
Through the Looking Glass

**A Design Vision for Publications
in an Era of Massive Fragmentation**

Jungho Kim
School of Design and Crafts, University of Gothenburg
Gothenburg, Sweden
Spring 2012
Degree Project, MA Programme in Design, 120 hecs

Table of Contents

Abstract	Page 3
Introduction	Page 3
Background & Context	Page 3
More Detailed Background	Page 4
Objectives	Page 6
Aim	Page 6
Problem Formulations	Page 7
Delimitations	Page 7
Implementation	Page 8
Research	Page 8
Work Methods	Page 8
Results	Page 11
Breakpoints (Responsive Design)	Page 11
Interaction Models & Templates	Page 12
UI/Interface	Page 14
Grid	Page 15
Individual Articles	Page 16
Evaluation of Results	Page 16
Reflections	Page 17
Summary of Examination	Page 17
Implications of Designing with Code	Page 18
Future Implications of Knowledge	Page 19
Reference List	Page 21

Abstract

In the publishing industry where 0s and 1s have replaced an archaic material called paper as the primary form of distribution, the struggle to bring content from analog to digital has been fraught with questions about how to create digital-native publications. Of existing solutions, most can be put into two camps; those that look like websites, and those that replicate print layouts on screen. Unfortunately, these approaches lessen the integrity of content by forcing it to live in the shell of a website sans consideration for editorial context, or are committing usability faux pas by simply mimicing print-like forms on a screen. Through the Looking Glass is a prototype concept for a true 21st century publication, from information architecture to design and development.

This project was conducted at a time in between the app revolution - where publishers rushed en masse to deliver content via applications made for specific devices - and a return to the browser and open web as the preferred content delivery platform of choice for publishers. A key catalyst for a shift in interest back towards web sites or web apps for content delivery was the advent of responsive web design, or an approach to designing and developing sites that automatically adapted to the screens on which they were viewed. This method of creating a content delivery platform appears to be, after discussion with a number of professionals who have experience designing and building both native applications and web sites and web apps, less time consuming and certainly less frustrating when considering the necessity to distribute content across various screen resolutions, platforms, and devices while trying to reach design parity.

Of course, the trade offs of moving to the web from apps aren't 1:1, additional pros and cons to each approach exist and this project simply sheds light on some of those areas. The year 2012 is still just the beginning of the migration back to the browser for the publishing industry, and knowledge gained from this degree project will hopefully help to provoke thought on this subject and be carried forward in my own future endeavours.

Introduction

Background + Context

This project's title, Through the Looking Glass, refers to the second story about a girl named Alice (yes, the same one that went to Wonderland) in which the world as she expects to see it becomes distorted with changes in time and space. It is, in fact, representative in many ways of the challenges facing those who are designing for screen based media with a very specific emphasis on the publishing industry.

With the rapid development of technologies for distributing information in the past twenty to thirty years publishers of magazines and newspapers have struggled to adapt their publications to make the transition from printed paper to screen-based devices. When publications started appearing on the web, they were, like most of the early internet, a failure in design. Though their aesthetics slowly came around, the user experiences were still unrefined. Where websites had previously been unable to deliver pleasurable viewing experiences due to a number of factors such as the need to download pages individually and browsers reloading on each page change, native applications or “apps” filled the gap with the inclusion of rich media, smooth transitions, and interfaces that were appropriate for the smaller screens on mobile devices. An explosion of devices and operating systems during the late 2000s to early 2010s in the mobile and tablet market contributed to the erosion of any semblance of a maintainable app development environment which then paved the road for a return to browser based publications.

More Detailed Background

Publications began their lives as printed material but technology developments through the latter part of the 20th century shaped and reshaped publications at a quick pace. Manual printing processes originally contributed to bespoke design for magazines and newspapers with type and other elements set by hand, a trade that was only made easier with the introduction of the computer and desktop publishing in the 1980s which encouraged iteration and sped up the production process.



Fig. 1: NYTimes.com in 1996.

Then came the internet. At first it was ugly, clunky, and hard to access, but eventually publishers realized that bringing content online would become a necessity.

In 1996 the New York Times launched NYTimes.com (Fig. 1), and despite some attempt at keeping visual integrity intact it turned out to be an aesthetic train wreck, preserving only a few design elements from the print edition and by and large falling in line with the look and feel of the internet of the 90s. Other publications succumbed to the same fate, departing from their high level of design standards in exchange for the uncharted newness of the internet. This pattern would continue to repeat over the next decade even as design trends and standards for online design evolved while publishers desperately sought new revenue streams to support their outdated content platform — print media.

As publication design evolved from a tedious job of manual labor to computer enabled desktop publishing and then the digitally controlled expanses of the internet, at each junction publications were at first heavily affected by these new technologies but eventually overcame the initial setbacks to become examples of contemporary design of their times. Each change, from manual type setting to desktop publishing, from desktop publishing to the internet (which marked the beginning of the end for print) provided the preface to the current scenario facing the publishing industry today.

While distribution has certainly gone digital, the issue of choosing an appropriate content platform has become an overwhelmingly complex mess.

If the first 10 years after NYTimes.com went live was web-based publishings infancy, the following few years marked its awkward toddler stage. The iPhone had sparked a revolution in the smartphone market when it came out in 2007, but the internet was in no way ready for what catering content to small screened devices would bring about. In fact, in Steve Jobs' introduction of the iPhone, NYTimes.com is used to illustrate the ability to hold the web in your hands and along with it came a telling quote. Jobs said, "Now this is really great and I can see the whole page, but of course I can't read it, it's a little to small..." Though the age of devices had been ushered in, the world in which it would live in had yet to be built.

In the couple of years after the first iPhone was introduced competition from Google's Android along with the rise (and demise of) WebOS, the 2011 Nokia/Microsoft partnership on Windows Mobile and subsequent near death of Nokia's Symbian OS (Dediu, 2012), along with continued (and probably futile) persistence of RIM/BlackBerry all together exacerbated the issue of complete and utter Operating System/platform fragmentation for software developers and content creators.

Paired with the rising of the tablet device category which was again dominated

by Apple's 2010 introduction of the iPad, the mobile device landscape had become a truly unforgiving environment to operate in. For example, in 2012 releasing an application that would run only on Apple's iOS operating system could require the design and development of 4 separate apps; a low resolution (iPhone 2G/3G/3GS) and high resolution version (iPhone 4) for iPhone, as well as low (iPad 1, iPad 2) and high (The new iPad) resolutions for iPad. Trying to reach full design parity across Android devices would be completely impractical with one app maker logging over 4,000 variants of Android and over one hundred screen resolutions in a six month period ([Open Signal Maps, 2012](#)). Even designing a small number of the most popular Android device resolutions yields less than optimal design results with liquid layouts not fully optimizing themselves for the various screen resolutions and aspect ratios.

Concurrently, developments in the web world such as HTML 5, CSS 3, and advanced Javascript paired with the advent of the responsive web design methodology paved the way for a return to web-based digital distribution that could provide much of the experience of native apps with the accessibility of the web.

Responsive web design is an approach to web design coined by Ethan Marcotte ([Marcotte, 2010](#)) that says content should adapt or respond to the screen size or device that it is being viewed on. In essence, instead of designing and building for individual devices and screens, do it once but build into it flexibility which mimics the constantly shifting landscape of technology. In many ways, it is perfectly suited to address the needs of publications by providing a mindset for designing to look good on anything and to make development more efficient - to design for the internet one must accept the ebb and flow of technology and space, and accept an almost Daoist state of mind ([Allsopp, 2000](#)).

Objectives

Given the previously introduced scenario, the following three points were created as metrics by which to judge the final result on:

1. The project should provide a more design-centric platform for digital publications and revitalize the relationship between design and content.
2. The project should utilize a way for interfaces and layouts to adapt to the various screens and contexts it could be viewed in.
3. The project should deliver a uniform experience across multiple screens and browsers.

Aims

In addition to the above specific points, there was a desire to spark dialogue about the realities and limits of the concept of a web-native publication.

Despite the successful re-launch of online publications such as the Boston Globe [<http://bostonglobe.com/>] as a responsive site, research didn't unearth specific examples where responsive design had been used in conjunction with a print like approach to design to exploit the possibility of varied layouts per article, especially in regards to editorial design. That signaled an opportunity to individually design articles on top of a grid and bring some of the "design" aspect back into designing online publications.

Another interest was getting hands on experience challenging the notion that apps and not the browser was the best place for editorial and journalistic content to live. Taking the stance that web apps in the browser were in many ways superior to native apps was siding with an idea that was just coming to fruition in the tech world, so making it a purpose to understand this with first hand experience seemed to guide the project towards a valuable learning outcome.

Problem Formulations

For the purpose of this degree project the stance was taken that responsive web design, enabled by new technology, would provide a good conceptual foundation from which to explore the question, "What might a modern publication at the intersection of print magazines, native apps, and the web look like?" It was a way of trying to keep the project both in a design and technology space, as early criticism of this project was that it leaned too much towards tech and had a deficiency in its level of design focus.

Though the approach of responsive design helped to target specific screen resolutions, it did not identify what kind of devices the content would be viewed on — a browser with a width of 320px could be a smart phone or it could be a narrowed browser window. The result of this realization was to design an interface that would be device agnostic or usable across a variety of devices, both touch screen and traditional, leaving behind the idea of device specific optimizations.

At the beginning of the project the final result was to be some form of flat comps. Initial reasoning for delivering flat mockups was to put emphasis on design over development due to past experiences where inability to implement design features resulted in a degraded product, but as soon as explorations into page and interaction models began and experimentations with layouts started it became apparent that a demo-able working prototype would be needed.

Delimitations

The execution of this project took place within a significantly shortened time period. The original degree project that was planned got dropped at the beginning of week 8 and due to difficulties in gaining approval for this project and obtaining a suitable tutor, work didn't officially start until week 11. This effectively cut the working time from 13 weeks to 7 weeks and as such did have an affect

on output.

After deciding on an approach to designing and being well aware of the potential consequences of choosing to work with code, a clear set of limitations were developed to aid in preventing scope creep on the project and give more focus to the primary goals. Though the list (below) was not exhaustive it did provide a strict set of guidelines to work within:

1. The project will be an initial attempt to see if the concept of a part app, part website, print influenced publication realistically addresses the needs of users and the industry, and a fully functional and complete application, website, etc., will not be attempted during this degree course.
2. Features such as integration with social networking and smarter/more complex ways of navigating, displaying, and exploring content will not be designed in order to preserve time for other portions of the project.
3. The brand and content will be based upon an existing publication, Seoulist [<http://seoulistmag.com/>], instead of being built around a purely fictional scenario.
4. The project will be envisioned as an issue based publication with a fixed number of articles as opposed to a feed based publication receiving newly published material as it is ready.
5. The project will be prototyped to be used on mobile Safari, Safari, and Google Chrome (webkit-based) browsers for the purpose of being able to demo across a range of devices.
6. It is acknowledged that design fidelity and depth will be sacrificed in order to develop a more evaluable final product, though a proper balance between the two should be maintained.
7. The overall focus will be on Information Architecture, User Interface design, and Grid and Layout Concept for interior (article view) pages; the homepage, while important, will be a secondary consideration.

Implementation

Research

The formulation of the idea for this project took place over the course of 8 months prior to the degree projects inception and research included monitoring and taking place in discussions over that period of time as some of the concepts were challenged and new issues came to light, additionally, my own experience with Seoulist, an online publication I have been working on for the past year and a half helped provide much of the foundation for understanding and formulating

the context that the project exists within.

A large part of the research phase was sampling publications that existed both as traditional websites, web apps, and native apps. Some of those examples included The Guardian (UK), The New York Times (NYTimes.com, Times Reader desktop app, NY Times iPhone/iPad apps), The Daily (daily editorial publication that is digital only), The Financial Times (HTML 5 offline web app), and several others.

Reviewing these apps helped formulate opinions on how to approach digital publication design and ways of organizing and laying out pages — namely, some of the fundamental aspects such as horizontally divided articles with continuously vertically running text in a singular column were observed and then more thoroughly understood after reading an article (Reichenstein, 2010) discussing the merits of this model as opposed to a column-based model that mimics a printed book with page changes occurring on the horizontal axis.

The challenge was that putting content into long running columns removed the existence and notion of pages within an article. This caused even medium length texts to start feeling heavy with more lengthy stories becoming undesirable to read due to fatigue.

The solution was derived from the same article discussing the difference between the scroll model and page model and the discussion of the difference between “semantic breaks” and “rhythmic breaks” was a key takeaway. Instead of letting articles become one long running piece of text, breaks could be inserted inline where the content itself wanted to take pause, such as at the end of a section or in an area where related photos wanted to be displayed.

Work Methods

The production of this project was by traditional terms unorthodox and counter to a typical design process. Because the decision to produce a prototype was made early on, the entire design process consisted of moving directly from ideas to sketches and from there straight into code (Fig. 2). Design applications and tools such as Adobe Fireworks, Photoshop, and Illustrator were used only to draw interface elements that were better rendered as graphics than code and as a way to quickly prototype ideas by taking screenshots and using Photoshop or Fireworks to cut up those screenshots and create rough digital sketches. After an idea had been worked through in that manner it was back to code to produce and test it in the browser.

Designing in the browser allowed for quick iteration, especially when applying styles to large lists, adjusting global colors and other items, and an easy way to test out interactions on the fly. By using source code control it became even simpler to branch off into new ideas since rolling back to a previous commit was

never more than a command line entry away. This helped take away some of the fear of going off in drastically new directions by automatically preserving any previously committed state and keeping the working directories clean and easy to manage.

Several technologies and services were involved in the building of this project. All of the coding was done in HAML and SASS, then compiled to HTML and CSS. Javascript was also used for putting together the application and handling the animations and interactions. It was built inside of an application framework called Middleman [<http://middlemanapp.com/>] that was used as a simple method for managing views and includes. At the end, the entire project was exported as an HTML site and uploaded to a server. Source code was tracked and managed with Github [<http://github.com/>], fonts were delivered via the web from Typekit [<http://typekit.com/>], and Dropmark [<http://dropmark.com/>] provided a handy repository for storing screenshots to track and share progress.

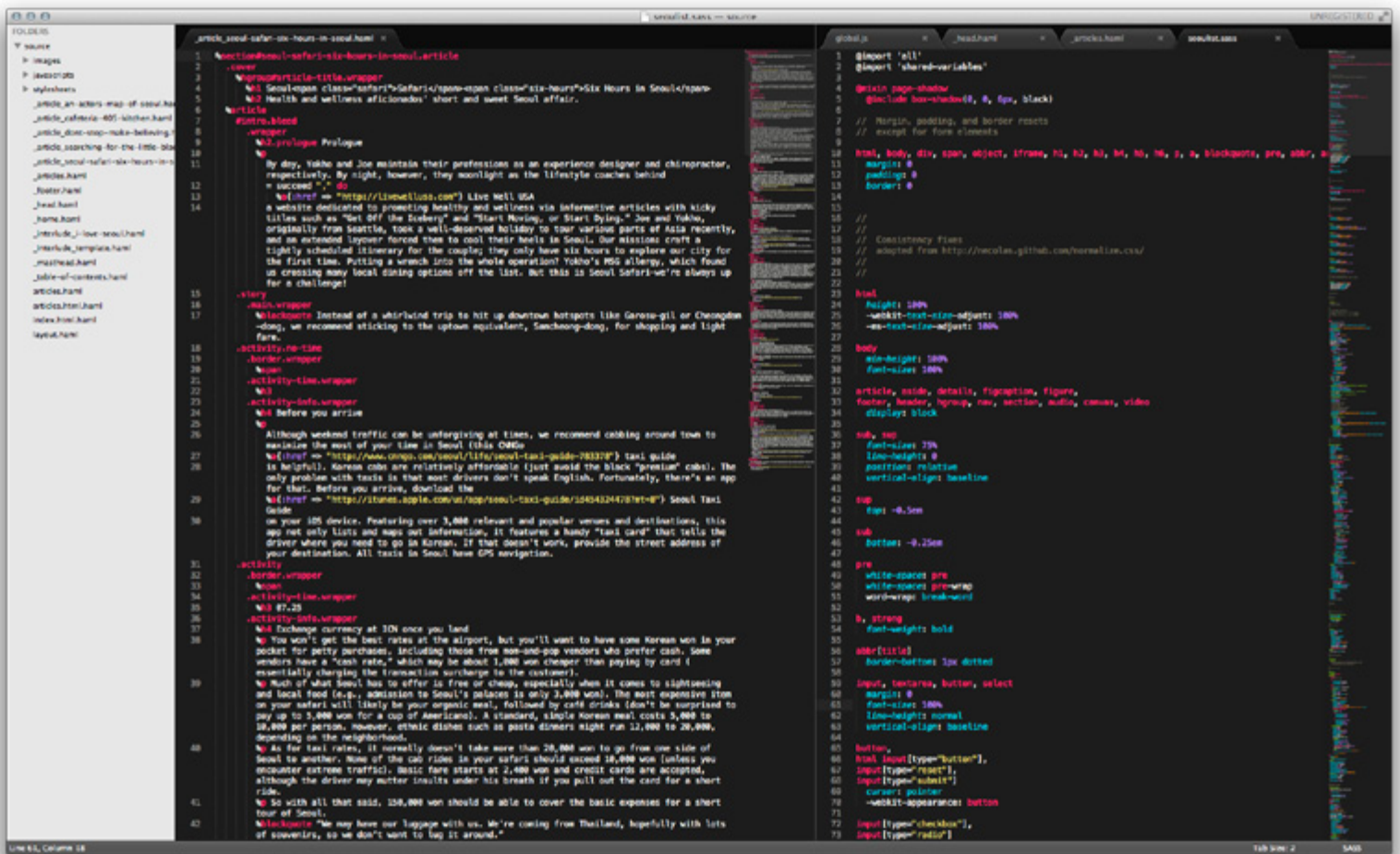


Fig. 2: Designing with code.

Results

The execution part of the design process broke down into the following phases:

Information Architecture: Gathering of content assets and subsequent organization of them, as well as developing an interaction model and creating a high level view of layout.

Wireframing: Designing the structural foundation that the rest of the app would live in and around, including the interface elements.

Design: Applying a visual styling to elements and creating layouts for individual articles.

Prototyping: Building out basic functionality and interactions, putting together separate views and individual articles into the prototype.

Though these phases were used to mark milestones in the process, in general the workflow was to come up with a way to address an issue, research it, sketch it out, then build and test it, and then repeat that process until the particular piece had been sufficiently refined. Overall it was a highly iterative way of working which generated multiple ways of looking at any one aspect.

The final result was a functional prototype that was able to be demoed with a script or clearly defined interactions in a specific order, also known as using “smoke and mirrors.”

Breakpoints (Responsive Design)

One of the main points that this project addressed was the need to work across a multitude of devices in the browser, and as such the responsive design methodology was used. Though the desire was to work on any device, for the purpose of the prototype three popular browsers were chosen: mobile Safari for iOS which runs on iPhone and iPad, and Safari for laptops and desktops as well as Google Chrome. The range of screens with widths of 320px to over 1900px was a good balance of having enough variety in resolutions to illustrate the reach of responsive design while still being manageable to design for.

Choosing the break points, or browser widths that designs respond or change at was one of the most integral part of the responsive design process. Because of the devices chosen to demo the publication on the following resolutions were selected; 320px for iPhone, 768px for iPad, and 984px for laptops and desktops. (Fig. 3)

When working with responsive design even though there may have be a single site or app, in effect the designer has had to come up with different solutions and adjustments for each page, template, or view and at each break point chosen.

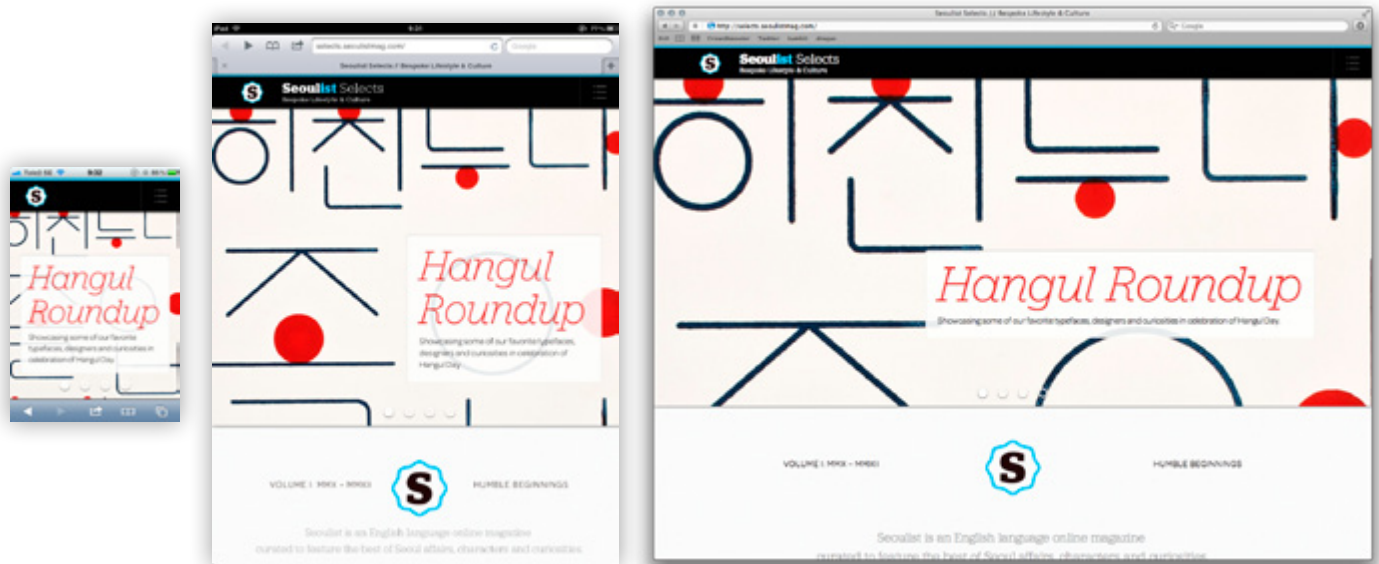


Fig. 3: Designs respond to three screen sizes

This project called for one homepage and four individual articles to be produced which yielded 15 variations (3 breakpoints * 5 individual pages). At times the translation from one break point to another is straightforward and other times it is more complex, and this is a point at which attention to detail is very important. For example, in several of the articles images that were happy to have extra breathing room around them on larger screens benefitted heavily by being extended to the edge of small screens and making the most use of the available real estate.

Other instances where small but significant adjustments were made include the removal of the title from the main navigation bar on small screens, an automatically adjusting set of lines in the background of the cover for “An Actor’s Map of Seoul”, and carefully tweaked type sizes to get the title to fit into the white space on the cover of “Searching for the ‘Little Black Pen.’”

Interaction Model & Templates

The interaction model defined the way that users moved through all of the content, it was the framework which tied all the bits and pieces together into a unified whole (Fig. 4). The primary goal was simply to compile a set number of articles together into an organized grouping akin to a single issue.

The chosen model for this project provided a clear separation of overview state and reading states, an intuitive way to move through and between pages, and a logical method for reading content.

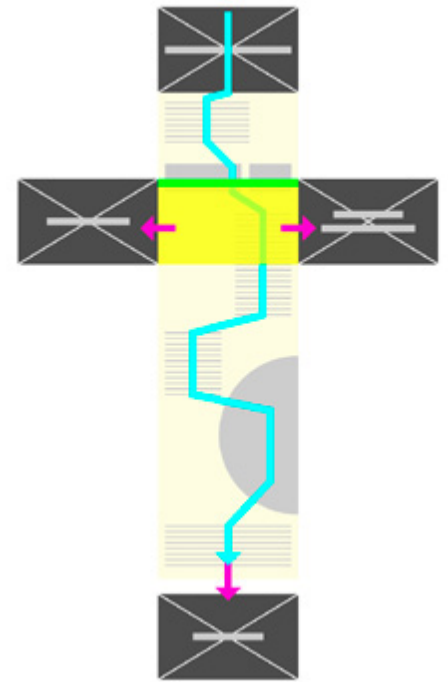
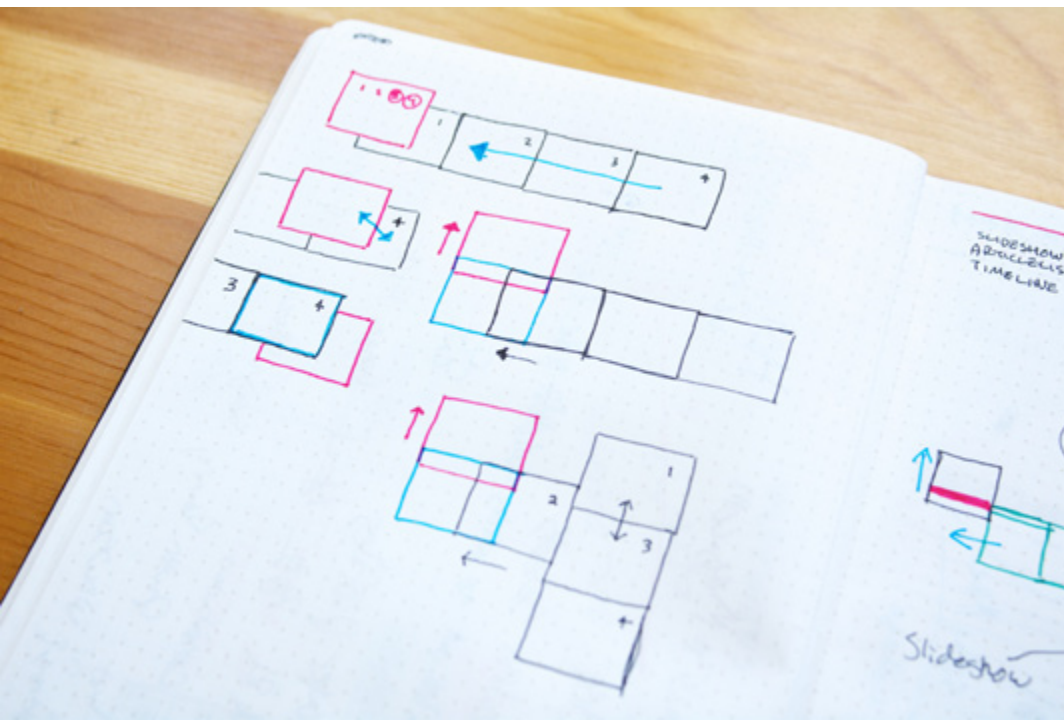


Fig. 4: Sketching different aspects of the interaction model.

The views were divided by two distinct purposes. The first view or template was the homepage and contained a single main element — a carousel with several featured articles that rotated automatically and linked to articles in the article view. The homepage also played double duty by holding anecdotal information about the publication like a summary of the issue and staff bios. The second view contained the articles, each with a cover and article below, and lined up horizontally in a single row. (Fig. 5)



Fig. 5: Homepage template (left), Articles template (right)

In addition to articles, smaller tidbits of interesting information like quotes, links, single photos, etc., were to be interspersed amongst the articles to break up the constant flow of articles.

It didn't ultimately make it into the final product, but an approach to introducing in an air of serendipity to digital reading was considered. When reading printed matter an accidental turn of a page, reading from back to front, or flipping randomly can lead to the discovery of unexpected yet enjoyable articles. Though this was something very tempting to research and design, it had been declared previously to keep the interaction and navigation models simple in an effort to focus attention on some of aspects of the project. This resulted in a decision to forego this feature, and ended up being a point of critique during the examination.

UI/Interface

The interface was derived in part from the persistent elements that exist in native apps and also from examining the running headers and footers of print magazines. Both of these elements provided branding space as well as functional information to the user/reader, and re-invisioned as a persistent header provided space for the logo and publication title as well as played home to the navigation. Early sketches in the process showed this header as an integral part of the layout, but numerous iterations both in sketch and in code were gone through before the final result was reached. (Fig. 6, Fig. 7)

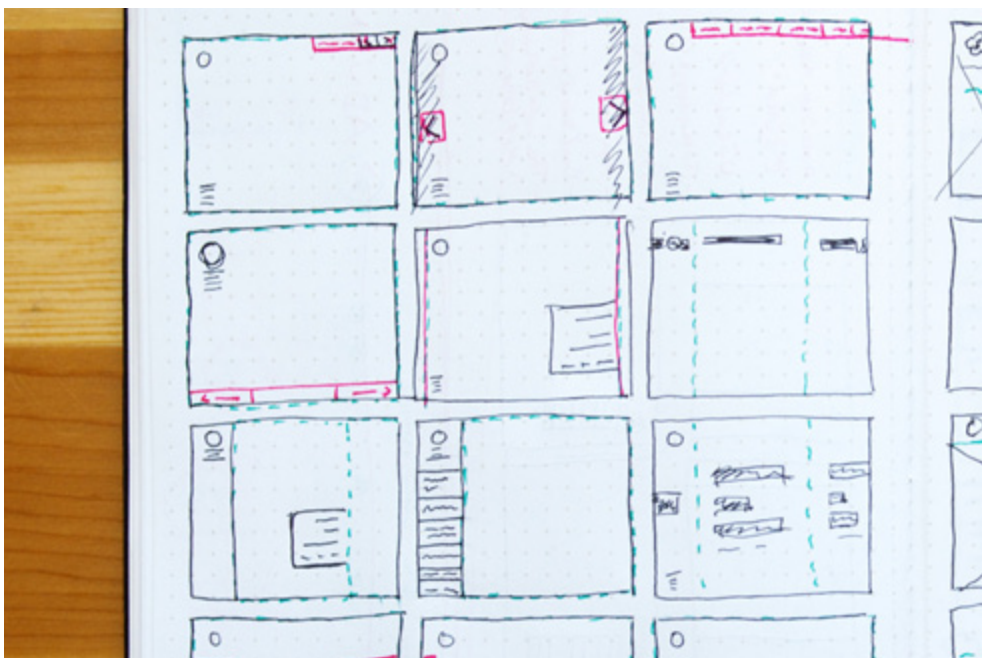


Fig. 6: UI Sketches

The navigation itself was a reflection of one of the delimitations; to maintain a simple interaction model in order to spend more time on developing an approach to layout, and though simple in its nature still allowed for basic movement through the publication in multiple ways. The first was to provide sequential navigation through a predetermined order of articles, essentially previous and next buttons. The second, to allow jumping between any two articles via a list.

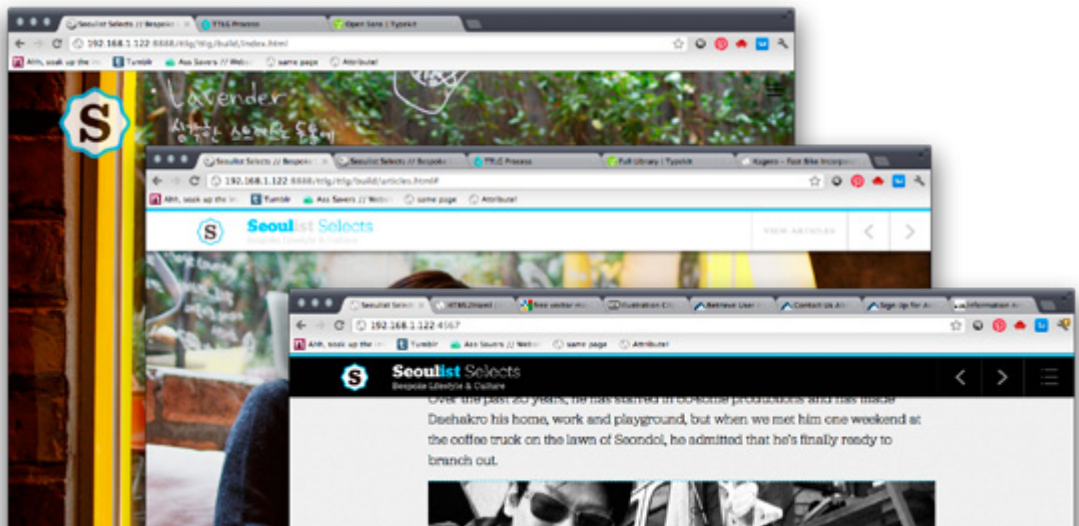


Fig. 7: Various iterations of the UI

Grid

A key component of any magazine is the existence of a clear grid system to assist in layout and establish a certain rhythm across all pages to visually tie the publication together as a unified whole. This concept while originating from print design has also become a staple of design for screens.

For the web, popular systems such as the 960 Grid System [<http://960.gs/>], 1140 Grid System [<http://cssgrid.net/>], and Skeleton [<http://getskeleton.com/>] are just a few of the numerous systems that have been used to give designers back the controlled chaos of print layout. Some of these grid frameworks are already setup to be responsive and reflow automatically for different screen sizes, but the approach of one system seemed more appropriate than the others.

The Golden Grid System [<http://goldengridsystem.com/>] divides the screen into 16 columns (plus two for the margins) which then collapses down to 8 and then 4 columns depending on the width of the device. It is also used best as a set of guidelines rather than a strict framework and instead than quick construction of pages with presentational classes, it encourages the designer or developer to

deconstruct the grid and the underlying system, and then to put it back together in a way that is suitable for their purpose.

This system was used as the framework for the articles and allowed the creation of bespoke templates for each article, as needed. It was useful when assembling the various types of content such as lists, and managing the display of images. Having the freedom to create separate designs on a per article basis was liberating, though the time required to put together a single article could be significant, depending on the complexity.

Individual Articles

A small group of the chosen twelve articles to appear in Seoulist Selects were fully designed. Rather than producing a larger quantity, it was encouraged by the tutor to focus on a higher quality set of fewer articles.

It was when designing these articles that the benefit of using the Golden Grid System began to shine. For example, the article “Seoul Safari: Six Hours in Seoul” provided an itinerary for a couple on a short layover in Seoul who still wanted to experience the city. The format of the text was largely an itinerary and being able to create a structured and columned list style helped convey an appropriate feel. Likewise, an article on a local cafe had two photos that worked great accompanying the copy in the main body of the story and the grid system allowed for setting these images side by side on larger screens and stacking them on the smallest screens.

The most significant choice surrounding the design of these individual articles was to not use a templated system for the layout of each article. As the relationship between content (text, images) and design can be very important to properly telling a story, the ability to design and move flexibly within a grid system is important for a proper publication and an area in which many web publications have historically failed. Instead, invoking the Golden Grid System allowed designs unique to each specific article.

Evaluation of Results

The final product, though challenged by the time constraints ultimately did satisfy all of the objectives, albeit with a few caveats.

Building not only the surrounding framework, but also the core layouts around a flexible grid system did alleviate the frustrations around designing for an online publication. Feeling free to be more creative and putting together layouts with specific intent gave a sense of design empowerment in a way that is quite different than the status quo, something experienced while working on Seoulist-Mag.com since 2010. During that time what was realized is that after the shell of the website was built, not much other design was required to push articles

out — the approach of using a grid system within customizable layouts really does change things and add a level of integrity back into the design that was previously lacking.

Additionally, the use of responsive web design was both successful in adapting content to various screen sizes and was viewable on both mobile and laptop/desktop platforms. What was lacking was an extension of support to other browsers which would have made a truly cross-platform experience, but this is something that was covered in the delimitations and would be able to be implemented by any seasoned web developer on a live project.

Despite the challenges and obstacles that appeared along the way, Seoulist Selects proved that a digital publication doesn't have to be an app and neither does it need to be modeled after typical news or editorial websites. The third option of a modern web app publications has been attempted as seen with the Financial Times, but a further step towards design refinement is still yet to come. As the industry moves forward there will hopefully be designers who also subscribe to this idea and publishers who are willing to put it into action.

Reflections

Summary of Examination

For the purpose of the examination the prototype was placed on a live server for demonstration during the final presentation. Due to a lack of suitable internet connectivity in the exam room a local copy was run off of a laptop, this had no negative affect on the presentation.

A strict script was followed in order to guide the viewers through the experience. This was necessary because a successful demonstration required clicking on different elements at specific times and walking through articles and features in a predetermined order. This is standard fare for demoing a product of this level of completion and successfully conveyed the project to the examiner, tutor, and opponent.

After the presentation and demo an informative and beneficial discussion took place. Overall, the reception was good from all parties and the evaluation was the most critical and most useful feedback that I had received at HDK.

The opponent, Filippos Arvanitakis, Creative Director at LBi, a digital agency in Gothenburg picked out two of the objectives that he believed were achieved and one he thought fell short. His positive remarks were that the solution of marrying responsive design and publication design was a good decision and successfully addressed the goals that had been set. He was much more vocal about the lack of bringing design integrity back to publications in the browser, and though

I believe there was some miscommunication about my specific intentions the resulting discussion was insightful and positive.

Fillippos' main points surrounded the lack of 'smart' navigation and taking advantage of the things that digital publications could do that paper ones couldn't. A few examples include being able to present the most popular, most read or other articles that would appeal in a specific way to readers, connecting with social media, and finding different ways of representing content — something he showed one of his studio's current projects doing.

Another point of contention was the debate around bespoke design for articles versus a developing a larger pool of predesigned templates. Fillippos argued that it wouldn't be financially viable for a publisher to pay a designer or developer to create every single article from scratch. There was then a reference to print magazines where a bits of more 'boring' content such as lists and listings, shorter articles, and boiler plate content get fed into templates and feature articles and more significant content get full design treatments — this was confirmed by Martin Farran-Lee who was my tutor. My acknowledgement is that what I proposed was certainly on the more extreme end of this argument, but with reason. Without making such a strong statement it's quite hard to make the point that editorial content often ants specialized layouts when many, if not most, publications are continuing to rely on heavily templated designs.

It's likely that the realistic answer is somewhere inbetween what Fillippos stated and what I believed, maybe with the same general balance held by print magazines that Martin put forth. Fillippos also suggested an array of templates so large (up to 50) that to the average reader each page would appear specially designed for that article was a solution, something he said is done on some projects today. When considering the points I wanted to make, any situation where an article recieves a considered design treatment is a win for web-based publishing.

Implications of Designing with Code

Throughout the process of this degree project a lot of thought was given towards defining the boundry between a designer and a developer, and what it means to design with code. As the role of web designers, interactive designers, and experience designers naturally involves producing designs that are rendered in a browser or on a device and then interacted with, it has become apparent that the model of designing flat comps in a design program and then passing it off to a developer to 'make real' is somewhat of a backwards process. Though flat design files designate a specific aesthetic, the actual usability and interactions are untestable until produced in the medium that they will ultimately appear within. Waiting too long in the design process to see how things function can be frustrating when presumptions about how actions and behaviors would feel

don't mesh with reality.

As stated in the delimitation part of this document, it became apparent after getting into the process that to properly show the publication experience I was building, a functioning prototype would be necessary. Though I am competent in both design and some front-end coding, bringing the two together can be a tedious process. For one, I routinely relied on my coding skills to execute my ideas that had been sketched out and in the few situations I was unable to get the code working I had to change course to accommodate. Another frustration was a perceived inability to 'get creative' while designing in code. It seemed that the logical side of the brain that also handled coding was overriding some of the creative problem solving that happened in the other half of the brain. Because by default code wants to behave in predictable linear ways, design that comes as a result of coding often falls into the trap of being overly simplistic to the point of boredom and visibly suffers from a lack of inspiration.

My solution was to take screenshots when those trouble times came up and then manipulated those screenshots in Photoshop or Fireworks before bringing it back into code once a satisfactory result was generated. From there it was easy to make tweaks and changes, but sometimes getting the heavy design lifting out of the way by using the more traditional methods seem to be more fruitful than slogging it out in CSS.

I am fully aware that the role of the dev-signer or designer/developer has arrived, and in fact that is more or less where my skill set places me. What is difficult is figuring out which methods are best at countering some of the negative effects of working in this manner and implementing them successfully. In my case, continued work on improving my coding chops and willingness to go back into design applications when needed seem to be the best tools in my belt right now.

Future Implications of Knowledge Gained

After having had my head in the space of digital publishing for almost two years and now having completed this degree project it has become quite clear to me that a true shift from an app-centric distribution model back to an open web distribution model is on the horizon. This was recently echoed in a post by the editor of MIT's Technology Review (Pontin, 2012) which really hit home a lot of the points I had been thinking on and was able to back a number of them up with data from their experience with native iOS apps. Even more convincing was the announcement by the Financial Times that they plan to completely abandon their native applications in favor of the HTML 5 offline web app that they had been testing over the past half year or so (Andrews, 2012).

Though the invention of Seoulist Selects was a fictional exercise for the sake of the degree project, the content was real and Seoulist as a publication will

continue to grow and develop independently. The vast amount of knowledge I gained from working processes to industry approaches to managing design and content, seeing first hand the proof of concept that I had bottled in my mind for quite a long time, and the recent moves by big publishers towards the web only fuel my desire to reinvent and relaunch SeoulistMag.com with an even more in-depth exploration and execution of these ideas.

I also hope to utilize the information that I take away from this project at Adobe where I'll have the opportunity to significantly influence the future of publishing. My work in this arena is really just beginning and this project has provided a number of valuable lessons that provide an inspirational point from which to build upon.

Reference List

- Allsopp, John. 2000. A Dao of Web Design. A List Apart, [blog] 7 April 2000. Available at: <<http://www.alistapart.com/articles/dao/>> [Accessed 20 May 2012].
- Andrews, Robert. 2012. Web journey complete, FT switching off iOS app. [online] Available at: <<http://paidcontent.org/2012/05/01/web-journey-complete-ft-switching-off-ios-app/>> [Accessed 20 May 2012].
- Dediu, Horace. 2012. The opportunity cost of Windows Phone. Asymco blog, [blog] 21 February 2012. Available at: <<http://www.asymco.com/2012/02/21/the-opportunity-cost-of-windows-phone/>> [Accessed 10 May 2012].
- Marcotte, Ethan. 2010. Responsive Web Design. A List Apart, [blog] 25 May 2010. Available at: <<http://www.alistapart.com/articles/responsive-web-design/>> [Accessed 20 May 2012].
- Open Signal Maps, 2012. Android Fragmentation Visualized. [online] Available at: <<http://opensignalmaps.com/reports/fragmentation.php>> [Accessed 20 May 2012].
- Pontin, Jason. 2012. Why Publishers Don't Like Apps. [online] Available at: <<http://www.technologyreview.com/business/40319/>> [Accessed 20 May 2012].
- Reichenstein, Oliver. 2010. iPad: Scroll or Card?. Information Architects blog, [blog] 21 October 2010. Available at: < <http://www.informationarchitects.jp/en/ipad-scroll-or-card/>> [Accessed 20 May 2012].