Unit 1 Reflection

I've completed my first unit on how to make a website. I was surprised at how much pre-work has to be done in order to code. I've seen a lot of Hello World tutorials but never to start with the purpose, personas, flows, site map, and mocks. My excitement has been suspended until Unit 2. I originally only had one persona in mind when I came up with my website idea, realizing that it may be useful for others was interesting to explore.

If I had to do it again, I would have spent more time choosing an appropriate scope for the site. It is difficult to think of something that is broad enough to be interesting and will be extensible, but narrow enough that I can create or find enough content to fill it up in a useful way. I think it would be easier to create a website for someone else, because they would be able to answer all of the questions this unit has and I could focus on translating their vision into a website. I have a sales/business background and being able to get into the head of other users was fairly simple.

I'm glad that I have all this structure now, all I have to do is learn how to turn my plan into a reality now!

Learning Outcome:

Apply a structured approach to identifying needs, interests, and functionality of a website.

I've identified the needs of my users by creating their personas, what they're looking for, and creating flows of how they would access that content. The website is based on my personal interest and solves the problem of aggregating content across different media types and platforms. The functionality of the website is outlined in the wireframe images and mockup.

Unit 2 Reflection

http://student.athabascau.ca/~zacharyle/

What aspects of these tasks were most difficult for you, and why?

Creating the HTML without CSS at the same time was rather difficult. I had to envision all the CSS I would do in the future when I wrote the HTML.

If you had to do the tasks again, what would you do differently, and why?

If I had to do the tasks again, I would create less pages with more content. I would also use a server-side scripting language to help me with templates. Not having the header and footer as separate files included in the main page is a lot of work.

How did your previous experience help and/or hinder you in completing the tasks?

My previous experience is mostly using Bootstrap as a guide for my HTML, it has made me dependent on other people's optimized code.

What was the most surprising thing that you learned?

Apparently the tag was not deprecated until HTML5.

What was the most useful thing that you learned?

While I was getting content for the site, it was reinforcing the idea that organizing the media (movies, documentaries, and videos) would be really helpful. One of the top YouTube searches was "Roman History Documentaries".

What did you learn about yourself as a result of completing these tasks?

I am easily frustrated by large amounts of content creation. It's taken me months of procrastination to finally get around to filling the site with content.

In addition to the pages themselves, you should provide (in your learning diary) a critique of the templates you have been given, explaining why they were poorly written and how you have improved them.

**Links to sample html pages available only from home page of the site in the menu.

Basic starting and ends tags were mismatched. This can be corrected easily by writing , then filling in the middle rather than something … and hoping to remmber the at the end.

Make sure you start with the basic template of <html>, <head>, and <body> tags before you start filling with content.

Tag attributes can be frustrating to remember all the quotation marks, easy to avoid this if you do attribute="" instead of attribute="something … and hoping to remember the ending ".

In your reflective learning diary, you should also provide an explanation of how your pages fit with the needs of the personas and scenarios you identified in Unit 1.

Use of Tags

Image - Index page
Menu - Yes
Hyperlink - Yes, to videos on timeline page
List - Sidebar left on categories page
Divs - Categories page
Span - Timeline page for civilizations and also footer
Table - Timeline page
Form - About page

Personas Used

Persona #1 Horatio the Hobbyist, Scenario #2 and Scenario #6

Horatio has videos to watch embedded into the site on the Categories Page, links to movies he can download on the Timeline Page, and high level descriptions of categories and leaders of Rome if he wants to go deeper.

Persona #2 Silvia the Student, Scenario #3 and Scenario #4

All the sources have been included so that Silvia can read high level information about different topics on the Timeline or Categories pages and go deeper or cite correctly.

Persona #3 Prem the Professor, Scenario #5

There's tons of media on the Gallery and Timeline pages that Prem can use in his classroom to engage his students.

Persona #4 Quinn the Questioner, Scenario #1

Quinn is using a search engine to find information, all of it sourced so that he can go deeper if his question isn't answered right away.

Unit 3 Reflection

http://student.athabascau.ca/~zacharyle/
CSS tested in latest Chrome and latest Safari

Record your observations in your Learning Diary blog. Make the relationship of you Unit 3 activities with your Unit 1 plan explicit.

Personas Used

Persona #1 Horatio the Hobbyist, Scenario #2 and Scenario #6

All of this information has been presented in a more attractive, more trustworthy way so that Horatio can enjoy the site and not just use it as an information source, it can be part of his learning experience. The media content of images and videos is centered and tiled to make them easier to watch and view.

Persona #2 Silvia the Student, Scenario #3 and Scenario #4

Quinn is using a search engine to find information, all of the sources are very easy to find because I made them into buttons, so you can dig deeper easier on the Timeline or Categories pages and go deeper or cite correctly.

Persona #3 Prem the Professor, Scenario #5

I made the site visually appeal and emphasized the images and videos by centering them and made them visually appealing so he can use in his classroom to engage his students.

Persona #4 Quinn the Questioner, Scenario #1

Quinn is using a search engine to find information, all of the sources are very easy to find because I made them into buttons, so you can dig deeper easier.

Map your Unit 3 activities to the appropriate outcomes.

Some Examples of CSS Used

- Custom font style used
- Custom font colors
- Custom borders, used for Categories Page nav and also Timeline markers
- Lots of positioning with columns (float, clear, height, etc) and for centering (images, videos)
- Images for backgrounds
- Pseudo classes for buttons
- Various selectors used as appropriate
- And more!

HTML Changes

- Added a column in the table for the timeline to hold the markers
- Added various classes and ids to elements

Questions

What aspects of these tasks were most difficult for you, and why?

The columns were quite difficult to build from scratch. I'm used to using Bootstrap for my columns. I had trouble floating the columns left and keeping the row height proper. Solved it with overflow auto tho.

If you had to do the tasks again, what would you do differently, and why?

I would have added the classes and ids before doing the CSS, it was kind of a pain to go back and add them all in.

How did your previous experience help and/or hinder you in completing the tasks?

I've been doing front-end development for years, however, I had a huge reliance on Bootstrap and other various front-end frameworks so I knew all the CSS selectors, classes, etc. to use, but it has made me dependent on the frameworks.

What was the most surprising thing that you learned?

The quiz had some questions with more complicated selectors using wildcards and whatnot. I typically stick to the standard selectors and change my HTML for better readability, but it was a nice refresher to see the other selectors again.

What was the most useful thing that you learned?

I learned that there are now only 3 main border-radius CSS properties which is nice.

What did you learn about yourself as a result of completing these tasks?

I learned that I can create a basic CSS template from scratch which is great. Helps for more difficult positioning to redo all the work form the frameworks.

Unit 4 Reflection

http://student.athabascau.ca/~zacharyle/
CSS tested in latest Chrome and latest Safari

Code Explanation

The snippet takes in a selector and changes all HTML elements of that selector to do a rollover effect on hover. It does this by looking for a data-title attribute in span class inside the selector, and also by dynamically adding content (class="roll") to the spans. It changes the text of the selector to the data-title value.

** This effect is only on the Categories page

The code didn't come with any comments, so I added them below.

```
/*
source: http://jsfiddle.net/hakim/Ht6Ym/
author: hakim
date: 20140723
*/
// on window load
window.onload=function(){
	// set true if the browser supports the perspective alteration
	var supports3DTransforms =  document.body.style['webkitPerspective'] !== undefined ||
document.body.style['MozPerspective'] !== undefined;
	// function to transform elements
	function linkify( selector ) {
		// if browser supports transform, else print failure to console
	  if( supports3DTransforms ) {
		// get all elements that match the selector
		var nodes = document.querySelectorAll( selector );
		// for each element
		for( var i = 0, len = nodes.length; i < len; i++ ) {
		  var node = nodes[i];
		  // if the class doesn't already roll
		  if( !node.className || !node.className.match( /roll/g ) ) {
			// dynamically add roll class to selector elements
			node.className += ' roll';
			/*
			uncomment next line to roll all elements that match selector, otherwise you'll
have to wrap text with
```

```
                    <span data-title="roll text"></span>
                    */
                    //node.innerHTML = '<span data-title="'+ node.text +'">' + node.innerHTML +
'</span>';
                }
            };
        } else {
            console.log ("failture, browser doesn't support 3d");
        }
    }
    // set which html elements you want to have scroll effect
    linkify( 'h1' );
    linkify( 'h2' );
    linkify( 'h3' );
}
```

Provide a critique of that code, indicating how and why it is good and/or bad.

- First of all the code wasn't commented at all, so that's not very good.
- I also added a console log if the browser doesn't support the effects for debugging purposes.
- The code wasn't called properly at first because it didn't have a window.onload=function(){}, so I added it.
- The code is pretty rad, it uses a cool effect that's not frequently used.
- The code worked pretty well, it was posted on js fiddle so having a barebones demo is always great.
- The code is rather elegant
- The code has some basic error catching which is great and degrades gracefully
- It was originally set up to affect all elements of the selector type, but I changed it so I had to explicitly state which elements I wanted it to work on in HTML.
- It was originally set up to do the animation with the same text as in the selector, but I changed with so that I could specify different roll-over text, in this case, I translated the words to Latin.

In your reflective learning diary, explain how the code improves the experience of the personas you created in Unit 1, and how it helps with the scenarios you presented.

Personas Used

Persona #1 Horatio the Hobbyist, Scenario #2 and Scenario #6

All of this information has been presented in a more fun way, hopefully to differentiate from the other sites out there with lots of information. It's a more engaging way to entertain Horatio because it had Latin, the language Romans spoke.

Questions

What aspects of these tasks were most difficult for you, and why?

Even copying and pasting from JS Fiddle you can get errors for browser-specific problems. In this case, debugging the windows onload was an issue. It's one I ran into before so no big deal, but frustrating.

If you had to do the tasks again, what would you do differently, and why?

I would have looked around at some forks of the code to get a better version for what I was trying to do with it, I had to change it a little to do what I wanted.

How did your previous experience help and/or hinder you in completing the tasks?

I've done lots of javascript work before, it made it really easy.

What was the most surprising thing that you learned?

There's a perspective library for mozilla and webkit based browsers. Very cool.

What was the most useful thing that you learned?

Most useful is that on JS Fiddle, things can still go wrong when copying and pasting.

What did you learn about yourself as a result of completing these tasks?

I usually go for a minimalistic design, the less bells and whistles the better, but in this case, I think it improves user experience for one of my users - might be a good idea to use it more.

Unit 5 Reflective Journal - Javascript Proposal

Based on your personas and scenarios developed for Unit 1, in your learning diary identify ways that JavaScript elements might improve your site. It is important that the code you develop meets the learning outcomes for this unit as well as being justified and useful in the context of the site.

This was submitted when I submitted my proposal. I mapped out the pseudo code and everything while it was all in my head so I wouldn't forget.

Idea #1

Gladiator Defence

The concept is that a gladiator is attacking you, and you have to defend yourself. You're shown an image of a gladiator attacking you for 2000ms, and you have to click on the appropriate defence or counter move before the time runs out. You have to defeat your enemy or you lose. This is meant to excite Horatio the Hobbyist in Scenario #6 to get him more engaged in Roman culture.

Variables

```
var attackType
var defenceType
var battlePairs =
{"stab":
        {"block": "shield",
        "counter": "sweeplegs",
        "image": "images/stab.png"},
"highswing":
        {"block": "sidestep",
        "counter": null,
        "image":"images/highswing.png"}
} etc.
var enemyHealth
var playerHealth
var reactionTimer
var battleOver
```

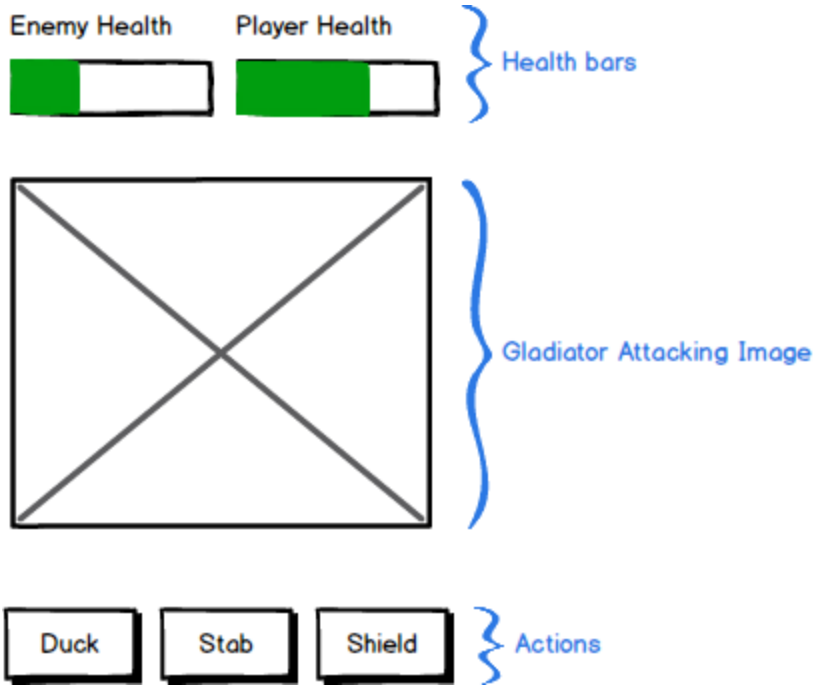
Add #gladiator-img id for the image of the gladiator
Add .action class to buttons
Gladiator health bars done with CSS classes. A div inside another div. A health bar having 33% width .health-1 with background green represents 1 health of max 3.

Battle Function

Every second until battleOver is true
        Pick a random key from battlePairs, set to attackType

Render the image to the id #gladiator-img
Render 3 defenceType buttons, the block and counter, and another random option.
switch
case: the defenceType is the value of the block inside the attackType object, no-one loses health
case: the defenceType is the value of the counter inside the attackType object, enemy loses -1 health
case: if the defenceType is neither value of the block or the counter of the attackType object, player loses -1 health
default: (the timer runs out before a defence is clicked), then player loses -1 health
render health classes for gladiator and player
1 health = .health-1 = healthbar width: 33%

Mockup



Idea #2

Currency converter

The high-level idea is that you are exchanging your money (in CAD) to the Roman currency at the time, Denarii. The converted uses an exchange rate to determine how much you'll have to spend. This provides some classroom content for Prem the Professor in Scenario #5 and engages Horatio the Hobbyist in Scenario #6 by making Roman history relevant to his life. The tool will be on the tools page.

http://student.athabascau.ca/~zacharyle/unit4/tools.html

Variables

var exchangeRate
var startAmount
var startCurrency
var endCurrency
var endAmount
var errors = []

On #convert-button click

if validation function returns true, run convert currency function
else loop through errors array and insert errors array strings into an unordered list #currency-errors

Check Validation Function validation(startAmount)

switch
case: amount is not a number, return false and add “Please input a number” to errors array
case: amount is < 0, return false and add “Please input a number > 0” to errors array
case: amount has > 2 decimal places, return false and add “Max 2 decimal places” to error array
default: return true

Convert Currency Function

if startCurrency is CAD
endAmount = startAmount / exchangeRate
endCurrency = denarii
else
endAmount = amount * exchangeRate
endCurrency = CAD
Insert into id #end-amount the endAmount
Insert into id #end-currency the endCurrency

Idea #3

Tabs for Categories Page

The tabs are for presenting the content in a neater way. This is useful for consuming large amounts of content as you are more focused on the content that is revealed. The user can thus learn more. It would help the persona #2 Silvia the Student in Scenario #4 be able to get a better general grasp of the information because she’d be more likely to read it that if the page were really long.
http://student.athabascau.ca/~zacharyle/unit4/categories.html

Flow

Give tab buttons a class of .tab-button

Give tab buttons ids relating to their names #tab-history
Give tab contents a class of .tab-content
Give tab contents ids relating to their names #content-history
On .tab-button click, run an anonymous function:
    get id of element var tabName = #tab-history
    split strings tab-history by - into an array var tabNameSplit = [“tab”, “history”]
    concatenate string “#content-” + tabNameSpit[1]
    display:none all .tab-content by adding class .hidden (CSS .hidden {display:none})
    display:block specific tab-content with id #content-history by adding class .visible (CSS .visible {display:block})
    add class .active to #tab-history for aesthetics

Unit 6 Learning Diary

In your learning diary, **reflect** on the process, the obstacles, and difficulties you faced, what you might do differently in future as a result, the strengths and weaknesses of using JQuery as opposed to coding it all yourself, and anything else you deem relevant or interesting about the process.

Reflection

I love jQuery, it's such a massive time and headache saver. Docs are great, example code is great, lots of dependent libraries that achieve specific tasks. My task's biggest issue was browser-differences in terms of copying to the user's clipboard. The jQuery-dependent library I used and modified used a Flash object overlay in order to achieve functionality in all browsers. There's still no perfect solution for this, as Flash is not supported on iOS. I had a ton of links I wanted to apply my code to, jQuery's advanced selector function was super useful in apply this functionality to all of the links, doing that in JS would have been a huge pain.

jQuery Strengths

Better syntax

$(".foo") vs document.getElementsByClassName('foo') and iterating over an array

Pre-built functions

Awesome for things like iterating .each()
Saves you a monstrous amount of time for more complex projects. This is the biggest benefit.

Browser differences

Ajax function in jQuery covers all browser differences

Mixing Code

It's fine to mix JS and jQuery which is great.

Better Docs

I find the docs and examples better than normal JS.

Dependent Libraries

Using jQuery opens you up to using tons of other jQuery-dependent libraries that achieve specific tasks. Super great for that.

jQuery Weaknesses

More overhead

Super minimal overhead, especially if you're using the minified version of jQuery, but overhead regardless.

Legacy examples

When Googling, you may stumble on older or newer code examples of jQuery library. Gotta make sure to check the dates.

CDN support

Always a big fear that CDNs may go down in the future. It'll just get cheaper to host, so I doubt they'll ever go down.

Learning curve

It's a new library, so you have to learn the functions and everything so you'll lose some time at the start.

Magic / Obfuscation

Unless you read all the docs, you're relying on some level of mystery to complete your tasks. This means if you run into a quirk debugging, you can't as easily trace-back because you can't know how every jQuery function works.

Unit 7 Learning Diary

As always, don't forget to write your reflections on the process, challenges, and things you found about yourself in your reflective learning diary.

Reflection

What aspects of these tasks were most difficult for you, and why?

Choosing between which jQuery functions to use .ajax, .get, .getjson, etc. was hard. Wasn't sure which one was the best in this scenario. I went with .getjson. Googling wasn't very helpful most of developer's problems were around specific use cases.

If you had to do the tasks again, what would you do differently, and why?

I wouldn't do anything differently really, maybe mashup some more data. The solutions for solving my user's problems were fairly simple.

How did your previous experience help and/or hinder you in completing the tasks?

All AJAX calls are pretty much the same. I still struggle with dealing with CSFR problems. There's a couple solutions like returning the data wrapped in a an auto-executing JS function and some APIs have special params for it, but I don't understand how it works still.

What was the most surprising thing that you learned?

I learned more about CSFR with AJAX, but that's about it, still haven't figured it all out.

What did you learn about yourself as a result of completing these tasks?

I've learned what most developers learn after a while which is that if you write great code, it's going to save you tons of time in the future when you need to do something similar. Comparing my two AJAX functions, I was largely able to use the same functions and everything, just had to tweak for post-AJAX formatting and the API URL formatting.