Option Chooser 1.0

What is Option Chooser ?

Option Chooser 1.0 is a drag-n-drop interface allowing users to sort items into multiple categories. It is javascript-based and built to use the DynAPI Cross-Browser DHTML Library.

Option Chooser exceeds the functionality of simple <SELECT MULTIPLE> form elements by allowing the user to sort their selected items into any number of categories. Also, the chooser is not bound by the traditional "*Add to Column A -->*" and "*<-- Remove from Column A*" buttons, needed to emulate the same functionality with html forms. Instead it provides an intuitive interface allowing the user to simply drag and place items into the category they would like and in whichever order they prefer.

Further surpassing html form selects, the chooser allows for a good deal of customization of it's appearance. Items can be images or images with text and their dimensions can be set. For instance, one might use square-items if there are relatively few to select from and wide, short items if there are many. It is also designed for a mouse-over description or explanation to appear when the user hovers over an item.

Because Option Chooser uses the DynAPI library, it is compatible with most browsers used currently.

Possible Uses :

The chooser was originally designed to allow end-users to customize where and which items would appear when they visited their personalized version of the AgNIC (www.agnic.org) website. After choosing the number of columns they would like in their page, the Option Chooser would allow them to organize the layout of various pieces of content into those columns.

Another use might be as part of a knowledge test (ie- "Sort these into their appropriate categories, ordering by importance"), or any number of other creative uses.

Of course, Option Chooser can also function with a single category to select into, simply serving as a better-looking select-multiple.

How to Use Option Chooser :

* Read instructions all the way through before beginning

Requirements

- DynAPI 2.5.6 library (with my one bug-fix) should be included in the same directory
- The "images" folder must be in the same directory and contain the files "bigLback.jpg", "options.gif", "selected.gif", "sepLine.gif", and "submit.gif"
- The user of the system must have javascript enabled on their browser
- The system should work on most browsers that support javascript 1.2, and has been tested successfully on Netscape 4.x,6.x(Mac & Windows) and IE 5.x(Mac & Windows),6.x (Windows)

How to Set Up

- Place the "OptionChooser.html" file in the same directory as the "dynapi" folder.
- Open "OptionChooser.html" in a text editor.
- This file is what you will add html into to make your final page when you've finished setting up the chooser
- First you must declare which options are available … goto lines 28 – 44 of the file:

```
startLeftChoices=new Array(
          "'Cow'"
                    +"|'<img src=\"images/cow1.gif\">'"
                    +"|'<img src=\"images/cow2.gif\">'"
                    +"|'<b>Laughing Cow</b> Spreadable Cheeses - delicious variety'"
                    +"|''|''",
          "'Calendar'"
                    +"|'<img src=\"images/cal1.gif\">'"
                    +"|'<img src=\"images/cal2.gif\">'"
                    +"|'A searchable <b>calendar</b> of events'"
                    +"|''|''",
          "'Vermont'"
                    +"|'<img src=\"images/vt1.gif\">'"
                    +"|'<img src=\"images/vt2.gif\">'"
                    +"|'<b>Vermont Cheddar</b> - perfect for any cracker'"
                    +"|''|''"
)
```

- There are already 3 example items that can be replaced.
- The “startLeftChoices” array is where all the initially unselected items will be declared.
- It is a comma-separated list of strings, where each string represents one item and is divided into six elements by the | character. Each string is also divided onto 5 lines and those sub-strings are joined with the + character. This separation is necessary because some browsers will choke on strings that are over 255 characters long and declared on a single line. We will ignore this separation and only pay attention to the places where the string is separated with a | .
- The first element is the item’s ID … it should be one word, no spaces, no special characters … it will be what the chooser sends out when you press submit.
- The second and third elements are the normal html for the item and the html for the item when it is being dragged, respectively. In the example items here, images are used for each state … you may combine images with text or have just text alone, but some browsers will not display more complex html properly, such as tables. The easiest way to check if your item-html will work is to test it with Internet Explorer 5.x on the Macintosh.
- NOTE: all double and single quotes that go inside our string must have the \ escape character before them, so javascript does not mistake them for the end of a string.
- The fourth element is again html; this time it is what will be displayed in the description box at the bottom of the chooser when you mouse-over that item.
- The fifth and sixth elements are the normal and “being dragged” item background colors, respectively. If you want the item to have a transparent background, set both these fields to the empty string ''. If you want them to be colored, set them each to an html color, like '#8899ff'.

- Next you must decide how many categories you’d like to have
  - For each category, put its name in the “allRightChoices” array:

```
allRightChoices=new Array(
          "A Good Brie","Cheddar","Meunster"
)
```

  - This name is what will be displayed at the top of the column for that category.
  - Also, for each category you create here, **you must make a corresponding array below**, like: stRightChoices0=new Array()
  - These arrays must be numbered starting at 0, and they correspond to their respective entries in the allRightChoices array.
  - You may also fill these arrays with items, if you wish them to initially appear in the corresponding category. In fact, the three example columns come with some example items.

- Next in the file is where you can customize some dimensions for your interface:

```
//User-provided constants
bigLHeight= 430;
bigLWidth= 600;
listsTopOMargin= 30;
listsLeftOMargin= 6;
listsLeftMargin= 11;
optionsTopMargin= 11;
selTopMargin= 25;
selTitleMargin= 3;
listItemHeight= 30;
listItemSep= 5;
descHeight= 20;
```

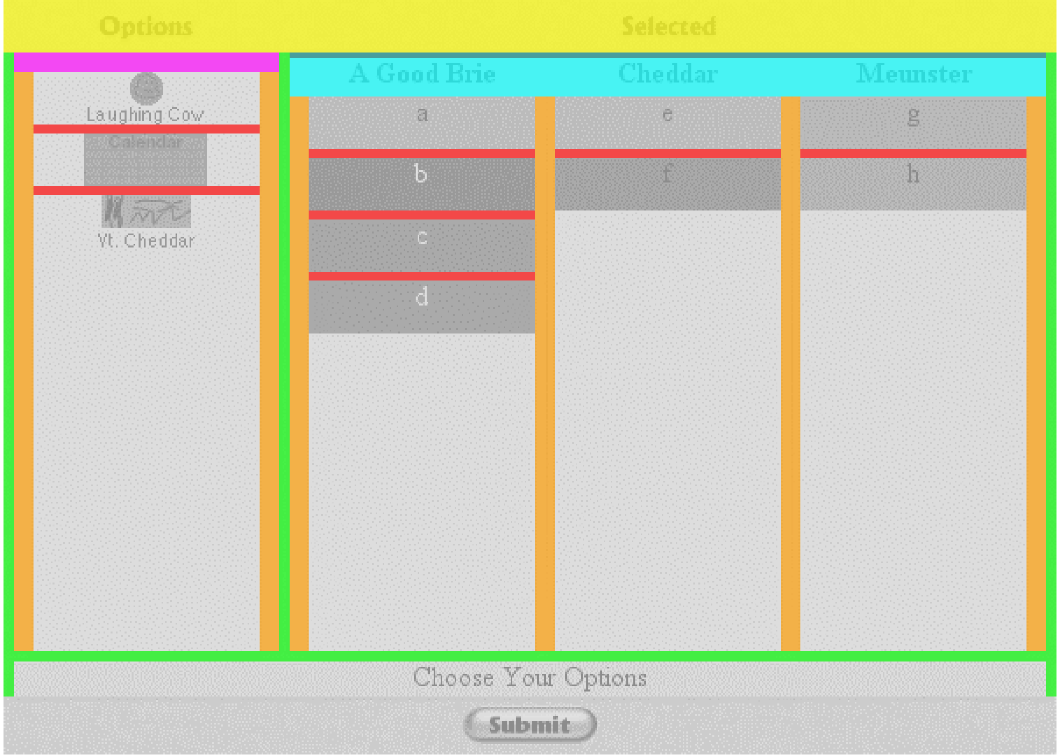
- bigLHeight and bigLWidth specify the height and the width of the entire interface.
- bigLHeight must also be repeated in these locations at the very bottom of the file:

```
<center><table height="430" width="600" cellpadding="0" cellspacing="0">
<tr height="430"><td height="430" width="600" valign="top">
        <div id="bigL" style="position:absolute;width:600;" align="center"></div>
        <div id="bigLN" style="position:relative;width:600;" align="center"></div>
```

- And bigLWidth must be repeated in these locations at the bottom of the file:

```
<center><table height="430" width="600" cellpadding="0" cellspacing="0">
<tr height="430"><td height="430" width="600" valign="top">
        <div id="bigL" style="position:absolute;width:600;" align="center"></div>
        <div id="bigLN" style="position:relative;width:600;" align="center"></div>
```

- NOTE: There is another file included with this distribution, "OptionChooserA.html," which uses different sizing methods (its width automatically stays sized to the page-width). You may wish to use that version instead, depending on your application. In that version, the width does not get specified at the bottom of the file.
- The constants for listsTopOMargin, listsLeftOMargin, listsLeftMargin, optionsTopMargin, selTopMargin, selTitleMargin, and listItemSep correspond to the thickness of these lines:



  - listItemHeight controls the height of every item.
  - descHeight controls the height of the description bar at the bottom.
- Next you must choose where you would like the information from Option Chooser sent
  - The chooser uses the POST method to send its information out
  - It comes set to send its contents to a script at dangeorge.com that simply spits back everything that was posted to it.
  - To change where the chooser posts, edit the action attribute of the form at the very bottom of the file from "http://pc.dangeorge.com/scripts/postback.php" to the location of your own script … or set it to "mailto:youremail@example.com"
  - The chooser's output comes in the form of posted arrays: There is one array called lchoices, which contains all of the item IDs for those items left in the Options column. Then, there are as many other arrays as there were categories. They are named rchoices0, rchoices1, … and on. They behave the same as the lchoices array.
- Add your own html to the page between the indicated lines near the bottom of the file
  - Also change the header and title that read "Option Chooser 1.0" to whatever you would like.

Use with Server-Side Scripting

This script may also be dynamically generated so as to set all of the above mentioned configurations from a database or other dynamic source. Just remember to make sure that the script produces the appropriate escape characters in its output instead of mistaking them for characters that it must escape when serving-up the page.

Known Issues

- Internet Explorer 5.x(Mac) cannot display more complex html in layers properly
- Most browsers err in some way or another when the window is resized, thus Option chooser is set to reload when this happens. However, Netscape 4.x for Windows produces an error and must be manually reloaded after a resize.
- When making a string (such as those used to define the items), errors may occur if there are over 255 characters all added to the sting on a single line.
- Netscape 4.x does not take the DIV's align attribute as applying to all of the layers it contains. Thus it does not center everything in the interface … However, if you really wish things to be centered on Netscape 4.x, simply enclose the html elements with a <center> tag when you define them.
- Explorer 5.0(Mac) (5.1 works fine) can't find the position of bigLayer in "OptionChooserA.html". Fix this by altering the line "if (browser4) {y-=55;}", changing the number 55 to the distance of the chooser from the top of your page.

License Information