackNamed("Viet Audio").SetEnabled(false)
TrackNamed("Spanish Audio").SetEnabled(FALSE)
TrackNamed("Language Switches").SetLayerTo(2)
```

This script demonstrates enabling the selected audio track and disabling the other tracks on a Mouse click.

We used the same Track names in all ten clips, so the scripts could be copy/pasted from one to the other.

Tutorial: Adding Subtitles with User Control

Our final (so far) step has been to build the text track(s) that would become the subtitles. In our case, we started with the scripts that we used for the original shoots. After reviewing the movie into which the text would be placed, we found that about 30% of the text had to be cut. It takes longer to read than it does to hear, especially when the viewer is trying to watch a video and read the subtitles at the same time. Much of the text required further editing for phrasing and timing.

The text will end up being another "track" in the movie, so we must add some formatting data to tell *QuickTime* how to handle it. In effect, *QuickTime* reads a form of tags similar to the way browser software read HTML tags. As with hand-coding HTML, we use a simple text processor to create this code. A sample script starts with a heading block:

```
{QTtext}{font:Arial}{plain}{size:12}{textColor: 0, 0,
0}{backColor: 65535, 65535, 65535}{timeScale:60}
{timeStamps:absolute}{language:0}{textEncoding:0}
```

This creates black Arial 12 point subtitles on a white background. The white background opaques some of the video but aids legibility; these aren't foreign art movies where the viewer can miss any of the subtitles. 12 point type fits our relatively small screen size, but is not absolutely 12 points; the font will scale up and down if the player window is resized.

We tested subtitles formated as moving "Times Square marquee" and as stationary blocks. The marquees were universally rejected, so we used blocks. The blocks are best edited into phrases about 5 words long. Our testers encouraged us to develop a non-standard set of punctuation to meet the requirements of frame-based reading. A script edited for subtitles might look like this:

Omit most commas between clauses…

...and use ellipses instead....
...at both ends of a block...
...to indicate to the reader...
...that text has preceded, and...
...that more text is to follow.

Tags for the times when the subtitles are to appear and disappear were then entered. The QuickTime timecode syntax is [hh:mm:ss.fff], where hh=hours, mm=seconds, ss=seconds, and fff =frames (60 frames to a second in our "timeScale" parameter above, but fine-tuneable to 600 frames per second). We found it best to allow 2 to 4 seconds for each text subtitle. Assuming a start time 8 seconds into the movie, the script above would now look like this:

[00:00:08:000]
Omit most commas between clauses...
[00:00:11:000]
...and use ellipses instead....
[00:00:13:000]
...at both ends of a block...
[00:00:15:000]
...to indicate to the reader...
[00:00:17:000]
...that text has preceded, and...
[00:00:19:000]
...that more text is to follow.
[00:00:22:000]

The timecode after the last subtitle is terminated with a final <Return>, and the file is saved as plain text. It is now ready to be converted into a movie track.

In the *QuickTime* player, the **File / New** command is used to create an empty file. Select **Import** to bring in the text file of subtitle, clicking the **Convert** button and then the **Options** button. For our clips, we chose options to set the width of the subtitles to 336 pixels and the height to 18 pixels to fit a preset area at the bottom of the existing frame. Preview the movie to verify that all of the blocks are present and the timings don't overlap; at this time you will see only the subtitles, presented for the durations specified. This is a text movie, so there is no video. This serves as a good legibility test; some phrases may not "read" well because of the number of characters on the screen, the phrasing, or the overall flow. We usually had to edit the textfile and re-import it several times to get it right.