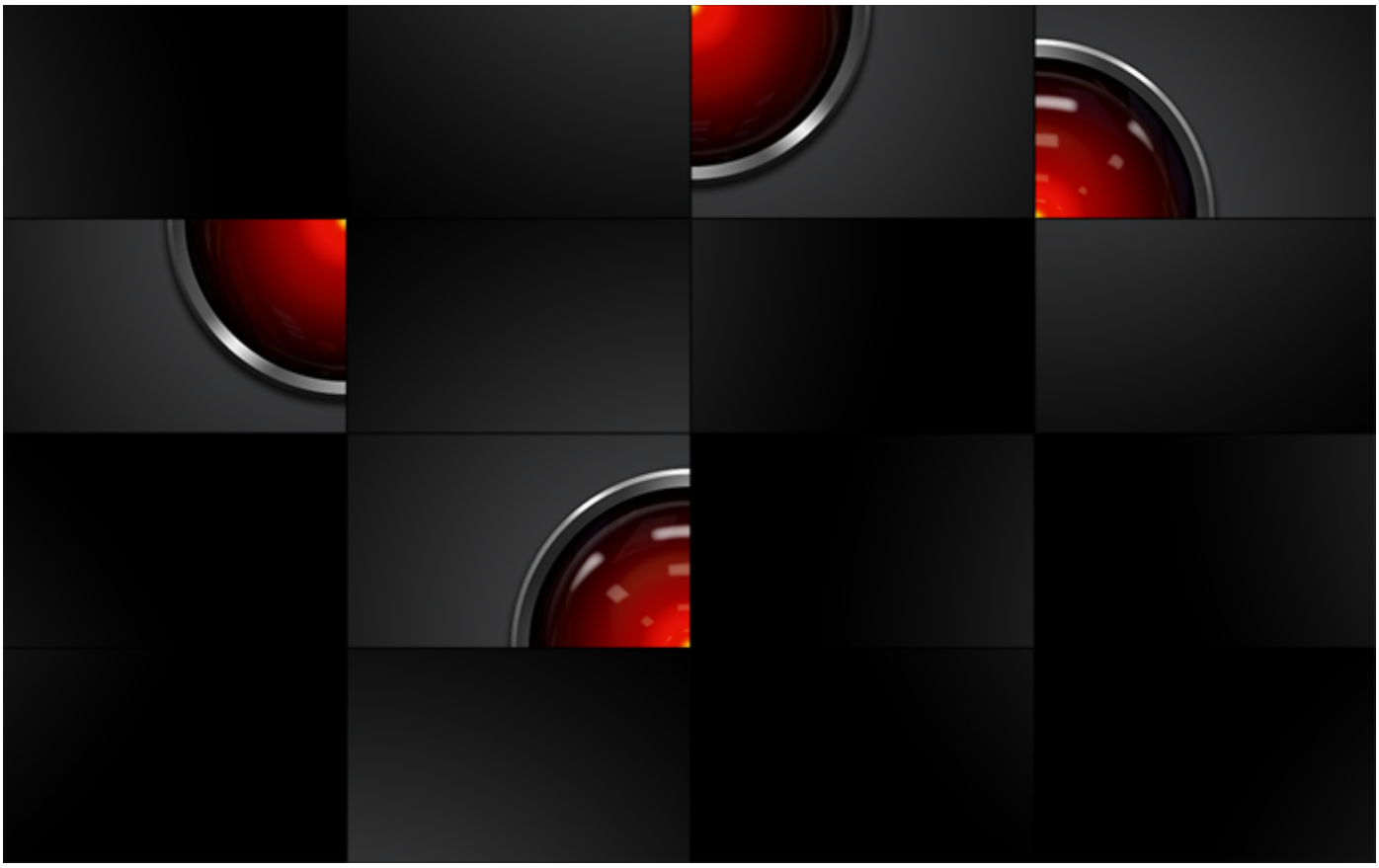


PUZZLE APP

Product Specifications

Authored by: **Armagan Tekdoner**, Web Developer

Ontario, Canada

TABLE OF CONTENTS

- The Criteria & How They Were Met
- Introduction
- Documentation
- User Experience
- Globally-set Styling Rules & Client-side Functionalities
- Technology & Extra Features
- Webmaster tools
- Test-to-fail the App
- Conclusion
- Appendix

THE CRITERIA & HOW THEY WERE MET

1. ***Create a web page allowing an image to be uploaded***

The custom-built web page for this purpose is http://www.grifare.net/examples/puzzles/puzzle
HTML5 and CSS3 presents the form, PHP uploads the image and runs the validations.
The directory that stores the images is [grifare.net/examples/puzzles/uploaded-images] with 777 permissions. All files there can be viewed using the app.

2. ***When the image is uploaded, split the image into at least 9 different pieces***

PHP collaborates with JS, native JS does the puzzle job, jQuery handles the overall interaction. Besides piece option, various fancy options have been provided as well.

3. ***Display the pieces on screen but in a mixed up order***

Any uploaded image is displayed exactly as described above, quickly.

4. **Allow the user to then move the pieces into the proper order to reform the image**

When any piece is dragged and dropped, it replaces the piece in the target square and the piece replaced goes to the moved piece's source location**.**

5. ***The page should recognize and produce an output when the pieces are back in the proper order***

When the pieces match, clear notifications are posted, options are provided, and puzzle is no longer editable when solved. Many cases were taken into account.

INTRODUCTION

This single-page app is located in the domain of **grifare.net**, a dedicated website, which I created as an experimentation platform and a coding showcase.

The way this **Puzzle App** has been handled, is meant to meet all 5 criteria listed above without a doubt, while introducing many extra functionalities. This study is an open book and inside this document, both front-end and back-end specifications of each functionality have been explained in detail.

However, it should be noted that the wording is not intended for nonprogrammers; therefore, the discourse is very technical.

DOCUMENTATION

Internal Documentation (Comments within the files)

All files involved are heavily commented and comments are written to be both meaningful and helpful to make the codes editable, including even the HTML codes and others that are not directly related to the requirements.

- [`puzzle.php`] file actually has only a few PHP/JS function calls and it can be regarded as the main HTML file. HTML comments are intended to mark the beginnings and endings of elements. Only relatively unusual elements are explained. My approach to inserting HTML codes using scripts is doing so, only when the content is dynamic

The following files can be found at this fancy index: http://www.grifare.net/examples/puzzles/system

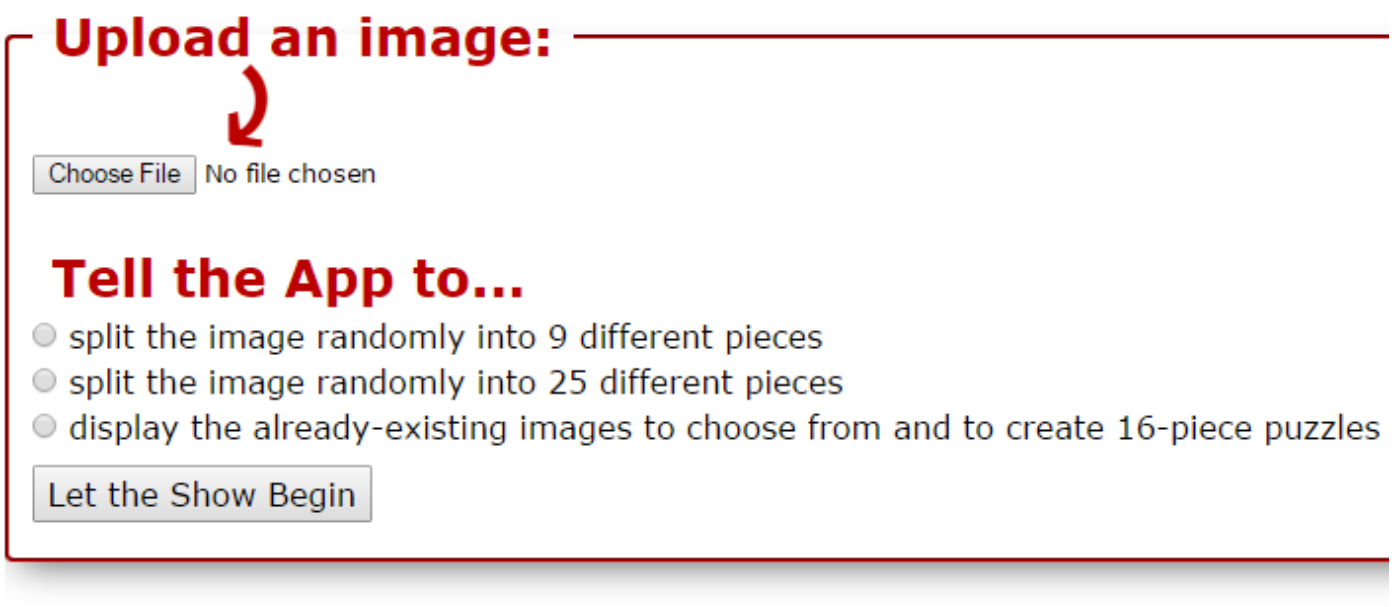
- The page-specific CSS file [`puzzle.css`] is very small in length due to the pre-existing CSS files. Comments inside that CSS file indicate which element is where only, as CSS properties and their values are easy to decipher for any web developer: I do not explain CSS rules
- The JS file [`puzzle.js`] inflated to twice its normal size because of comments. I tried to make them as clear and meaningful as possible for reusability purposes
- The [`puzzle-functions.inc`] file containing PHP functions is crucial and having used JS and HTML codes inside that file frequently, I tried to leave nothing ambiguous there

Please check the comments to see what any particular line of code exactly does.

External Documentation (This document)

Using plainer (but still jargonised) English, this document explains the concepts neither covered by internal documentation nor can be easily discovered by testing the web page.

USER EXPERIENCE



Using this puzzle is not puzzling: understanding what to do takes no time and does not necessitate a tutorial.

The subtitle below reads the 4 most relevant UX elements to this App and list items specify my approaches.

Visual design, Content, Functionality, Interaction design

- Call-to-action is easy to locate and uses more than one property: colour, size and graphics – Visual
- Wording is easy to understand with no language errors – Content
- The presentation page has some 160 words in total and the user has to read a maximum of 100 words to be able to start the puzzle – Content
- When puzzle opens, nothing but the puzzle is communicated – Content
- Starting takes the minimal number of clicks – Functionality
- A link to the image's unshuffled version made available that opens in a separate tag – Functionality
- Replaying or starting new puzzle takes the minimal number of clicks – Functionality
- Works like a charm! – Functionality
- Nothing unexpected or annoying happens (Pop-ups, warnings, redirects to irrelevant pages, legalese, auto-starting apps, disclaimers, terms of use and the like) – Interaction Design
- Clear instructions are dynamically provided throughout the journey – Interaction Design

- Once the puzzle is requested, an initially-hidden section fades in 0.8 seconds to make the user see the game commence, the uploaded image appears already-shuffled, requiring no further user input – Interaction Design
- All other sections disappear to facilitate concentrating on the puzzle – Interaction Design
- Navigation is not tricky: no dead ends. In fact, it all takes place in one page avoiding new page downloads – Interaction Design
- Required input kept at minimum at each stage – Interaction Design
- Stats; such as completion time and number of clicks, have been made available when the puzzle is complete, in order not to distract the user while solving the puzzle – Interaction Design

GLOBALLY-SET STYLING RULES & CLIENT-SIDE FUNCTIONALITIES

There are pre-existing files that empower the entire website, which could be good supplementary sources to refer to.

Styling

Inherits global styling from the existing CSS files. This website's overall design has 3 separate responsive layouts for various screens; from the largest, down to 640px width. Given this website's raison d'être, it has limited content in its mobile version. This is to say that there is no mobile version at all for this specific app.
Existing CSS files are made available at this fancy index:
http://www.grifare.net/css/

Behaviour

Existing JS files are made available at this fancy index:
http://www.grifare.net/js/

TECHNOLOGY and EXTRA FEATURES

Validations

HTML5

- Uses the built-in "required" attribute for both image uploads and radio buttons

jQUERY

- Removes the required attribute from the file input, on third radio select, before button click
- Restores the required attribute of the file input, on third radio deselect, before button click

PHP

- Validates out direct requests to the hidden section by first checking server request method using PHP: that section opens only after valid input submittal
- Validates out empty submissions
- Using regular expressions, checks if the uploaded file has one of the predefined (jpg/jpeg, png, gif) extensions
- Checks if the size of the uploaded file is normal, which is 3mb maximum
- Then checks the dimensions of the uploaded image, too small or too large images are not accepted
- If the input has been validated, the upload happens and the hidden canvas on the same page opens
- If the input has not been validated, warnings issued on the same page and the user is redirected to the presentation page by JavaScript, which in reality, is the same page
- In case the user prefers to start the puzzle using one of the images in the upload directory, without uploading an image, many validations are by-passed, thanks to collaboration of jQuery and PHP functions (Inclusion of this option and integration of validated input with unvalidated input is a notable study by itself and on its own. Please refer to details of this in the test-to-fail part.)
- To facilitate the use of existing images, a dynamically-populated table displays thumbnails and each thumbnail creates a new puzzle on mouse click event

Messages

There is no shortage of auto-generated feedback to user! Almost every event is listened to and new messages are communicated within the existing HTML elements, by adding/replacing/removing content using jQuery. Many of exceptional cases have been defined and handled as well.

Extra features

- Several options not mentioned in the core requirements list
- User-friendly layout and adaptive event-driven messages

- Advanced applications of HTML5, jQuery and PHP
- Armagan is emailed each time "Let the Show Begin" button is clicked and emailed again when a file is successfully uploaded

WEBMASTER TOOLS

https://tools.pingdom.com/#!/cdIQO/http://www.grifare.net/examples/puzzles/puzzle

Summary

Performance grade A **95**

Load time **536 ms** Faster than **97 %** of tested sites

Page size 92.9 kB Requests **15** Tested from **New York City** on Jul 6 at 18:54

Performance insights

GRADE		SUGGESTION
D	64	Remove query strings from static resources

Resources with a "?" in the URL are not cached by some proxy caching servers. Remove the query string and encode the parameters into the URL for the following resources:

- http://www.grifare.net/css/default.css?version=1
- http://www.grifare.net/css/small.css?version=1
- http://www.grifare.net/examples/puzzles/system/puzzle.css?version=1
- http://www.grifare.net/examples/puzzles/system/puzzle.js?version=1
- http://www.grifare.net/js/default.js?version=1

A	93	Leverage browser caching
A	100	Avoid bad requests
A	100	Minimize redirects
A	100	Minimize request size
A	100	Serve static content from a cookieless domain
A	100	Specify a cache validator
A	100	Specify a Vary: Accept-Encoding header

Response codes

RESPONSE CODE	RESPONSES
200	15
OK	

MY COMMENT:

D	64	Remove query strings from static resources

This error points out a deliberate choice I made, This is a website under continuous construction.

TEST-TO-FAIL THE APP

Please black-box test the app and play hard ball on it!

Suggested test cases:

- Directly accessing to the hidden section

http://www.grifare.net/examples/puzzles/puzzle#splitImage

- Empty submissions
- Uploading other file types, such as .exe, empty files, or files without extensions
- Uploading image when third button is chosen

(If JavaScript is disabled, for eg, that means the user actually is not trying to play any puzzle. This of course disables all client-side interactions. But then, PHP will not make any uploads of any type AT ALL, as the function fired when this third radio button has been chosen, has no upload codes.)

- Cross-browser testing (Tested and customised for Chrome, FF, IE (11-10-9) and Yandex)

(The latest 5.1.7 version of Safari for Windows, which dates back to 2012, does not recognise the HTML arrow entity. Hope Apple shows that red arrow correctly.)

- Any other methods I did not consider
- Please note that there is no cap in the target folder for image uploads currently as this is not a real product and as I do not want the tester to face any errors. I will deal with that later, if I decide to keep the App in working condition. That is why I do not see it as a vulnerability for now.

CONCLUSION

This has been a great piece of study for me, during which I have had the opportunity to refresh my knowledge while learning new material. I enjoyed it very much and hope everyone will like it.

APPENDIX

This is the partial list of URLs I found to be very helpful during the study. I visited hundreds of web pages during development but listed only the most useful ones for future reference.

HTML

http://www.w3schools.com/tags/ref_canvas.asp

http://mcdlr.com/8/

http://htmlarrows.com/arrows/right-side-arc-clockwise-arrow/

CSS

https://philipnewcomer.net/2014/04/target-internet-explorer-10-11-css/

https://css-tricks.com/snippets/css/css-hacks-targeting-firefox/

JS

https://msdn.microsoft.com/en-us/library/gg130967(v=vs.85).aspx

https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent/offsetX

https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/offsetLeft

http://code.tutsplus.com/tutorials/create-an-html5-canvas-tile-swapping-puzzle--active-10747

http://code.tutsplus.com/tutorials/quick-tip-how-to-randomly-shuffle-an-array-in-as3—active-8776

http://www.homeandlearn.co.uk/JS/html5_canvas_draw_scale_slice.html

PHP

http://stackoverflow.com/questions/15687363/php-access-global-variable-in-function

http://php.net/manual/en/function.glob.php

.HTACCESS

https://perishablepress.com/better-default-directory-views-with-htaccess/

UX

https://knowledge.hubspot.com/cta-user-guide-v2/call-to-action-best-practices

WEBMASTERS

https://tools.pingdom.com/#!/cdIQO/http://www.grifare.net/examples/puzzles/puzzle