

Creating Automated Tutorials for Online Technology Classes

Prepared by

Jason G. Caudill

Assistant Professor of Business Administration

Carson-Newman College

2130 Branner Ave.

Box 71887

Jefferson City, TN 37760

Prepared for

2008 Instructional Technology Conference

Middle Tennessee State University

Abstract: In order to transition technology classes from the classroom to an online environment it is necessary to provide students with examples of how to perform the operations that are being taught in the class. Many technology textbooks provide screen captures and explanatory text for how to do things but this is rarely as useful to students as seeing demonstrations of the same procedures. By using automated tutorials it is possible to give students the same benefits of watching a demonstration that traditional students receive in the classroom, only with the added benefit of an asynchronous environment.

Introduction

This paper is based in part on the case of the author's experience in teaching an introductory college computer class in a traditional classroom format one semester and then transitioning the same course, with the same texts, to an online format the following semester. The key challenge of the online environment was to provide students with examples of how to perform complex operations, even how to simply navigate what was for many of them an unusual software interface, while maintaining an asynchronous, 100% online environment.

The solution employed in this case was to take the textbook practice materials, which were presented in the traditional class format, and move those materials into self-paced automated tutorials that were accessible to the students 24 hours a day.

The Challenges of Online Technology Instruction

Many times it is difficult for students to translate static displays of software operations in a textbook or digital document into the steps they need to take to perform that same operation on their own. To supplement these static displays students need to see the operations performed in order to understand how to function in the live, dynamic environment of actual software use.

In addition to the need for these demonstrations in the initial teaching of technology there is also the issue of addressing student questions. In the online environment instructors do not have the luxury of simply taking a moment at the beginning or end of a class session to address student questions; they can come in at any time and may need to be answered at any point during the term. Automated tutorials can

help address this by providing instructors with a quick method to capture on-screen activity, annotate it, and distribute it for students to use as an additional reference.

One of the limiting factors of face to face technology instruction that is actually avoided in the online format is the need for some students to see an example more than once to understand it. While repeating examples in a physical classroom can be limited due to time constraints, tutorials prepared in advance and deployed online can be viewed by students as often as the student needs to see them, at any time of any day during the week.

A Freeware Solution to Automated Tutorials

For most institutions there are considerable issues with the availability of funding for software, particularly for faculty computers after the campus' computer labs and other technology needs have been met. Because of this it may not be possible for instructors to have commercial screen capture products provided. As a solution to this issue there is an excellent freeware program available, Debugmode's Wink.

Wink can be freely downloaded from www.debugmode.com and is available for Windows and Linux. As a freeware product instructors can acquire the program on their own and use it without needing to be funded through the institution. This also provides users of Wink with the option of installing the program onto multiple machines so they can have access in the office as well as on home computers or on laptops for applications away from their desks.

Wink is a full-featured screen capture program, offering support for multiple capture modes, audio integration, user-customizable editing features, and multiple export modes. The design considerations presented in this paper all apply to Wink, although they can also be applied to other screen capture products.

Automated Tutorial Design Decisions

There are many design decisions that need to be made when creating an automated tutorial. The type of capture to perform, area of the screen to capture when recording, the timing of the presentation when it plays, user navigation, the use of audio, and links to supplemental resources are all different design elements that must be considered when creating a tutorial.

The first decision to be made when starting a tutorial project is how to capture the images to be used in the tutorial. Wink offers three different options. The first is to capture single images with the press of a key. The second is to do what is referred to as an input-driven capture, where the press of a key or the click of the mouse will trigger a capture. The third and final option is to do a timed capture, where Wink records everything that happens on the screen much like a video camera. While it is often tempting to the beginner to use the timed capture option this is rarely necessary for creating software tutorials. The timed capture option records an incredibly large volume of frames, which not only makes the editing process difficult but also generates very large file sizes. For most software tutorials the input-driven capture feature will capture all of the important actions and in the export of the tutorial the mouse movements will be animated to move from point to point between captured frames. Other capture methods

can be very useful in the right applications, it is just a question of making the right choice for the right tutorial.

How much of the screen to capture for the tutorial is one of the earliest design decisions to be made because it has to be made before the capture begins. While it is easy to default to capturing the entire screen, and sometimes the entire screen is what needs to be captured, it is not necessarily the best way to create a tutorial. When making design decisions one of the key questions to ask is whether or not something contributes to the instructional purpose of the tutorial. Does showing the menu bar give users of the tutorial information they need to know? If part of the instruction includes using the Start menu, pointing out things in the system tray, or other menu bar actions then it should be included, but if not, it may be best to exclude that area of the screen from the capture. This same decision process applies to capturing applications. Does the learner need to see the entire application screen or do they only need to see a part of it? For most software tutorials it will probably be useful to show the entire interface, including both the work area and the toolbars, but in some cases this may not be necessary. By using a tablet PC or digitizer screen captures can be used to create sample problems and explanations; in this case the learner probably only needs to see the work area, not the toolbars or other items being used while the tutorial is created. Regardless of what is or is not included the intent is always the same, to provide learners with the best possible instructional experience.

After capturing the screen images that will be used in the tutorial decisions must be made about how the tutorial plays for the learner. While a frame rate setting will

determine the speed at which the frames are played there are also options to stay on a specific frame for a user-defined period of time. This may be used to emphasize a specific point or to give learners time to read over information that is displayed on that frame. Setting the appropriate amount of time for a frame to stay on the screen takes a great deal of experience and, regardless of experience, a certain amount of guesswork. How long is long enough for every user to be able to see what they need to see but not so long that users' attention fails? Determining these times is challenging and before attempting to time a presentation it is best to consider whether the tutorial needs to run on its own with set times or be user-navigated. Sometimes a simple demonstration can be best illustrated by a movie that plays from start to finish and provides an example of what needs to be done. In cases like this timing a presentation can be a good option. As the tutorials get longer and more complex, however, it becomes more difficult and less productive to try to time a presentation to play on its own. As more information is presented it is necessary to consider the added time that learners need to process what they are seeing on the screen and the time that it may take them to read the information presented in notes with the tutorial. As a general rule the more complex the tutorial is the less suitable it is to run as an automatically timed, versus user-navigated, presentation. The caveat to this, however, is that each situation and each audience is different so the ultimate decision is always with the designer to make the choice that best suits the environment in which the tutorial is being deployed.

As has already been mentioned the alternative to timing a presentation is to include user navigation. This navigation can include options for learners to go directly to specific points in the presentation or to go forward or backward one frame at a time. As

with other design options the decision of what kind of user navigation to use, if any, is dependant on the content and purpose of the tutorial. If the tutorial is a large one, or if it covers multiple topics, it can be very useful to provide links from the start to different locations in the tutorial; this allows learners to navigate directly to the topic they need to review. These same direct navigation tools can also be integrated into a tutorial, providing learners with navigation options at the end of each section. Within the tutorial it can be useful to add forward and back navigation buttons. By adding a forward navigation button, simply a user-initiated command to move to the next frame, it is possible to add informative content to a page and know that the learner will have all the time they need to read it, but that once they have finished reading they can continue the tutorial. If there are certain parts of a tutorial that may need to be repeated a previous button can also be added, which gives the learner the option of going back and repeating what they have just seen before continuing on to the rest of the presentation. With the inclusion of user-controlled navigation a tutorial can be much more tailored to the individual viewing it and provides them with the opportunity to move at their own pace through the information. Exactly what that information consists of will be the next set of decisions.

Tutorials that are created by capturing on-screen images are by default visual presentations, and these can be enhanced by adding text boxes, arrows, and other visual information for learners. Sometimes this visual information can be enhanced by the inclusion of audio. One of the first considerations when making the decision of whether or not to include audio is the environment in which your learners will be working. Can the learners access audio from their most common work locations? Many computer lab

areas do not have speakers available because of the disruption that would be caused to others in a public area and headphones may or may not be made available. This is also true of many office environments, particularly open, cubicle-styled areas where audio can be disruptive. If the learner can not access audio then it obviously will not be an effective addition to the presentation. Also, the use of audio is tied directly to the decisions about how the tutorial will be timed or navigated. If user navigation is used then the audio will need to be aligned with those navigation options so that the audio the learner hears is synchronized with what they are seeing on the screen and they get complete sentences and complete sets of information from every point to which they navigate. If audio is an appropriate addition to the tutorial then the next question is how it will be captured. The choice will be between capturing audio as the actions are performed and the screen captures are completed or capturing the screen images, editing the visual component of the presentation, and then recording audio after the fact. While any work with audio can prove challenging, it is important to recognize that narrating operations while simultaneously performing them can significantly complicate things for the person creating the tutorial. This certainly is not to say that narration during screen capturing is not a powerful tool, simply something to be noted in the decision-making process.

The final major decision in tutorial creating is whether or not to link from the tutorial to supplemental information outside of the tutorial. It is possible to link to websites outside of the tutorial; the question is whether or not linking to additional information will enhance the learner's experience. Many times it can be useful to include a link at the end of the tutorial that points back to a course homepage or learning management system (LMS) so that the learner can return to the online location where

from which other information is accessed. Within the presentation it may be helpful to point to related tutorials or other resources for learners to explore more about a given subject to enhance their understanding of the tutorial. If this is done it is important to consider what moving to the link will do to the tutorial; ideally links will launch in a new browser window so that the learner can return to the same place in the tutorial after viewing the external material.

Tutorial Deployment

After a tutorial has been created it needs to be deployed for use by learners. Many times these tutorials are posted as embedded Flash files online. The HTML export option with Wink exports three files, one that is the HTML page that the user sees when they go to the assigned web address, one that is the actual Shockwave Flash (.swf) formatted media file, and a third that serves to provide functionality to the interface. All three of these files should be uploaded to the directory for the tutorial and learners can be directed to the .html address.

As an extension to the web page deployment method, since the .swf file is uploaded as part of the online package it is also possible to link learners to the .swf file for download to play on their computers as a file rather than as an embedded media object in a web page. The caution to this is that often the Windows operating system will not immediately recognize a .swf file and the learner will have to designate a player for it, most often a web browser. While this is a relatively simple operation it may be beyond the ready skills of normal computer users.

To solve the problem of player compatibility with distributing an .swf file for local use by learners there is another export option. Wink will export a self-contained executable (.exe) file that can be made available for learners to download or it can be distributed via CD or other method for play on local machines. As a self-contained program the .exe format does not require the learner to designate a Flash player on their machine, they simply need to run the program and things will play on their own.

Wink also offers two export options for static displays if deployment to a Flash player is not the right solution. By exporting to HTML or PDF Wink will automatically order the screen capture images, including any editing done to them, from top to bottom of a web page or PDF document. This can be helpful for deployment in areas with limited bandwidth or for printable tutorials to be kept as a hard copy.

Conclusion

Automated tutorials can be a valuable tool for teaching technology and with the proper software they can be created without an investment in commercial applications. Coupled with proper web access these tutorials can be deployed to support learners in a variety of environments at any time.