

B & E Games – AN SI 539 FINAL PROJECT

Website Documentation

Elizabeth Popiel – SI 539 FINAL

Prepared for: Colleen Van Lent

January 28, 2019

Minimum Content Checklist:

- ✓ 4 Pages, each with consistent **NAV, HEADER** and **FOOTER**
 - **Homepage/Index** – Landing page/Splash Page for site
 - **Products** – some of the upcoming products for release pre-order
 - **Events Page** – about the ongoing events in-store
 - **About Us** – includes contact point
- ✓ Each page must have a minimum of two (2) paragraphs of text or two (2) pictures of a combination of picture and photograph at minimum.
 - **Homepage/Index**
 - Basic Image Count: 9, Enhanced Images: 3 (footer), 2 logos, etc.
 - Paragraphs: these are short, but there's more than 1, technically.
 - **Products**
 - Basic Image Count: 8, Enhanced Images: 3 (footer), 2 logos, etc.
 - Paragraphs: these are short, but there's 3, technically. (not including the short sentences)
 - **Events Page**
 - Basic Image Count: 4, Enhanced Images: 3 (footer), 2 logos, 1 background image (see in-store demonstrations), etc.
 - Paragraphs: these are short, but there's more than 1, technically.
 - **About Us**
 - Basic Image Count: 3, Enhanced Images: 3 (footer), 3 logos, etc.
 - Paragraphs: these are short, but there's ~4 (not including the short sentences).
- ✓ NAV shows current page in both mobile and desktop views
 - NAV pages should be BLACK when active/current page, and change color when hovered over (pseudocode)
- ✓ Each page contains 'Jump to Content' anchors for accessible use.
 - `Skip to content`
 - ``
 - **BONUS FUN:**

```

<div class="scroll-to-top" style="display: block;">
  <a href="#mainpage"><i class="fa fa-angle-up" aria-label="up chevron for
snap to content when at end of page </i><br>Return to Top</a>
</div></div>

```

✓ At Least 3 Different Media Queries

Responsive design is discussed throughout the document.

- **Portrait** (content has changed significantly for final)
 - Largely dynamic content, ensures taller screens are stacked appropriately.
- **Landscape** (content has changed significantly for final)
 - Largely dynamic content, ensures wider screens are stacked appropriately.
- **Screen-Size by max or min width** (content has changed significantly for final + added support for even smaller screens), majority of content will dynamically adjust based on the screen size (each query has a min, max, etc) and is designed so that at almost any point the content scales. This is especially important with the NAV java because it becomes a dropdown at smaller scale sizes.
- **@media (any-pointer: coarse)** – this changes the button sizes when there's no fine input such as a mouse pointer.

✓ Grid or Flexbox Content



- Each Page contains [mostly nested] Grids and is generally repeated where able. Some variations include color changes, format changes.
- **Homepage/Index**
 - Main content includes a large grid as this is the landing page, I didn't want to make it too complicated, so this seemed most logical. The grid contains a border box with a rounded radius, and then a pleasing color contrast (blue and white) containing general information about the business. There is a small change in mobile view, but as it is one major space, its fine.
- **Products AND Events**
 - I am combining these two as they contain multiple grids with borders (radius rounded included). The layouts are aesthetically pleasing, as well as contrasting.
 - In some cases content followed a general rule to enhance the look by nesting a border either on the bottom, right or left with 2 frames inside the grid so they would nest with images and css tables inside them.
 - In mobile view, the grids become horizontally stacked inside their respective bootstrap accordions.
- **About Us**
 - Desktop contains 2 side by side grids similar to the index page, but color coded and filled appropriately, adjust with screen sizes. Aesthetically pleasing and simple nested colored rounded radiuses.

- In Mobile or smaller screen view, these stack on top of each other and compliment the contents. Note the scaling images (px size changes depending on screen).



Contains at least two (2) Javascript/JQuery uses on site.

- **All Pages**

- **NAV** is a **Javascript** function called 'myNav' which functions as the NAV portal. This is not only to allow responsiveness, but also change dynamically depending on screen size.
- It also contains a small **JQuery** information in the code that would give the active page a different color.
- Cobbled together from W3, SI 539 and some internet searches.
- Problems appear in iOS Safari and Firefox but not Chrome, because Chrome would automatically re-size the NAV, even though there were formatting issues/breakpoints. Had to be careful to check all browsers.

- **Homepage/Index**

- **Flickity Javascript** (customized) so that I could have a dynamic carousel that you can scroll both backward and forward on.
 - Simple fix override? The Flickity Carousel didn't wraparound, it only went from Left to Right and stopped which isn't terribly accessible; I overrode the Flickity script in the local index file because I can't change the CDN file. It turned out to not be **too** hard – make a wee **JQuery** with wraparound=true instead of false, and keep centered, this way people can scroll they like for worldly accessibility (R to L). There was no need to make it scroll up and down and aesthetically it didn't look nice when I tested that.
 - In Mobile I scaled so it decreases to 1 image instead of 3 on-screen at a time (with tail ends depending on size) and you can swipe or tap.

- **Events Page**

- **The failed calendar.** Actually the **JS&JQ** on this works really well (<https://codepen.io/B8bop/pen/GhCAb>) but I could not get it to scale dynamically for mobile no matter what I tried, hence its exclusion from the final (it is there just commented out so you could see I did it). I changed a few things inside the JS so that it was color-tolerant WCAG. I also had to call an older cation of font-awesome (3.2.1) instead of (4.7.1 or newer) and therefore had to figure out a way to follow the fa fas prefixes that didn't exist in 3.x versions. So the selector iclass^[] changed to iclass*[] to help this issue along.
 - **Note:** I had previously had a Google calendar embedded in prior projects for 539, but this caused several AXE/WAVE issues and was not terribly friendly, so I also took this out. I Thought it important to show here, because although I was not able to get it to scale dynamically, everything else worked.

- **About Us**

- There's script here so that the card flips over. This looks cool and is a good way to stack content without over-crowding the page with extra DIVS or grids. This is detailed again in the CSS flip section.

✓ Does it pass WAVE as well as AXE?

- I checked each page as best I could while activating different types of content.
- AXE and WAVE were quite insistent on ARIA-LABELS especially for the NAV content (empty link) even though technically this shouldn't have mattered according to the WCAG.
- Firefox: AXE tried to read the following items which are not issues:
 - Contrast on carousel overlay with image-based background. Originally the text was white, but this was changed to a color safe red in case for some reason the image didn't load, or the white went against the lighter background page color. AXE was still confused because of the image node, but it follows WCAG black on white (in case background or img fail), so I wasn't sure how else to tackle this, except to dismiss it.
 - MAC computers in general (Safari/Firefox) often require TAB settings activated at user level. Tested with these settings on, they worked fine. (Skiplink non-issue. It's fully visible on activation). Ignore error.

We are not sure this is an issue, because:

Skip link target should become visible on activation

☐ This is an issue ☒ This is not an issue

SAVE

- AXE and WAVE Flipcard: Color contrast and target issues. There are not contrast issues (it just can't read them), and the target is not live because this isn't a truly live webserver to send the posted content.
- WAVE Contrast Errors:
 - Skiplink was disguised for tab, and set off-page, since this isn't dependent on visualization of tabbed accessibility. I decided it could stay off page, but I could just change the color to accessible color schemes.
 - Skiplink was an issue for the flipcard, but the target is already present, so I added some alt to fix this, AND changed the # to a <form action="#"> to create a more permanent fix.
 - Firefox also had an issue with background images of all types, even if the contrast ratio was fine.

We are not sure this is an issue, because:

Element's background color could not be determined due to a background image

Related node:

[Inspect](#)

.borderbox_reverse

☐ This is an issue ☐ This is not an issue

- It also had an issue with color determination for the flip cards (send us a message on the back and front) but the color is color safe, so it allowed me to dismiss.

We are not sure this is an issue, because:

Unable to determine contrast ratio

☐ This is an issue ☐ This is not an issue

SAVE

- Return to Top was color safe, but couldn't be determined in Firefox:

Element's background color could not be determined because it partially overlaps other elements

☐ This is an issue ☒ This is not an issue

SAVE

- Many color schemes come from The US Design Systems, but not all. This was largely to ensure that contrasts could still “look good” as well as check WCAG for 4:5:1 ratios. (<https://designsystem.digital.gov/components/colors/#text-accessibility>)

✓ Any other Validation issues (W3)

- Just with the .css file, because the validator doesn't believe that regular expressions exist for some reason (line 1033). As I was unable to complete the calendar (even though it looked great), it should not appear as a validation issue. You can un-comment them to test if need be.
- Also in the margin-block field. (<https://developer.mozilla.org/en-US/docs/Web/CSS/margin-block-end>). This exists and was used in the NAV for each webpage.

Extras Content Checklist:

✓ CSS Extras

- **All Pages**
 - I Had a lot of fun investigating and peppering little CSS things throughout the work (font weight instead of strong,
 - **The NAV** contains an oval overlay (well technically underlay) called .circle-overlay that goes under the B & E Games logo. It let me use some of the drawing work we learned in class, but also enhanced the logo's presence in the NAV bar and allowed it to remain color-accessible.
 - **The Footer** contains 3 icons for social media. These icons are responsive for desktop (there's not really a way to do this for mobile), so that when you hover over them they will change position within the image to a different color/location (from 0-40). There's ARIA labels to help identify these items for screen readers. I got the idea from seeing it somewhere in the past, but the initial coding came from here (<https://developer.mozilla.org/en-US/docs/Web/CSS/background->

[position](#)). I adapted it to the :hover, and I also learned that if I were to go further and specify x,y coordinates I could get more control over positioning (https://snook.ca/archives/html_and_css/background-position-x-y). I had no reason to implement this version of it since it is not supported by enough browsers to be considered accessible.

- **Homepage/Index**

- CSS - I built a **CSS 'container_overlay'** tile based off one of the W3 schools examples and set it over the Flickity Carousel. Changing from W3 was tricky because the initial CSS was reliant on the *{} tag, it was much simpler to incorporate it into {}. I also had to adjust padding and other minor details, but this worked really well with the Flickity Carousel.
 - Other issues: see above (flickity script override & axe colors)

- **Events Page**

- **CSS Enhanced Tables.** They looked MUCH better than a list did. I added nth-child, colors and last child selectors.

- **Products Page**

- CSS Enhanced Tables. See above.
- **Lightbox** (from <http://fancyapps.com/fancybox/3>). Originally I wanted to use something like this on the index page because it had some nice enhancements (different views on click, but the Flickity carousel I built looked SO much better, so I put it here in a smaller capacity instead. This enhanced lightbox uses a Modal type display and has "data-caption" defined as its captioning method. They have good documentation, so it helped when I needed to learn how to enhance this and if I wanted to do more. I don't need much, so I kept it pretty simple.

- **About Us**

- **CSS – Flipcard** (adapted from <https://css3playground.com/3d-flip-cards/>). This was no small amount of trouble. I wanted to add another element, like a contact card, without changing the footer details. The code itself was in github, but the packages were combined, so it took a bit of testing to make sure I had all the right pieces for the flipcard. Since I already had a .panel allocated on the Events Page for bootstrap (explained later), had to make changes to the CSS again to support it, without breaking things.
 - **JQuery** content also present.



THE BOOTSTRAP CHALLENGE (ACCORDIONS OF DOOM)

The bootstrap I chose to add to the website occurs on two (2) pages, both are the same code, just applied to the pages original content.

- These are collapsible **accordions** used to open and close content sections. I thought I would keep it simple. Plus, I wanted to do this with a fontawesome icon.
- Links to bootstrap are needed twice on the page (top and bottom).

- Accordions are VERY sensitive to layout or script changes. Move one thing, and it would break other layouts (less mobile layouts and more desktop issues, namely grid placement and then later mobile justifications -- think hunting through CSS to find what I am breaking in the bootstrap).
- I loved incorporating the dice as a 'roll' when the accordion expanded or contracted. This replaced the boring banner I had in a previous assignment. Keeping them in alignment is a struggle and they broke often when playing with the content.
- Sometimes the inner 'white' borders would throw off the layout of the outer sides, which resulted in some layout changes to prevent that from happening. Had to be careful not to override colors from different accordions.
- I had originally planned to make the accordions ONLY mobile use, but they worked so well, and no one likes a crowded page, that I decided to keep it for desktop as well.
- The accordions were adapted from (<https://getbootstrap.com/docs/3.3/javascript/#collapse-example-accordion> AND specifically the Gleb Kamarsky example here - <https://stackoverflow.com/questions/38037274/bootstrap-nested-accordion-font-awesome-icons>).
- I had originally intended to one from use (<https://mdbootstrap.com>), but there are issues with paid content. I did try to re-create it without the CSS, but it only worked partially and in the end I opted for something simpler to go with the layouts.
 - **Products**
 - Releasing This Month
 - Releasing Next Month
 - Pre-Order Products
 - **Events Page**
 - Q & A Session
 - Tournament of Splendor
 - In-Store Demonstrations