Seven Secrets of Successful Software Demos

Matthew Ellison

16th – 17th June, 2011
Brighton, UK
What we’ll cover in this session

• Screen-based capture compared to full-motion recording
• Demos compared to simulations
• Strengths and weaknesses of the available tools
• #1 Size matters
• #2 Allow learners to control the pace
• #3 Apply the modality principle
• #4 Use zooming and panning for demonstrations
• #5 Focus learners’ attention on the right area
• #6 Avoid spoon-feeding learners in simulations
• #7 Aim to match behaviour of application exactly
Two ways of capturing/recording software tasks

- Full-motion recording
- Screen-based capture

Most tools specialise in one of these two types of capturing/recording
Simplified workflow for full-motion recording

1. **Record Full Motion Video**
2. **Edit within tool (limited capability)**
   - Trim unwanted sections, add callouts, add transitions, add Zoom-n-Pan effects, etc.
3. **Publish to Flash or video format**
   - Formats typically include FLV, MP4, FLV, SWF, EXE
The advantages of capturing full motion video

• All screen activity is faithfully recorded

• Custom mouse movement is possible

• Demonstrations may appear more natural
Simplified workflow for screen-based capture

Capture Static Screenshots

Manual or automatic recording

Edit within tool

Add transitions, captions, highlights, interactions, etc.

Publish to Flash or video format

Demonstration | Simulation
Advantages of capturing static screenshots

• Don’t have to worry about timing or mouse movement during recording

• Relatively easy to edit the recording:
  • Add new or missing slides
  • Remove unwanted slides
  • Edit slides
  • Control the pace of the playback
Advantages of capturing static screenshots

• Possible to publish as interactive simulation:
  • Successful user action causes transition to next slide

• Mouse not captured as part of static screenshot
  • Mouse movement can be removed for simulation
  • Tool adds “perfect” movement based on the mouse position on consecutive slides
  • Re-position the mouse during editing
  • Speed up / slow down the mouse movement
  • Easily add audible and visual click effects
### Demonstrations compared to Simulations

<table>
<thead>
<tr>
<th>Demonstration</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learner <strong>watches</strong></td>
<td>Learner <strong>participates</strong></td>
</tr>
<tr>
<td>Passive</td>
<td>Active</td>
</tr>
<tr>
<td>Mouse pointer moves and clicks automatically</td>
<td>Learner moves and clicks own mouse pointer</td>
</tr>
<tr>
<td>Typing happens automatically</td>
<td>Learner types any required input</td>
</tr>
<tr>
<td>Pace may be dictated by demo</td>
<td>Pace dictated by learner</td>
</tr>
</tbody>
</table>
• Full-motion recording

<table>
<thead>
<tr>
<th>Tool</th>
<th>Key Strength</th>
<th>Potential Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camtasia Studio</td>
<td>• Zoom-n-Pan</td>
<td>• Can’t import .swf files from other tools</td>
</tr>
</tbody>
</table>
Tools

- Screen-based capture (SCORM compliant)

<table>
<thead>
<tr>
<th>Tool</th>
<th>Key Strength</th>
<th>Potential Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Captivate</td>
<td>• All-round feature set</td>
<td>• Zoom and pan</td>
</tr>
<tr>
<td>Atlantic Link Capture Point</td>
<td>• Automatic playback modes</td>
<td>• Zoom and pan</td>
</tr>
<tr>
<td>Qarbon ViewletBuilder</td>
<td>• Win, Mac, and Linux versions</td>
<td>• Uptake</td>
</tr>
</tbody>
</table>
**Tools**

- Screen-based capture (no SCORM support)

<table>
<thead>
<tr>
<th>Tool</th>
<th>Key Strength</th>
<th>Potential Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComponentOne</td>
<td>• Range of output formats</td>
<td>• Zoom and pan</td>
</tr>
<tr>
<td>DemoWorks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MadCap Mimic</td>
<td>• Similar UI to MadCap Flare</td>
<td>• Similar UI to MadCap Flare!</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other low-price alternatives are available:
- Tanida Demo Builder
- FlashDemo
#1 Size matters

- Flash files normally play back at fixed size
- Recording size becomes playback size

- Aim for the lowest screen resolution that your users may have
Creating resizable Flash demos

• Use 100% for width and height settings in calling code

• For example:

    SWFObject("1CustomerJourney.swf","Captivate","100%", "100%", "10", "#CCCCCC");

Note: My experience is that this does not work in Firefox unless you omit the DOCTYPE declaration on the calling HTML page
#2 Allow the learner control the pace

- Not an issue for simulations

- Demonstrations: insert regular “pause points”
#3 Apply the Modality Principle

• Learners benefit from multimodal instruction (visual + spoken)

• Demo + audio

is more effective than

Demo + text

Mayer (2005c)
The Redundancy Principle

• **Demo + audio**

  is more effective than

  **Demo + audio + text**

• If you have to use text and audio, make sure they match
#4 Use zooming and panning for demonstrations

- Use to give context **and** detail for large-screen apps
- Not possible for simulations
- Should not require full-motion video recording
Zoom-n-Pan – Camtasia 7.0’s killer feature

### Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Start time</th>
<th>Clip</th>
<th>Zoom speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoom 1</td>
<td>0:00:03:05</td>
<td>makecall_1_Screen.avi</td>
<td>1 second</td>
</tr>
<tr>
<td>Zoom 2</td>
<td>0:00:05:20</td>
<td>makecall_1_Screen.avi</td>
<td>1 second</td>
</tr>
<tr>
<td>Zoom 3</td>
<td>0:00:07:13</td>
<td>makecall_1_Screen.avi</td>
<td>1 second</td>
</tr>
<tr>
<td>Zoom 4</td>
<td>0:00:09:10</td>
<td>makecall_1_Screen.avi</td>
<td>1 second</td>
</tr>
</tbody>
</table>

### Diagram

- **Zoom area**
- **Transition time**

- Scale: 154%
- Duration: 1.00 sec
- Zoom out
- Zoom in
- Faster
- Slower
Zoom-n-Pan keyframes on the timeline

Keyframe can be dragged, modified, or removed
Key features of Zoom-n-Pan keyframes

• Keyframes added post-recording

• Keyframes only affect the final published video, and don’t touch the source video
How about Captivate’s Zoom and Pan capabilities?

- Zoom Areas can be added post-recording
  - Magnify selected area of the slide
How about Captivate’s Zoom and Pan capabilities?

- Panning can be used during the recording
  - Transitions recorded as full motion video
Why you might need both tools

• Use Captivate for:
  • Easily updatable demonstrations and simulations where a full screen view is acceptable throughout

• Use Camtasia for:
  • Rich demonstrations where panning and zooming is required
Case Study – background

• User assistance for contact management and communication software

• Used by agents in high pressure Contact Centre environment

• Require simple step-by-step instructions with optional demonstrations

• User assistance displayed in vertical pane within application UI
Example of user assistance topic

Set your Online Status

By setting your *Online Status* you can choose whether or not you will send and receive communications.

You can do either of the following:

- Set your overall online status, which affects all communication channels
- Set your online status for a specific channel, such as Telephone

**To set your overall online status**

- Click one of the three status indicators 🟢, 🟡, or 🟠

**To set your online status for a specific channel**

- Use the drop-down list beneath the channel name

Starts demonstration
Issues determining recording strategy

- No demonstration recording tool was installed on application computers
- Rapidly evolving application UI
- Very little screen estate available for display of demonstration

As a result, I needed:

- Static screenshot capture
- Post-recording zoom and pan
My workflow for creating demonstrations

1. Capture screenshots using Alt + PrintScreen
2. Create Image SlideShow from screenshots using Captivate
3. Fine-tune timing and mouse movement within Captivate
4. Publish to Flash
5. Play within Adobe Flash Player, and record using Camtasia Recorder
6. Add Zoom-n-Pan keyframes and Callouts within Camtasia Studio
7. Publish to Flash
#5 Focus the learners’ attention on the right area

- Don’t use conventional “static” highlight boxes
- Prefer instead:
  - Spotlighting (greying out everything else)
  - Animations
- For simulations, consider:
  - Time-delayed pointers
  - On-request highlights
#6 Avoid spoon-feeding learners in simulations

Instead of this:

- **Click File**

Consider this:

- **Select the required menu**

Use in conjunction with an on-demand “Hint” facility
#7 Aim to match behaviour of application exactly

- Include mouse-over effects
- Include drag-and-drop actions
- Show pauses, transitions, and intermediate “Loading” screens

May require full-motion recording

- Less easy to achieve for simulations
#8 Bonus tip! Avoid re-inventing the wheel

- Use templates
  - Introductory and summary slides
  - Project settings
  - Publishing settings
  - Useful multimedia items and resources
- Or base new projects on existing projects
- When creating demonstrations and simulations, use a single source
  - Multimode recording
  - Show Me / Try Me / Test Me
References and further reading


• *Trainers guide to authoring tools: users tell you what works, what doesn’t, and why*, 2010, TMR Publications, http://tinyurl.com/6g7s3zx

• *Want to Create Engaging Screencasts?* TechSmith, http://tinyurl.com/2ertj6v

• Matthew’s blog on Adobe Captivate, http://blogs.highlander.co.uk/tag/captivate/
Questions?

Matthew Ellison
UA Europe
matthew@uaeurope.com

16th – 17th June, 2011
Brighton, UK