Interactive Map Manual

Table of Contents:

tool version : 1.2.0

Embedding the map into a webpage

I do recommend using swfObject (http://code.google.com/p/swfobject/wiki/documentation) for embedding the Interactive Map into a webpage. Here's an example:

```
<script type="text/javascript" src="swfobject.js"></script>

<script type="text/javascript">

      var flashvars = {};
      flashvars.map      = "map";       // path to directory with map
      flashvars.mapFile  = "map.jpg";   // filname with map
      flashvars.ui       = "ui";        // path to folder with ui swfs
      flashvars.hotspots = "hotspots";  // path to hotspot.xml and hotspots

      var params = {};
      params.scale = "noScale";
      params.salign = "lt";

      swfobject.embedSWF("imap-free.swf", "flashContent", "720", "500",
"9.0.0","expressInstall.swf", flashvars, params);

</script>


<div id="flashWrapper">
      <div id="flashContent">
      Detecting Flash....
      </div>
</div>
```

As you can see in the above example, it is possible to set custom flashVars parameters. If you don't set them, default values (the same as in the example above) are used.

Please note that you need to have the swfobject.js downloaded and included in the code. The paths are relative to the webpage where the swf is embedded. It is also possible to use absolute pahts, e.g. `flashvars.map = "http://www.mydomain.com/tour/map";`

Interactive Map Manual

Directory structure of the data

The data is sturctured in three folders, to keep the hotspot data and skin files separate:

```
- hotspots
- map
- ui  (has 2 subfolders)
  - preloader
  - screens
```

-> Folder **hotspts** contains the ****hotspots.xml**** file and the hotspot data (hotspot images). Hotspots.xml defines all about the hotspots: their location, title, description and images. For easier data management I recommend to create a folder fore each of the hotspots, and keep the images of each hotspot in it's folder.

-> Folder **map** contains the map or the background image, on which the hotspots are placed. Currently the scripts loads by default the file ****map.jpg**** which is placed in the map folder. If you want to use some another map file, such as map.jpg, or map.png, or mymap.jpg, you must specify this file name in the flashVar parameter.

-> Folder **ui** contains flash movies, which build up the user interface or the skin. By replacing these files with yours, you can theme your presentation with an another skin. The ****preloader**** folder contains the preloader movie, the ****screens**** folder contains the screens for the map and for the detail view.

The folder names "hotspots", "map", and "ui" are default and the flash movie looks for the data in these folders, if no custom folders are defined using the flashVar parameters. See the chapter : Embedding the tool int a webpage

Plugging in your map image

To plug in your map image replace **map.jpg** in the **map folder** with your custom image. It is also possible to plug in a swf file. If the fileName is different than "map.jpg", specify it in the mapFile flashVar parameter.

Interactive Map Manual

Editing hotspots

The hotspots are defined in ****hotspots.xml**** file, which is located in the **hotspots** directory. Each hotspot has the attributes:

- **x** : x position of the hotspot from the top-left corner of the map
- **y** : y position of the hotspot from the top-left corner of the map
- **category** : Name of the category the hotspots belongs to (displayed in map legend)

Each hotspot node has the subnodes:
- **title**
- **item** (one or more)
- **javascript** (optionally)
- **url** (optionally)

The ****title** node** defines the title which is displayed on hotspot mouse-over (this behaviour can be changed in screen-map.swf). The hotspot number reflects it's position in the config.xml. The first hotspot has the number 1 etc.

You specify either **item**, **javasript** or **url** node:

The ****item** node** defines an image and it's description in the detail view (after a hotspot is clicked on). A hotspot can have one or more item nodes. Each item node has this subnodes:
- title : image title
- description : image description
- image : path to image, relative to the hotspots folder
- thumb : pat to thumb image, relative to the hotspots folder

The ****javascript** node** defines which JavaScript function to call, example:
<javascript parameter="blank">myJavaScriptFunction</javascript>

The ****url** node** defines which url to open, example:
<url target="_blank">http://www.sme.sk</url>

The hotspots.xml file contains also the information on startposition - which place on the map should be displayed when the interactive map is first time loaded.

Creating custom skins and modifying the movie size

The size of the movie and its appearance are defined in two (at least two) files: screen_map.swf and screen_detail.swf.

Screen_map is the screen which displays the map, hotspots, navigator and map legend. By editing the screen_map source file you can create a new custom skin or change it's size.

Screen_detail is the screen, which is displayed after a hotspot is clicked. Currently, the interactive map is shipped only with one type of the detail screen which is a picture viewer. By editing the screen_detail source file you can change the look of the picture viewer or change it's size.

See next page for details.

Interactive Map Manual

Editing screen_map

The source file screen_map.fla is located in the folder "ui/screens/". Editing this fla file you can create a new skin or change the size of the flash movie. The file has these layers:

```
- Footer
- Legend       : is displayed when the "Show legend" button is pressed
- Navigator    : displays a thumbnail of the map and a highlighted rectangle which
                 represents the visible area
- Map Target   :  a movieclip into which the map is loaded
```

If you resize the assets, I do recommend resizing the actual content of the movieclips, not the movieclips themselves. E.g. do not resize the mapTarget movieclip directly, but edit it and resize it's content. In other words the size of the movieclip must always be 100%.

The screen_map.fla does have also some assets which are not on the stage, but just in the library.

- hotspot : is the hotspot which can be clicked on. It does have an over state. In the example which is shipped in the download the "over" hotspot consists of three parts, the center part get's scaled according to the length of the title textfield. It should be pretty straightforward if you want just to change colors or fonts of the hotspots. If you want a completely different hotspot, you will need to edit the actionscript class "assets.hotspot.Hotspot" of the hotspot.

Editing screen_detail

The source of screen_detail. is located in the folder "ui/screens". Editing this fla you can modify the look of the detail view or it's size. The layers of the fla are structured self-explanatory so it should not be a problem to locate the assets you want to update.

The default screen_detail provided in the download is a simple picture browser. It does have the main image with it's description displayed + a filmstrip of thumbnails displayed below. If there are more thumbnails, the filmstrip is scrollable.

All actionscript classes are provided for the screen_detail, so it is possible to make larger changes to the picture viewer than changing colors, positions, fonts which you can do just by editing the fla. file.

Events and Interface

Both screen_map and screen_detail communicate with the parent application with events + they expose some public functions (interface) like setData(). That means that you can plug in a completely new application for your screen_map or screen_detail. E.g. if you don't like the provided picture viewer for the screen_detail, you can plug in your own. It just needs have the same interface (public functions) and dispatch the same events.

Support and contact

Feel free to contact me at the support forum http://www.yofla.com/forum/index.php?action=vtopic&forum=11 or directly at info@yofla.com with your questions. I provide a timely response for my customers, for those who are using the free version I answer as soon as it is possible to do so without affecting the quality of service for the customers who paid for a non-free version. Thanks for understanding!

CREDITS
****

Bulkloader : http://code.google.com/p/bulk-loader/
PureMVC : http://puremvc.org/
TweenLite : http://blog.greensock.com/tweenliteas3/