AcroTEX.Net

Using Stamps in a Matching Game
Custom Messaging, Part 2

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*.

AeB Blog August 28, 2016

http://www.acrotex.net
Published: August 28, 2016

1. Introduction

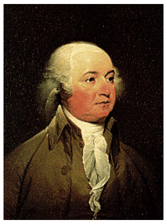
This article is a continuation of Using Stamps in a Matching Game: Custom Messaging, Part 1.[1] It is recommended that you read that article first. In the first article, we demonstrated how to modify the messages the player views as he/she drags and drops the stamps to their targets. The messages were all the same (static), except for a dynamic part of the message that presented some statistics.

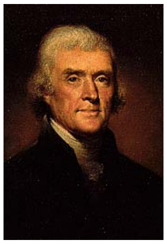
In this article, we show how to offer a different message for each of the stamps.

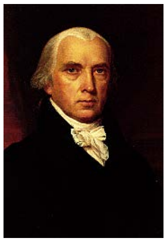
2. Custom messaging: Different messages for each stamp

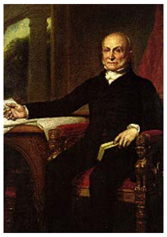
The presidential drag and drop game is presented below. As you drag and drop presidential images, observe the messages in the alert boxes that appear, both when you drop a president in the correct rectangle and when you drop him in an incorrect rectangle.

As usual, the rules are to drag and drop each president into the rectangle containing that president's name as the caption.









John Quincy Adams

Thomas Jefferson

James Madison

George Washington

John Adams

Reset

As it was with Part 1, the key to success is in the definition of the two JavaScript functions named `ddCustomCorrectAction()` and `ddCustomWrongAction()`.

[1] This link works in a browser on the AcoTeX Blog website, or on your local computer provided Part 1 and Part 2 are in the same folder.

```
1  \begin{insDLJS}{dad}{Custom Notifications}
2  var alertTitle="Presidents of the United States";
3  var oCorrMsg=new Object();
4  oCorrMsg[\ddTrueName(Presidents,George Washington)]
5      ="George Washington, the first President.";
6  oCorrMsg[\ddTrueName(Presidents,John Adams)]
7      ="John Adams was the second President.";
8  oCorrMsg[\ddTrueName(Presidents,Thomas Jefferson)]
9      ="Thomas Jefferson was the third President.";
10 oCorrMsg[\ddTrueName(Presidents,James Madison)]
11     ="James Madison was the fourth President.";
12 oCorrMsg[\ddTrueName(Presidents,John Quincy Adams)]
13     ="John Q. Adams was the fifth President.";
14 function ddCustomCorrectAction(event,ddName) {
15     app.alert({cMsg: oCorrMsg[event.targetName],
16         nIcon: 3, cTitle: alertTitle});
17 }
18 var oErrMsg=new Object();
19 oErrMsg[\ddTrueName(Presidents,George Washington)]
20     ="Wrong! Some called him George.";
21 oErrMsg[\ddTrueName(Presidents,John Adams)]
22     ="Mistake! His name is John.";
23 oErrMsg[\ddTrueName(Presidents,Thomas Jefferson)]
24     ="Bad, bad, bad! Thomas would be disappointed.";
25 oErrMsg[\ddTrueName(Presidents,James Madison)]
26     ="Oh, come on! " +
27      "His name is James, ask me again and I’ll tell you the sames.";
28 oErrMsg[\ddTrueName(Presidents,John Quincy Adams)]
29     ="You can’t get anyone right! " +
30      "His name is John too, and son of the other John.";
31 function ddCustomWrongAction(event,ddName) {
32     app.alert({cMsg: oErrMsg[ddTrueName(ddName,activeStampName)],
33         nIcon: 0, cTitle: alertTitle});
34 }
35 \end{insDLJS}
```

We define two new objects `oCorrMsg` and `oErrMsg` (lines (3) and (18)). We then declare the messages that we want to appear for correct answers (lines (4)–(13)) and for wrong answers (lines (19)–(30)). To associate the message with the stamp, the property name is given using the `\ddTrueName` command. Lines (2)–(5) are reproduced below for comment:

```
oCorrMsg[\ddTrueName(Presidents,George Washington)]
    ="George Washington, the first President.";
```

The syntax for `\ddTrueName` is `\ddTrueName(`⟨*game-name*⟩`,`⟨*stamp-name*⟩`)`, where the argument is enclosed in parentheses and separated by a comma (`,`); there must be no space following the comma. The ⟨*game-name*⟩ argument is the name of the drag and drop game, as declared by the `\initDDGame` command, `\initDDGame{Presidents}` for this game. The ⟨*stamp-name*⟩ is the name of the stamp as declared by the first argument of `\ddTargetOfIcon` commands. The George Washing stamp was declare as `\ddTargetOfIcon{George Washington}{George Washington}`; the first argument here is the stamp name and second argument is the caption name, they may or may not be the same.

Finally we come to the meaning of `\ddTrueName(Presidents,George Washington)`. This command is a convenience command that expands to the JavaScript function `ddTrueName` defined by `aeb_dad`. The expansion is

```
ddTrueName("Presidents","George Washington")
```

that is, it inserts the double quotes around each of the argument, hence the necessity that there be no spurious spaces. The `ddTrueName` function returns the true name of the underlying pushbutton field. In this case, `ddTrueName("Presidents","George Washington")` returns `"Presidents@#George Washington"`, which is the name of the pushbutton field where George Washington belongs.

Now for a discussion on the `app.alert()` method of the custom actions:

```
function ddCustomCorrectAction(event,ddName) {
    app.alert({cMsg: oCorrMsg[event.targetName],
        nIcon: 3, cTitle: alertTitle});
}
function ddCustomWrongAction(event,ddName) {
    app.alert({cMsg: oErrMsg[ddTrueName(ddName,activeStampName)],
        nIcon: 0, cTitle: alertTitle});
}
```

In the case of `ddCustomCorrectAction()`, the player has dropped the correct stamp onto the correct rectangle, so we issue the message `oCorrMsg[event.targetName]`. The `event.targetName` property contains the name of the field the player has dropped the stamp onto (for example, the George Washington stamp is dropped onto the first whose name is `"Presidents@#George Washington"`. The associated message is then displayed in the alert box.

When `ddCustomWrongAction()` executes, that means the player has dropped a stamp onto the wrong rectangle. To get the correct message we must determine the name of the stamp the player has moved, that name is stored in the JavaScript variable `activeStampName`. We issue the error message for the name returned by

```
ddTrueName(ddName,activeStampName)
```

which might expand to `"Presidents@#George Washington"` if the player dropped the George Washington stamp. By the way, `event.targetName` is the name of the field the stamp was dropped onto, we don't want this associated message.

To summarize: (1) `event.targetName` is the name of the field the stamp is dropped onto; and (2) `ddTrueName(ddName,activeStampName)` is the name of the field the stamp is supposed to be dropped onto.

Now, back to my retirement. DPS