

Using Stamps in a Matching Game Custom Messaging, Part 3

D. P. Story

The source file for this article is attached to this PDF. Click [here](#) to open/close the attachments panel. Within the attachments panel, you can save the source file for the example presented in this article.

Important: This document needs to be viewed using Adobe Reader XI, or later; otherwise, the functionality of the game presented here is lost. The document will behave as designed when viewed using Adobe Acrobat, any version. Also, for the drag and drop matching to work, this file needs to be viewed *outside a browser*.

1. Introduction

This is the third article in the series of customizing messaging, as needed, review [Part 1](#) and [Part 2](#).¹ In this article, we show some methods of having multiple games in one document, while maintaining control over the messaging.

2. Custom messaging: Multiple DAD games

In this document, we present three DAD (Drag And Drop) games, their names are Presidents1, Presidents2, and Presidents3. We use custom messaging for first and third games, but use the default messaging for Presidents2.

We begin by presenting the three DAD games. Play with them and observe the messages on the alert boxes as they appear.

2.1. The 'Presidents1' game board: Custom messaging



**John Quincy
Adams**

**Thomas
Jefferson**

**James
Madison**

**George
Washington**

John Adams

Care must be taken to insure the game icons (stamps) and their targets (push buttons) are all on the same page. Here is issue a `\newpage`.

¹These links work in a browser on the AcoTeX Blog website, or on your local computer provided Part 1, Part 2, and Part 3 are all in the same folder.

2.2. The 'Presidents2' game board: Default messaging



John Quincy
Adams

Thomas
Jefferson

James
Madison

George
Washington

John Adams

2.3. The 'Presidents3' game board: Custom messaging



John Quincy
Adams

Thomas
Jefferson

James
Madison

George
Washington

John Adams

3. Discussion of methods of messaging

In the partial verbatim listing below, we suppress the messages.

```

1  \begin{insDLJS}{dad}{Custom Notifications}
2  var aCustomResponses=new Array("Presidents1","Presidents3");
3  var oAlertTitle=new Object();
4  oAlertTitle["Presidents1"] = "The Presidents of the United States";
5  oAlertTitle["Presidents3"] = "The Presidents of the United States";
6  var oCorrMsg=new Object();
7  oCorrMsg[ddTrueName("Presidents1","George Washington")]="Game 1:...";
8  ...
9  oCorrMsg[ddTrueName("Presidents1","John Quincy Adams")]="Game 1: ...";
10 oCorrMsg[ddTrueName("Presidents3","George Washington")]="Game 3: ...";
11 ...
12 oCorrMsg[ddTrueName("Presidents3","John Quincy Adams")]="Game 3: ...";
13 function ddCustomCorrectAction(event,ddName) {
14     var addStats=" (" +ddStats.nCorrect+" out of "+ddStats.nTries+)"
15     if (aCustomResponses.some(function(element){return(ddName==element)}))
16         app.alert({cMsg:oCorrMsg[event.targetName]+addStats,
17                 nIcon: 3, cTitle: oAlertTitle[ddName]});
18     else ddCorrectActionDef(event,ddName);
19 }
20 var oErrMsg=new Object();
21 oErrMsg[ddTrueName("Presidents1","George Washington")]="Game 1: ...";
22 ...
23 oErrMsg[ddTrueName("Presidents1","John Quincy Adams")]="Game 1: ...";
24 oErrMsg[ddTrueName("Presidents3","George Washington")]="Game 3: ...";
25 ...
26 oErrMsg[ddTrueName("Presidents3","John Quincy Adams")]="Game 3: ...";
27 function ddCustomWrongAction(event,ddName) {
28     var addStats=" (" +ddStats.nCorrect+" out of "+ddStats.nTries+)"
29     if (aCustomResponses.some(function(element){return(ddName==element)}))
30         app.alert({cMsg:oErrMsg[ddTrueName(ddName,activeStampName)]+addStats,
31                 nIcon: 0, cTitle: oAlertTitle[ddName]});
32     else ddWrongActionDef(event,ddName);
33 }
34 \end{insDLJS}

```

As mentioned in the earlier parts of this topic,² to obtain custom messaging the functions `ddCustomCorrectAction()` and `ddCustomWrongAction()` need to be defined as appropriate; the new wrinkle is that we want to use the default messaging, which is the JavaScript functions `ddCorrectActionDef()` and `ddWrongActionDef()`. In line (2) an array is defined that lists the DAD games that are to have custom messaging.

We define custom messages for those two games in lines (7)-(12) and in lines (21)-(26).

In lines (15) and (29) we test whether `ddName` is an entry in the array `aCustomResponses` array using the `Array.some` method, the method returns `true` if the `ddName` is listed

²Part 1 and Part 2

in the array. Now, `true` is returned, we executed our custom alert (lines (16)-(17) and lines (30)-(31); otherwise, the default message handler, `ddCorrectActionDef()` or `ddWrongActionDef()`, is executed.

Now, I simply must get back to my retirement. ~~DS~~