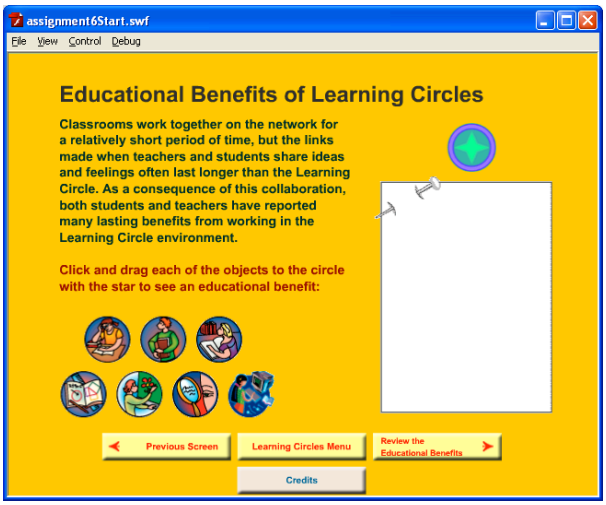
For this assignment we're going to reverse engineer the drag and drop interaction from a portion of one of our exemplar projects. Note you have a couple of files for this. The first is assignment8Start.fla, which contains all of the media already set up. The second file is assignment8Text.txt which contains all of the prose you'll need to plug into the appropriate places. To be specific, you'll rebuild the interaction found on Frame 3 of the contentAndModelMC (which can be found on Frame 2 of the main timeline, or you can just open it in the library).

I'll leave it up to you as to where you want to place your functions for these draggable objects. In the spoiler I put them on frame one of the main timeline. You might want to put them on Frame 1 of the contentAndModelMC movie clip—or put them with your listeners on Frame 3 of the contentAndModelMC movie clip. I care only somewhat, as long as you are consistent. I do put them on frame 1 of the contentAndModelMC timeline to follow our convention of functions up front, listeners where you need them. Some might choose to put them on the main timeline though, just know that will change the pathing for some of your commands.



Note that because the project was originally created for actionscript 2.0 I stripped out almost all of the code (recall you cannot mix and match actionscript 2.0 and 3.0). As a consequence you have basic navigation, but none of the actionscript for individual pages works. That's where you come in.

You might want to take a tour through and see how the navigation is organized—this is one possibility for you to put together your own final project.

Requirements:

1. Make each of the circles with pictures draggable objects, with the idea that the user will drag them over to the circle with a star in the middle as the drop target. (*looking ahead to your final project, note that the draggables are already movieClip objects with instance names, and the drop target is also a movieClip with an instance name—you'll have to do this with your own media before you can get this to work with your own embedded items—assuming you choose drag and drop that is*).
2. Check for the appropriate drop target (circle with star in the middle).
3. If one of the circles is dropped onto the star, then show the relevant text in the dynamic text field immediately below the star (again, the text can be found in the assignment9Text.txt file)

    a. After showing the text, do something so that the learner knows that circle has been used. Options include setting like alpha = .5 or setting visible = false.
4. If the circle is not dropped onto the star, then show the invalid response text prompting them to do so again (this is the last line found in the .txt file).
5. As you are dragging and checking for the appropriate target:
    a. Make sure you can accurately detect the drop target (*hint: use the lockCenter option of the startDrag() function. You should also make sure the registration point is in the middle of the draggable object, advice more for your final projects again, since these have all been set up for you*).
    b. Make sure your object won't run off the edge of the screen (hint: pass the startDrag() function a rectangle). Note if the learner persists and runs their pointer off the screen that's an acceptable bug.
6. Provide comments for **one pair** of functions or for your custom drag and drop functions if you go the custom function route.

Note that this drag and drop is related to the most simple example we covered in the lecture video (the sugar packet). The primary reason for this is I want you to start focusing less on assignments and more on your final projects.

If you are already well versed in Flash: Try to implement an approach to making sure the circle that the user is dragging is always on "top" of the other circles. Use the ability to attach CSS to employ additional text formatting. Bag the idea of Barry's drag and drop in favor of something really cool: revist the photo viewer again, have it randomly present a picture and a set of names, then ask them to drag the appropriate name onto the picture (best to limit the names to 3-4), you might want a stratified random sample of names too (e.g. track not only picture and name but gender, so that the presented names meaningful distractors since showing Erik Hjorten's name as a possibility for Jennifer Jorgenson's picture is a dead giveaway).

- Deliverables: flash development file (.fla)
- Submit to: course website
- File Naming convention: assignment8{YourName}.fla (so if your name were Sam Walker you would submit assignment8SamWalker.fla).

Assessment Rubric

Your assignment will be assessed using the following rubric:

Criteria	Points
Do you use a consistent naming convention for layers, symbols, and pseudo-symbols—in this case the image bitmaps? Did all of your layers have a meaningful name? (e.g. "layer 1" is not an option)	1 points
Is your project easy to change and update? • you should have only the number of instances you absolutely need for each symbol. • you should use consistent tab stops for your code—don't be shy about using the autoformat button in the actions window. • **Finally, you should not have any "magic numbers."** For the purposes of this class, a magic number is defined as a value in ActionScript that is used in more than one piece of code, but not updatable in one place. Note: "magic numbers" do not consist of just numbers, but any kind of data—including String variables.	2 points
Do you have a well organized timeline (related layers are near each other, elements are where they are promised—e.g. student photos are in the pictures layer, not the buttons layer).	2 points
Are all of the required elements (see above) present and working correctly?	4 points
Is the assignment personalized (e.g. not a reproduction of the spoiler video).	1 point
Total	10 points