COM1008: Web and Internet Technology

Assignment: Website (80%)

Dr Steve Maddock, s.maddock@sheffield.ac.uk

Part 1: Planning and Design (10%)

Part 2: The Website (60%)

Part 3: The development and testing document (10%)

Deadline: 3pm, Tuesday 13 December (week 12)

Handin: zip file of all your documents via MOLE.

1. Introduction

This assignment will test your ability to create a website using Responsive Web Design. It will test your understanding of stages of creating a website, as well as your coding skills in HTML5, CSS, JavaScript and the Canvas.

This is an individual project. Do not collaborate with anyone else. The work you submit must be your own work.

2. The Website

You will develop a website for a fictitious company that develops web-based children's puzzle games. The following is a list of the requirements for this web site:

- It should have a home page that welcomes people to the website. This page should serve as what is commonly called the 'splash page' for the website, and should say something positive about the company. This page should load quickly, and should be immediately intuitive, as well as being attractive. Invent a logo for your fictitious company. (Hint: This logo should appear on every page of your website.) It is your choice what to put on the home page, but whatever decision you make it should be part of your planning document.
- A page with details of the founders of the fictitious company. This should include photographs and short biographies for each person. Three people, including yourself, will suffice. All the information can be fictitious, so as to match the fictitious company you are creating. Consider carefully how you lay out images (photographs) in relation to surrounding text. (Note: You should make it clear that the profile information is fictitious by including a note to this effect on the page.).
- A news page. Each fictitious news item should be accompanied by a photograph. Pay careful attention to which HTML5 semantic element it would be appropriate to use for news items.
- A recommendations page – this should include recommendations and reviews for puzzle games, which could include links to other websites.
- A separate page that includes the puzzle games your company offers. You should list a range of puzzles (a list of five puzzles will suffice), but you only need to develop one puzzle game using JavaScript and the Canvas. You should choose one from the following list to implement, with details of each given in Section 3:
  - A match-the-pairs puzzle
  - A sliding blocks puzzle
  - A spot-the-difference puzzle

  The game must be interactive and the interaction must be done with the mouse interacting on the canvas. (Note: this is perhaps the most difficult part of the assignment, so you may wish to concentrate on other pages before tackling this.)
- A contact page, which includes the fictitious contact details of the company (e.g. address) and a form for users to send comments to you via your e-mail address. A user should input their e-mail address in a text box in the form, and input their comments in another text area on the form. The form should also include a button, which, when clicked by the user, sends the contents of the two text areas to your company's e-mail address. (Note: Since the contact details for the company are fictitious, the form should be sent to your University e-mail address.) Make sure the form is stylish – you are again demonstrating that your company can make a form look good.
- An accessibility page. On this page you should give the accessibility statement for the web site. (You may have addressed accessibility in a number of ways on the website and if so you should state that on this page.)

You must satisfy the following when constructing your website:

- The overall website design must be consistent.
- The website must be legible, e.g. is there good contrast between text and background? What happens when stylesheets are turned off?
- The website must take into account accessibility issues.
- The <head> element must include an element identifying the author.
- You must use the HTML5 semantic elements when structuring each webpage. These must then be styled in the relevant stylesheet(s).

- A navigation area must be included for the Web site.
- Appearance must be controlled by the linked stylesheet(s), *not* by inline styles.
- Use of @media queries
- Economic use of properties in a stylesheet, e.g. margin a b c d, rather than setting the top, right, bottom and left margin separately
- (*Note*: You should link the same stylesheets to all of your pages. However, for the JavaScript Canvas demo page, you may need an extra stylesheet for any specific effects you use on that page.)
- The HTML, CSS and JavaScript code should be well organised and neatly laid out, e.g. using *indentation*.
- You must include comments in your HTML, CSS and JavaScript files. As an example, a comment might describe the purpose of a particular CSS rule. We will look at these comments carefully since the comment text will be unique to you and reflect your understanding. However, do not comment simple pieces of HTML, CSS or JavaScript where the meaning is obvious – use your common sense here. Remember to include comments in the <head> element that state that *you* wrote the code.
- (*Note*: If you want to use videos on your website, then put them on youtube and just put a single image and a link to the youtube video on your web site. Do not include the video on the web site itself, as this will make it impossible to hand in the final work – see Section 3.)

3. The puzzle game

(Warning: Whilst the puzzle game will be fun to implement, please do not spend inordinate amounts of time on this. Check out how many marks this is worth in Section 8. Marking.)

3.1 Match-the-pairs

Multiple copies of the same image (e.g. the company logo or some other image) should be displayed on the canvas as a 2D grid of images – a 4x4 grid of images, giving a total of 16 images, will suffice. We can refer to these as 'cover images'. When the user clicks on a cover image it is replaced by something else (a 'thing'), e.g. a picture of an animal, or a number, or a word. The user then clicks on a second cover image and it is also replaced by a thing. If the two things match, then the things stay displayed on screen. If the things do not match, then the cover images are redisplayed.

The game finishes when all the pairs of things are matched, so that no cover images are displayed. When the game finishes, there should be a short animation effect involving a congratulations message, e.g. the message flies onto the screen or appears and flashes in different colours. There should be a button that can be pressed to start again with a new randomised arrangement of the cover tiles.

3.2 Sliding blocks puzzle

This should be a 3x3 sliding blocks puzzle game (https://en.wikipedia.org/wiki/Sliding_puzzle). A 3x3 grid of cells is created, in which eight cells have pictures in them and one is empty. Clicking on a cell next to an empty cell causes the picture tile in that cell to move into the empty cell. When all the pictures tiles are in their correct cells, they create a complete picture across the 3x3 grid.

Once all the picture tiles are in their correct position, the empty cell should turn into a picture tile that completes the picture across the 3x3 grid.

When the puzzle is complete, there should be a short animation effect involving a congratulations message, e.g. the message flies onto the screen or appears and flashes in different colours.

There should be a button that can be pressed to start again. Each time the game starts a different final picture is used. (You only need to implement three different pictures.)

3.3 Spot the difference

An example is given at:
https://en.wikipedia.org/wiki/Spot_the_difference

Two pictures should be displayed on the screen – either line drawings or images. The second picture should have five differences from the first picture. The user must use the mouse to click on or near to the differences. If the user clicks in the right position, s/he should be rewarded with a short animated effect, e.g. a pulsating circle around the difference.

When the puzzle is complete, there should be a short animation effect involving a congratulations message, e.g. the message flies onto the screen or appears and flashes in different colours.

There should be a button that can be pressed to start again. Each time the game starts a different pair of pictures is used. (You only need to implement three different pictures.)

4. Part 1: Planning and Design (10%)

The Word or pdf document for this part should be between 500 and 1000 words long, and may include as many images as you wish to use.

You must create a document that describes how you planned and designed the website using 'Responsive Web Design'. This document should be completed before implementation starts. It should *not* include screen shots of your implementation.

(Note: Requirements is given in section 2 above, so you don't need to write about this.)

You must include the following in separate headed sections:

- General ethos: A brief statement about the general ethos behind your design and why it suits the requirements, which is for a site for children, focussed on children's puzzles.
- Site Map: draw the site map for your website and justify the structure.
- Accessibility: describe how you address accessibility issues on your website
- Design mock-ups: You must use a mobile-first approach for the assignment. Create design mock-up diagrams that show, as a minimum, the mobile design and the desktop design. You could use wireframes for this, or you could use paint software (e.g. Photoshop) or even hand-drawn sketches (that are then scanned in) to produce diagrams similar to the following examples: The one labelled "*The "extreme" versions of the new website design"* in the following article: http://www.smashingmagazine.com/2013/03/building-a-better-responsive-website/; The one labelled "Normal, Narrow, Mobile" in the Overview section at http://webdesignerwall.com/tutorials/responsive-design-with-css3-media-queries/comment-page-1
  You must decide how many breakpoints to use in your design and write a short justification of this. (Note: the exact position of the breakpoints could be changed during the development stage, and you may also add a few tweakpoints.) Justify your design decisions.
- Menu System: A consideration of the menu system that is being used, e.g. consider the articles at http://responsivenavigation.net/index.html and http://bradfrost.com/blog/web/responsive-nav-patterns/ and give a justification for the menu system you decide to use.
- Puzzle Game Design: A design for the puzzle game you intend to implement. How will this work on the mobile version of the site? Will it work, or is it just for desktop users? Will it consider 'accessibility'?

In discussing each of the above things, I expect you to justify your decisions using references to relevant websites. For example, the following website should be referred to: https://developers.google.com/webmasters/mobile-sites/

You only need to consider the last two major versions of web browsers in your planning process. Do not consider old browsers.

You should include plenty of diagrams in your document (and each figure should be numbered and have a caption), where the pictures can be general illustrations (e.g. a design hierarchy or a page layout diagram). Pictures can be hand-drawn and scanned in, or can be produced using appropriate software tools.

Surprise me with your own comments on different aspects of the process. The aim of this document is to make sure you have thought about the process of planning and designing a website and carried out the process in a structured way.

5. Part 2: The Website (60%)

The Website should include all the requirements given in Section 2 and should match your design document. If you make any changes to the design, these can be discussed in Part 3.

6. Part 3: The development and testing document (10%)

The Word or pdf document for this part should be between 500 and 1000 words long, and may include as many images as you wish to use.

The development and testing document should cover the following, each in headed sections:

- Changes: Has the design changed? If so, briefly describe the changes and why you made them.
- Organisation: Did you create templates for the HTML and CSS? If so, why, and how did you use them? Consider the use of CSS reset or normalize – why would you use these? How did you organise the file structure for the website? What debugging tools have you used? Discuss.

- Optimisation: Have you considered optimisations (e.g. image loading times)? Discuss.
- Security: For the contact page, which contains a form for users to complete, discuss any security issues that you think are relevant and how you deal with these. Also discuss any other relevant security issues for your website.
- Testing: Tests on different devices and different browsers – you only need to consider the last two major versions of web browsers in your testing.

(Note: I am not including 'delivery' as part of the assignment, since that is just handing the assignment in through MOLE.)

You should include plenty of diagrams in your document (and each figure should be numbered and have a caption), where the pictures can be general illustrations, as well as screen shots from your own website (e.g. screen shots of tests on different devices).

Surprise me with your own comments on different aspects of the process. The aim of this document is to make sure you have thought about the process of developing and testing a website and carried out the process in a structured way.

7. Handin via MOLE

There are three separate parts to handin, via the assignment link on MOLE. All three parts should be put in a single zip file called *name*.zip, where *name* is your name, e.g. JohnSmith.zip. This zip file should contain the following:

- Part 1. The planning and design document. This should be a Word document called design.doc or a pdf document called design.pdf.
- Part 2. This is the website itself. The home page of the website must be index.html, so that it is easy for us to identify which file to load first. Make sure you include every file, including all relevant images. (Remember: using relative addresses on your website for the links between pages and resource files such as images is important so that the website can be easily copied onto a different server.) Do not include videos, as including these will create a large zip file, which will crash MOLE – there is previous experience of this and it wasn't pretty. If you want to use videos, then put them on youtube and just put a single image and a link to the youtube video on your web site.
- Part 3. The development and testing document. This should be a Word document called developandtest.doc or a pdf document called developandtest.pdf.

Note: Remember to include comments in each and every HTML and CSS file to identify that *you* wrote the code

8. Marking

8.1 Part 1 (10%)

You must include each of the things asked for in the above specification. Justifications should be given for each part, but these should be brief as there is a maximum word limit.

The accompanying description should give reasons for choices, e.g. the discussion of the site map should not describe what the site map diagram already shows. Reasoning is more important. Justify accessibility statements by citing references. The design mock-ups should be neat and reasons for the breakpoint(s) given.

8.2 Part2 (60%)

The majority of marks are for producing a website that fulfils all the requirements. Read them *carefully*.

- General (15%) – includes look & feel, RWD behaviour, content, turning features off, use on different browsers;
- HTML (15%) – includes comments, layout, content of <head> element, semantic elements, menu, validation;
- CSS (15%) – includes organised, layout, comments, economic use of properties, RWD, @media, validation;
- Puzzle game/JavaScript (15%) – includes comments, layout, code structure, completeness of game, creativity.

8.3 Part 3 (10%)

You must include each of the things asked for in the above specification. Justifications should be given for each part, but these should be brief as there is a maximum word limit.

9. Practical considerations

9.1 Keeping your work private

Your website should be developed in a local folder in your CiCS managed desktop file space, *not* in your Department mypublic_html folder (visible on the Department's Web server). If you develop the

website in your mypublic_html folder, it will be visible to the world and all other students will be able to see your work and have the opportunity to copy it.

When your work is marked and returned to you: you can copy your Web site to your Department mypublic_html folder (see instructions in the lab class for setting up a network link to the mypublic_html folder from a CiCS machine) and thus make it visible to the world via the Department's Web server. This is similar to the standard process of developing a Web site offline and then uploading it to make it visible online. By making it public *a few weeks after the handin deadline*, other students will be able to see your work, and to learn from what you did. You can also advertise the link to other people anywhere in the world.

9.2 Relative links

Make sure you use relative URLs in your HTML code when referring to your own resources or to other HTML files that you are developing – the reasons for this are explained in lectures. Links to external URLs should still be given in full, e.g. http://www.w3.org/.

9.3 Unfair means

The standard Department rules for use of unfair means will be applied:

http://www.dcs.shef.ac.uk/intranet/teaching/public/assessment/plagiarism.html

I am aware that there are lots of HTML, CSS and JavaScript tutorial sites on the Web (including websites that implement puzzle games). Do *not* copy them, since that would be plagiarism.

9.4 Code reuse

Do NOT use Bootstrap or any other similar frameworks for creating web sites.

You may reuse HTML, CSS and JavaScript code that I wrote that is given in lecture notes, as long as it is not code that is from another source that is being used to illustrate something – I may have used it to illustrate something, but you will not have permission to reuse it. If there is any doubt, then assume you cannot reuse it.

If you are using CSS reset or normalize, you need to make sure you comply with the license for each of those and make clear that they are not your work.

9.5 Text editor

There are plenty of Web design tools available. The expectation for this assignment is that you will use a text editor to develop your website. You might use more sophisticated tools to support your design process, but not to develop your code for you – typically, it is obvious when one of these sophisticated tools has been used in code development as the files created contain lots of extraneous HTML and CSS, rather than only including what is required. As part of the deliverables you will have noted that I expect code that contains detailed comments. These will demonstrate whether or not you understand the HTML, CSS and JavaScript that you have produced.