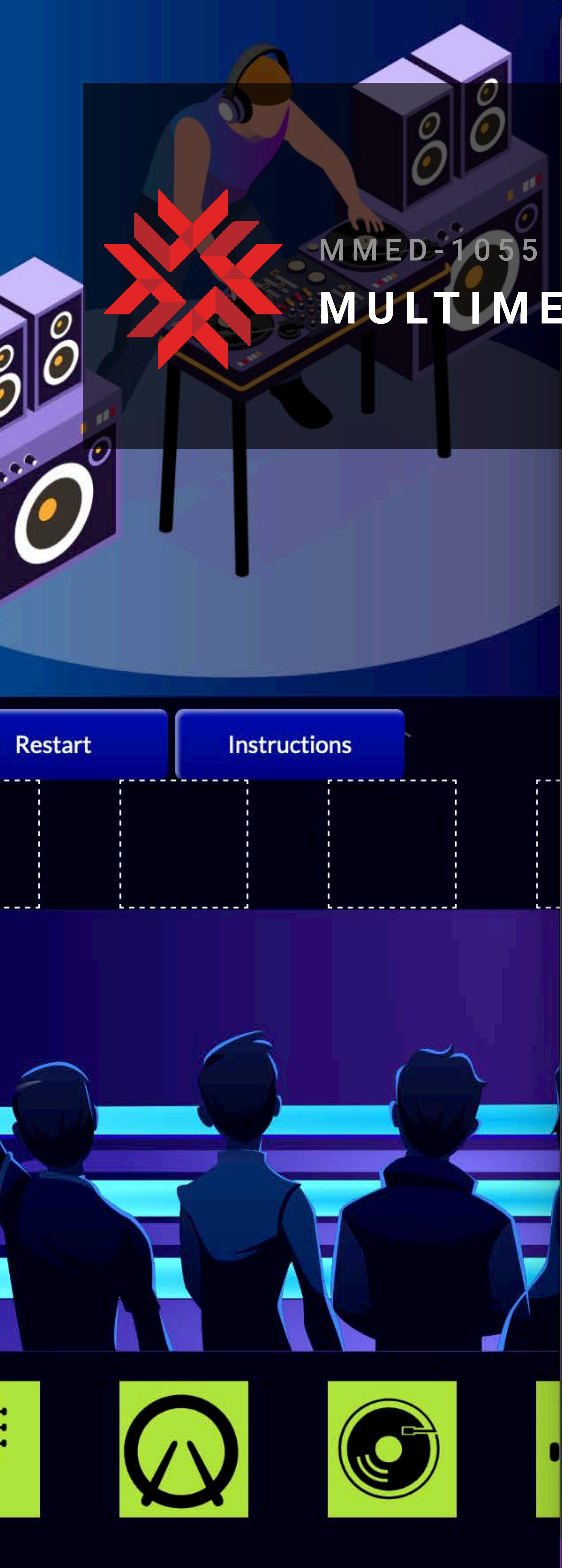
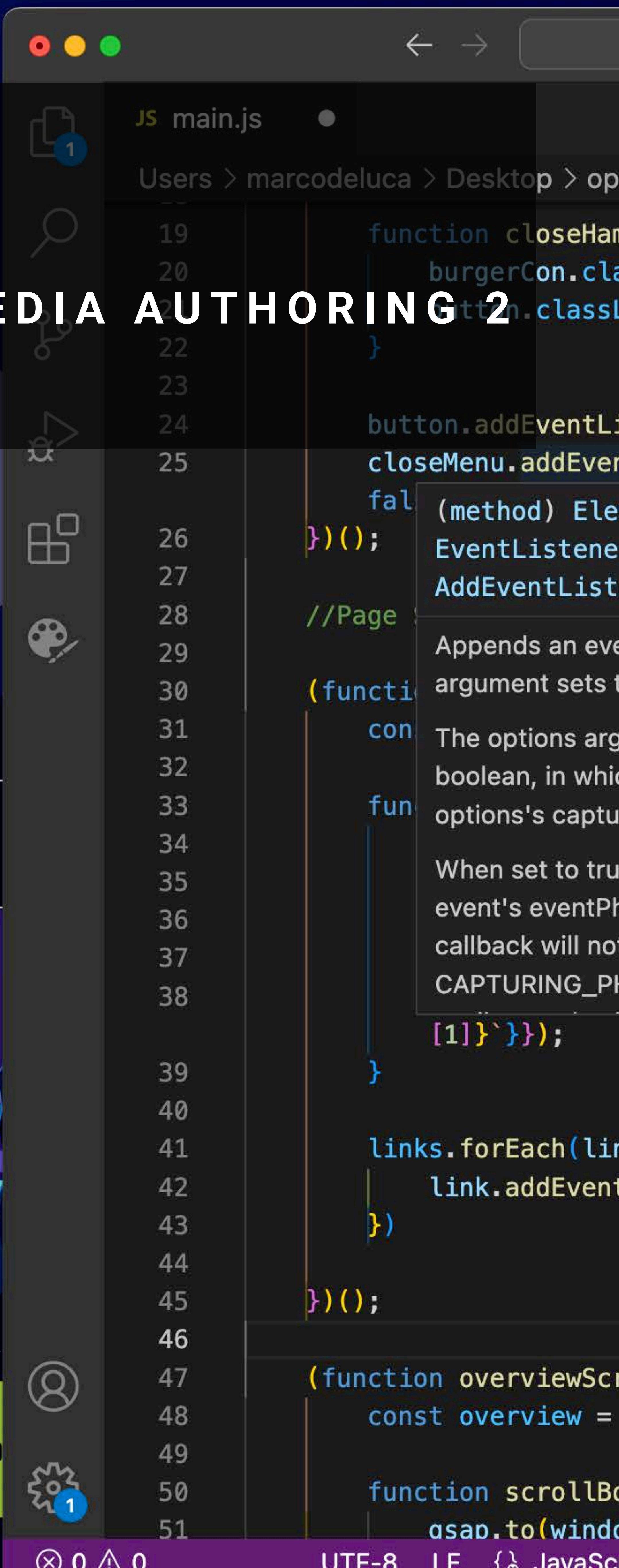
MMED-1055

MULTIMEDIA AUTHORING 2





```
function closeHam
    burgerCon.cla
    butt n.classL
}

button.addEventLi
closeMenu.addEven
fal
})();

//Page

(functi
    con

    fun



                                        
    [1]}`}});
    }

    links.forEach(lin
        link.addEvent
    })

})();

(function overviewScr
    const overview =

    function scrollBo
        gsap.to(windo
```

(method) Ele
EventListene
AddEventList

Appends an eve
argument sets t

The options arg
boolean, in whic
options's captu

When set to tru
event's eventPh
callback will no
CAPTURING_PH

TABLE OF CONTENTS

MMED-1055

COURSE DESCRIPTION

This course will continue to explore current tools and techniques introduced in Authoring 1. Students will be introduced to the fundamentals of programming with JavaScript and then move to Front-End JavaScript-based interactive media.

COURSE LEARNING OUTCOMES

Upon successful completion of this course, you will be able to reliably demonstrate the following Course Learning Outcomes which will be taught and evaluated:

1.) Use JavaScript programming fundamentals to solve basic problems;
2.) Use graphic assets as components as well as plan interactivity and UX/UI at the design stage;
3.) Use JavaScript to create dynamic content and interactivity on a web page;
4.) Leverage browser development tools and apply debugging techniques through the browser console;
5.) Explore and implement best practices when developing rich media applications (authoring JavaScript, HTML5, CSS3)
6.) Apply animation through the use of JavaScript libraries;

GRADE BREAKDOWN

Assignments/Projects - 70%
Final Integrated Project(FIP) - 30%

COURSE BREAKDOWN

MODULE 1 - SVG’S AND JAVASCRIPT FUNDAMENTALS

WEEK 1 * *

- Course Introduction & Intro to Stack
- Music Mixer Project Assigned (Due week 12)
- FIP Brand Revival Assigned (Due Week 14)

WEEK 2 *

- Intro to SVG and Responsive Design
- Interactive SVG Icons Assigned (Due Week 5)

WEEK 3 *

- Intro to Javascript, the DOM, variables, functions, Querying the DOM
- In Class Assignment (Based on class content) 5%

WEEK 4

- Events, Attributes, functions, dynamic content, the dev console

MODULE 2 - JAVASCRIPT INTERACTIVITY

WEEK 5 * *

- Intro to Drag and Drop
- Interactive SVG Icons Submission
- Crushing Bugs Assigned (Due Week 8)

WEEK 6 *

- Drag and Drop Continued
- Music Mixer Checkpoint

WEEK 7

- Logic and Debugging

MODULE 3 - DYNAMIC CONTENT AND MEDIA

WEEK 8 * *
- Intro to working with Media (Audio)
- Virtual Jukebox Build
- In Class Assignment (Based on class content) 5%
- Crushing Bugs Submission

WEEK 10-11
- Dynamic Asset Creation, working with data

WEEK 12 *
- Content Review, Strategies for the FIP
- Music Mixer Submission

WEEK 13
- Open Lab

WEEK 14 *
- FIP Submission and Presentations

WEEK 15
- No Classes

ADDITIONAL LEARNING MATERIALS

Javascript Essential Training

Javascript as a Second Language

Learning GitHub

Additional Reference

Javascript Reference

LinkedIn Learning is available for free through https://fcsw2.fanshawec.ca/portal/index.htm. Look for LinkedIn Learning under My Bookmarks and Campus Links on the right side of the page. You will be prompted to login, using your FOL login credentials to access LinkedIn Learning. You will need to login first before you can click or copy and paste the above links.

HOMEWORK BRIEFS

HW #1 – INTERACTIVE SVG ICONS

Each Student must build a branded, scalable themed icon set using Adobe Illustrator. Export each icon from Illustrator as an SVG. Use the SVGs as assets on the FIP/Brand Revival site (think icons for different flavours or ingredients). Add some simple interactivity with JavaScript. Use HTML (and the appropriate elements) to include a description of the icon set.

Assignment Requirements

Use Terminal, Git and Github to create the project structure and a linked repository. Create a minimum of 6 SVG icons in Illustrator. Use CSS to organize the icons on an HTML page (products)in your FIP. You can choose how you want to implement this - as an image tag source, as an SVG tag, or using the object tag.

Use CSS interactions or transitions to change the icon's appearance (on hover, as an example, reverse all of the colours or add a drop shadow, etc). When clicked log a message to the console with the ID of the element that you've clicked. The ID should be retrieved and displayed with a JavaScript string template and variable (refer to the class example).

Push the finished files to your repo; include a readme that describes the project. Merge everything to the main branch and submit a link to the dropbox on FOL - a .txt file is fine.

SUBMISSION REQUIREMENTS & DUE DATE

Project must be submitted Week 5: ONLY THE MAIN BRANCH WILL BE GRADED.NOT ZIPPED FILES!

- Github repo: readme.md file, master branch, design branch and development branch
- Correct structure for a web project (css, js, images and index.html)
- DO NOT INCLUDE AI FILES - just the SVGs in the image folder will do.
- Name the dev branches appropriately per feature IE des.tvr.artwork, dev.tvr.script

GRADING RUBRIC /20

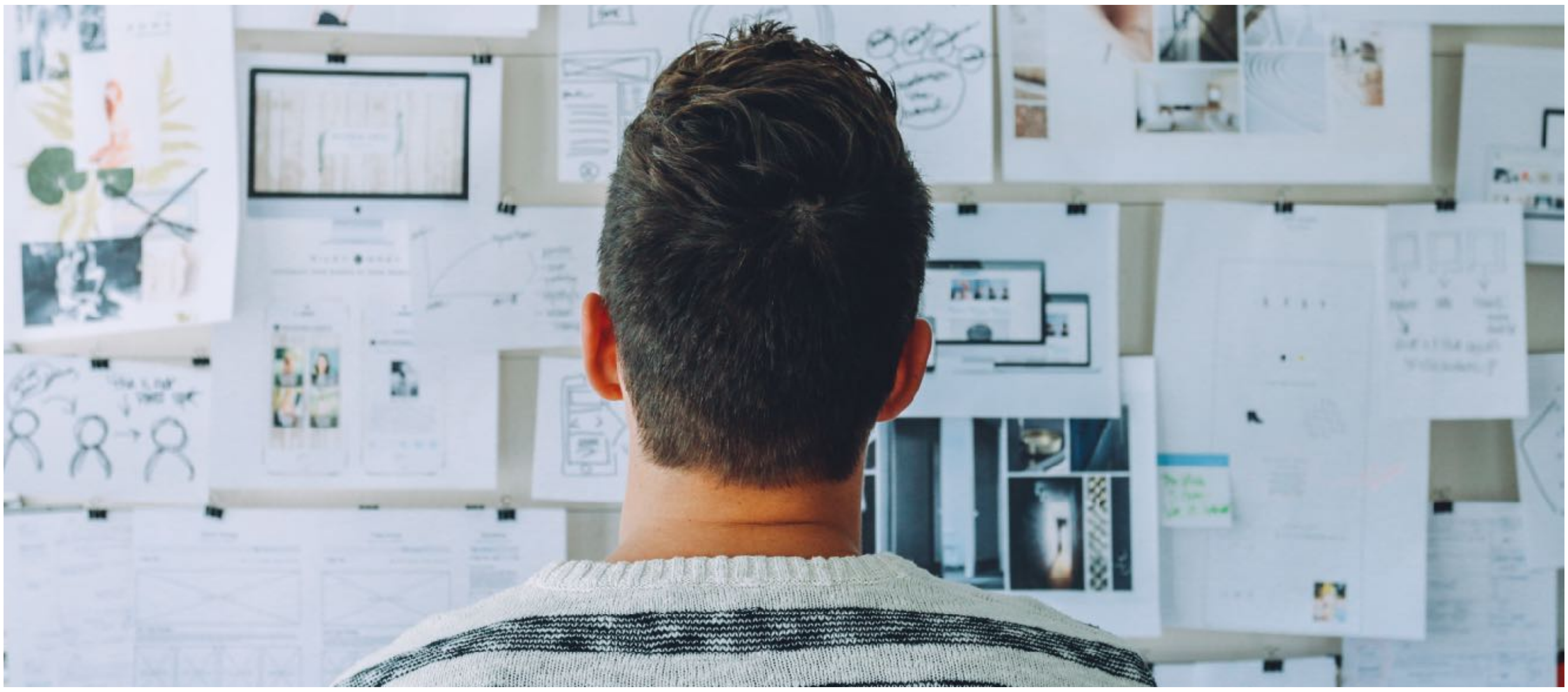
Folder Setup & Naming /1

GitHub Best Practices /5

Valid HTML and CSS /3

CSS Interactions /5

JS Functionality /6



CHECKPOINT – MUSIC MIXER

The checkpoint is a way to report to your Studio Managers (your teaching team - which is just me for now) your current progress / how you intend to continue to implement your strategy for your assignment. For the Authoring class, I expect to see the following:.

1. Github repos for each project with team access (both members contributing). Initial commits should be from Week 2 / 3 at the latest. The more the better!

2. Readme file that describes the project, and any technical requirements for local setup / development. DO NOT use the repo name as the heading of that document. You should have a description, etc. NOT just a couple of names.

3. Initial commits from the developer and the designer: site shells / project structure (files and folders up on the repos - can be empty with .gitkeep files); CSS and HTML (obviously not finished, but at least a start)

4. Dev notes for the projects (any initial research into events, audio, drag and drop etc)

5. Pinterest or other collections of examples / inspiration so that I can see the visual direction you're going with your project; sample audio files, if any

SUBMISSION REQUIREMENTS & DUE DATE

Project must be submitted within Week 6 class time: ONLY THE MAIN BRANCH WILL BE GRADED.NOT ZIPPED FILES!

- Github repo: readme.md file, master branch, design branch and development branch
- Correct structure for a web project (css, js, images and index.html)
- Submit Dev notes, research, visuals, audio files, etc, to the FOL Dropbox

GRADING RUBRIC /10

Folder Setup & Naming /1

GitHub Best Practices /3

Dev Notes /3

Asset Collection /3

HW #2 – CRUSHING BUGS (TROUBLESHOOTING JAVASCRIPT)

Use the in-class build files (the Puzzle Drag and Drop) and fix the bugs that were discovered in class. This is a research assignment - solve the problems outlined in the brief.

Create new branches (named appropriately) and update the functionality to improve the game UX.

Assignment Requirements

In its current state, our puzzle game has at least a couple of bugs. You can drag and drop more than one puzzle piece into a drop zone - this shouldn't happen. There should only be one piece in one drop zone at a time.

The second bug is the problem with pieces appearing in the drop zones on reset / choosing a new puzzle. Those should be removed / reparented back to the drag zone as well, so that the player has a fresh board to drop onto.

Solve these two problems. You can use the original drop function to solve the first problem, and another function for the second. Try experimenting with calling one function from another - you might not need to add any event handling to solve these issues, just extend the functionality.

Write out your plan FIRST. You can't solve a problem - coding or otherwise - if you don't understand what you need to do. Reason about the problem - think it through. Articulating the solution first will be a great help in completing the assignment successfully.

Additional Requirements

1. Include a written plan / path with your submission. Analyze the problem and provide the solution.

2. Create a Readme.md document for the repo with detailed information about the project.

3. Create a branch for each bug and its solution. Provide the JS for each.

4. Merge everything to the master branch (keep all branches in your repo).

SUBMISSION REQUIREMENTS & DUE DATE

Project must be submitted within Week 8 class time: ONLY THE MAIN BRANCH WILL BE GRADED.NOT ZIPPED FILES!
- Github repo: readme.md file, master branch, design branch and development branch
- Correct structure for a web project (css, js, images and index.html)

- Name the dev branches appropriately per feature IE des.tvr.artwork, dev.tvr.script

GRADING RUBRIC /10

Folder Setup & Naming /1

GitHub Best Practices /3

Documentation/Plan /2

Fixing Bugs /4

MUSIC MIXER

The focus of this project is to collaboratively research, experiment with and ultimately design and build a multimedia application using design and motion tools as well as HTML, CSS and JavaScript. Your team will implement drag and drop functionality to load audio clips and create a dynamic audio experience, trigger animation, and play a compiled audio track (see www.incredibox.com as a reference).

Try to implement this project in sprints - part of an agile workflow best practice. A sprint consists of a scoped mini-project that focuses on one particular deliverable or feature. The work you do lives on a specific branch; everything required to complete that deliverable is done on that branch and then merged to the main branch when finished.

Then it's on to the next sprint!

Assignment Requirements

Sprint 1: Setup, analysis and initial design

Set up the Github repo with a readme (make sure this is more than just the project name - refer to readme examples online). BOTH team members should be contributing at this point - the designer can create the layout, generate the html and the css; the developer should be researching functionality, creating JavaScript files, updating the HTML to add classes or other tools you might need to make things work. Testing events, loading audio, etc can all happen here - it doesn't have to be final/finished but you should have an idea of how you want things to work, how you want it to look and start designing, testing and experimenting.

IMPORTANT - *designers MUST contribute to the repository directly, on their own design branch. You MAY NOT deliver design work via social media, or your partner's commits, or deviate from the workflow best practices in any way. Create your artwork, add and commit it via Github. Failure to follow the correct workflow as outlined in the Authoring class will result in a (substantial) grade reduction.*

Sprint 2: Drag and drop, audio, design improvements

Implement drag and drop functionality; get your audio synced and playback working. Designers should be replacing any FPO (for position only) assets with real artwork and doing any other design changes that need to be finished.

Sprint 3: Final touches / improvements
At this point you should be crushing any final bugs that pop up, doing any final design tweaks, and lots of testing / fixing. Responsive fixes would occur here as well - make sure you're testing on lots of different screen sizes.

Checkpoint

There will be a graded checkpoint at week 6. (Please see page 8 for details)

SUBMISSION REQUIREMENTS & DUE DATE

Project must be submitted within Week 12 class time: ONLY THE MAIN BRANCH WILL BE GRADED.NOT ZIPPED FILES!

- Github repo: readme.md file, master branch, design branch and development branch
- correct structure for a web project (css, js, images, audio etc and index.html)
- DO NOT INCLUDE AI OR PSD FILES - just the SVGs / .png / .jpg in the image folder will do.
- Name the dev branches appropriately per feature IE des.tvr.artwork, dev.tvr.script

GRADING RUBRIC /20

Folder Setup & Naming /1

GitHub Best Practices /3

Valid HTML and CSS /2

Design and Layout /4

Functionality and Interactivity /10

FINAL INTEGRATED PROJECT

Please Refer to FIP document on FOL for detailed Information about this project.

Create some UI on the products / promotions page that can be used to load relevant information from an array. Think about interactivity, micro-interactions / transactions, user experience, etc.

Your team can use a combination of advertisements or promotions and product features. Create an interactive graphic / icon for each (create at least 3) and load content onto the page when the user requests that information. Consider scalability and responsive design considerations for your graphics, and choose your format(s) accordingly.

Option 1:
Consider a “2 for 1” promo (or any other promotion of your choice). Create a “Click for promo details” button or graphic; clicking on that graphic should retrieve the content for that promo from an array and add or replace existing content on the page using a lightbox, popover, etc.

Option 2:
Create “hot spots” on your product images and reveal exciting information about a given feature on a click. As an example, you could reveal information about the health benefits of Orbitz or Quatro using a series of clickable graphics overlaid on your product image.

For Both Options:
Store the text information for each feature or promo in an array in your JavaScript file. Develop a way to retrieve that information on a user interaction (a hover, a click) and show it somewhere on the page. You can use a lightbox, a popover, a “see more” link… the UI choices are entirely up to you.

All of your JavaScript code must be your own. You are free to re-use (but change) any assets from the Authoring class files. If you would like to use a JavaScript library for animation please ask first.

SUBMISSION REQUIREMENTS & DUE DATE

Due Week 14 - Monday, August 7th, by 5pm EST, in FIP Dropbox
ONLY THE MAIN BRANCH WILL BE GRADED. NOT ZIPPED FILES!
- Github repo: readme.md file, master branch, design branch and development branch
- Correct structure for a web project (css, js, images, audio etc and index.html)
- Name the dev branches appropriately per feature IE des.tvr.artwork, dev.tvr.script

This is the naming convention you must use for your submission / repository:
LastName_FirstName_FIP
e.g. (Smith_John_King_Sally_FIP)

Presentations will take place August 7th and 9th.

GRADING RUBRIC /30

Folder Setup & Naming /1

GitHub Best Practices /4

Design and Layout /5

Research and Planning /5

Functionality and Interactivity /10

Presentation /5