Using Adobe Captivate and Flash Simulations in e-Learning

Doug Peterson, Questionmark
Meet the Presenter

Doug Peterson

- 1989 - 2011 with GTE/Verizon, the last 12 in Workforce Development (training).
- Researched, developed, and delivered technical training, mostly in Internet technologies, specializing in Captivate and Flash.
- Project lead for developing online pass/fail testing for call center agent training.
- Currently Product Owner & Demonstration Systems Manager at Questionmark.
Objectives

By the end of this session, you will be able to:

- Define “simulation”
- Explain the advantages of using software simulations in training and assessments
- Decide when to use a Captivate simulation and when to use a Flash/ActionScript simulation
- Publish a Captivate simulation for Perception and create a question
- Publish a Flash simulation, write the necessary QML, and create a Perception question
Simulations: Definition and Advantages of Use
Simulation: A Definition

- Application simulation: an interactive object (Flash SWF) that looks and behaves like the real application.
- Can also emulate hands-on activities when teaching over the Web.
- Types of simulations:
  - Demonstration – show me
  - Training – let me drive, but give me hints
  - Assessment – I have to prove I know what I’m doing
Advantages of Using Simulations

- Interactive (mousing, typing, clicking)
  - Holds interest
  - Improves recall
- Authentic experience
- Application experience/training without:
  - Learner having app installed
  - Licensing expense/tracking
  - Access credentials (IDs/passwords)
  - Connection to actual system
- Control active/non-active elements
- Allow “hands-on” topics to be taught remotely
Example: Call Center Testing

- Call center agents use a variety of applications to access customer data, place/update/check on orders, etc.
- Captivate and Flash allow us to present the learner with an application simulation to verify their understanding of how to use the application.
- Also allows the use of audio – check for working headphones, test knowledge of greetings, etc.
Scenario 1 - Opening [Question 1 of 6]

Assume that the customer’s information has populated on your screen via his IVRU response. Place your mouse over each of the four icons and listen to the entire greeting for each one. Select the best greeting from the list below.

- Greeting A
- Greeting B
- Greeting C
- Greeting D

Submit
Example: Set-Top Box Wiring

- Call center agents must know how to cable together various components (e.g. set-top box, TV, and DVD player) in order to be able to answer customer questions and provide help.
- DEMO: STB Wiring Simulations
Creating Simulations Using Captivate
Poll #1

What level of experience do you have with Captivate?

- None
- Some
- A lot
Captivate: Overview

- Allows you to record activity taking place on your computer monitor.
- Create training content from scratch – text, graphics/images, video, audio, etc.
- DEMO: Create demonstration, training, and assessment simulations for searching for a Knowledge Base article that includes Flash, QML, and Choices.
Captivate: Quiz Settings

- In addition to learning, use Captivate for assessments
- Click boxes/text entry boxes included in quiz
  - equivalent of a question
  - by clicking on a click box, the learner is “answering the question correctly.”
- Quiz elements/settings can be customized.
  - Interaction ID (used in choice/outcome labels)
  - Pass/Fail threshold
- DEMO: Customizing quiz elements and setting quiz settings.
Captivate: A Problem

- Very linear
  - Branching possible, can get complex quickly
- By default, only valid interaction is where learner is supposed to click/type – bit of a give-away.
- Multiple interactive objects on one slide can be trouble.
- Hard to create complex navigation/relationships between slides.
Creating Simulations Using Flash
Poll #2

What level of experience do you have with Flash (not counting ActionScript)?

- None
- Some
- A lot
Poll #3

- What level of experience do you have with ActionScript?
  - None
  - Some
  - A lot
Flash: The Solution

- Use ActionScript to create complex behaviors
  - Selecting a radio button populates a field with a default value
  - Selecting a value in one drop-down menu populates the available values in another drop-down menu

- DEMO: NetSuite issue
Publishing Simulations
Publishing for the Web

- Basically, create a SWF file and some HTML to invoke the Flash player to play the SWF, then move HTML and SWF to a server
- Demonstrations and quizzes – low stakes, no security needed
Publishing for SCORM - Captivate

- Enable reporting and set it for SCORM
- Fill out the Manifest
- Be wary of sending Resume Data
- Default template problem
Publishing for SCORM - Flash

- SCORM is accomplished via JavaScript calls in the HTML, there are SCORM HTML templates (1.2 & 2004) available in Flash Publish Settings
- Use FSCommand to communicate between your Flash code and the JavaScript in the HTML
- WWW.ADLNET.GOV for SCORM info
- http://pipwerks.com/2008/02/10/flash-demos-for-scorm-actionscript-classes-now-available/
Publishing for Perception - Captivate

- Enable reporting and set it for Questionmark Perception.
- The publishing process produces a Flash SWF file and a QML file. Authoring Manager uses the QML file to build the question in Perception.
- DEMO: Publishing the simulation and reviewing the QML.
Captivate: Creating the Question in Perception

- DEMO: Adding a Captivate question in Perception Authoring Manager
While there is an extension that can be installed in Flash to publish a QML file, it only allows for a single pass/fail message to be sent to Perception. No individual outcomes/partial credit.

Possible to write QML and ActionScript that allows a Flash simulation to report individual outcomes.

The Key: the question must be added as a Captivate question, not a Flash question. The Flash question type will not accept QML that has more than just pass/fail information.
Flash: ActionScript

```actionscript
var grey:Boolean = false;
var orange:Boolean = false;
var red:Boolean = false;
grey_btn.addEventListener(MouseEvent.CLICK, button_clicked);
orange_btn.addEventListener(MouseEvent.CLICK, button_clicked);
red_btn.addEventListener(MouseEvent.CLICK, button_clicked);
function button_clicked(evt:MouseEvent):void {
    switch(evt.currentTarget.name) {
        case "grey_btn":
            grey=false;
            grey_txt.text="false";
            break;
        case "orange_btn":
            orange=false;
            orange_txt.text="false";
            break;
        case "red_btn":
            red=false;
            red_txt.text="false";
            break;
    }
    send_score();
}
```

2013 European Users Conference  ♦  Barcelona
function send_score() {
    if (grey) {
        fscommand("MM_cmiSendInteractionInfo", "00;01;grey;mytest;04;05;06;grey correct;08;09");
    } else {
        fscommand("MM_cmiSendInteractionInfo", "00;01;grey;mytest;04;05;06;grey wrong;08;09");
    }
    if (orange) {
        fscommand("MM_cmiSendInteractionInfo", "00;01;orange;mytest;04;05;06;orange correct;08;09");
    } else {
        fscommand("MM_cmiSendInteractionInfo", "00;01;orange;mytest;04;05;06;orange wrong;08;09");
    }
    if (red) {
        fscommand("MM_cmiSendInteractionInfo", "00;01;red;mytest;04;05;06;red correct;08;09");
    } else {
        fscommand("MM_cmiSendInteractionInfo", "00;01;red;mytest;04;05;06;red wrong;08;09");
    }
}
send_score(); //initializes scoring
## MM_cmiSendInteractionInfo Parameters

<table>
<thead>
<tr>
<th>Parameter position (zero-based)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Date</td>
</tr>
<tr>
<td>01</td>
<td>Time</td>
</tr>
<tr>
<td>02*</td>
<td>Option (Interaction ID)</td>
</tr>
<tr>
<td>03</td>
<td>Objective ID</td>
</tr>
<tr>
<td>04</td>
<td>Question Type</td>
</tr>
<tr>
<td>05</td>
<td>Correct Answer</td>
</tr>
<tr>
<td>06</td>
<td>Participant’s Answer (Comments)</td>
</tr>
<tr>
<td>07*</td>
<td>Outcome</td>
</tr>
<tr>
<td>08</td>
<td>Point Value</td>
</tr>
<tr>
<td>09</td>
<td>Elapsed Time</td>
</tr>
</tbody>
</table>

*Required.
Generic QML Example

```xml
<?xml version="1.0" standalone="no"?>
<!DOCTYPE QML SYSTEM "QML_V3.dtd">
<QML>

<QQUESTION ID="6205643879861730" DESCRIPTION="T/F question" TOPIC="Adobe Flash" STATUS="Normal">
  <CONTENT TYPE="text/html"><![CDATA[This is the stimulus.]]></CONTENT>
  <ANSWER QTYPE="TF" SUBTYPE="VERT">
    <CHOICE ID="0">
      <CONTENT TYPE="text/html"><![CDATA[True]]></CONTENT>
    </CHOICE>
    <CHOICE ID="1">
      <CONTENT TYPE="text/html"><![CDATA[False]]></CONTENT>
    </CHOICE>
  </ANSWER>
  <OUTCOME ID="0 True" SCORE="1">
    <CONDITION>"0"</CONDITION>
    <CONTENT TYPE="text/html"><![CDATA[That's correct!]]></CONTENT>
  </OUTCOME>
  <OUTCOME ID="1 False" SCORE="0">
    <CONDITION>"1"</CONDITION>
    <CONTENT TYPE="text/html"><![CDATA[The correct answer is "True".]]></CONTENT>
  </OUTCOME>
</QQUESTION>
</QML>
```
Flash: QML (part 1)

<?xml version="1.0" standalone="no"?>
<!DOCTYPE QML SYSTEM "QML_V3.dtd">
<QML>
<QUESTION DESCRIPTION="fg1 qml1" TYPE="text/plain">
  <ANSWER QTYPE="FLASH" COMMENT="Y" SHUFFLE="N">
    <CHOICE ID="grey_button">
      <OPTION>grey</OPTION>
      <CONTENT WIDTH="175" HEIGHT="150" TYPE="NULL">greybutton</CONTENT>
    </CHOICE>
    <CHOICE ID="orange_button">
      <OPTION>orange</OPTION>
      <CONTENT TYPE="NULL">orangebutton</CONTENT>
    </CHOICE>
    <CHOICE ID="red_button">
      <OPTION>red</OPTION>
      <CONTENT TYPE="NULL">redbutton</CONTENT>
    </CHOICE>
  </ANSWER>
</QUESTION>
</QML>
<OUTCOME ID="grey true" ADD="1" CONTINUE="TRUE">
  <CONDITION >"grey_button" MATCHES NOCASE "grey correct"</CONDITION>
  <CONTENT TYPE="text/plain">Grey is true.</CONTENT>
</OUTCOME>

<OUTCOME ID="grey false" ADD="0" CONTINUE="TRUE">
  <CONDITION >"grey_button" MATCHES NOCASE "grey wrong"</CONDITION>
  <CONTENT TYPE="text/plain">Grey is false.</CONTENT>
</OUTCOME>

<OUTCOME ID="orange true" ADD="1" CONTINUE="TRUE">
  <CONDITION >"orange_button" MATCHES NOCASE "orange correct"</CONDITION>
  <CONTENT TYPE="text/plain">Orange is true.</CONTENT>
</OUTCOME>

<OUTCOME ID="orange false" ADD="0" CONTINUE="TRUE">
  <CONDITION >"orange_button" MATCHES NOCASE "orange wrong"</CONDITION>
  <CONTENT TYPE="text/plain">Orange is false.</CONTENT>
</OUTCOME>

<OUTCOME ID="red true" ADD="1" CONTINUE="TRUE">
  <CONDITION >"red_button" MATCHES NOCASE "red correct"</CONDITION>
  <CONTENT TYPE="text/plain">Red is true.</CONTENT>
</OUTCOME>

<OUTCOME ID="red false" ADD="0" CONTINUE="TRUE">
  <CONDITION >"red_button" MATCHES NOCASE "red wrong"</CONDITION>
  <CONTENT TYPE="text/plain">Red is false.</CONTENT>
</OUTCOME>

</QUESTION>
<CHOICE ID="grey_button">
  <OPTION>grey</OPTION>
</CHOICE>

<CONTENT TYPE="text/plain">Red is false.</CONTENT>

<OUTCOME ID="grey true" ADD="1" CONTINUE="TRUE">
  <CONDITION >"grey_button" MATCHES NOCASE "grey correct"</CONDITION>
</OUTCOME>
JavaScript Considerations

- ActionScript “fscommand” invokes function in qmDelivery.js on Perception server.
- qmDelivery.js requires an override.
- Create new template (.xtmp) for use with Flash questions.
- Add a HOOK_SCRIPT at the bottom to override JavaScript in qmDelivery.js.
- Delivery is working on a permanent fix.
Providing Feedback & Assessing Performance
Providing Feedback & Assessing Performance

- Feedback can be built directly into the simulation, e.g., the Success/Failure/Hint text boxes in Captivate
- Deploying via Questionmark Perception provides other options
  - Automatic feedback to the learner at the end of the question or block of questions
  - Reports
Perception: Immediate Feedback

```javascript
function send_score() {
    if (copay_correct) {
        fscommand("MM_cmiSendInteractionInfo", "00;01;copay;mytest;04;05;06;C;08;09");
    }
}
```

```
<OUTCOME ID="C Copay" ADD="2" CONTINUE="TRUE">
  <CONDITION >"copay" MATCHES NOCASE "C"</CONDITION>
  <CONTENT TYPE="text/plain">Copay is correct.</CONTENT>
</OUTCOME>
```
function send_score() {
    if (cat_correct) { fscmd("MM_cmiSendInteractionInfo", "00;01;category;mytest;04;05;06;Report Category Correct;08;09"); } else { fscmd("MM_cmiSendInteractionInfo", "00;01;category;mytest;04;05;06;Report Category Wrong;08;09"); }
}

<OUTCOME ID="C Report Category" ADD="2" CONTINUE="TRUE">
    <CONDITION >"category" MATCHES NOCASE "Report Category Correct"</CONDITION>
    <CONTENT TYPE="text/plain">Report Category is correct</CONTENT>
</OUTCOME>

<OUTCOME ID="W Report Category" ADD="0" CONTINUE="TRUE">
    <CONDITION >"category" MATCHES NOCASE "Report Category Wrong"</CONDITION>
    <CONTENT TYPE="text/plain">Report Category is wrong</CONTENT>
</OUTCOME>

Assume that you have brought up the customer's account in iCAD. What Report Category and Trouble Type would you use to initially classify this call, given what you know so far? Use your mouse to navigate the iCAD simulation. Open the Report Category selection box and select the correct code. Then do the same for the Trouble Type field. When you are satisfied with your selections, click the Submit button.

Outcome:

<table>
<thead>
<tr>
<th>Category</th>
<th>% #</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Report Category</td>
<td>50.00%</td>
</tr>
<tr>
<td>W Report Category</td>
<td>0.00%</td>
</tr>
<tr>
<td>C Trouble Type</td>
<td>16.67%</td>
</tr>
<tr>
<td>W Trouble Type</td>
<td>33.33%</td>
</tr>
<tr>
<td>Not answered</td>
<td>0.00%</td>
</tr>
<tr>
<td>Total</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

Times presented | 3 
Max weight | 4 
Times answered | 3 
Mean weight | 2.67 
Min weight | 0 
Median weight | 2
Perception: Individual Learner Report

```javascript
function sendScore() {
    if (one.value == "$20") {
        totalPoints += 1;
        fscommand("MM_cmiSendInteractionInfo", "00:01;copay;week4test1_1g;04:05;$20;C;08:09");
    } else {
        studentResponse1 = "Co-pay = " + one.value;
        fscommand("MM_cmiSendInteractionInfo", "00:01;copay;week4test1_1g;04:05;" + studentResponse1 + ";W:08:09");
    }
}
```

---

<ANSWER QTYPE="FLASH" COMMENT="Y" SHUFFLE="N">

Coaching report

<table>
<thead>
<tr>
<th>Question wording</th>
<th>Flash Testing</th>
<th>Adobe Flash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic</td>
<td>2 - Not Correct</td>
<td>4 - Copay Incorrect</td>
</tr>
<tr>
<td>Question type</td>
<td>6 - LT Max Incorrect</td>
<td>8 - LTM Met Incorrect</td>
</tr>
<tr>
<td>Outcome(s) chosen</td>
<td>10 - Other Info Incorrect</td>
<td></td>
</tr>
</tbody>
</table>

| Actual score | 0 |
| Maximum score | 1 |
| Feedback shown | You made some mistakes. Check your Coaching Report for feedback. |

Participant comment
MM_cmiSendInteractionInfo 00:01;copay;week4test1_1g;04:05;Co-pay = $15;W:08:09
MM_cmiSendInteractionInfo 00:01;ltmax;week4test1_1g;04:05;Lifetime Max = $20,000;W:08:09
MM_cmiSendInteractionInfo 00:01;ltmet;week4test1_1g;04:05;Lifetime Max = 400;W:08:09
MM_cmiSendInteractionInfo 00:01;oi;week4test1_1g;04:05;Other Information = $100 reimbursement on hardware;W:08:09
MM_cmiSendInteractionInfo 00:01;week4test1_1g;week4test1_1g;04:05;06;W:08:09
In Conclusion...
What Could Possibly Go Wrong?

- SWF generated by Captivate and/or Flash plays on the learner’s machine. Ensure that the learner’s machine has the correct version of the Flash player, as well as adequate resources (RAM, speed, etc.)

- If using simulations in an assessment, since simulations might be new to the learner, consider providing a practice assessment to give the learner a sense of familiarity with the testing method before it really counts.
Summary

- Captivate is excellent for simple, one or two step simulations.
- For more complex simulations, use Flash and ActionScript.
- For individual item scoring in Perception, Flash must be deployed as if it were a Captivate question.
Thanks for Attending!
The Grandmaster Flash Approach

- If you have several activities all based on the same application, maintenance can be a nightmare if the application changes.
- Consider using a centrally located “core” Flash video that is called by individual activity Flash videos.
- Use variables in the individual activity to control the behavior of the core video.
GMF: In the Calling Activity

// When loaded to QM, allows grandMasterFlash.swf to access variables in this SWF
Security.allowDomain("fsctrainingvideos.verizon.com");

// Define the basic question type
var questionType:String="closeOut"; // closeOut, createTicket
var serviceType:String="data"; // voice, data, or video

// Define which controls to make active.
var useQuickTicket:Boolean=false; // voice, video, data
var useReportCategory:Boolean=true; // voice, video, data
var useTroubleType:Boolean=true; // voice, video, data
var useTroubleInformation:Boolean=true; // voice, video, data

// Text to display if field is clicked on.
var troubleInformationText:String="Cx can't get email. Email acct OK - VzUID=res12345, no portal";

// Set to true if user is to click field to populate.
var makeTroubleInformationClickable:Boolean=true;

// Set to true if user is to type in the field to populate.
var makeTroubleInformationUpdateable:Boolean=false;

var useSuspOOS:Boolean=true; // voice, video, data
var useAllCalls:Boolean=false; // voice
var useAllPhones:Boolean=false; // voice
var useIsolation:Boolean=false; // voice
var useConnection:Boolean=true; // data
// Load grandMasterFlash.swf
var loader:Loader=new Loader(); // used to load external SWF.
var gmf_mc:MovieClip; // used to store the loaded SWF.
loader.load(new URLRequest("http://fsctrainingvideos.verizon.com/data/FSC_Test_Automation/grandMasterFlash.swf"));

loader.contentLoaderInfo.addEventListener(Event.COMPLETE,swf_loaded);
function swf_loaded(e:Event):void {
    gmf_mc=loader.content as MovieClip; // load to _mc so play() can be invoked
    this.addChild(gmf_mc);
    gmf_mc.play();
}

function executeQuestionSpecificCode():void{
    // Place any question-specific code here. This function is called after all objects are in place.
    gmf_mc.reportCategory_mc.reportCategory_txt.text="CR - Cust Call about Exist Svc";
    gmf_mc.reportCategory_mc.reportCategory_btn.mouseEnabled=false;
    gmf_mc.reportCategorySet=true;
    gmf_mc.troubleType_mc.troubleType_txt.text="MEM - Memory Services/Features";
    gmf_mc.troubleType_mc.troubleType_btn.mouseEnabled=false;
    gmf_mc.commitmentDTtable_mc.visible=true;
    gmf_mc.troubleType_mc.checklistNDT_mc.visible=true;
}
GMF: In the Core Flash

// set the appropriate background
var backGround_mc:MovieClip;
switch (serviceType) {
    case "voice":
        backGround_mc=new createTicketVoice();
        break;
    case "data":
        backGround_mc=new createTicketData();
        break;
    case "video":
        backGround_mc=new createTicketVideo();
        break;
}
addChild(backGround_mc);

// Load appropriate controls.
if (MovieClip(parent).useRemarks) {
    var remarks_mc:remarks=new remarks();
    addChild(remarks_mc);
    remarks_mc.x=10.00;
    remarks_mc.y=139.30;
}