Overview

This project is based on some ideas from Dr. Ravi Palanivelu in the U of A School of Plant Sciences. In the initial phase, we are working with some of his slides that depict plant structure along with descriptive labels. Initially, we are making those static slides into interactions that students can use for study and review. A fairly simple process of drawing red rectangular boxes around the text labels along with a combination of JavaScript and HTML5 transforms the graphics into an interaction as may be seen from some of the examples below. We initially got the interactions working using ActionScript and PHP, but had to switch to ActionScript, JavaScript, and HTML to work within the D2L environment. This latest iteration of development has eliminated the now obsolete and unavailable ActionScript (Flash™), instead building on the HTML5 canvas along with JavaScript libraries as replacements.

We also developed a quiz capability using drag and drop, using the same sort of slides as input. With that functionality in place, we added **S**harable **C**ontent **O**bject Reference **M**odel (**SCORM**) capability to enable integration with D2L, including automatic gradebook updates.

Preparing Images for the Interactive Viewer

An easy option for preparing images is through the use of PowerPoint™, since it provides a rather easy-to-use set of tools for manipulating images, adding text boxes, drawing lines and boxes, etc. In fact, often some of the graphics that are useful for interactive viewing may have already been imported into PowerPoint™ for use in classroom presentations. Once the graphic is imported into a slide and text labels have been created, proceed as follows for each text box:

1. Click on a text box
2. In "Drawing Tools" click on the "Shape Outline" pull-down and select the "Standard Colors" red (second from left). Then, on the same pull-down, hover over "Weight" and select 1/4 pt. Finally, hover over "Dashes" and select the solid line at the top of the dialog.
3. Once this is closed, verify that you see a very thin red rectangular box surrounding the text.
4. With the same text box selected, click on the "Arrange" button and select "Bring to Front" to ensure that the text box is not partly concealed by lines or other graphics, since the interactive viewer must see the unbroken rectangle to recognize it and provide the interaction.
5. Repeat steps 1-4 for each text box.
6. View your picture. It should have red rectangles with thin 1-pixel width lines surrounding the text boxes.
7. Finally, right-click on the overall picture (i.e. not one of the text boxes) and select "Save as Picture" from the context dialog. Be sure that PNG is selected in the "Save as Type" dialog. Then, put in a name for the file. It would be best to avoid using spaces or other special characters in the file name. Acceptable special characters are underscores, dashes, and parentheses.

Afterward, you will need to email the graphic to me for validation and some additional processing steps that are somewhat more complicated. Alternatively, just email the un-prepared images to me as described in the following section and I will be happy to prepare them.

OIA SCORM Viewer

Getting your Images on the Site

Please send your images via email attachment to: cphill@arizona.edu. Please also indicate the type of interactions you would like. If the links have not been prepared, we will do that for you. Either way, we will check them, upload them to the site and will email you back the links for the types of interactions you wish to use. In this way, we can ensure that the interaction links will function properly with the prepared images. Insertion of these links in the D2L environment is described further below.

Link to the Interactive Viewer

The following discussion and examples assume that trunk.png, and root_cross_section.png have been prepared and uploaded to your content folder on this app's site (i.e. not in D2L):

/users/rpalanivelu

The 2 links shown below show how these specially prepared PNG graphics can become interactions. Currently, there are 2 display options for the non-quiz mode of interactions. The first allows only a single label to be uncovered at a time. Each time a label is clicked, it will be revealed and all other labels will be covered. The second option is to allow progressive clicking to reveal additional labels until all labels have been revealed.

Currently, the default rectangle box color that we recognize is pure red (0xFF0000), however, one of these slides uses an alternate color to illustrate that we can configure to recognize any box color.

The *clipname* parameter (see example below) is used to specify the graphic image to view this parameter conveys both the path to the graphic and its name.

Since, as mentioned above, these slides us a non-standard red color (0xED1C24) for illustration purposes, we must specify its color when calling up the viewer, by using the *boxcolor* parameter (see sample link below). Note that the boxcolor parameter should not be needed if the color is true pure red as is the case with PowerPoint (see instructions above).

The default interaction mode is to allow the user to progressively click to uncover multiple box labels. If one desires this mode, no parameter need be specified. However, to enable the mode where only a single label may be uncovered at a time, one may specify the *mode* parameter with the value of *single* (see example below). In single mode, one may also optionally specify *showButton=1* that will enable a button to toggle the visibility of all labels. This only works in *single* mode (see example below).

Sample link (brings up trunk.png using default *boxcolor* 0xFF0000 for interaction to reveal multiple boxes as the user progressively clicks on labels):

https://apps.oia.arizona.edu/SCORMView/dev/view.html?clipname=users/rpalanivelu/trunk.png&mode=multiple&width=750&height=550

Note that the above links are examples of what would be entered in a D2L Content "Quicklink" dialog that would then create a clickable link. In fact, I used D2L "Quicklinks" to provide the working example links above.

Now, everything is ready to add "Quicklinks" to content under the "Manage Content" section of "Content Areas" in D2L.

Using Quiz Mode

With a bit of care, one may also use such prepared images in a drag and drop quiz mode. To be most effective in this mode, some additional considerations should be observed when creating and preparing images:

1. For this mode it is best to avoid intersections of graphical portions of the image with the text boxes. Otherwise, the quiz will contain "jigsaw puzzle" like clues that may render the quiz results less meaningful.
2. For the same reason, it is also best to make the text boxes identical in size.

OIA SCORM Viewer

URL Query String Parameters in Quiz Mode

The viewer supports the following URL Query String parameters:

- **width:** The **_width_** parameter should specify the width of the image in pixels.
- **height:** The **_height_** parameter should specify the height of the image in pixels.
- **quiz:** When the parameter **_quiz=1_** is present, the viewer behaves in quiz mode to support dragging and dropping labels to their correct position. Correct placement is signified by a light green filter on the dropped label, while incorrect placement will display with a light red filter.
- **showButton:** The **_showButton=1_** parameter enables a "Show All" button that will uncover all labels. When this is done, the "Show All" button toggles to "Hide All".
- **boxwidth:** The **_boxwidth_** parameter may be used to specify the number of pixels of width to allocate to the right of the main image for the quiz control panel, along with the draggable text labels. If less than 80 pixels are specified, then the minimum value of 80 will be used to ensure a satisfactory appearance of the quiz control panel.
- **debug:** The debug=true parameter may be specified to display testing/development debugging messages from both JavaScript and ActionScript code.
- **timelimit:** The **_timelimit=xxx_** may be provided to specify the time limit for the quiz (where xxx is the number of seconds allowed for the student to complete the quiz after clicking on the start button).
- **quizname:** The **_quizname_** parameter may be used to specify the quiz name. This is necessary when using SCORM mode within a learning management system.
- **points:** The optional **_points=xxx_** parameter (where xxx=the number of points to be awarded for this quiz) only applies to quiz mode. It will normally also only be used in SCORM to scale the correct percentage to the number of points allocated to this quiz.
- **retries:** The optional **_retries=xxx_** (where xxx is the number of retries allowed) parameter may be used to specify the number of attempts that a student may make for this quiz. If not specified, the default is one attempt. This parameter only works when using SCORM mode within a learning management system.
- **score:** The parameter **_score=average_** may be used to specify use of the calculated average of attempts as the scoring method in lieu of defaulting to the score of the last attempt. This parameter only works when using SCORM mode.
- **scorm:** One may use the **_scorm=1_** to specify use of SCORM mode to communicate with SCORM compliant Learning Management Systems.
- **drill:** When the parameter **_drill=1_** is present the interaction allows clicking to reveal individual labels. When a new label is clicked, covers are restored for all other labels.

Sample Quiz Mode link (brings up root_cross_section.png using default **_boxcolor_** 0xFF0000 in Quiz Mode with a 60 second time limit):

https://apps.oia.arizona.edu/SCORMView/dev/view.html?clipname=users/rpalanivelu/root_cross_section.png&quiz=1&timelimit=60&boxwidth=150&width=515&height=550&debug=1&quizname=Quiz_5&attempts=3&score=average&points=100

About SCORM Mode

SCORM is an acronym for **S**tructured **C**ontent **O**bject **R**eusable **M**odule. Rather than being an actual standard, SCORM is a collection of standards that have been assembled into a useful framework for communicating with **L**earning **M**anagement **S**ystems (LMS). There are several versions of SCORM, with the latest being SCORM 2004, however, SCORM 1.2 is still in common usage.

The SCORM capabilities of our quiz mode in conjunction with the current version of D2L in use at the University of Arizona as of now are as follows:

- Specify and control the number of attempts that will be allowed for each SCORM quiz.
- Update the SCORM reports for individual students with either the latest or the average score of multiple attempts at taking a quiz. We use the SCORM 2004 API to communicate with D2L.

Here are some technicalities that explain how we implement some of our quiz option using SCORM:

- What D2L's version of SCORM considers an attempt differs somewhat from and is too amorphous to be useful for tracking quiz attempts. Instead, we use what is known in SCORM terminology as ***cmi.progress measure*** to track the attempts made on a quiz. This measurement is in the range [0..1]. Upon entry into the quiz, we retrieve the stored value for this SCORM variable. If the stored value is < 1, then additional attempts on the quiz are allowed. Upon exit, we save this value as the fractional ratio of the number of attempts taken divided by the total number of attempts allowed. By reading this value on entry, in conjunction with the number of allowed attempts, we derive the number of attempts that have already been made. Note, that we do not update this variable unless one clicks the "Start" button. Thus, a user may enter the quiz, simply to review their previous score.

OIA SCORM Viewer

Preparing a SCORM Module

A SCORM module is a set of files enclosed in a Zip compatible compressed container. Typically, the zip file will contain a single folder in its root directory, with the folder name having the same name as the quiz name. For our SCORM content, the folder hierarchy and contents will look like the following (assuming a quiz name of "Quiz_01"). Note that bold/underlined items below represent folders, while other entries represent files:

```
Quiz_01

        imsmanifest.xml
        view.html
        assets
                oia.png
                (other image resources)
        javascript
                consolelog.min.js
                jquery-3.5.1.min.js
                jquery-3.5.1.min.map
                jsutils.js
                easeljs.min.js
                preloadjs.min.js
                soundjs.min.js
                tweenjs.min.js
                SCORM_API_wrapper.js
```

OIA SCORM Viewer

Notice the special file imsmanifest.xml is an XML file for SCORM version 2004 that describes the SCORM content (see the following example for Quizzes 4 and 5 combined in a single package):

```xml
<?xml version="1.0"?>
<manifest identifier="OIAViewer"
        version="2004"
        xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
        xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2 imscp_rootv1p1p2.xsd
http://www.imsglobal.org/xsd/imsmd_rootv1p2p1 imsmd_rootv1p2p1.xsd  http://www.adlnet.org/xsd/adlcp_rootv1p2 adlcp_rootv1p2.xsd">

  <general>
   <title><langstring xml:lang="en-US">OIA Viewer SCORM Content</langstring></title>
   <language>en-US</language>
   <description>
    <langstring xml:lang="en-US">OIA Viewer SCORM Content</langstring>
   </description>
  </general>
  <metametadata>
   <metadatascheme>ADL SCORM 2004</metadatascheme>
   <language>en-US</language>
  </metametadata>

  <organizations default="oia.arizona.edu">
   <organization identifier="oia.arizona.edu">
   <title>OIA SCORM Quizzes</title>
    <item identifier="Quiz_04" identifierref="SCO_Quiz_04" isvisible="true">
     <title>Quiz_04</title>
    </item>
    <item identifier="Quiz_05" identifierref="SCO_Quiz_05" isvisible="true">
     <title>Quiz_05</title>
    </item>
   </organization>
  </organizations>

  <resources>
   <resource identifier="SCO_Quiz_04" type="webcontent" adlcp:scormType="sco"
href="view.html?clipname=users/rpalanivelu/trunk.png&amp;quiz=1&amp;timelimit=180&amp;boxwidth=143&amp;width=750&amp;height=550&amp;quizname=Quiz_04&amp;attempts=10&amp;score=average&amp;points=10&amp;scorm=1" target="_new">
     <file href="view.html"/>
     <file href="javascript/jquery-3.5.1.min.js"/>
     <file href="javascript/consolelog.min.js"/>
     <file href="javascript/jsutils.js"/>
     <file href="javascript/easeljs.min.js"/>
     <file href="javascript/tweenjs.min.js"/>
     <file href="javascript/soundjs.min.js"/>
     <file href="javascript/soundjs.min.js"/>
     <file href="javascript/preloadjs.min.js"/>
     <file href="javascript/SCORM_API_wrapper.js"/>
   </resource>
   <resource identifier="SCO_Quiz_05" type="webcontent" adlcp:scormType="sco"
href="view.html?clipname=users/rpalanivelu/root_cross_section.png&amp;quiz=1&amp;timelimit=180&amp;boxwidth=150&amp;width=515&amp;height=550&amp;quizname=Quiz_05&amp;attempts=10&amp;score=average&amp;points=10&amp;scorm=1" target="_new">
     <file href="view.html"/>
     <file href="javascript/jquery-3.5.1.min.js"/>
     <file href="javascript/consolelog.min.js"/>
     <file href="javascript/jsutils.js"/>
     <file href="javascript/easeljs.min.js"/>
     <file href="javascript/tweenjs.min.js"/>
     <file href="javascript/soundjs.min.js"/>
     <file href="javascript/soundjs.min.js"/>
     <file href="javascript/preloadjs.min.js"/>
     <file href="javascript/SCORM_API_wrapper.js"/>
   </resource>
  </resources>
</manifest>
```

OIA SCORM Viewer

It is worthwhile to note that the items highlighted in green are the quiz name and the item highlighted in yellow is the URL with query parameters. Note that the normal ampersand between query parameters has been escaped in HTML format to **_&_** in order to comply with the XML specification. It is worthy of note that the actual graphics that are being referenced are not listed, but are contained in the SCORM zip container.

Also, worthwhile to note is that in addition to graded content, one may also package interactions that are ungraded.

Troubleshooting

If you are having problems here are some suggestions to help:

Problem: When clicking on a link, I see the red rectangles and the image is not interactive.

Resolution: If you did not specify a boxcolor, you must not be using the default pure red (0xFF0000) and need to find the color and specify it using a boxcolor parameter.

Problem: When clicking on my link, I never see my graphic image.

Resolution: Check the path and filename for your image and be sure they are specified properly in the clipname parameter of your link.

Problem: When interacting with a graphic, I have to click multiple times before the text label is revealed.

Resolution: More than likely the line width of your red rectangles is not properly set. The line should be 1 pixel. In PowerPoint, this is .25 pt.

Problem: I am using Adobe Fireworks to make the rectangles around the text labels, but the viewer is not finding the rectangles (i.e. not covering them). What am I doing wrong?

Resolution: When working with Fireworks, the problem was with the "Stroke Category" (to the right of the "Tip Size" that is set to 1 pixel. In any event, the "Stroke Category" should be selected as "Basic ☐ Hard Line". It had been set to "Pencil ☐ 1-Pixel Soft" and that was the option that I showed you that worked only if you laid down the rectangle twice. Apparently, the "Pencil ☐1-Pixel Soft" stroke category dithers the pixels somewhat with the surrounding pixels, causing the pixels to be slightly off-color. Laying down the rectangle twice with the "Pencil ☐1-Pixel Soft" setting will often (if not always) produce a working result, but obviously not in the best way. At least this explanation makes it less mysterious.

Thus, the settings we need in Fireworks are as follows:

- Rectangle Tool
- Foreground color: 0xFF0000
- Fill Color: Transparent
- Tip Size: 1 (pixel)
- Stroke Category: "Basic ☐ Hard Line"

OIA SCORM Viewer

Then, when you lay down a rectangle, it should work with our viewer!

If your problem is not listed please send me an email, describing the problem you are encountering so that I can assist you further: Charlie Hill