Assignment 9 – Drag And Drop+, Custom Functions
Guidelines and Assessment Rubric

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 assignment9Start.fla. This is Kendra Hall's project, with much of the Verbs and Pronouns section actionscript removed. The second is fivePointStarFeedback.rtf, which contains the messages, draggable instance names and drop target instance names for each frome of the Verbs and Pronouns section. Here is what has been done so far:

Custom start drag function (frame 1 of main timeline, line 78);

```
//functions for drag and drop
function myStartDrag(draggable:MovieClip) {
	startX = draggable.x;
	startY = draggable.y;
	draggable.startDrag(true, dragBounds);
}
```

Custom stop drag function (frame 1 of main timeline, line 85):

```
function myStopDrag(draggable:MovieClip, myTarget:String,
myTextField:TextField, correctResponse:String) {
	if(draggable.dropTarget.parent.name == dropTargetMC.name) {
		this.contentMC.feedbackText.text = myTarget;
		draggable.alpha = inactiveAlpha;
	} else {
		myTextField.text = invalidIncorrectResponseText;
		draggable.alpha = activeAlpha;

	}
	draggable.stopDrag();
	draggable.x = startX;
	draggable.y = startY;
}
```

There are already instance names and button modes set for each of the draggable objects. For frame 2 of the Verbs and Pronouns section, there are already functions and button listeners for each of the draggables.

The progress gague is working partially. The original function (frame 1 of main timeline, line 20):

```
//custom function for progressGauge
function updateProgressGauge() {
	// variable for amount completed.
	var percentDone;
	// calculate percentage of timeline at or finished.
	percentDone = 100* (contentMC.currentFrame / contentMC.totalFrames);
	// force to a whole number (needed for gotoAndStop).
	percentDone = int(percentDone);
	// update playhead of progressBarMC.
	this.contentMC.progressBarMC.gotoAndStop(percentDone);
}
```

Right now it gets called when learners click on the previous button or the next button. This is a fairly complicated project, and the assignment is one of the more difficult in that you need to be working closely with code that someone else put together. If you want an additional overview of the project, you might look back through the progress gauge screencast for this week.

Requirements:

1. Finish the drag and drop for the Verbs and Pronoun section, making all of the relevant objects draggable for frames 4, 5, and 6. (2 is already done).
2. Note that in later screens Kendra wants learners to drag only some of the objects. Thus, there is correct feedback for only some of the objects, the others will use the "incorrect feedback."
   a. For example, on frame 5 of the Verbs and Pronouns section there are five words but information for only 4 of them. One (tener) is a verb, it will always be incorrect.
   b. This will require a little thinking on your part—as a hint, think about giving them a drop target they'll never be able to get to for those items that are always incorrect.
3. Get the progress gauge working consistently.

If you are already well versed in Flash: Improve the drag and drop functions so that the draggable object being manipulated is always on "top". As a code hint, the following can be adapted to work. Keep in mind that this refers to the parent of the draggable object, so you'll need to pass a reference to the parent as well or perhaps path to it:

```
this.setChildIndex(readingWritingMC, this.numChildren-1);
```

Use the TLF to employ additional text formatting (perhaps highlighting the repetitious words of "correcto!", "Oops" and "No, that is not correct."). You might also use Kendra's strong naming convention to implement both incorrect and invalid response feedback—hint if you find the dropTarget instance name begins with "dropTarget" you are 90% of the way there.

- Deliverables: flash development file (.fla)
- Submit to: course website
- File Naming convention: assignment9{YourName}.fla (so if your name were Sam Walker you would submit assignment9SamWalker.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 • **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? **Free points for this assignment** since this part is already done.	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	1 point

the spoiler video).	
Total	10 points