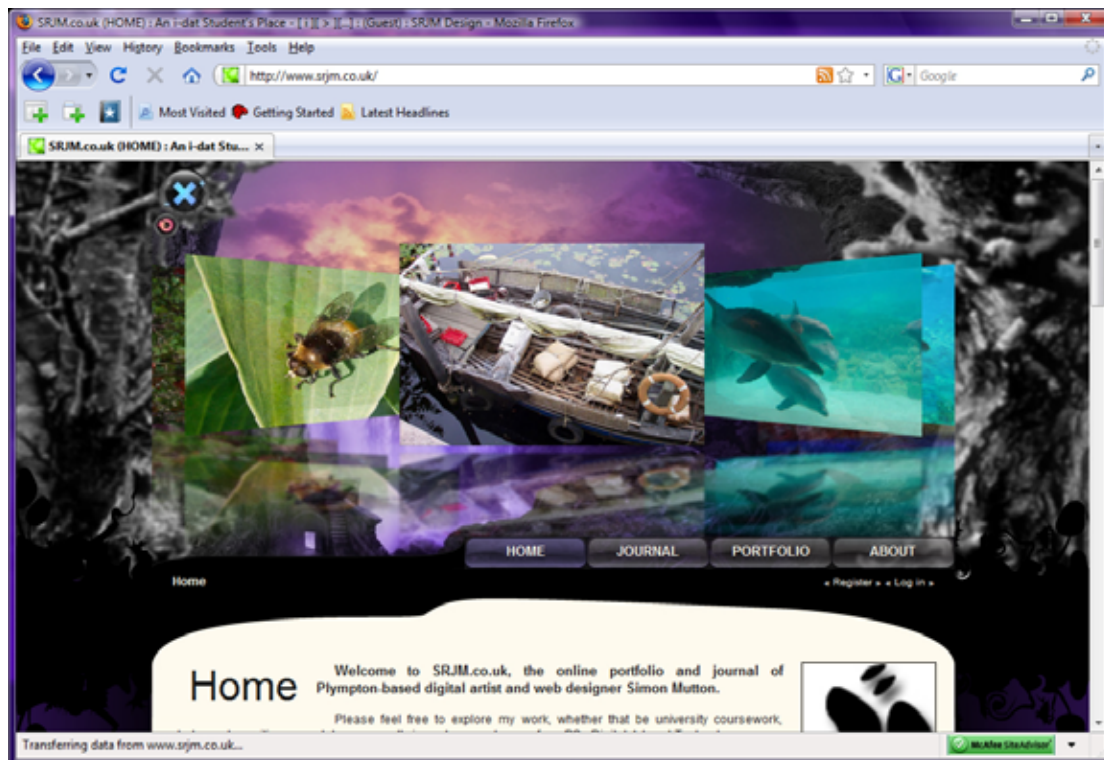


SRJM.co.uk Portfolio Gallery

Scripting for Digital Media: Flash Application Documentation



This gallery is meant to provide a browser-based visual showcase for various images such as photography or artwork, all loaded dynamically off of a server when the user requests the data by interacting with the piece. Since the intention was to make it part of a much larger portfolio-esque solution, I have also included a previous flash project in the form of a puzzle game, with its own set of easily portable Actionscript functions.

End user interaction has been made as straightforward as possible, with additional modes of interaction (such as the keyboard arrow keys) readily available as well, if not immediately noticeable. Due to the size constraints of the website's header, the flash content remains partially hidden until activated by clicking on the blue icon. Once activated the user can browse which gallery (or eventually portfolio item) they wish to view, and open it.

A live demo will always be available, but possibly updated in the future in the header of each page on <http://www.srjm.co.uk/>.

The non-live version features a larger test photo gallery.

All code within the main .fla file and the .as includes was written by myself with the notable exception of DistortImage.as, which is an open source class for performing skews on a bitmapData object, something which I was unable to do myself. Additionally in the puzzle game there is a borrowed 'class' that prototypes a sine wave motion to movieclips. Otherwise both parts of the project (gallery and puzzle) were developed from scratch with a lot of help from the Macromedia/Adobe Livedocs help site.

Project.fla

- The entry point for the galleries, it loads the data as an XML object. This means the gallery can be freely manipulated with other server technology, for example PHP generated XML. Therefore beyond the flash, this gallery could include a serverside administration panel that allows a site owner to modify its contents or alternatively it could display an automated script such as an RSS feed of my latest Flickr photos.
 - The flash gallery also communicates with the client's browser via AJAX in order to activate CSS changes and check that its container <div> is the correct size.
 - Further possible extension could include AJAX/javascript navigation and browser history to simulate an URL structure for Flash, so that users can bookmark various sections of the portfolio, for example #galleries/gallery2/image8.

MediaGallery.as – Gallery Class

- The gallery class will create multiple instances of MediaObject.as that load and display the images (discussed in detail later). The galleries provide the clock for the scripted animation, telling each Media Object where they are meant to be.
 - When there are multiple galleries they will rotate from top to bottom, in a wheel shape and therefore each gallery is 'aware' of the others, keeping track of where they are in the order.
 - Images will rotate from right to left (as you might pages of a book) when on auto-play and there are more than one image in the gallery.

- This setup is entirely flexible, and it is possible to go from having one image in a single gallery (no animation will occur) through to hundreds of images in many galleries.
- It is possible for the user to interact by using the direction keys to override the auto-play. Similarly, pressing Esc will unload the galleries.

MediaObject.as – Media Object Class

- The Media Object class contains and loads the image. My intention here was to expand it beyond just displaying images but rather being able to load multiple types of media such as video clips and audio. While I have no doubt that this could be achieved alas time constraints dictated otherwise.
- The visual representation of the object relies heavily on the bitmapData class object that is used to dynamically manipulate the pixels. Similarly, bitmapDatas of both the reflection and image were converted into the distortImage class so that they could be realistically skewed to simulate depth and perspective.
- The Media Object calculates its position from the public variables in its parent Media Gallery.
- Obviously CPU usage was a serious issue for such dynamic animation so I have taken care to only run functions in the onEnterFrame event when they are absolutely required.
- All the images are loaded dynamically for the dual purpose of making the gallery contents completely customised and flexible while also conserving bandwidth on the server when the gallery (and therefore images) isn't loaded.

_mcPuzzleGame – Puzzle Game

- The puzzle game randomly generates a puzzle of a range of sizes and then jumbles the pieces a given number of times (however this is currently set to once).
- The generation function will create all the pieces within the world size. It also sets the various matching sides as it works through each piece by checking the surrounding pieces. For example if the current piece has already set sides to the top and left of it, these will be made to match and the remaining sides will be set randomly. This ensures there is always a solution.
- The jumbling function rolls through each piece and swaps it with a random new location. If by the powers of randomness the puzzle is immediately solved the puzzle jumbles itself again, but this is highly unlikely except for the 2x2 size.
- The swap function (used initially to jumble the pieces) is then used by the player to swap pieces until they find a solution.
- Every time a piece is moved the puzzle runs a search function to check that all the sides are matching, and if they are, the player is congratulated and wins the game.
- Additionally, I included functionality for rotating pieces as well as swapping them, as the cardboard versions of such puzzles ultimately allow. However I thought this was an additional difficulty too far – it is certainly challenging enough as it is and I know of only one person with the patience and brain to have solved the Ultimate 5x5 size board.