Title	Re-imaging the UI for metadata entry & media production
Sub Title	
Author	Almahr, Ali(Ota, Naohisa) 太田, 直久
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2013
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2013年度メディアデザイン学 第274号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002013-0274

Master's Thesis

Academic Year 2013

Re-imaging the UI for Metadata Entry & Media Production

Graduate School of Media Design

Keio University

Ali Almahr

A Master's Thesis

submitted to Graduate School of Media Design, Keio University

in partial fulfillment of the requirements for the degree of

MASTER of Media Design

Ali Almahr

Thesis Committee:

Professor Naohisa Ohta	(Supervisor)
Associate Professor ,Kazunori Suguira	(Co-Supervisor)
Professor Adrian Cheok	(Co-Supervisor)

Abstract of Master's Thesis of Academic Year 2013

Re-imaging the UI for Metadata Entry & Media Production

Abstract

Digital technology allows a great amount of content creation but people usually get burdened with the curation and organization of content. Research shows that tagging content allows for better content management and discovery. Yet, most of the solutions right now are catered towards simple tagging functionality especially for videos. Touchscreens have the potential to offer a better experience for tagging content. Unlike mice, which are pointing devices, multi-touch devices excel in natural and continuous movements. This new range of movement along with natural gestures could help alleviate some of the burden on users when tagging content. This paper describes the different design and prototyping methodologies that were explored to develop and test two high-fidelity touchscreen prototypes for tagging content.

Keywords:
Video Production, Metadata, Keywords, Tagging, Touchscreen

Graduate School of Media Design, Keio University
Ali Almahr

Table of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

1.1. Background and Motivation

Digital technology sometimes takes as much as it gives us. Looking at the era of the film camera, tangible photos at that time tend to have more meanings and memories associated with them. Perhaps it is the concept of scarcity and limited film capacity that made people more attentive to capture the amazing moments in life rather than be shutter-happy as they sometimes are these days. This poetry seems almost gone with the digital age, where the cameras are in constant capture of people's daily lives.

What is more disheartening than the abundance of photos, is that they are usually there but not there at the same time. Unlike physical photos, which occupy a known space, digital photos could get lost not because of deletion or storage failure but rather because the human link for them was never there to begin with. People do not hold the photos, carry them, or stash them away as the older generations did with their photos. In a sense, the physical and tangible photos provided the older generation with "real metadata" just by holding and moving the photos around.

Now taking that scenario and applying it to video productions, and it is becomes clear that digital technology could be both an enabler and destroyer at the same time. Since costs for digital filmmaking are probably a fraction of the film days, people will tend to film more than they need. Thus comes the problem of content curation and organization in the digital era. While there are many attempts and even current solutions that provide automatic metadata and organization, the human factor is still an important element that could bring back the poetry of the old days to form a better connection between people and their content.

The use of touchscreen interfaces seems befitting for such task as they allow people to use natural gestures to interact with content directly. This paper investigates the idea of using touchscreen devices with a keyword tagging concept and explores how it affects the people involved in such tasks.

1.2. Aims and Goals

The research question this thesis aims to answer is "Can new visual interface styles that employ a blend of graphic-driven interfaces, touchscreens or second-screens enhance keyword tagging and selection, to aid media browsing and creation?" Based on this research question, the aim of the paper is to:

- Demonstrate the impact of using touchscreen devices on keyword and metadata selection
- Test and evaluate using touchscreen devices as a second-screen setup for keyword tagging

1.3. Thesis Overview

The structure of this thesis is comprised of 8 chapters covering the following topics:

- Chapter 2: Background Review

  Some of the topics related to the research at hand including literature review on narratives and video documentaries, metadata tagging roles, and touchscreen interfaces. Related works discussed include the "places + perspectives" documentary. This section includes some of the design inspirations for the used prototypes.

- Chapter 3: Design Requirements

  Defining what the prototype and concept needs to address from the users' perspective.

- Chapter 4: Prototype Design

  The methodologies used to understand current practices in desktop UI design and touchscreen applications. It explores how the concept developed from these findings and the Design Requirements.

- Chapter 5: Implementation

  A summary covering prototype development and test methodologies.

- Chapter 6: Results and Evaluation

  This chapter illustrates the usability testing process, observations, and results.

- Chapter 7: Improvements and Scenarios

  Discussion about prototype enhancements and the various scenarios of implementation.

- Chapter 8: Conclusion

  A summary of the paper and discussion about future work.

1.4. Scope and Limitations

The focus of this paper is more towards visual user interfaces with computer human interaction a key area of interest. Moreover, media production was the main inspiration to pursue new user interface and interaction patterns. Thus, the prototypes and investigations will take that scenario into consideration but the results could have other uses and applications beyond media production.

Chapter 2

2. Background Review

This chapter highlights some of the subjects that inspired the project, which include resources on documentary production, digital asset management and metadata practices, touchscreen user interfaces and current software design trends. Despite the wide-ranging material covered, this array of topics mentioned intertwines together and serves as the foundation behind the concepts discussed within this paper.

2.1. Literature Review

2.1.1. Narratives & Video Documentaries

Storytelling is an intrinsic part of society that has evolved throughout the years to take various forms and shapes. Landau writes that science and research are in fact another form of storytelling and in order to understand the scientific history of humanity, it must be treated as a story [Landau]. Furthermore, looking at documentary filmmaking in general as a narrative method, it offers an interesting blend of production techniques combining an array of traditional and advanced narrative methods [Pimenta]. In that same vein, video documentaries have stories to tell but unlike fictional works, the full story typically forms after the production of content rather than before. Consequently, it becomes up to the filmmakers to navigate through the content and find how these fragments fit together to create a cohesive narrative [Sørensen].

There are many techniques to tie the stories together ranging from traditional methods, such as handwritten annotations to storyboarding techniques that allow plotting and rearranging of the story. But with the rise of digital video production, more systems are introduced into the video production workflow that help in organizing and finding content more easily. One of these systems is the Digital Asset Management (DAM) system, one of the backbones of digital content creation nowadays. The content production lifecycle consists of coordinated actions, tasks, and collected assets. DAM systems were introduced as a method to organize and manage assets, and streamline production. Both automated and user-defined metadata play an important role in DAM and determine the assets' effectiveness for current and future productions [Hodgetts].

2.1.2. Metadata Importance, Content Creation & Discovery

Metadata exists in many forms, some can be gathered either automatically, such as the digital camera data (EXIF: Exchangeable Image Format data), derived using computers or created manually through user input [Hodgetts]. Metadata tagging could also be social as highlighted in a study published by the European Broadcasters Review. The study experimented with the employment of "social tagging" as an input and search method for educational video retrieval [Melenhorst]. The study asked 140+ high school students to answer a set of questions by searching through a video library. Various search techniques were employed ranging from standard metadata search, to social tagging, and professional metadata search. Results concluded that social tagging was more effective than professional structured metadata both in searching for answers and correctly responding to the questions. Their results show that content tagged by users tend to have better search outcomes since there is no "terminology" barrier between the creators and users. Additionally, the study sheds light on an important aspect that despite the advances in video technology, words are still more commonly used to search for videos. Thus, there is indeed a great potential to link between the content creation and discovery process through the use of keywords, tagging, and annotations to organize, archive, and retreat content. In fact, many online services encourage people to tag, annotate, and describe their videos more thoroughly to get more views and responses from people [Schloss].

Additionally, with the rising popularity of smartphones and digital devices, the process of media content creation became significantly easier and accessible to a wider range of people. In a user study conducted in the pre-smartphone era [Ames], an application called ZoneTag was highlighted for its tagging functionality and its use of Flickr for uploading tagged photos via a regular camera phone. The study highlighted some of the motivations that made users tag their content. Aside from the personal and social aspects, the inclusion of suggested metadata tags within the program encouraged users to annotate and add more metadata to their photos.

Another significance of tagging is that it creates a "common central memory" that connects different people and events together [McCune]. This phenomenon could be seen in social networking services such as Instagram. By uploading and sharing photos through Instagram, people are sharing their personal views with others. In the case study done by McCune, he reported that users tend to use the tagging function not only to describe an image, but to target a photo towards a certain group or theme. Thus, tagging content is not limited to describing the content itself, it takes on a delivery and targeting role as well.

2.1.3. Touchscreen User Interface

With the proliferation of smartphones and tablets into everyday life, touchscreen input has become one of the most used input interfaces. Despite the popularity of this method, it is still in its infancy compared to other established methods of input such keyboards, mice, trackpads, etc. Moreover, touchscreens and gesture-driven interfaces still rely on visual and graphical cues to be more effective [Shedroff]. Of course touchscreen interfaces are not limited to smartphones, tablets or PCs, they extend to include larger screen sizes such as multi-touch tables, TVs and public interactive displays.

One of the advantages of the touch interface is its reliance on changing graphics/states to convey different functions. Capitalizing on this aspect allowed a team of researchers to build a multi-touch device meant for public use to browse museum images [Cioccaa]. The system was designed to be used by virtually everybody so they opted to simplify the interactions and in addition to gestures, users were able to use tangible objects on the table's surface. In this case, the touchscreen functionality allowed the users to "explore" and navigate through content more freely compared to traditional interfaces.

2.2. Related Works

2.2.1. Places + Perspectives

The short documentary "places + perspectives"[1] is the 2nd production of the Growing Documentary (Figure 2.1). It is a remote collaboration production between Keio Media Design and University of California, San Diego. In typical documentary fashion, the final storyboarding sessions of "places + perspectives" were done after the content has been gathered from both teams. At the storyboarding stage, file storage and DAM applications played an important role. Not only was archiving important, but value-added data was needed to complete the story and tie the movie together. Thanks to the support of the technical IT crew at KMD and UCSD, a new system was devised to

[1] http://vimeo.com/47917016

accommodate those needs along with the existing cloud system and PIX (Figure 2.2). The professional PIX system is used by many professional production studios to allow for better project management and more streamlined online collaboration. The implementation of PIX within the production of "places + perspectives" came at the later stages after gathering the majority of the assets. Both the cloud sharing and PIX System allowed the UCSD and KMD production teams to focus more on the story and less about file and asset management.

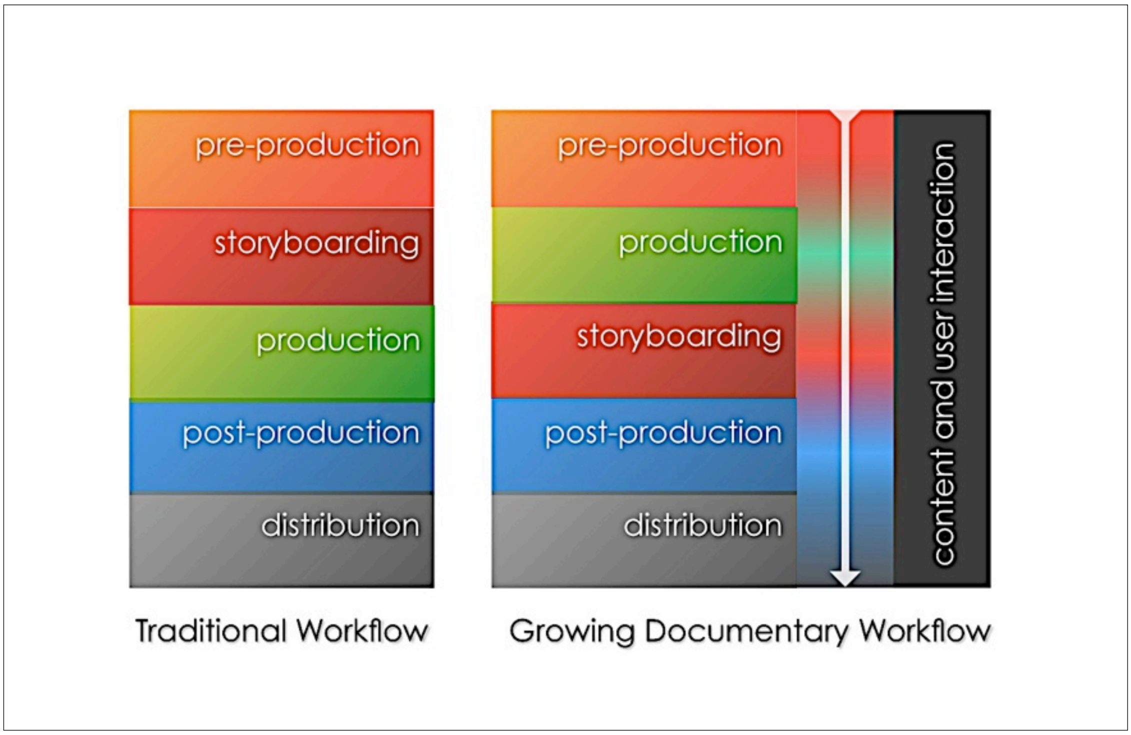


Figure 2.1. Traditional film workflow compared to the Growing Documentary workflow.

To get the themes of the final movie, metadata entry was an essential activity. This process was done by going over the videos and taking notes within the PIX program. PIX allowed the synchronization of files and metadata between KMD and UCSD members, where multiple users can view and comment on the videos at any time.

2.2.2. SAGE (Scalable Adaptive Graphics Environment)

SAGE is a graphics streaming architecture with a focus on high-resolution and data-heavy applications that permit the same level of sharing, display, and accessibility of content allowed by the web on professional level [Renambot]. This environment was used during the production of "places + perspectives". The scalability, interactivity, and power of SAGE along with the large real estate allows the simultaneous view of different media assets. By applying these strengths during the production of “places + perspectives”, KMD and UCSD experimented with SAGE as a “storyboarding” tool in discussing collected content where the video clips and images could be rearranged in different sequences, permitting both sides to work on the flow of the movie. Furthermore, the capability to share multiple laptop screens on the unit’s display gave team members the opportunity to use different software applications that were not natively supported by the SAGE OptiPortal environment such as non-linear editors and DAM systems [Bhimani].

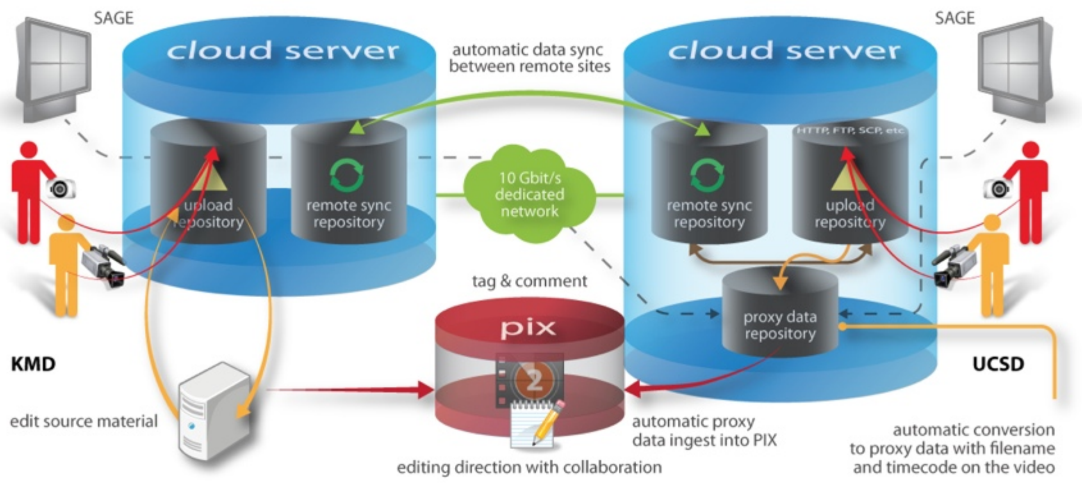


Figure 2.2. The network infrastructure, tools, and hardware used in “places + perspectives”.

All of the aforementioned uses were done on a version of SAGE that was catered for scientific data and research. As a result, there were many usability issues that arose when it comes to media handling and management. Some of these issues are being addressed by the Electronics Visualization Lab (EVL) in collaboration with KMD and UCSD (Figure 2.3). Moreover, the use of touchscreen technology as a central interface method in SAGE means that some of the traditional concepts of UI and user interaction needs to be altered or rethought to fit such input methodology.

Figure 2.3. Shared applications from a PC/Mac can now be controlled on SAGE by different users using the SAGE pointer

2.2.3. The Mill Touch Application

Advertising and visual effects agencies tend to procure a large amount of produced content. While this adds credibility to the agency's work portfolio, it also presents some difficulty navigating through various content. This was one of the motivations at The Mill [2] creative agency to provide clients with access to their portfolio in a more interesting and unique way (Figure 2.4).

The creative team at The Mill began working with the Cinder [3] programming library, which they had released as an open source C++ library. Through the various iterations and prototypes, many of the graphics and animation lessons learned from Adobe Flash and Actionscript were implemented in the library. The Mill's outcome for their gallery program was an unconventional UI with a mix of animations and multi-touch interactions. In addition to the software component, they built their own customized touchscreen device using infrared lasers and sensors, a projector, and a switchable glass panel. The final result was very well received by their clients and they enjoyed the experience more than sitting on a small PC screen using the keyboard and mouse.

[2] http://www.themill.com/work/mill-touch/behind-the-scenes.aspx

[3] http://libcinder.org/

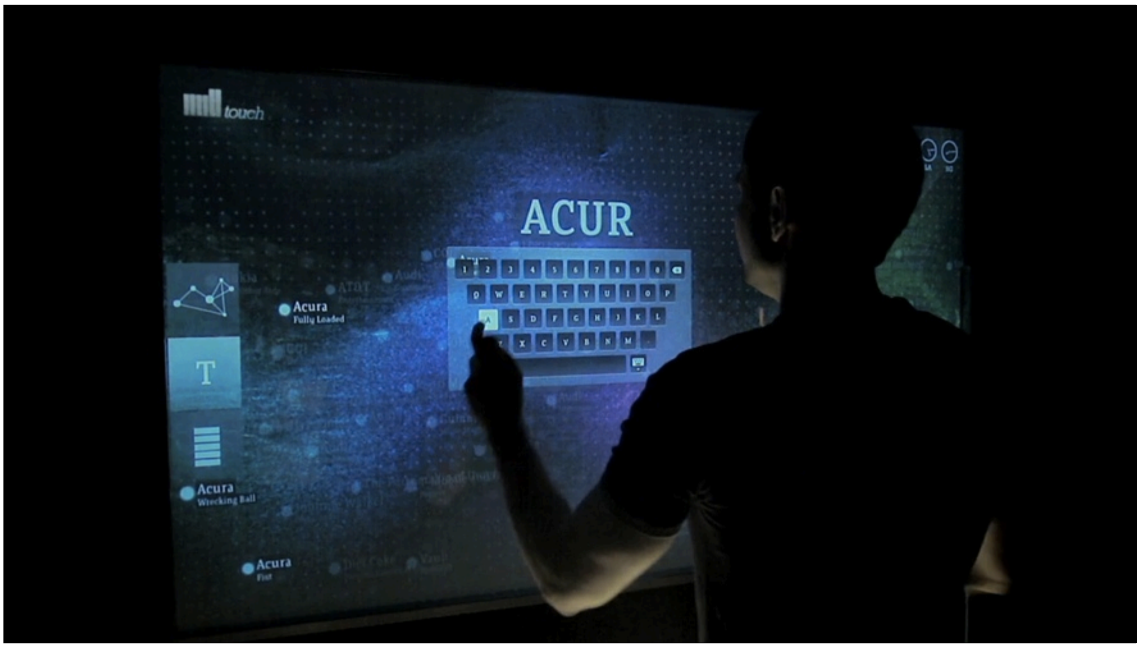


Figure 2.4. A user navigating the Mill's content library using their touchscreen application.

2.3. Design Inspiration

With the current advancements in graphics technology and with the introduction of new device form factors such as tablets and smartphones, some applications (including desktop ones) eschewed the conventional GUI to create both beautiful and functional products. Such applications take advantage of the changing demographics of users along with the technological advancements to create unique experiences.

2.3.1. DaisyDisk Application UI

There are many solutions on the market for visualizing disk space usage but few have the aesthetics and functionality of DaisyDisk[4]. Many users praise the application's design and functionality as DaisyDisk allows users to view the contents of their storage devices and organize or delete files/folder via a vibrant starburst-style graph navigation (Figure 2.5). Software Ambience, the application's developers, transformed viewing and interacting with large data from boring and casual into something more exciting [Carlson]. What is inspiring about this application is that the developers focused on the *experience* more than the app's features[5]. They also started with the design concept and worried about the code later allowing them to explore different ideas and not settle for standard ones.

[4] https://itunes.apple.com/us/app/daisydisk/id411643860?mt=12

[5] http://www.daisydiskapp.com/blog/2011/05/29/10-basic-principles-behind-daisydisk/

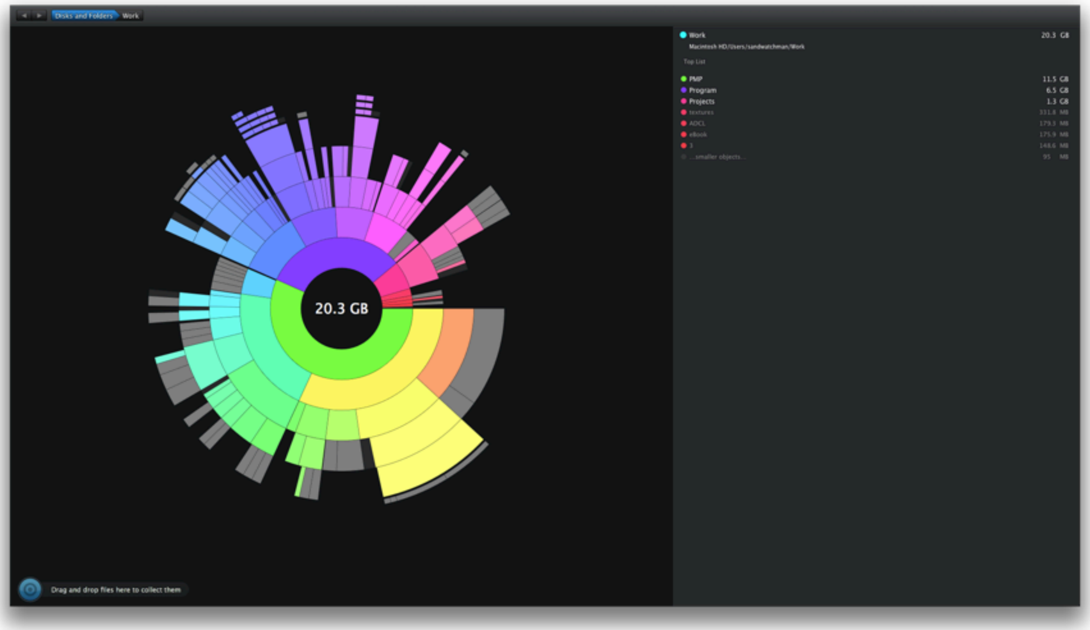


Figure 2.5. DaisyDisk uses a graph view as both an infographic and navigation UI.

2.3.2. Wipeout 2048 Game UI

With the increasing power of videogame consoles, the user interface takes on a different shape. Games rather than having a single GUI, they offer various interaction methods that experts describe as a FUI (Fun User Interface), which is more seamless than standard GUIs [Wigdor]. One videogame franchise that helped fuse traditional graphic design with interactive graphics is the Wipeout franchise, a series which debuted on the first PlayStation in 1995.

Throughout the years, the franchise has been praised not only for its gameplay, but for its graphic design and use of unique interfaces. The most recent game is Wipeout 2048, which debuted in 2011. The game provides a unique UI display system and enhanced interaction through a multi-touch interface. Both the visual style and method of interaction play a key role in navigating the game's content and provide excellent interaction using the PlayStation Vita sensors. Designer Edd Wainright[6] used both soft and physical prototypes to test the idea behind the interface before committing it to the game (Figure 2.6). Nevertheless, the results were quite eye-pleasing and functional at the same time.

[6] http://1280x720.co.uk/WipEout-2048-User-Interface

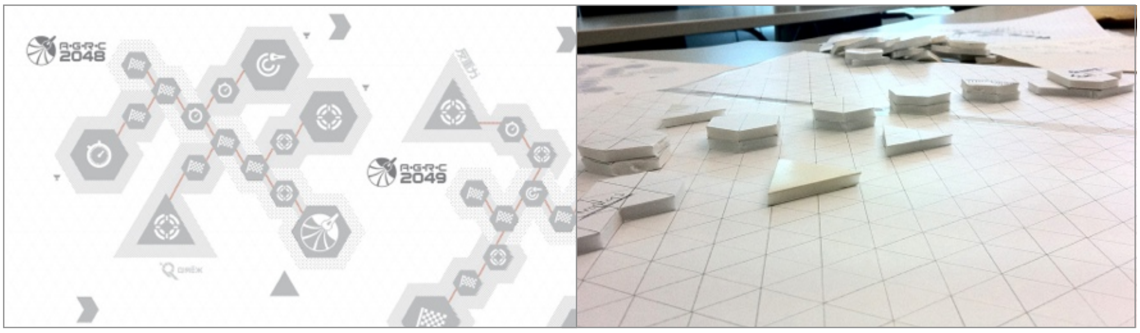


Figure 2.6. Prototyping methods used for the new Wipeout 2048 game.

2.4. Tying it Together

Not only does media production involve many people, it also requires various tools and methods to create exciting content. There is a lot of potential in improving the production cycle by integrating, merging, or rethinking some of the techniques mentioned above to create new ways of content interaction. The focus of this paper will be around the theme of visual interaction and how to primarily improve and enhance media production though improved interaction concepts.

Chapter 3

3. Design Requirements

In order to investigate the topic of this paper, various research methods were employed. This section highlights the methodologies used to define the design requirements and considerations for the target audience:

1. Highlight some of the observations made during the production of "places + perspectives" such as workflow and production bottlenecks.
2. Look into some of the experiments done using touchscreen devices.
3. Define the target demographic through an interview.

3.1. Observations from "Places + Perspectives"

The production of places + perspectives relied on video interview and B-roll footage shot by KMD and UCSD. The interviews themselves are approximately 10-12 minutes long and span different topics. Most of the clips were shot in two to three takes, each lasting the maximum amount allowed by the DSLR cameras for HD video recording (approximately 12 minutes). A rough edit was done on each clip to have only the interviewee's answers within the interview. After gathering a sizable amount of footage, the PIX System was used for organizing and annotating the interview clips (Figure 3.1). Consequently, the teams on KMD and UCSD sides started categorizing the files depending on the interviewee's reaction and stories. Here is a brief overview of the categorization process:

1. First Browse-Through

   In this stage, the interviews are brushed through by different team member and the general themes were established based on this quick overview. The themes were then tallied to see how how frequently they occur within the interviews.

2. Identifying the Flow

   The general themes were then organized in a thematic timeline that served as a guide for the story flow and later, the final editing of the movie.

3. Establishing Tagging Rules

   Tags were then generated for each portion within the storyboard timeline and they served as the standard tags when browsing through the content for the second time. These tags helped identify interviews that were relevant to the story.

4. Assigning the Tags and Annotating

   During this phase, the content was thoroughly annotated by transcribing the relevant scenes and the tags were assigned accordingly. The different team members on both sides worked on the footage they shot and tagged the content using the timeline tagging and annotations within PIX. Thus, a single video clip could have multiple tags/ annotations assigned to it at specific time codes allowing for easy retrieval.

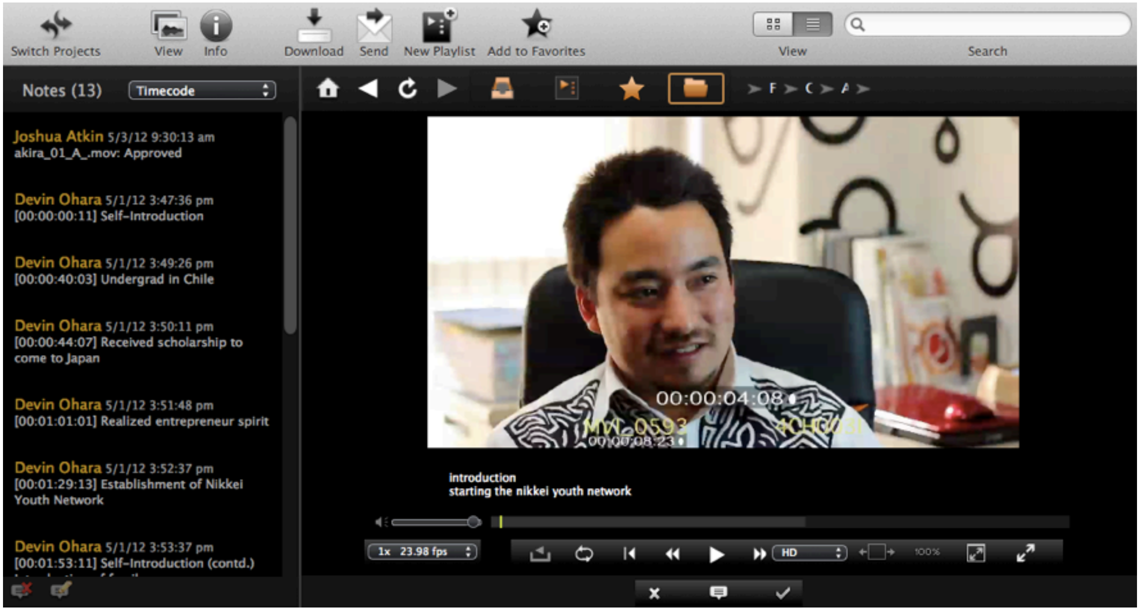
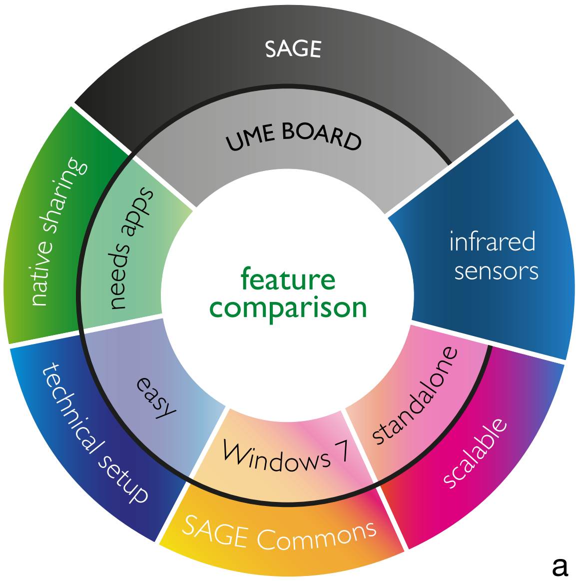


Figure 3.1. The PIX System annotation and video player interface.

3.1.1. SAGE and Universal Media (UME) Board

As a continuation of "Places + Perspectives", more events and demos were conducted using multiple remote SAGE units to showcase the collaborative sessions. But as the KMD team was planning for the EuroITV 2012 Conference in Berlin, they faced some hurdles for a demo session due to budget and the transportation restrictions attributed to the size and weight of the SAGE unit. As an alternative, the KMD members experimented with the Universal Media (UME) Board interconnectivity with SAGE in Yokohama through a one-way communication using SAGE Pointer (Figure 3.2 b).



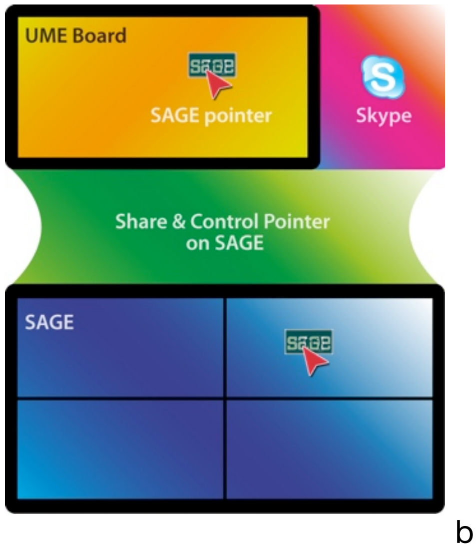


Figure 3.2. a) Feature comparison between SAGE and UME Board. b) The connection between SAGE and UME Board.

UME Board (Universal Media Board) is a multimedia device that can be used with little setup compared to SAGE. Similar to SAGE, UME Board uses infrared sensors allowing for multi-touch interaction by users (Figure 3.2. a). UME Board is (currently) preloaded with Microsoft Windows 7. It contains a free-writing memo feature allowing real-time annotation. Due to its size and (relative) affordability, UME Board was a practical and feasible tool for the demo session (Figure 3.3). This test proved to be a good testbed for new collaboration models using a multi-device framework. The demo session led to trial and test new Windows 8 tablet devices with unique form factors (e.g. the 20" Sony Tap 20) along with other touchscreen input devices.

3.1.2. Touchscreen Devices Usage

After acquiring the new Sony Tap 20, some of the KMD members started tinkering with the device to see the features that it provided. The device does come with a keyboard and mouse but most of the test interactions were done using the touchscreen. One of the quick observations noticed when testing different multi-touch devices is the user's tendency to use familiar gestures for certain outcomes. Users tend to think of the goal (e.g. expanding) and recall a certain gesture (spreading). This sometimes causes problems as gestures differ from one device to another and could throw some people off. So the interactions need to be more clear to the user to use them more effectively.

Figure 3.3. Demo session between SAGE (Yokohama) and UME Board (Berlin) at EuroITV 2012.

3.2. Target Demographic

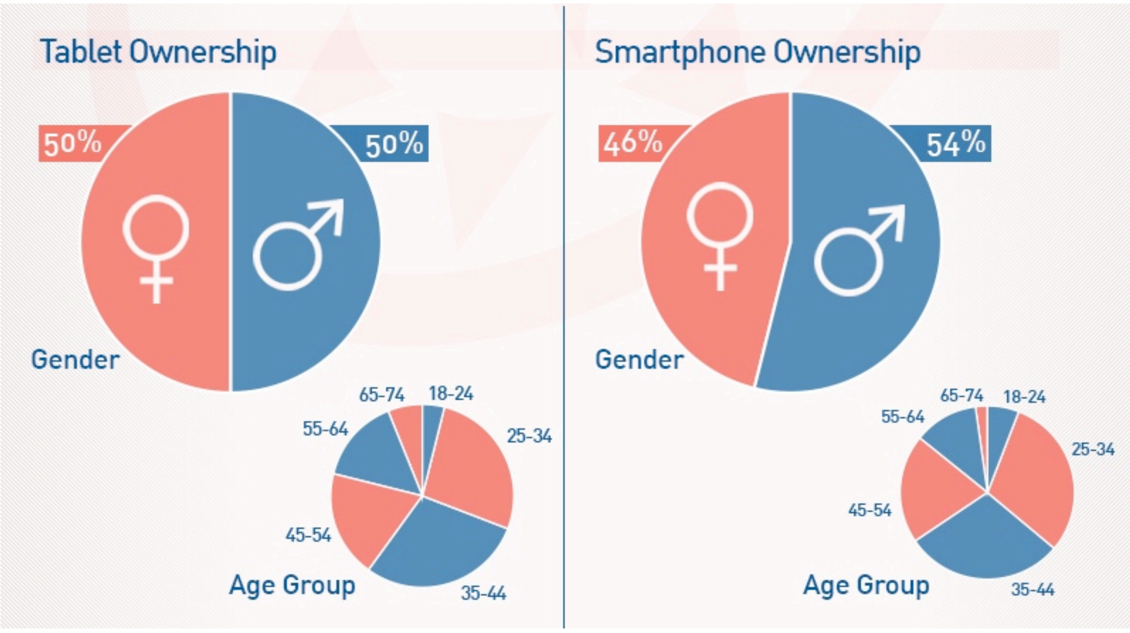


Figure 3.4. Touchscreen device ownership amongst target demographic. (eDigitalResearch, 2012)

The age of the target audience falls between 20~35, they are motivated individuals who are quite adept at new technologies. Their interests include video production whether it is in casual or semi-professional capacity. They do not have to necessarily understand the underlying technologies behind the products, but know how to operate them and which tools to use. In addition, the target demographic should have access to a touchscreen capable device. According to market research, smartphone and tablet ownership is high within the age group specified (Figure 3.4).

This demographic probably has enough exposure to social media aspects. In order to understand the target audience more, an interview was conducted to better understand the user's motivations and methodologies when editing videos.

3.2.1. Interview and Insights

The interview was conducted with a female participant within the 20~25 age demographics. She has good experience in film production and editing with a background in film arts. For the most part, the questions revolved around her production workflow, here is a summary gathered from her interview (Appendix a):

- Always interested in creating new video contents.
- Started gradually with iMovie then Final Cut Pro then Adobe Premiere.
- Cameras: DSLR and Prosumer equipment.
- When filming videos, she does is it according to a schedule, and the number of takes depends on the length of the video e.g. a 40 seconds video would take two hours, and twenty takes in general.
- She usually shoots and edits by herself mostly but she works with her friend, and work via the cloud when needed.
- Starts editing usually after the shoot date. Her first browse through the files is about the quality of the clip, and is usually done one by one (Figure 3.5).
- She remember things visually such as the place, the scene, etc.
- Multiple takes: She remembers the sequences and takes in a chronological order.
- The naming of the file is not that important, length of the movie is (e.g. 2 seconds clip, does not want to bother watching it).
- When browsing files, UI is a major drawback, maybe an iPad to play videos more smoothly. Looking at thumbnails would help and make browsing easier.
- Metadata use is light as it is a bit disruptive, maybe due to unintuitive UI.

- She needs “a way to organize content more properly.”

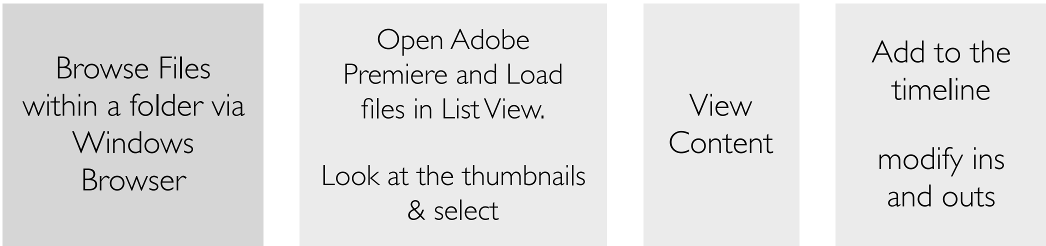


Figure 3.5. Browsing the video files for editing.

Based on the interview outcome, it provides a good scenario to build more visual-driven interfaces. By mentioning the iPad, this tends to show the user’s acceptance of new methods of interaction, if they are useful and intuitive. Furthermore, the visually driven nature of the interviewee is reflected in what she values in a user interface. Some of the concerns voiced here such as remembering places could be solved, for example, through automated metadata along with manual input.

3.3. The Design Requirements

Based on the previous discussion, these are the user requirements to address:

1. UI Scalability & Adaptability
   The interface should take into consideration scalability in terms of device deployment (large screen, tablet, smartphone) and also should be able to accommodate single and multiple users. The concept should be used not only in environments such as SAGE but also regular setups such as desktops and tablets.

2. Enhancing Productivity
   One of the most important elements in a production environment is time. with many deadlines to meet. All of the people involved, and even the hardware and software components should facilitate delivering content on time. So, the concept and prototype should enhance efficiency and productivity.

3. Utilize the Touchscreen Advantages
   The concept should take into consideration the advantages that touchscreens add and utilize them for a better user experience.

Chapter 4 investigates touchscreen interaction within the context of these requirements

Chapter 4

4. Prototype Design

After defining the users and the design requirements, this chapter examines the unique attributes of touchscreen devices themselves in order to realize those requirements. The following topics will be discussed throughout this chapter:

1. Review some of the background research done on touchscreen interfaces, keyword tagging practices, and how the keyword tagging concept came to fruition as a result of these findings.
2. Discuss the soft prototyping and testing that lead to the development of the high-fidelity prototype.

4.1. Survey of Current Keyword Tagging Practices

4.1.1. Entry Method

In most desktop applications, metadata tagging involves keyboard input for new metadata fields and keywords. Afterwards, users are presented with checkboxes or predefined tags to assign the relevant keywords to content. For small batches of files with few keywords, this process is simple but becomes cumbersome with multiple tags and a larger file set. Moreover, having detailed keywords for motion picture fames becomes a burden as videos sometimes contain various scenery, sound, and dialogue cues.

Unlike photos, the time element complicates matters further and long video clips become harder and tedious to annotate/ tag properly. This became a problem when generating tags and transcribing interviews for "places + perspectives", and thus came the motivation for researching alternative input methods. The KMD team experimented with different software solutions for automated voice transcription but the results were inaccurate and were sometimes far from the topic discussed by the interviewee. The team opted not to use this method as it could result in more effort than manual transcription.

As good as manual transcription is, it is a time consuming process. In order to cut the time, both teams at KMD and UCSD filtered the most interesting clips from the gathered footage and then decided to only transcribe the selected footage.

Now, here is a look at the current keyword selection methods used in today's software.

4.1.2. Selection Method

There are a number of solutions that already provide keyword tagging interfaces both on an OS level (such as the Mac OS X), application level (Adobe Creative Cloud, Apple Final Cut Pro X), and web-based services (such as tumblr, Instagram, YouTube). Looking at these interfaces closely, they typically fall under these general categories show in Figure 4.1:

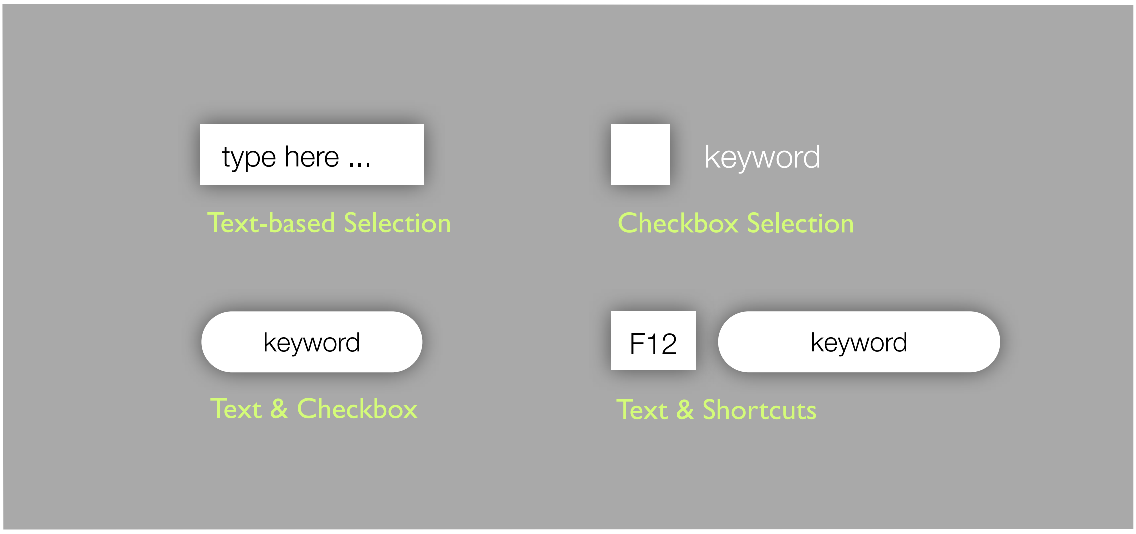


Figure 4.1. Four general methods of data selection.

- Text-based Selection

  Users type their desired tags separated by commas. The program would detect the individual tags. Some software would encapsulate the tags within a box. Also, web services such as tumblr allow autocomplete to easily select pre-existing tags.

- Checkbox Selection

  Multiple standard checkboxes that are widely used since the first computer GUIs. They are usually represented by an empty box that gets filled when the user makes the selection. Most systems limit checkbox selection to one at a time.

- Text & Checkbox Selection

  This method combines both of the aforementioned methods by displaying the tags encapsulated within a box. When the user clicks on a tag, it becomes selected and added to the desired content.

- Text & Keyboard Shortcut

    Found in Final Cut Pro X, which now offers more robust metadata management tools than before. As the user types the keywords, they get added and the user could be assign keyboard shortcuts to add keywords within the video clip timeline.

The majority of the aforementioned input methods are catered towards desktop solutions and most mobile applications still adopt some of these methodologies.

4.1.3. Timeline Segments as Events in Final Cut Pro X (FCPX)

With the addition of the metadata management in FCPX, individual frames or a range of frames from a movie clip could be tagged with different keywords. Hence, the addition of the keyboards shortcuts (Figure 4.2), along with other methods such as smart folders, to allow for more efficient and precise keyword assignment. These scenarios are similar to those experienced in "places + perspectives" where the interview clips span different topics within their 10-12 minutes runtime. Therefore, keywords in this context would allow for a more concentrated effort to segment the clips via keywords rather than slicing them. This would allow for a different approach to content management and movie editing that would preserve the original clips though the use of metadata.

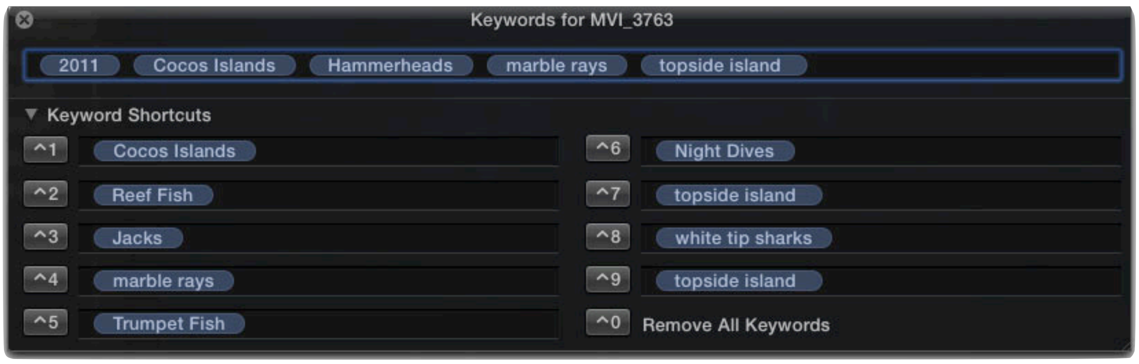


Figure 4.2. FCPX shortcut assignment for keywords.

Still, shortcuts and other advanced features are usually utilized by advanced users that use this system on a regular basis as they build their different libraries, workflow and methodologies for their projects. So, the challenge now is to find efficient ways to fully utilize the touchscreen's strengths in the same manner to how FCPX uses the keyboard shortcuts to capitalize on the advantages of the keyboard. At the same time, this method should also be accommodating to the different user levels.

4.2. Research and Development

4.2.1. Technology Background Research

Looking at current and previous software design patterns, it becomes clear that the methods used to develop an interface are closely related to the way the user interacts with that interface. Mouse interactions, as an example, have a rapid and precise movement, they are *pointing* devices and are not suited for path following [Wigdor]. Additionally, the mouse follows Fitts's Law, which states the time it takes to move to a target is a function of target size and distance to that target [Lidwell]. These examples help explain how the current desktop GUI patterns evolved as an interaction method.

Looking at the touchscreen environment, it employes and imposes a new set of rules and "primitives" to user interaction according to the Brave NUI World book [Wigdor]. Primitives are defined as the foundation rules of an interaction language that take into consideration the types of interactions that a device is capable of receiving and transmitting. Touchscreen devices allow for more smoother and continuous movement compared to the mouse. People could easily, for example, draw on an iPad or Android tablet, but the mouse makes it more difficult to do so.

Another example illustrating some new design primitives is shown through the use of a new primitive for pen stylus interaction with checkboxes. Instead of using a click primitive, which is a mouse action, a *cross* primitive could be used to select one or multiple checkboxes at the same time [Wigdor]. This would allow the best use of the pen technology, which is better for continuos motion.

4.2.2. Concept Development

By taking the previous principles into consideration, the 1st prototype was aimed at creating a different checkbox shape and mechanism that is catered toward touchscreen devices. Based on various observations, horizontal swipes appear to be more common in touchscreen interaction than vertical ones. This lead to the stacking of the checkbox elements together horizontally to allow multiple selection through the use of swipes (Figure 4.3). Furthermore, many studies and articles show the drawbacks of using vertical navigation in websites as it contradicts with the way people read, which makes horizontal checkboxes an interesting test scenario [Lazaris].

4.2.3. Early Prototype Testing

The prototype was first tested in paper form. The aim at this stage is to identify any design complications with the concept. For this stage, the RITE (Rapid Iterative Testing & Evaluation) method was used to test and gauge the feedback from users. RITE is used as a way to identify any usability problems before committing any code to the program. The method is aimed to test low-fidelity version prototypes with various users, experts, and fellow team members [Hanington].

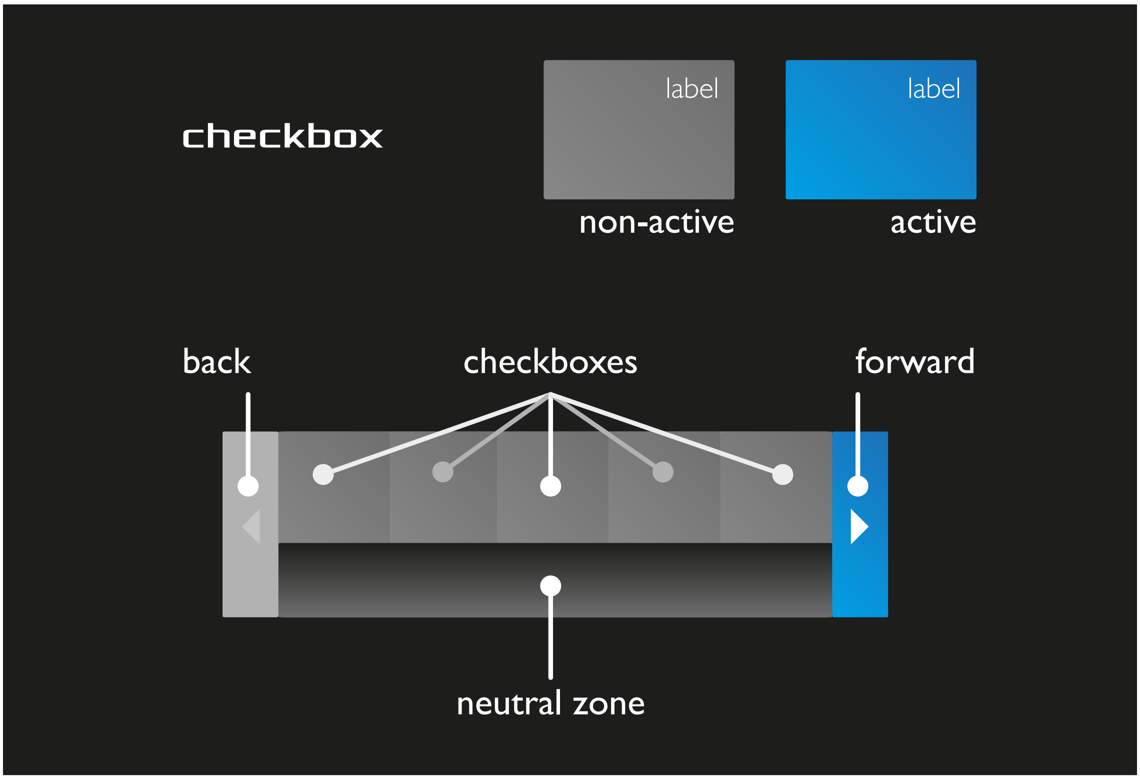


Figure 4.3. First concept prototype.

The test concentrated at this stage on the idea of using swipes, taps, and other single gesture patterns. Thus, the comments that were directly related to the design and functionality of the checkbox selection were considered. Based on how positive or negative the comments are, they determined whether the prototype failed or succeeded. However, other comments/ suggestions were taken into consideration to test or apply during the high-fidelity prototype stage. The set goal was to have 5 consecutive successes.

The final result of this experiment is 6 participants and two paper/ low-fidelity prototypes:

Participant 1: Fail

When the prototype was presented to the first participant, he thought the swipe motion was constricted to a certain region on the screen, which might hinder the movement for different individuals. Also, the participant thought that swipes were the only method of interaction. Thus, he pointed that clicking or pointing should also be included within the supported input methods. Figure 4.4 shows the revised version.

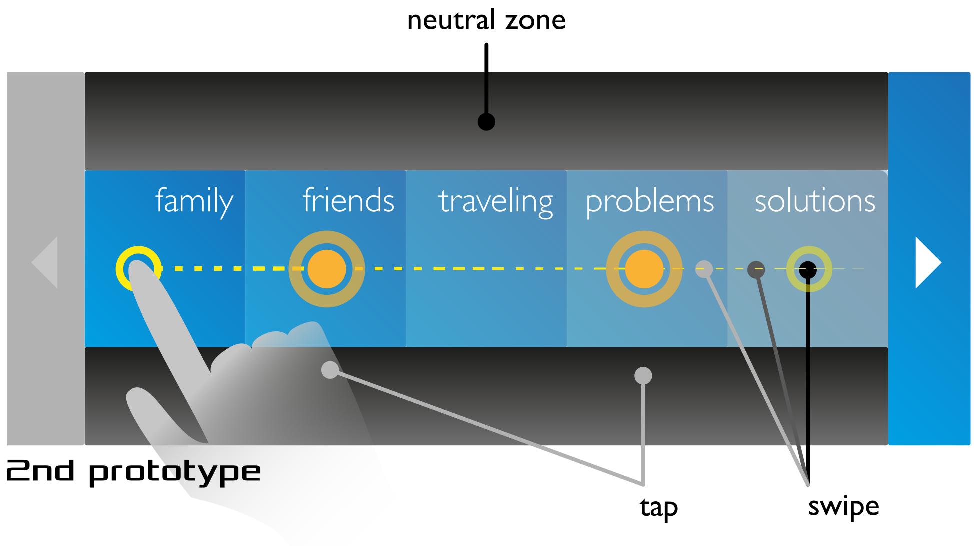


Figure 4.4. Revised prototype.

Participant 2: Success

There was interest in the concept from the second participant. The only concerns were defining scenarios to which this prototype could be tested and used.

Participant 3: Success

Response was positive but they also commented about situations with too many tags and how it could be implemented with gestures.

Participant 4: Success

The participant showcased a different approach to handling tagging on the checkboxes by using two thumbs instead of the index finger to select them. Also, they showcased some of their favorite applications with unconventional UI navigation.

Participant 5: Success

One of the comments highlighted that the checkbox size is big and could be used for different age demographics. Additionally, the participant wondered about the two screen scenario in terms of usability and added value.

Participant 6: Success

The participant liked the idea of swiping instead of tapping each box but did note that an indicator or method is needed to show that swiping could used as a selection method.

4.2.4. Prototype Results

Consequently, the participants' feedback provided valuable insights about the usability of the concept while at the same providing scenarios for its implementation. In particular, the first and fourth users offered new perspectives on how the concept could be used. Overall, the concept behind the prototype was cautiously accepted by the participants as most of them wanted to see the concept in action.

Furthermore, this stage has also added another consideration for touchscreen interaction: the size of objects within an interface needs to accommodate for thumb usage. The initial concept was designed primarily for index finger interaction and not thumb use.

Traditionally, UI elements are measured by pixels, which is quite evident in web design and GUI design where the screen resolution determines item sizes and the spaces between the different elements. However, research shows that measuring UI elements in a touchscreen environment should be done with real world units rather than pixels. Various resources state different minimum sizes for interaction elements such as 1.6 cm in large touchscreen displays to accommodate for user movement [Wigdor]. Another research states that the size should be around 9.2 mm for individual tasks and 9.7 mm for serial tasks on smaller screens [Parhi]. This same paper was targeting single-thumb use on smartphones, which could be adapted to meet the two-thumb input scenario proposed by the fourth participant.

The next step is to create a series of prototypes and tests building on the design requirements, touchscreen technology, and early usability testing. Chapter 5 discusses the methods used for creating the prototypes and tests.

Chapter 5

5. Implementation

Development of the high-fidelity prototypes started shortly after gathering the design requirements and user feedback from the initial testing. The goal of this phase is to find the most suitable environment to test the prototype

5.1. Prototype Aim

One of the aims of the prototype is to test touchscreen interaction, such as the use of swipes, taps, etc. The prototype would use images as simplified "events" to simulate the FCPX events but in a much smaller and controlled scale. Additionally:

1. Photo tagging is more commonly used in social apps (Instagram, etc).
2. Photos are much easier to test than video; there would be no timeline-related issues. The focus would be on the keyword selection only and not on frame or timeline selection and keyword selection simultaneously.
3. The prototype and tests are more focused on interaction. Other scenarios could be implemented later based on the results and modifications.

5.2. High-Fidelity Prototype Development

At the beginning of the development, the Processing development environment was chosen due to its simplicity, accessibility, and overall familiarity with it. The initial tests proved to be satisfactory when using Processing with some of the free libraries but the results were deployed on a regular desktop environment. Processing's latest version allows for Android deployment but after some tests and due to the limited availability of the testing device (a 2012 Nexus 7), this testing environment presented some stability problems and troubleshooting problems. Due to the unfamiliarity with the mobile development environments, another testing methodology was needed.

5.2.1. Prototype Development

Upon further investigation, the quickest and easiest way to proceed with the prototype development relied on second-screen applications for the iPad. By using the iPad as an extended monitor, this allowed the deployment of the prototype on a touchscreen device. Thankfully, the majority of these applications not only allow the extension of the desktop onto the iPad but also allow interactivity through the touchscreen directly.

Different applications were tested for their fidelity, response time, and overall stability. The application that had acceptable performance in the aforementioned criteria was the XDisplay alongside with its desktop companion, Splashtop Streamer. With this application, the testing environment was established as seen in Figure 5.1.

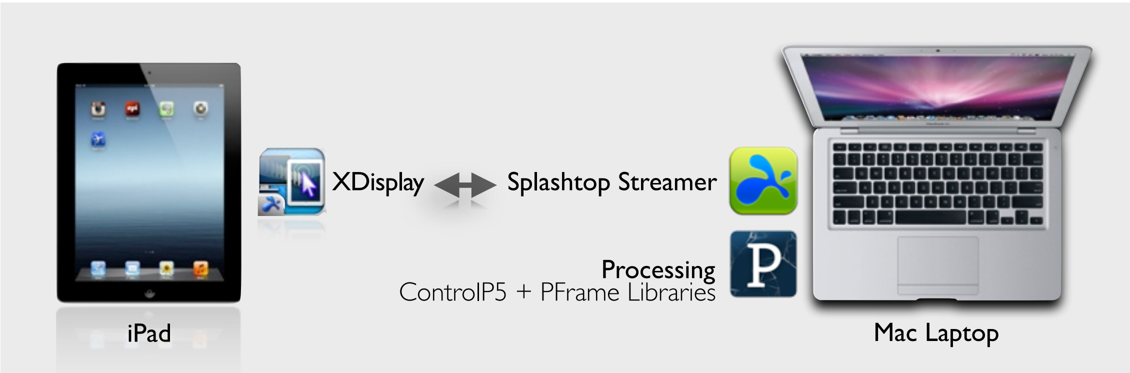


Figure 5.1. System schematic showing the hardware and software used.

In addition to the main Processing libraries, the ControlP5 library was used for the UI development. The ControlP5 library provides a robust array of tools ranging from GUI elements such as buttons and checkboxes to more advanced techniques such as event handling and mouse cursor replacement. Furthermore, the library contains very detailed examples of almost every function within ControlP5. Finally, the documentation was also quite helpful to understand how ControlP5 operates.

5.2.2. Limitations of the System

The initial test started with the checkbox function to see how it behaves on the XDisplay application. As it turns out, XDisplay presented the following limitations:

- Selection Behavior
  Upon further investigation with the XDisplay application, it uses *drag* instead of *point and click* for mouse simulation on iPad. So as the users moves their hands across the iPad, the application registers that movement as a "dragging" movement. When the users point and click on a checkbox, the mouse cursor moves to that area but also stays there. Thus, the behavior added a sense of confusion when using a combination of swipes and taps to select the checkboxes.

- No Multi-touch Gestures

  Since XDisplay simulates the iPad interactions as mouse movements, it didn't allow for multi-touch input. So, the test was limited to single-touch gestures such as tapping and swiping on the checkboxes.

Due to these limitations, the checkbox within ControlP5 function was not used since its behavior within XDisplay's limitations was not as intended. Based on that, the mouse cursor was disabled in the Processing window using ControlP5 functions and replaced with a custom Pointer from the library. This custom cursor allowed for easier control over the prototype's behaviors within XDisplay.

5.2.3. Modifying the Prototype's Behavior

Additionally, a ControlP5 button example was extended to override some of the behaviors of the mouse to simulate the taps, swipes, and the behavior of checkboxes. Here is an example of the overridden functions in the custom button class (Appendix g):

- `onEnter()`

  When the user taps on a checkbox, it will change depending on its state: a non-active box will activate and vice versa. An IF statement was used to flip the checkbox state and set the trigger state to TRUE.

- `onDrag() and onRelease()`

  When the user swipes over a checkbox, it will activate/ deactivate the checkbox.

In addition to these class modifications, the main program also used the ControlP5 Pointer functions to override the cursor function:

- `mouseReleased() and mouseDragged()` functions

  Both of these functions were set to: `cp5.getPointer().released().` This also overwrote the mouse behavior on XDisplay.

With these modifications and others done, the prototype behaved as intended; taps or swipes over a row of custom checkboxes could be done smoothly just as they would on a native touchscreen application. And in addition to these modifications, the prototype code included timer and logging features. The data-logs contained the users' activities such as total time for the session, time spent between clicks or selections, and name of chosen keyword tag (Table 5.1). Moreover, all this data is written to a text file to process it later using spreadsheet applications. The name of the file had to be done within the Processing coding area itself and there was no GUI for it.

Table 5.1: A sample data set showing the logged items within a test session.

Image ID	Time	Elapsed	Ms	Box status	Box No.	Tag Name
1	7:59 PM	0:00:02	2365	1.0	11	E g y p t

Two touchscreen prototypes were created using the Processing development environment. Aside from the standard libraries and the ControlP5 library, the PFrame library was utilized to allow multiple windows in Processing. In addition to the two aforementioned prototypes, another program was created to simulate existing checkbox-driven interfaces. Again, all these prototype programs contain data-logs of the users' activities.

5.3. Prototype Test Design

In order to test the design of the prototype, a total of 30 images were gathered from morgueFile[1] (Appendix b). The collected images were then categorized depending on their complexity and amount of objects within them. This pertains to the scene details within each image such as the number of people and vehicles, or landmarks of famous cities (Figure 5.2). Afterwards, the images were split into two groups (Set A and Set B) with 15 images in each.

The keyword tags were then generated depending on the scene objects, people and locations. They are categorized into three different groups as shown in Table 5.2:

Table 5.2: 15 keyword tags used for the test.

Vehicles	Boat	Bus	Car	Train	Truck
People	Boy	Girl	Man	Woman	Crowd
Places	Egypt	France	India	UK	USA

[1] http://www. morgueFile.com

Figure 5.2. A comparison between the images in Set A and Set B.

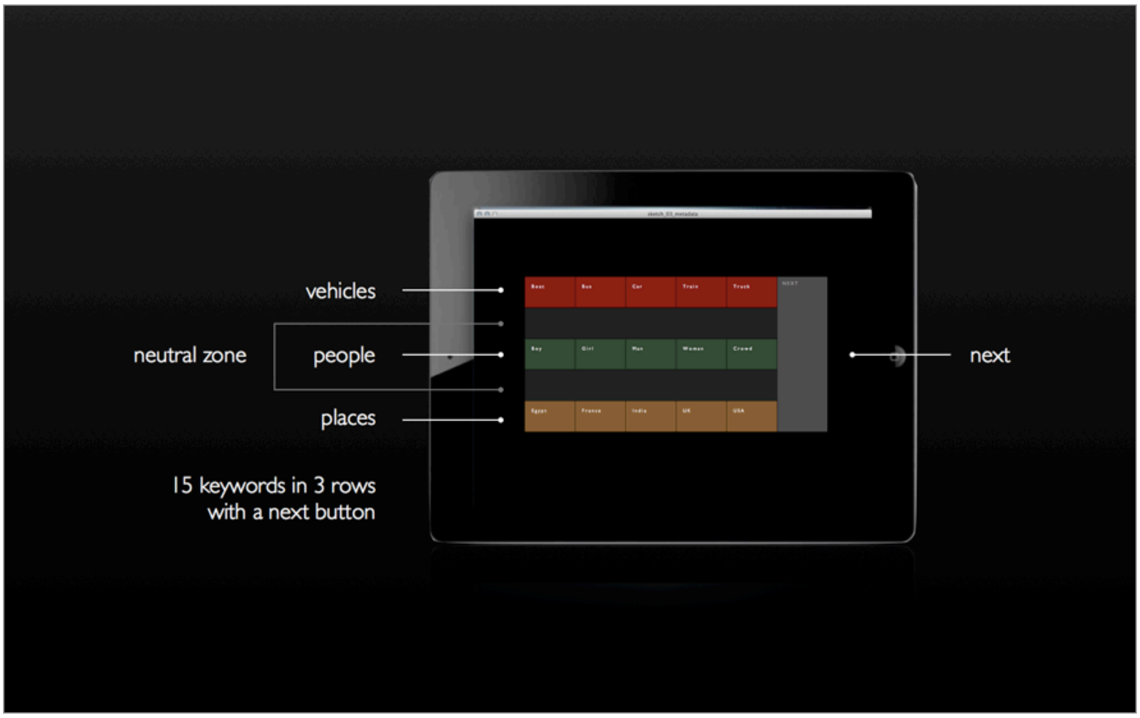


Figure 5.3. Layout for the touchscreen applications.

The test was designed to investigate the usability of the prototypes through different scenarios. *However, the usability testing was not intended to test the information design or organization methodologies of keyword tags.* Rather, the interaction was the key testing aspect here. The sequence of the keywords was the same for all test scenarios and the only difference was the horizontal rows in the touchscreen applications (Figure 5.3). The three prototypes had the same order of keywords and that order was done based on alphabetical sequence except for the "People" row. Despite the primitive nature of this arrangement, it allowed to test different user actions such as individual taps, horizontal swipes, and even vertical swipes between two rows.

Additionally, in order to focus the usability testing, only the Next button was introduced to the users with no Back button. There were different reasons for this decision:

- Technical Limitations

  The limitation of the prototype in its current state of testing would not allow the users to view what they had done before. They would need to start the keyword tagging for that image from scratch despite the fact that all their choices are saved and exported through the data-log. Yet, there was not any function to display these choice within the current interface.

- More Focused Approach

  Adding the Back button would make the test longer as users would flip back to the previous images to correct or check on their choices. Also, it would be more difficult to determine whether the interface is the reason behind the error or the user's habit.

5.3.1. Prototype Testing Scenarios

Three different scenarios were devised for the testing:

Scenario 1: Traditional UI

The Adobe Bridge CS5 served as the base model for this scenario. A group of checkboxes from the ControlP5 library were used to imitate the checkboxes found in the aforementioned application as depicted in Figure 5.4.

Test Setup:

This task uses the first set of images (Set A) and introduces the keyword tags.

Purpose:

Provide users with a familiar user interface and serve as a benchmark for the later tests.

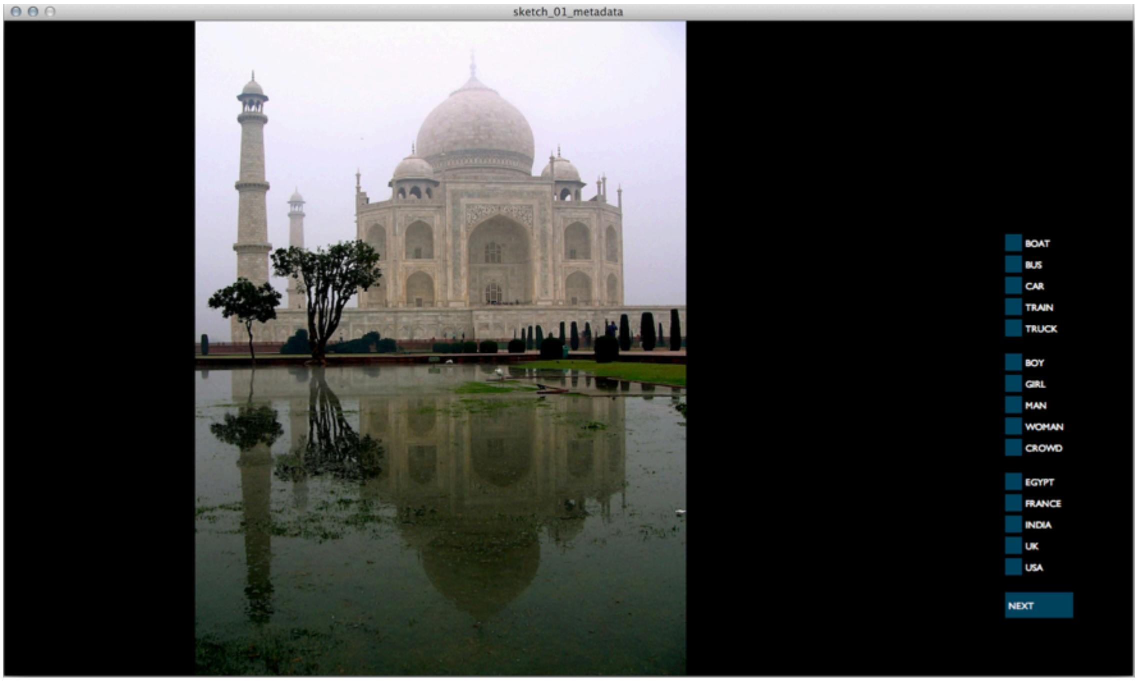


Figure 5.4. Traditional UI

Scenario 2: iPad UI

The users are presented with an iPad that includes a new UI. Participants use only the iPad to view and tag the images (Figure 5.5). Using the XDisplay application, the Processing code was directly displayed on the iPad.

Test Setup:

This task uses the second set of images (Set B) and introduces the new interface.

Purpose:

Test how well and how fast would the users adapt to the new UI.

Figure 5.5. iPad UI.

Scenario 3: Second-screen UI

The users are presented with controls on the iPad and the images on the main screen (Figure 5.6).

Test Setup:

This task uses the first set of images (Set A) and an expanded iPad UI in a horizontal position compared to the vertical one in Scenario 2.

Purpose:

Test how users adapt to second-screen input and gauge any improvement due to UI and image recognition.

Figure 5.6. Second-Screen UI.

5.4. Evaluation Sheet

With these three test scenarios, an evaluation sheet was created to collect various user data (Appendix c). Some of the questions were related to the ease of use and enjoyability of the interfaces. Additionally, the I-PANAS-SF was included to capture the participants' mental condition before they start the tasks and after each task [Thompson]. Here is an example of some of the questions:

How do you feel today?

1	2	3	4	5
Very slightly or Not at all	A little	Moderately	Quite a lot	Extremely

Active	_______	Nervous	_______	Upset	_______	Determined	_______
Hostile	_______	Inspired	_______	Attentive	_______	Ashamed	_______
		Alert	_______	Afraid	_______		

Some sample questions from the evaluation sheet include these recurring questions after each of the three tasks:

How easy was it to tag the images?	Difficult	Ok	Easy
How much fun was the process?	Not fun	Ok	Fun

At the end of the usability testing, the following questions were asked to the participants:

What was more enjoyable?	Traditional	iPad only	Two screens
What aspect did you like the most?	Design	Functionality	Touchscreen
How difficult will this be to master?	Difficult	Easy	Don't know
How often would you tag content if it was more enjoyable?	Less often	Same Amount	More often

Chapter 6 includes a detailed view of the testing sequence and the use of the evaluation sheet along with the results and evaluation.

Chapter 6

6. Results and Evaluation

After finishing the high-fidelity prototypes and the testing scenarios, a small usability test was conducted. The following sections highlight the methodologies, observations, and results of the user testing.

6.1. Usability Testing

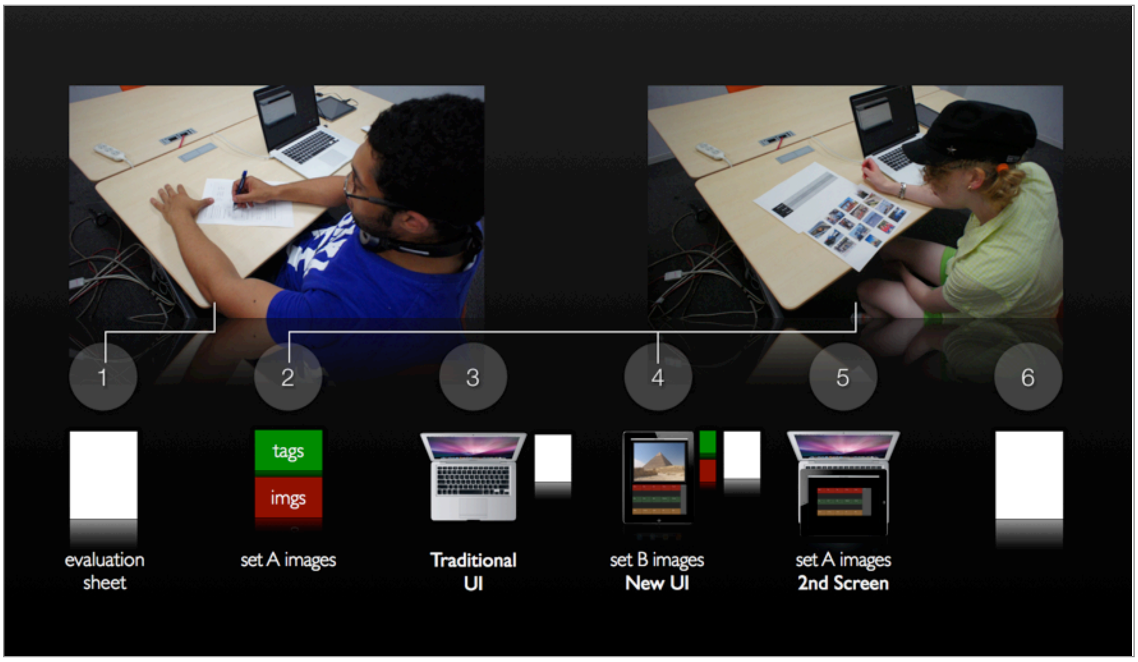


Figure 6.1. Testing and Evaluation sequence.

The test group comprised of 7 participants within the target demographics' age range. All subjects come from diverse backgrounds. Both male and female participants were part of the test. Location was set in dedicated rooms and the test spanned two days with four participants in the first day, and three on the next. Here is the sequence of the conducted tests that each participant went through:

4. Evaluation Sheet Distribution

   At the beginning, the test participants were given a paper with a set of multiple-choice questions. They rate how they feel today (Figure 6.1 step 1).

5. Tasks are Defined with First Task explanation

   This phase informs the participants about the three tasks they will be doing. The first is the traditional UI test. Users were given a sheet of paper with the first set of pictures (Set A). They were asked to get familiar with them and afterwards, the participants were given a sheet with the 15 keywords from Table 5.2 (Figure 6.1 step 2).

6. First Task: Traditional UI

   The users are shown the UI to get familiar with the environment. As soon as they are ready, the program timer starts (Figure 6.2). After the users conduct this task, they are given the evaluation sheet again and they rate how they feel.

Figure 6.2. Traditional UI (task 1).

7. Second Task: iPad UI

   Users were given another sheet of paper with the second set of pictures (Set B) and are shown the new user interface on the iPad afterwards (Figure 6.3). After they get familiar with the interface, the timer starts. The evaluation sheet is given again after they are done and they rate how they feel.

Figure 6.3. iPad UI (task 2).

8. Third Task: Second-Screen UI

   The same images from the first task are assigned (Set A) and the UI is a slight modification from the second task but employs the second-screen scenario (Figure 6.4). The timer starts when the users are ready. The evaluation sheet is given again after they are done and they rate how they feel.

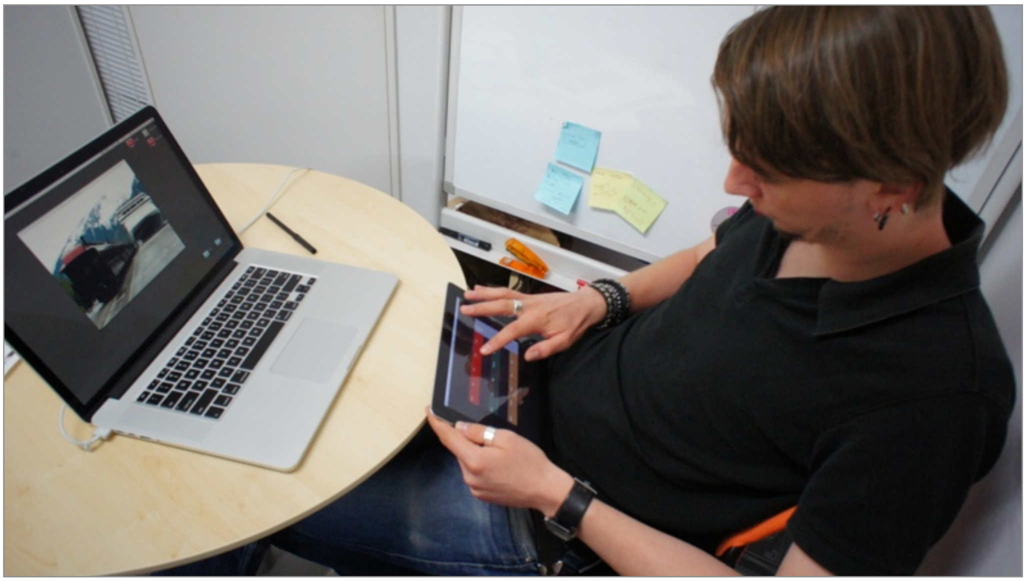


Figure 6.4. Second-Screen UI (task 3).

9. Informal Interview and Feedback

   In this final stage, the evaluation sheet is assessed and users could add their feedback. Additionally, the participant's time estimates were contrasted with the actual recorded time.

The purpose of showing the participants the images and keywords before the application is that it simulates the browse-through and tagging rules establishment steps that were discussed in Chapter 3.

6.1.1. Other Evaluation Methods

With the addition of the evaluation sheet, each of the users were observed during their sessions, and video and images were taken during their sessions. The focus of the observation is to see how the users interact with the prototypes and if there are other factors in play during the test.

6.1.2. Evaluation Sheet Results

The participants were asked a series of repeated questions after each task to gauge how difficult it was to tag content and how much did they enjoy the process as mentioned in the previous chapter. Figure 6.5 shows the tally of their answers. While the traditional UI was mostly easy for the participants *(2 Ok and 5 Easy responses)*, they enjoyed it the least *(2 Not Fun and 5 OK)* compared to the iPad *(3 Ok and 4 Fun)* and second-screen tasks *(2 Ok and 5 Fun)*. The second-screen system scored higher in both ease of use and enjoyment. The task time results show that the second-screen UI had an average time of 1:37 compared to the other methods. Still, the iPad prototype fared better time-wise than the traditional UI.

All the participants agreed that the interface would be easy to master with 4 liking the functionality of the application and its affordance to accept different taps and gestures, while 3 participants liked the touchscreen interface as a method for data selection.

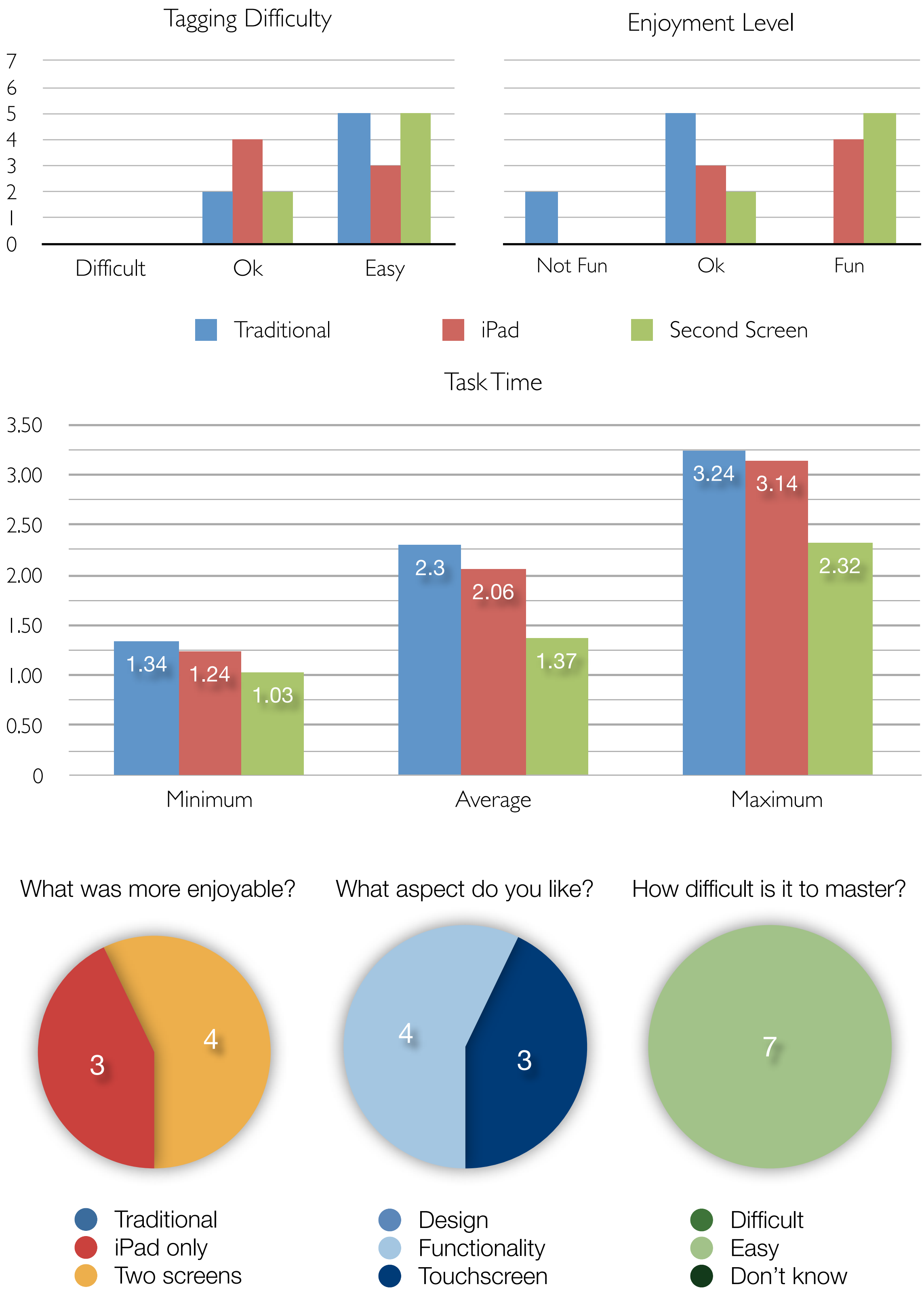


Figure 6.5. Summary of users' responses.

6.1.3. Primary Findings

Most of the participants had their own unique way when selecting the checkboxes. Some people “flicked”, others tapped, or swiped horizontally only. While others swiped vertically, or went in straight line patterns between the checkboxes. The interface allowed for these actions to be executed without imposing any limits to the user interaction. There were a few pauses at times as users were looking at the tags on the touchscreen, this happens more noticeably on the second-screen tests.

Additionally, there were some differences in terms of tagged items between the different users but it is mostly consistent within the participants themselves. Some users included more relevant tags than others. Since the first task and third one include the same images, consistency was expected from the participants. Yet, there was some small variance between the tags chosen at the first task and third one.

Despite having a shorter average time, the two-screen method was not preferred by almost half of the participants (*4 Two-Screens, 3 iPad only*). They cited how their eyes kept traveling back and forth between the image and finding the correct tags. Even though both the images and the interface were introduced before, they found this aspect to be a bit annoying. The others who preferred the two-screens liked the flexibility that it allows and how much real-estate it provides.

Other users commented that even though the design and application are fun, it might attributed to the fact that it is something new and that is why people are getting excited about it. Thus, a longer study experiment would likely show if the system is effective in increasing how often users tag their content.

6.1.4. Positive and Negative Affect

Looking at Figure 6.6, the positive affect is at its highest on the third task and the negative effect is at its lowest, which means the prototype increased the enjoyment level while decreasing the negative affect. The negative affect peaked after the first task and gradually decreased towards the end of the usability test. [1]

[1] Note: The first participant did not complete the positive and negative affect portion correctly. Thus, the results in Figure 6.6 only represent the six participants whom completed the evaluation sheet correctly.

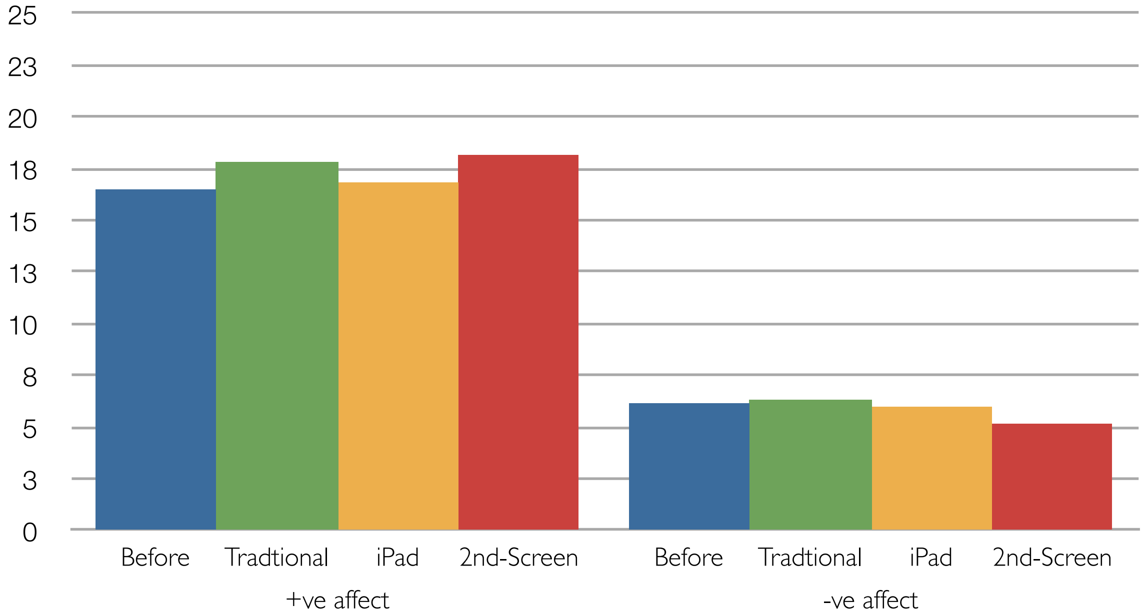


Figure 6.6. positive and negative affect results.

6.1.5. Secondary Findings

One of the observations noticed across the different participants is how their body posture and their handling of input devices change with the different interfaces. When using the traditional input via the mouse or trackpad, the participants had an almost similar posture. But with the iPad, the participants had more freedom and each had their own way of handling the devices (Figure 6.7). Also, one participant commented that the two screen interface would allow him to control the screen more comfortably where he could move around or sit on the couch while doing so.

Figure 6.7. Comparison between users' postures with tablets and PCs.

Another observation regarding task times is that the two female participants (Table 6.1) spent the most time checking the images and keywords before proceeding to the first task compared to the other male participants. Also, looking at the task times for each individual (Figure 6.8), their times tend to decrease as they progress through the tasks, with the lowest time in the second-screen task as mentioned before. The exception is User 7, who had a higher time in the second task. Looking at the footage again, User 7 tends to select the keywords on the iPad using taps and swipes efficiently but he tends to pause before clicking the Next button as he looks back at the image.

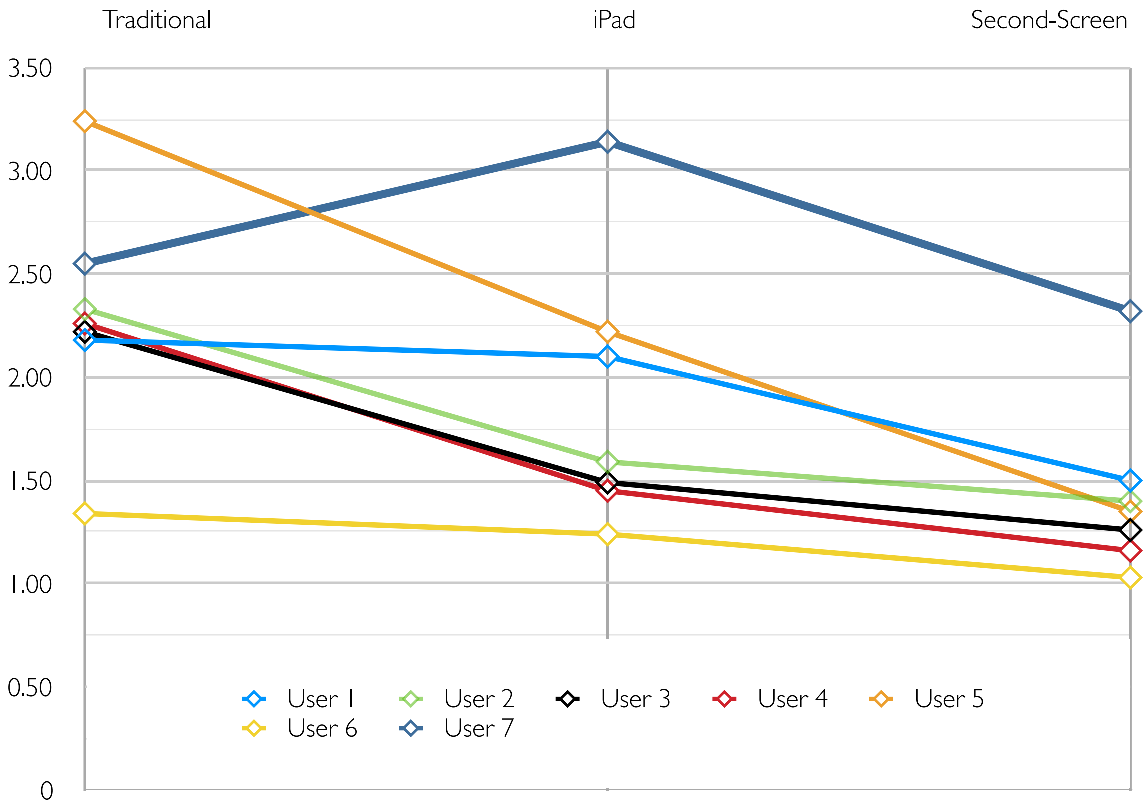


Figure 6.8. Task times for each individual user. [2]

With regards to perceived errors by the users, the majority of users did not have any corrections to the tags (changing their minds to select/ deselect or by accident). As for the others that had such errors, it averaged at about one select/ deselect in one session. Furthermore, these errors were not unique to any task and they occurred in all.

[2] Table data is available in Appendix e.

Table 6.1: Small bio and info about the participants.

participant 1	male
The user has a background in TV production both on-location shooting and studio broadcast experience. Additionally, his field of interest is in human factors in remote collaborative production	
Level of Video Production Experience	Mid to High
Hobbies, Interests, & User Comments	
Watching TV shows, sometimes in a multi-screen environment with multi-tasking at the same time. The user also commented on how he browses for his apps on his smartphone by using icons and location ques	

participant 2	male
A background in policy, the user is also interested in blending video technology and policy aspects to create better products for the users	
Level of Video Production Experience	Basic to Mid
Hobbies, Interests, & User Comments	
Playing in a band, socializing, and videogames mostly on smartphones and PC	

participant 3	male
The user's background is in computer science and AI. Currently, he is working in a technology lab dealing with robotics and telepresence	
Level of Video Production Experience	Basic to Mid
Hobbies, Interests, & User Comments	
Reading, and interest in video production techniques and issues.	

continued on next page...

participant 4	male
The participant worked on the “places + perspectives” documentary as a director and content producer. He worked with the PIX system and participated in the full production cycle of the documentary	
Level of Video Production Experience	Mid to Hi
Hobbies, Interests, & User Comments	
Karaoke, fashion and style, photography	

participant 5	female
Work experience in videogame development with a computer science background	
Level of Video Production Experience	Basic to Mid
Hobbies, Interests, & User Comments	
Traveling, playing videogames, attending live concerts	

participant 6	female
Art-related major and she has experience working as a flight attendant	
Level of Video Production Experience	Basic to Mid
Hobbies, Interests, & User Comments	
Socializing, smartphone use, playing games in different forms from smartphone to game consoles such as the Playstation 3. Also, involved in game commentary production with a channel on YouTube dedicated for these videos.	

participant 7	male
The user’s background is in computer science and computer technology	
Level of Video Production Experience	Basic to Mid
Hobbies, Interests, & User Comments	
Weightlifting, traveling, socializing. He commented on how he browses for his apps and does not memorize where his applications are	

6.2. Technical Evaluation

From a technical standpoint, the prototype system was stable throughout the different tasks. There were no major difficulties during the tests but here are some of the observations:

- Latency Issues

  XDisplay has two options for displaying content: one focuses on performance while the other on fidelity. The experiments were done on the fidelity setting since the keyword text was not legible when in performance mode. For the most part, latency was not high throughout the tests and did not hinder the test. However, as the XDisplay application ran for an extended period (over one hour and a half), there is a level of latency introduced. This was noticed during the end of the first and second testing sessions.

- Connection Issues

  One of the disadvantages of using XDisplay and Splashtop Streamer is the connection issues. There were some occasions where the iPad application would not connect to the Splashtop Streamer application. After browsing the troubleshooting pages of the Splashstop website, some of the solutions include reinstalling the iPad application or trying to update it. This added a level of inconvenience as it becomes difficult and frustrating to fix an element that is out of one's control.

- Processing Stability

  The programs performed very well, there were no crashes or errors for the most part. Many of the errors were in fact due to human errors, such as renaming the data-log for each user's session or making sure not to move the mouse when using the XDisplay application with the prototype in session.

The next chapter discusses some of the enhancements derived from the usability testing and future applications of the concept.

Chapter 7

7. Improvements & Scenarios

After analyzing the evaluation and results, there are a few improvements that could be implemented in the future. Additionally, this section discusses some of the environments the interface concept could be used in.

7.1. Design Improvements

One of the aspects that was not addressed in the user testing was the arrangement of the keywords. The three prototypes had the same order of keywords. After looking at the general user behavior during the usability testing, it became clear that there could be more enhancements that to make the experience even smoother. An example for improving keyword selection is through the addition of icons, in addition to the colors, for easier tag recognition.

7.1.1. A Better Experience: A More Visual One?

At the end of the evaluation test, some of the participants were asked about how they browse their app library on their mobile devices. Some answered that they memorize the application's location (Participant 1 and 2). While Participant 7 said that he relied more on recognition rather than memory. He would flip through his application library to find the desired application.

The home screen premise in a nutshell represents apps with text and icons on multiple pages. Users browse through these pages and select their desired app. In terms of ordering, the default for most devices would order them based on their installation date, the newest are at the end of the app list. However, users could reorder the apps and group them into folders if they would like to.

Building on the home screen model, the keyword selection prototype could be improved through the addition of categorization by the user. Moreover, the use of icons and colors along with text, would probably make keyword selection more easier especially in the second-screen scenario. Relying on text only has its disadvantages because many considerations need to be accounted for such as font type, size, uppercase or lowercase, clarity, position, etc. Also, sometimes reading text can be disruptive, which might cause unnecessary pauses [Johnson]. But having both an image and text would allow users to scan the page and identify the tags more easily, and users could confirm their choice further by reading the keyword tag (Figure 7.1).

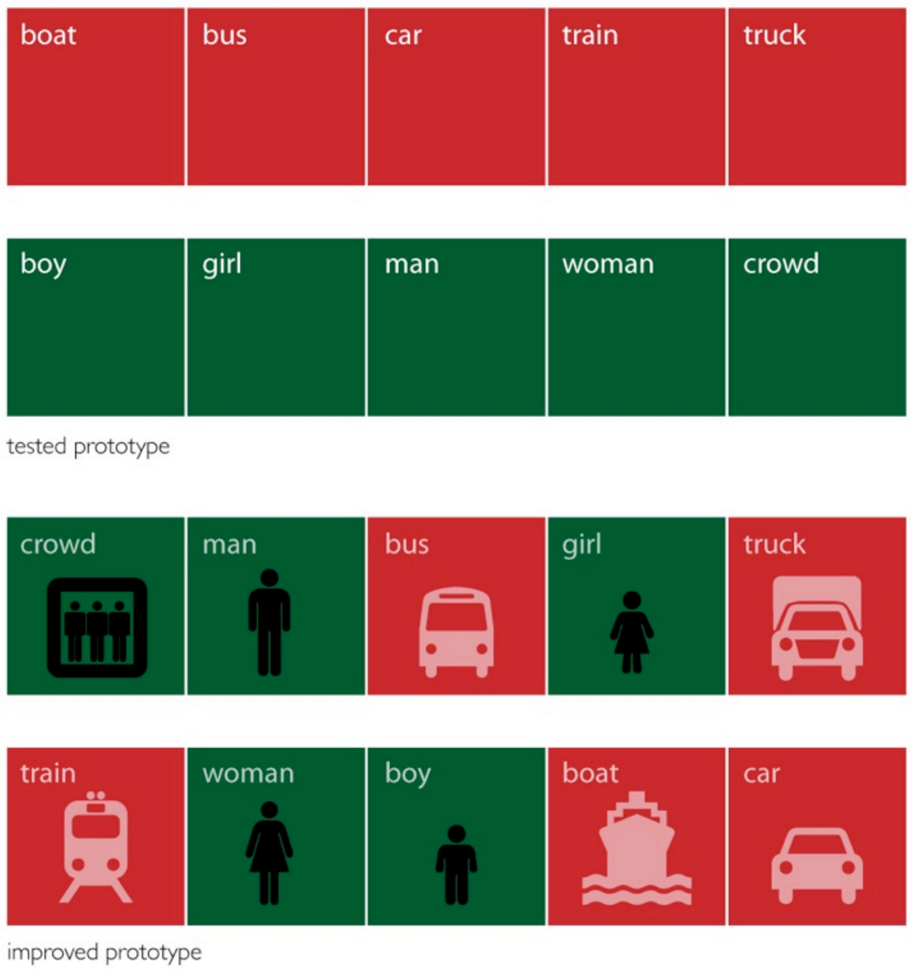


Figure 7.1. The tested prototype and proposed visual improvements.

7.1.2. Selection Methodologies

With the inclusion of the latest metadata tools into FCPX, the program introduced many advanced features to organize content. In addition to the backbone technology, the UI also features new enhancements. For example, users could apply multiple keywords at the same time using either keyboard shortcuts or typing them through the keyboard with auto-complete for each individual tag. Yet, the majority of these actions are manual, in fact, the user needs to have a “keyword strategy” or keyword collections to start assigning them to content [Hodgetts]. Based on that, the users could have their own collections within the touchscreen application organized based on their preferences. This allows the users to customize keyword input in a way that works for them and would be similar to keyboard shortcut assignment.

Another scenario of assigning keywords using the touch-based prototype could include main category selection that leads to more specialized categories. The concept could use a category system similar to the Dewey Decimal Classification or the ACM Computing Classification System. This might fit interview-based documentaries where there are general themes for the movie clip but the clip has more specific topics within.

One of the features that is becoming more prevalent in video editing applications and could be used with the iPad or second-screen prototypes is auto-analysis metadata. There are different approaches to metadata extraction with transcription and speech-analysis being a first choice for unscripted content [Hodgetts]. The technology is getting better but it is still far from perfect and manual data-entry is sometimes necessary. The touchscreen application could in theory work together with transcription and offer the users the most optimal keyword selection arrangement automatically.

7.2. Implementation Scenarios

Since the concept interface is flexible, it could be applied to different scenarios ranging from single user applications to multiple users.

7.2.1. Single User Experience and Social Tagging

The simplest form of implementation, integrating the interface with a DAM system. This would allow the users to utilize the best of the PC environment along with the touchscreen interface. The mouse still represents an accurate control tool, which is still heavily used in video editing. And in tandem, the touchscreen allows for more natural control movements.

Also, this concept could be used in TV second-screen applications to allow users to tag the content more easily. It could be expanded to allow social tagging where users would be able to build on the original keywords assigned by other users and in turn add their own keywords and details. Doing so will create a more social experience and could help in content discovery.

7.2.2. New Applications and Remote Collaboration Software

With the ongoing research in remote collaboration at KMD, a recent experiment was done in mid-2013 using the Open Source software, Novacut[1] (Figure 7.2). The software provides interesting opportunities in remote collaboration and it also showcases a new way of video editing thanks to its unique UI. Novacut is gathering the attention of amateur filmmakers, especially those who shoot with DSLR cameras.

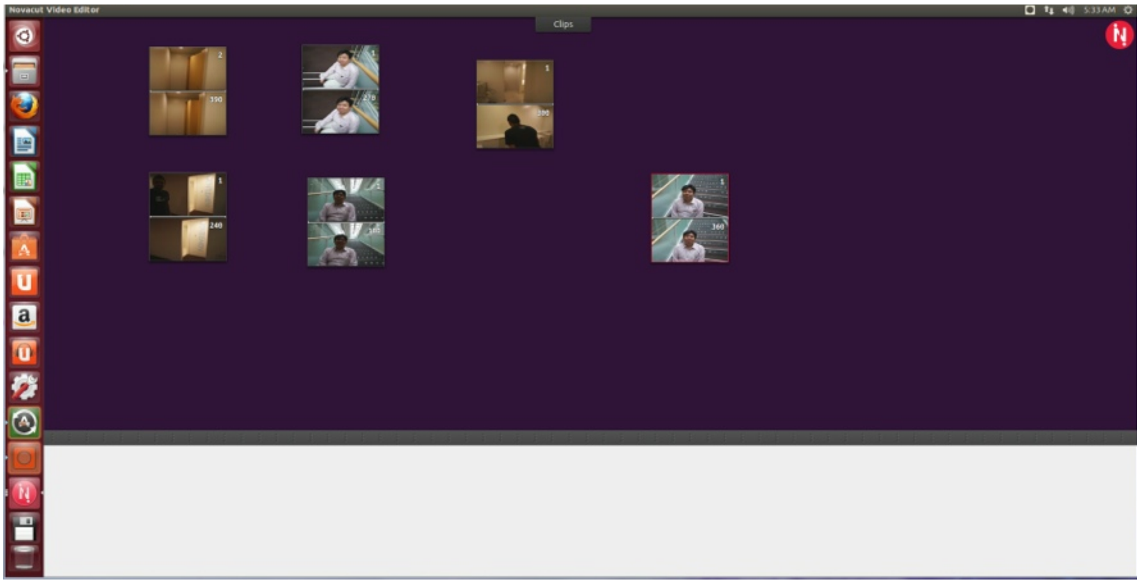


Figure 7.2. The UI of Novacut is a departure from standard video editors.

In addition to video editing, the software includes a data management software called dMedia, which offers basic digital asset management. Due to the open nature of the Novacut environment, the touchscreen concept could be implemented into this system with a few modifications. This would allow to test the interface in a one-to-one remote situation where users could discuss, annotate, and tag content together.

7.2.3. Large-Screen Applications and SAGE

Another implementation scenario is integrating the concept within a large touchscreen environment. For this scenario, the keyword tagging could be done on the large screen

[1] http://www.novacut.com

and/ or on a smartphone and tablet (Figure 7.3). Multiple users could collaborate at the same time to add keywords and tags to the content. For example, some users could handle the video while others the audio portion of a clip with annotations and keyword assignment.

Figure 7.3. Integrating touchscreen with DAM into the SAGE environment.

Chapter 8

8. Conclusion

Touchscreen applications are challenging our perception of computer human interaction. Both the form-factor and interaction methods are quite different from the stationary workstations and even the portable laptops. Moreover, this form-factor highlights the importance of the visual aspects in touchscreen UI design.

The prototype development methodologies and the usability test discussed in this paper produced two different scenarios, one through single-touchscreen keyword selection and the other through a "second-screen" interface. Both scenarios have their advantages as perceived by different users. Based on the results and evaluation, it would be recommended to implement both of these methods and allow the users to choose based on their preferences. Even though performance can be optimal in the two screen scenario, it was not unanimously accepted by everybody.

The implementation and freedom of use of these concepts were the result of the research methodologies that focused on the people. Through observation and interviews, the design requirements were defined at first. With the paper and soft-prototypes, different participants offered their own input about how they would interact with the touchscreen prototype. Thus, the "accommodating" aspect of the interface was realized in the RITE phase of testing. With each participant providing their input and method(s) of interaction, they allowed for better understanding of the user requirements. Additionally, the test scenarios were defined based on the feedback received in the RITE phase. The system was designed to fit those scenarios accordingly.

There are some shortcomings to look into, especially in the two-screen interface. And such scenario shows some of the issues that could be faced in a multi-user environment with one shared display. A real production test would allow to test the system fully in an uncontrolled environment under pressure.

Finally, even though the inspiration and motivation behind the prototypes was driven by video production, these concepts could be used in other areas. Form selection or system settings could improve on this concept to allow the users to select or change their settings more easily. It also could be used outside the target demographic to accommodate older people. Just as long the backend of the system is designed with flexibility towards users and focus on the tasks, there would be good opportunities for this interface concept to improve the overall experience.

8.1. Future Works

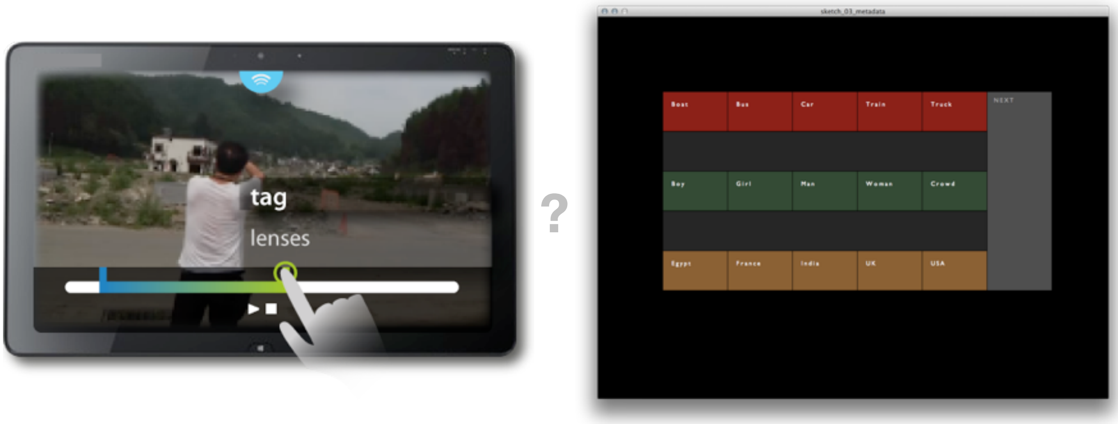


Figure 8.1. How to integrate the concept with video applications.

The improvements discussed in the previous chapter need to be fully tested to proceed to the next step: implementing this concept to video tagging. This test scenario will present a few challenges when considering the timeline element and range selection methodology (Figure 8.1). Moreover, displaying content is just as important as inputting and selection content. This will also present a new challenge.

A navigation prototype (Figure 8.2) is still in the RITE phase at the time of writing (two more users are needed to validate it and move it to the prototype development stage). This navigation concept was inspired by the interview mentioned in Chapter 3 and relies on visual cues more than text. So far, the three people that piloted the concept thought that it would suit them. They liked how the interface is event and date-driven with an emphasis on visual clues.

One such clue is the inclusion of the movie duration as a vertical bar instead of being represented by text only. For example, the blue bar in Figure 8.2 represents a percentage from a reference duration. If the reference is 15 minutes long (similar to the average HDSLR movie lengths), and the video duration is 7.5 minutes, the blue bar takes 50% of the vertical bar. Based on that, a quick test was assigned to users to see how fast they could detect the longest and shortest videos. They were able to do so quite fast without relying heavily on reading the text. This concept could be integrated in the future with the current prototypes to offer a new integrated solution for viewing and organizing content more effectively.

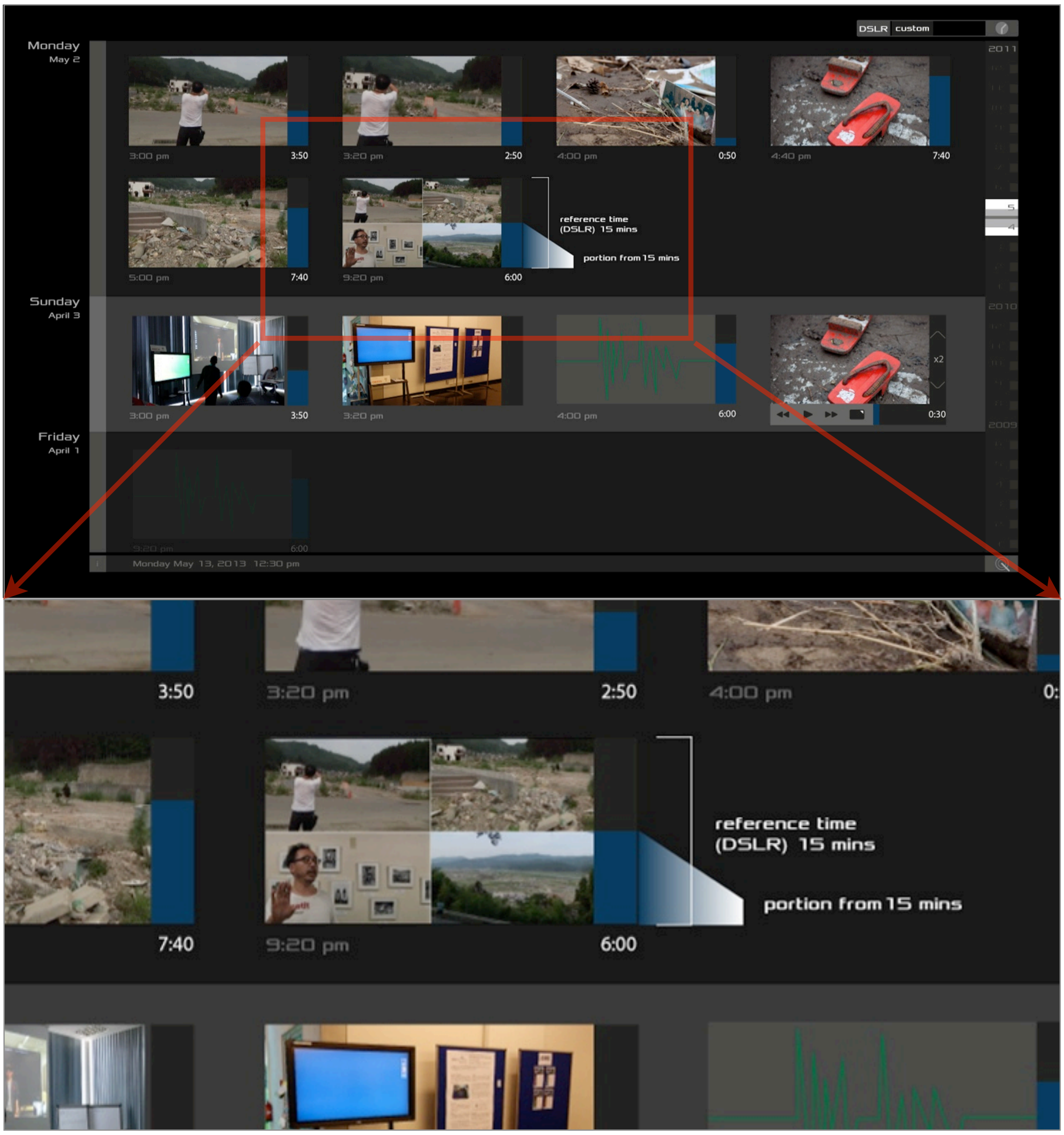


Figure 8.2. Concept navigation UI under development.

Acknowledgements

I would like to thank Professor Naohisa Ohta for his patience, support, and guidance with my research. I would also like to thank Associate Professor Kazunori Sugiura and Professor Adrian Cheok for their help and assistance during my tenure at KMD.

Finally I would like to thank all of the Power of Motion Pictures members for their support and feedback.

References

[1] Landau, M. 1984. Human Evolution As Narrative. American Scientist, 72 (May-June, 1984), 262-268.

[2] Pimenta, S. And Poovaiah, R. 2010. On Defining Visual Narratives. Design Thoughts (August 2010), 25-46.

[3] Sørensen, K., Bahnsen, M., Holch, H., Hvid, G. And Otte, L. 2002. Guidelines For Producing A Short Documentary, P.o.v.: A Danish Journal Of Film Studies, No. 13 (March 2002),126-127.

[4] Hodgetts, P. 2011. Conquering The Metadata Foundations Of Final Cut Pro X. (June 21, 2011). CreateSpace Independent Publishing Platform.

[5] Melenhorst, M., Grootveld, M. And Veenstra, M. 2008. Tag-based information retrieval for educational videos. (Q2, 2008). EBU Technical Review.

[6] Schloss, D. 6 Easy Steps To Optimizing YouTube Videos For Search Rankings. 2011. SITEFling. http://sitefling.com/blog/6-easy-steps-to-optimizing-youtube-videos-for-search-rankings/

[7] Ames, M. and Naaman, M. 2007. Why we tag: motivations for annotation in mobile and online media. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI 2007), (San Jose, USA, April 28 - May 3, 2007), pp. 971-980.

[8] McCune, Z. 2011. Consumer Production in Social Media Networks: A Case Study of the "Instagram" iPhone App. (24 May, 2011). http://thames2thayer.com/blog/wp-content/uploads/2011/06/McCune_Instagram_Dissertation_Draft.pdf

[9] Carlson, J. 2012. Where did all that disk space go? DaisyDisk helps you tidy up. The Seattle Times. (January 6, 2012). http://seattletimes.com/html/businesstechnology/2017162877_ptmacc07.html

[10] Shedroff, N., & Noessel, C. 2012. MAKE IT SO: Interaction Design Lessons From Science Fiction. Rosenfeld Media.

[11] Lidwell, W., Holden, K., And Butler, J. 2003. Universal Principles Of Design. Rockport Publishers.

[12] Wigdor, D. And Wixon, D. 2011. Brave NUI World: Designing Natural User Interfaces For Touch And Gesture. Morgan Kaufmann.

[13] Cioccaa, G., Olivob, P., And Schettinia, R. 2012. Browsing Museum Image Collections On A Multi-touch Table. Information Systems, Volume 37, Issue 2, (April 2012), 169–182.

[14] Renambot, L. Et Al. 2004. SAGE: The Scalable Adaptive Graphics Environment. In Proceedings Of WACE 2004, (4th Workshop On Advanced Collaborative Environments)

[15] Bhimani, J., Nakakura, T., Almahr, A., Sato, M., Sugiura, K. Ohta, N. 2013. Vox Populi: Enabling Community-based Narratives Through Collaboration And Content Creation. EuroITV '13 Proceedings Of The 11th European Conference On Interactive TV And Video, (June 24 - 26, 2013), 31-40.

[16] Parhi, P., Karlson A., And Bederson B. 2006. Target Size Study For One-Handed Thumb Use On Small Touchscreen Devices. In Proceedings Of MobileHCI'06.

[17] Lazaris, L. 2010. The Case Against Vertical Navigation. Smashing Magazine. (January 11th, 2010) . Http://www.smashingmagazine.com/2010/01/11/the-case-against-vertical-navigation/

[18] Hanington, B., And Martin, B. 2012. Universal Methods Of Design: 100 Ways To Research Complex Problems, Develop Innovative Ideas, And Design Effective Solutions. Rockport Publishers.

[19] Thompson, E. 2007. Development and validation of an internationally reliable short-form of the positive and negative affect schedule (panas). Journal of Cross-Cultural Psychology, 38(2):227–242,.

[20] Johnson, J. 2010. Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules. (May 20, 2010), Morgan Kaufmann.

Appendix

a. Interview Questions

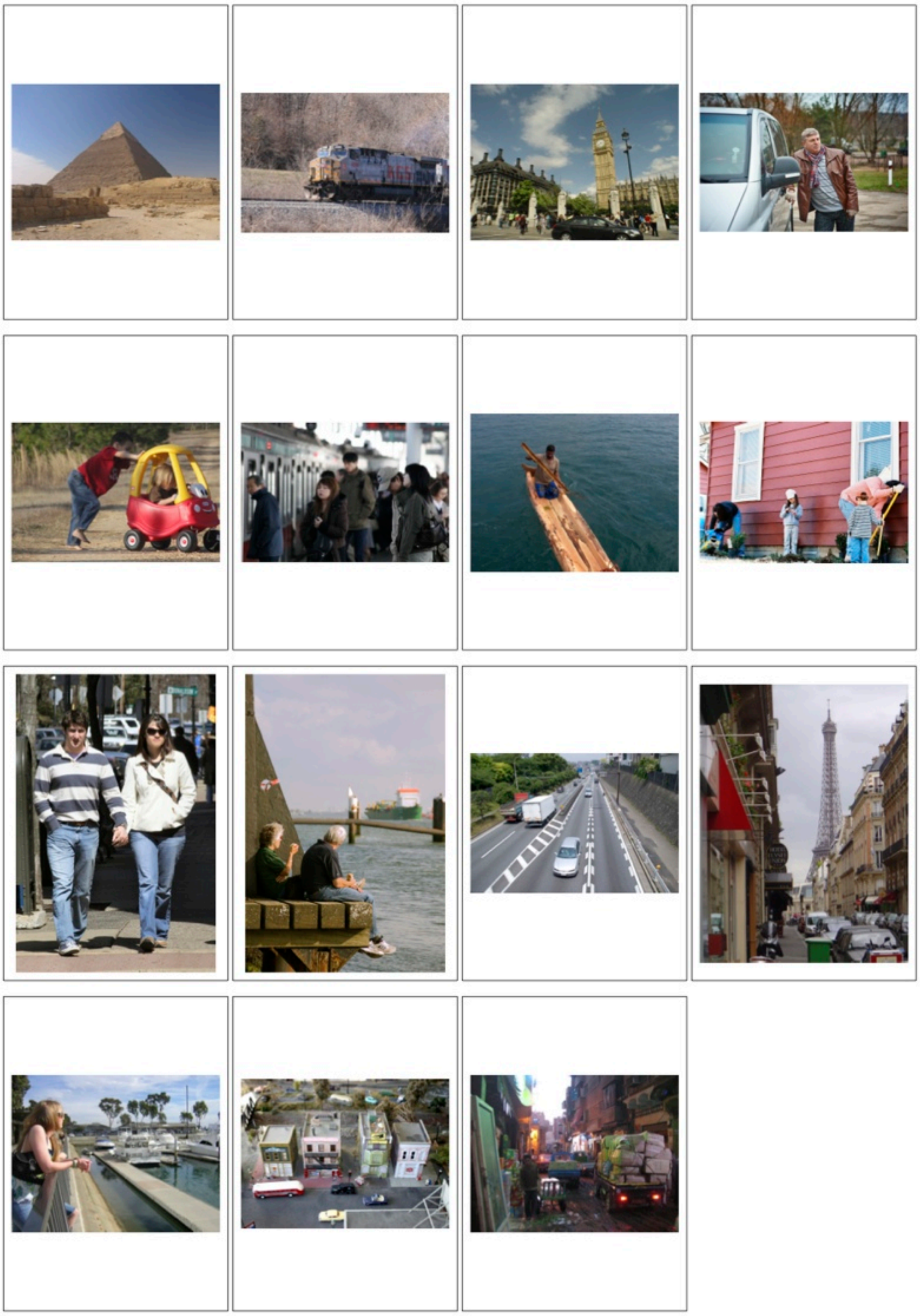
Background Information

- Previous Major/ Work Experience / School / Hobbies / Production(s)
- Operating System / Software / Equipment

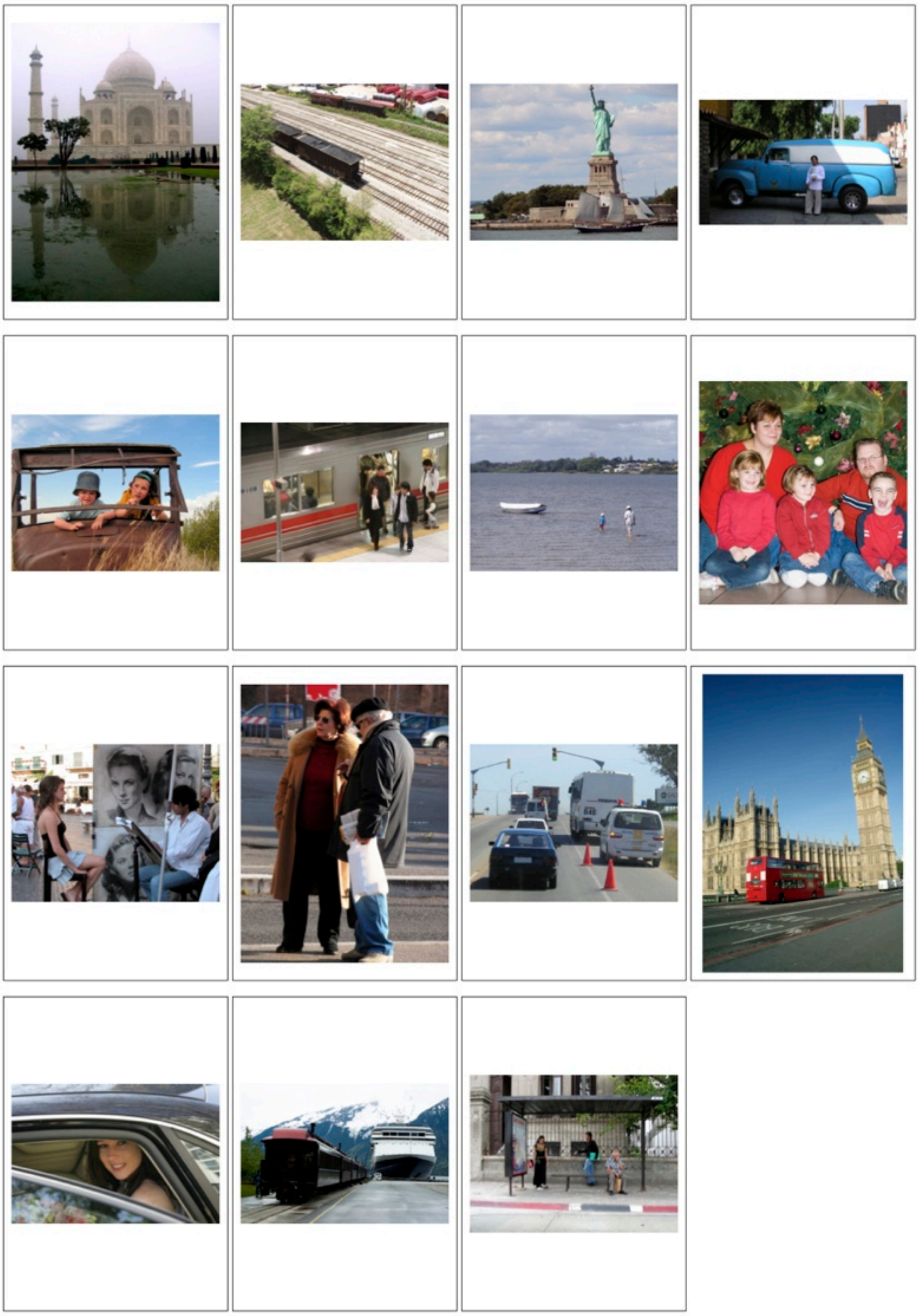
Workflow Questions (How they work/ Do their tasks)

- What types of video production? How is it shot? How many takes?
- Do you film + edit or do either one exclusively?
- Is editing done in the same day or afterwards?
- How do you browse files and pull them off the camera?
- Would you say you remember video shoots as events? True or False?
- Do you have an idea of the footage you want to use before browsing or afterwards?
- What do you look for in a video you are searching for?
- Do you rely on text or visual cues to browse and check files?
- What parameters of the file are important (e.g. file size, length)?
- What are some of the difficulties of the process? UI? Workflow? Data amount?
- Do you look at each file individually or all of them via thumbnails then individually?
- Do you use metadata tags? Why or Why not?

b. First Image Set (Set A)



Second Image Set (Set B)



c. Keyword Tagging Evaluation Sheet

Keyword Tagging Prototype

Please circle one of the following:

How often do you create videos?	○ Never	○ Sometimes	○ Often
How often do you tag your videos?	○ Never	○ Sometimes	○ Often
Do you enjoy tagging content?	○ No	○ Moderately	○ Yes

How do you feel today?

1	2	3	4	5
Very slightly or Not at all	A little	Moderately	Quite a lot	Extremely

Active	______	Nervous	______	Upset	______	Determined	______
Hostile	______	Inspired	______	Attentive	______	Ashamed	______
		Alert	______	Afraid	______		

Thank you, this is the end of the first portion of this test. Please await further instructions.

How do you feel now?

1	2	3	4	5
Very slightly or Not at all	A little	Moderately	Quite a lot	Extremely

Attentive	______	Afraid	______	Hostile	______	Inspired	______
Ashamed	______	Upset	______	Determined	______	Active	______
		Nervous	______	Alert	______		

How easy was it to tag the images?	○ Difficult	○ Ok	○ Easy
How much fun was the process?	○ Not fun	○ Ok	○ Fun
How long do you think it took you?	______ Minutes	______ Seconds	

Thank you, this is the end of the second portion of this test. Please await further instructions.

Keyword Tagging Evaluation Sheet 2

How do you feel now?

1	2	3	4	5
Very slightly or Not at all	A little	Moderately	Quite a lot	Extremely

Inspired	______	Alert	______	Ashamed	______	Upset	______
Determined	______	Hostile	______	Afraid	______	Attentive	______
		Active	______	Nervous	______		

How easy was it to tag the images?	O Difficult	O Ok	O Easy
How much fun was the process?	O Not fun	O Ok	O Fun
How long do you think it took you?	______ Minutes	______ Seconds	

Thank you, this is the end of the third portion of this test. Final test coming.

How do you feel now?

1	2	3	4	5
Very slightly or Not at all	A little	Moderately	Quite a lot	Extremely

Attentive	______	Afraid	______	Hostile	______	Ashamed	______
Inspired	______	Upset	______	Determined	______	Active	______
		Nervous	______	Alert	______		

How easy was it to tag the images?	O Difficult	O Ok	O Easy
How much fun was the process?	O Not fun	O Ok	O Fun
How long do you think it took you?	______ Minutes	______ Seconds	
What was more enjoyable?	O Traditional	O iPad only	O Two screens
What aspect did you like the most?	O Design	O Functionality	O Touchscreen
How difficult will this be to master?	O Difficult	O Easy	O Don't know
How often would you tag content if it was more enjoyable?	O Less often	O Same Amount	O More often

Thank you very much for your time.
For any comments, please use the space below or relay it to the test observer.

d. Sample Data Output

Image ID	Time	Elapsed	Ms	Box status	Box No.	Tag Name
1	7:59 PM	0:00:02	2365	1.0	11	Egypt
1	7:59 PM	0:00:06	6732	1.0	99	NEXT
2	7:59 PM	0:00:09	10004	1.0	4	Train
2	7:59 PM	0:00:12	13022	1.0	99	NEXT
3	7:59 PM	0:00:16	16468	1.0	10	Crowd
3	7:59 PM	0:00:19	19768	1.0	3	Car
3	7:59 PM	0:00:21	22235	1.0	14	UK
3	7:59 PM	0:00:22	23002	1.0	99	NEXT
4	7:59 PM	0:00:25	25699	1.0	3	Car
4	7:59 PM	0:00:25	25832	1.0	8	Man
4	7:59 PM	0:00:27	27399	1.0	99	NEXT
5	7:59 PM	0:00:31	31917	1.0	6	Boy
5	7:59 PM	0:00:31	31983	1.0	7	Girl
5	7:59 PM	0:00:32	32800	1.0	99	NEXT
6	7:59 PM	0:00:35	36050	1.0	4	Train
6	7:59 PM	0:00:37	37368	1.0	10	Crowd
6	7:59 PM	0:00:39	39652	1.0	99	NEXT
7	7:59 PM	0:00:43	44094	1.0	1	Boat
7	7:59 PM	0:00:44	44545	1.0	8	Man
7	7:59 PM	0:00:45	46044	1.0	99	NEXT
8	7:59 PM	0:00:49	49910	1.0	6	Boy
8	7:59 PM	0:00:49	49977	1.0	7	Girl
8	7:59 PM	0:00:49	50044	1.0	8	Man
8	7:59 PM	0:00:49	50110	1.0	9	Woman
8	7:59 PM	0:00:54	54411	1.0	99	NEXT
9	7:59 PM	0:00:56	57156	1.0	8	Man
9	7:59 PM	0:00:56	57207	1.0	9	Woman
9	8:00 PM	0:00:59	60142	1.0	99	NEXT
10	8:00 PM	0:01:02	62363	1.0	8	Man
10	8:00 PM	0:01:02	62414	1.0	9	Woman
10	8:00 PM	0:01:05	66031	1.0	1	Boat
10	8:00 PM	0:01:07	67799	1.0	99	NEXT
11	8:00 PM	0:01:12	72852	1.0	3	Car
11	8:00 PM	0:01:13	73319	1.0	5	Truck
11	8:00 PM	0:01:15	75607	1.0	99	NEXT
12	8:00 PM	0:01:21	81427	1.0	12	France
12	8:00 PM	0:01:21	82194	1.0	99	NEXT
13	8:00 PM	0:01:25	86001	1.0	1	Boat
13	8:00 PM	0:01:26	86384	1.0	9	Woman
13	8:00 PM	0:01:28	89219	1.0	99	NEXT
14	8:00 PM	0:01:34	95000	1.0	2	Bus
14	8:00 PM	0:01:34	95051	1.0	3	Car
14	8:00 PM	0:01:37	98185	1.0	5	Truck
14	8:00 PM	0:01:38	98734	1.0	99	NEXT
15	8:00 PM	0:01:41	102079	1.0	5	Truck
15	8:00 PM	0:01:42	102564	1.0	8	Man
15	8:00 PM	0:01:44	104730	1.0	13	India
15	8:00 PM	0:01:45	105747	1.0	99	NEXT

e. Task Time Data

	Traditional	iPad	Second-Screen
User 1	**0:02:18**	**0:02:10**	**0:01:50**
User 2	**0:02:33**	**0:01:59**	**0:01:40**
User 3	**0:02:22**	**0:01:49**	**0:01:26**
User 4	**0:02:26**	**0:01:45**	**0:01:16**
User 5	**0:03:24**	**0:02:22**	**0:01:35**
User 6	**0:01:34**	**0:01:24**	**0:01:03**
User 7	**0:02:55**	**0:03:14**	**0:02:32**

f. Traditional UI Program Code

```
import controlP5.*;

//Setup the controlp5 class
ControlP5 cp5;
ControlTimer c;
Textlabel t;
CheckBox cbox1, cbox2, cbox3;

PrintWriter output;

//Setup image, event trigger, counter, seconds, minutes, hours,
PImage bigImage;
String URL = "imageA1.jpg";
boolean eventTrig = false;
int counter = 2, s, m, h;
int timertime = 0;
float[] comparison = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

int bwidth = 140, bheight = 120;
int bposX = 1174,  bposY = 250;
int testing = -1;

void setup() {
  //Setting up screen size, controller, and mouse settings
  size(1324, 768);
  cp5 = new ControlP5(this);
  frameRate(60);  smooth();

  // Create a new file in the sketch directory
  output = createWriter("userdata7.txt");

  //Timer
  c = new ControlTimer();
  t = new Textlabel(cp5,"--",100,100);
  c.setSpeedOfTime(1);

  //Image
  bigImage = loadImage(URL, "jpg");
  int x = bigImage.width;

  //Changing the Font
  PFont p = createFont("GillSans",20,true);
  cp5.setControlFont(p,11);

  //Create the checkboxes
  cbox1 = cp5.addCheckBox("VEHICLES")
              .setPosition(bposX, bposY)
              .setColorForeground(color(120))
              .setColorActive(color(255))
              .setColorLabel(color(255))
              .setSize(20, 20)
```

```
                .setItemsPerRow(1)
                .setSpacingColumn(30)
                .setSpacingRow(5)
                .addItem("BOAT", 1)
                .addItem("BUS", 2)
                .addItem("CAR", 3)
                .addItem("TRAIN", 4)
                .addItem("TRUCK", 5)
                ;
  cbox2 = cp5.addCheckBox("PEOPLE")
                .setPosition(bposX, bposY+ 140)
                .setColorForeground(color(120))
                .setColorActive(color(255)).setColorLabel(color(255))
                .setSize(20, 20).setItemsPerRow(1)
                .setSpacingColumn(30).setSpacingRow(5)
                .addItem("BOY", 6)
                .addItem("GIRL", 7)
                .addItem("MAN", 8)
                .addItem("WOMAN", 9)
                .addItem("CROWD", 10)
                ;
  cbox3 = cp5.addCheckBox("PLACES")
                .setPosition(bposX, bposY+ 280)
                .setColorForeground(color(120))
                .setColorActive(color(255)).setColorLabel(color(255))
                .setSize(20, 20).setItemsPerRow(1)
                .setSpacingColumn(30).setSpacingRow(5)
                .addItem("EGYPT", 11)
                .addItem("FRANCE", 12)
                .addItem("INDIA", 13)
                .addItem("UK", 14)
                .addItem("USA", 15)
                ;

  //Create the next button
  cp5.addButton("NEXT")
     .setId(99)
     .setPosition(bposX, bposY+ 420)
     .setSize(80, 30)
     ;
}

void draw() {
  background(0);

  //Time
  s = second();
  m = minute();
  h = hour();
  t.setValue(c.toString());
  timertime = millis();

  //Draw the image
```

```
  imageMode(CENTER);
    if(eventTrig == true){
      bigImage = loadImage(URL, "jpg");
      eventTrig = false;
    }
    image(bigImage, 1024/2,  bigImage.height/2);

  fill(0);
  rect(1024, 0, 300, 768);
}

public void controlEvent(ControlEvent theEvent) {
  if(theEvent.getId()== 99){
    //This is the next button
    eventTrig = true;
    if(counter<16 && eventTrig == true){
      URL = "imageA"+counter+".jpg";
      counter +=1;
      resetButton();
      for(int i=0; i<15; i++){
        comparison[i]=0;
      }
    } else if(counter == 16){
        output.flush(); // Write the remaining data
        output.close();
      }
      println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c + ", "+
timertime + ", " + theEvent.getValue() + ", "+
theEvent.getController().getId() + ", "+
theEvent.getController().getName());
      output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c +
", "+ timertime + ", " + theEvent.getValue() + ", "+
theEvent.getController().getId() + ", "+
theEvent.getController().getName());
   }
  //println("controlEvent : "+theEvent.getName());
  if((theEvent.getName()=="VEHICLES" || theEvent.getName()=="PEOPLE"
|| theEvent.getName()=="PLACES") && eventTrig == false && counter<16){
  for(int i=0; i<5; i++){
  if (cbox1.getArrayValue()[i]!= comparison[i]){
     println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c + ", " +
timertime + ", " + cbox1.getItem(i).value() +", "+ (i+1) + ", " +
cbox1.getItem(i).name() );
     output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c +
", " + timertime + ", " + cbox1.getItem(i).value() +", "+ (i+1) + ", "
+ cbox1.getItem(i).name() );
     comparison[i] = cbox1.getArrayValue()[i];
//     output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+tim
ertime + ", "+theEvent.getController().getId());
      }
    if (cbox2.getArrayValue()[i]!= comparison[(i+5)]){
```

```
     println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c + ", " +
timertime + ", " + cbox2.getItem(i).value() +", "+ (i+1) + ", " +
cbox2.getItem(i).name() );
     output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c +
", " + timertime + ", " + cbox2.getItem(i).value() +", "+ (i+1) + ", "
+ cbox2.getItem(i).name() );
     comparison[i+5] = cbox2.getArrayValue()[i];
      }
    if (cbox3.getArrayValue()[i]!= comparison[(i+10)]){
     println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c + ", " +
timertime + ", " + cbox3.getItem(i).value() +", "+ (i+1) + ", " +
cbox3.getItem(i).name() );
     output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c +
", " + timertime + ", " + cbox3.getItem(i).value() +", "+ (i+1) + ", "
+ cbox3.getItem(i).name() );
     comparison[i+10] = cbox3.getArrayValue()[i];
      }
    }
  }
}

void keyPressed() {
    if (key == 'b' || key == 'B')
    {
      if (counter>2 && counter<=16)
      {
        //Use this to get back to the previous image
        counter -=2;
        URL = "imageB"+counter+".jpg";
        //println(counter);
        counter +=1;
        resetButton();
      }
     eventTrig = true;
    }
    if (key == 'r' || key == 'R')
    {
      //Reset the timer
      c.reset();
    }

}

void resetButton(){
    cbox1.deactivateAll();
    cbox2.deactivateAll();
    cbox3.deactivateAll();
    }

void printApp(){

}
```

g. iPad UI Program Code

```
import controlP5.*;
import com.shigeodayo.pframe.*;

//Setup the controlp5 class
ControlP5 cp5;
ControlTimer c;
Textlabel t;

PrintWriter output;

//Setup image, event trigger, counter, seconds, minutes, hours,
PImage bigImage;
String URL = "imageB1.jpg";
boolean eventTrig = false;
int counter = 2, s, m, h;
int timertime = 0;

//Button labels values & color & size
nuCheckBox[] nuCB, nuCB2;
nuCheckBox2 nuCB3;
String[] data= {"B o a t", "B u s", "C a r", "T r a i n", "T r u c k",
             "B o y", "G i r l" , "M a n" , "W o m a n" ,  "C r o w d",
              "E g y p t", "F r a n c e", "I n d i a", "U K", "U S A"};
String[] dataCon= {"V E H I C L E S", "P E O P L E", "P L A C E S"};
int[] buttonCon = {0xFF660000, 0xFF003300, 0xFF663300};
int buttonCon2 = buttonCon[0] ;
int bwidth = 130, bheight = 80;
int bposX = 20,  bposY = 600;

void setup() {
  //Setting up screen size, controller, and mouse settings
  size(768, 1024);
  cp5 = new ControlP5(this);
  frameRate(60);  smooth(); noCursor(); //noStroke();
  // Create a new file in the sketch directory
  output = createWriter("userdata7.txt");

  //Timer
  c = new ControlTimer();
  t = new Textlabel(cp5,"--",100,100);
  c.setSpeedOfTime(1);

  //Setting up the second window
  bigImage = loadImage(URL, "jpg");

  //Pointer replacing the mouse cursor used for touchscreen testing
  cp5.setAutoDraw(false);
  //cp5.getPointer().enable();
  //cp5.getPointer().set(width/2, height/2, true);

  //Changing the Font
```

```
  PFont p = createFont("GillSans",20,true);
  cp5.setControlFont(p,12);

  //Button position, size, colors
  int[] buttonCol = {0xffff0000, 0xff336633, 0xffFF9933, 0xff0080FF,
0xff660033};
  //int buttonRed = 0xffff0000, buttonGre = 0xff336633, buttonYel =
0xffFF9933, buttonMag = 0xff660033;
   nuCB = new nuCheckBox[15];
   nuCB2= new nuCheckBox[3];

  //Second group of selection
  Group control = cp5.addGroup("control")
                .setPosition(bposX,bposY+bheight*2)
                //.setPosition(0,0)
                //.setBackgroundHeight(bheight*1.5)
                .setWidth(bwidth*5)
                .hideBar()
                .close()
                .hide();

                for(int i = 0; i<3; i++)
                {
                    nuCB2[i] = new nuCheckBox(cp5, dataCon[i],
buttonCon[i]).setPosition(i*bwidth, 0).setSize(bwidth, bheight/
2).setId(i+100).setGroup(control);
                }

                nuCB2[0].setActive();

  //The previous and next buttons
  nuCB3= new nuCheckBox2(cp5, "N E X T"   ).setPosition(bposX
+bwidth*5, bposY).setSize(bwidth-50,
int(bheight*5)).setId(99); //.setGroup(control);

  //First group of selection
  Group g1 = cp5.addGroup("g1")
                .setPosition(bposX,bposY)
                .setBackgroundHeight(bheight)
                .setWidth(bwidth*5)
                .hideBar()
                .setId(40)
                .setBackgroundColor(color(50,50));

                for(int i = 0; i<5; i++)
                {
                    nuCB[i] = new nuCheckBox(cp5, data[i],
buttonCol[0]).setPosition(i*bwidth, 0).setSize(bwidth,
bheight).setId(i+1).setGroup(g1);
                }

  //Second group of selection
  Group g2 = cp5.addGroup("g2")
```

```
                .setPosition(bposX,bposY+bheight*2)
                .setBackgroundHeight(bheight)
                .setWidth(bwidth*5)
                .hideBar()
                //.close()
                //.hide()
                .setBackgroundColor(color(50,50));

                for(int i = 5; i<10; i++)
                {
                    nuCB[i] = new nuCheckBox(cp5, data[i],
buttonCol[1]).setPosition((i-5)*bwidth, 0).setSize(bwidth,
bheight).setId(i+1).setGroup(g2);
                }


  //Second group of selection
  Group g3 = cp5.addGroup("g3")
                .setPosition(bposX,bposY+bheight*4)
                .setBackgroundHeight(bheight)
                .setWidth(bwidth*5)
                .hideBar()
                //.close()
                //.hide()
                .setBackgroundColor(color(50,50));

                for(int i = 10; i<15; i++)
                {
                    nuCB[i] = new nuCheckBox(cp5, data[i],
buttonCol[2]).setPosition((i-10)*bwidth, 0).setSize(bwidth,
bheight).setId(i+1).setGroup(g3);
                }
}

void draw() {
  background(0);
  fill(20);
  rect(bposX, 20, bwidth*6-50, 560);
  stroke(3);
   imageMode(CENTER);
    if(eventTrig == true){
      bigImage = loadImage(URL, "jpg");
      eventTrig = false;
    }
    image(bigImage, 768/2, bigImage.height/2.6, bigImage.width/1.5,
bigImage.height/1.5);

  fill(50, 150);
  rect(bposX, bposY, bwidth*5, bheight*5);

  //Time
  s = second();
  m = minute();
```

```
  h = hour();

  //Drawing the CP5 controls and custom pointer
  cp5.draw();

  t.setValue(c.toString());
  timertime = millis();
  //t.draw();

  cp5.getPointer().set(mouseX, mouseY);

}


public void controlEvent(ControlEvent theEvent) {
  //println("controlEvent : "+theEvent);
  for(int i=0; i<15; i++){
  if (theEvent.isFrom(cp5.getController(data[i])) && eventTrig== false
&& counter<=16){
     println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c + ", "+
timertime + ", " +  theEvent.getValue() +", " +
theEvent.getController().getId()+ ", "+ data[i]);
     output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c +
", "+ timertime + ", " +  theEvent.getValue() +", " +
theEvent.getController().getId()+ ", "+ data[i]);
    } else if(i<3 && theEvent.isFrom(cp5.getController(dataCon[i])) &&
cp5.getValue(dataCon[i])==1){
     println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c + ", "+
timertime + ", " +  theEvent.getValue() +", " +
theEvent.getController().getId()+ ", "+ dataCon[i]);
     output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c +
", "+ timertime + ", " +  theEvent.getValue() +", " +
theEvent.getController().getId()+ ", "+ dataCon[i]);
    }
  }
  if (theEvent.isFrom(cp5.getController("N E X T")) && counter<= 16) {
    println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c + ", "+
timertime + ", " +  theEvent.getValue() + ", "+
theEvent.getController().getId() + ", "+
theEvent.getController().getName());
    output.println((counter-1) +", "+ h +":"+ m + ":" + s +", "+ c +
", "+ timertime + ", " +  theEvent.getValue() + ", "+
theEvent.getController().getId() + ", "+
theEvent.getController().getName());
  }
  switch(theEvent.getController().getId()) {
    case(100):
    if(cp5.getValue(dataCon[0])==1){
      cp5.getGroup("g"+1).setVisible(true);
cp5.getGroup("g"+1).open();
      cp5.getGroup("g"+2).close();
cp5.getGroup("g"+2).hide();
```

```
      cp5.getGroup("g"+3).close();
cp5.getGroup("g"+3).hide();
      buttonCon2 = buttonCon[0];
      nuCB2[1].setAlpha();
      nuCB2[2].setAlpha();
      //println("controlEvent : "+theEvent);
    }
    break;
    case(101):
    if(cp5.getValue(dataCon[1])==1){
      cp5.getGroup("g"+1).close();
cp5.getGroup("g"+1).hide();
      cp5.getGroup("g"+2).setVisible(true);
cp5.getGroup("g"+2).open();
      cp5.getGroup("g"+3).close();
cp5.getGroup("g"+3).hide();
      buttonCon2 = buttonCon[1];
      nuCB2[0].setAlpha();
      nuCB2[2].setAlpha();
      //println("controlEvent : "+theEvent);
    }
      break;
    case(102):
    if(cp5.getValue(dataCon[2])==1){
      cp5.getGroup("g"+1).close();
cp5.getGroup("g"+1).hide();
      cp5.getGroup("g"+2).close();
cp5.getGroup("g"+2).hide();
      cp5.getGroup("g"+3).setVisible(true);
cp5.getGroup("g"+3).open();
      buttonCon2 = buttonCon[2];
      nuCB2[0].setAlpha();
      nuCB2[1].setAlpha();
      //println("controlEvent : "+theEvent);
    }
    break;
    case(99):
    eventTrig = true;
    //println("controlEvent : "+theEvent);
    if(counter<16 && eventTrig == true){
    URL = "imageB"+counter+".jpg";
    counter +=1;
    //println(counter);
    resetButton();
    } else if(counter == 16){
        output.flush(); // Write the remaining data
        output.close();
      }
    break;
  }
}

void mouseReleased() {
```

```
  //if(cp5.isMouseOver()){
  cp5.getPointer().released();
  //}
}

void mouseDragged() {
  cp5.getPointer().released();
}

void keyPressed() {
    if (key == 'b' || key == 'B')
    {
      if (counter>2 && counter<=16)
      {
        //Use this to get back to the previous image
        counter -=2;
        URL = "imageB"+counter+".jpg";
        //println(counter);
        counter +=1;
        resetButton();
      }
     eventTrig = true;
    }
    if (key == 'r' || key == 'R')
    {
      //Reset the timer
      c.reset();
    }

}

void resetButton(){
for(int i = 0; i<15; i++)
                {
                    nuCB[i].setAlpha();
                }
    }
```

MyButton Class

```
// Create a custom Controller, please not that
// nuCheckBox extends Controller<nuCheckBox>, <nuCheckBox>
// is an indicator for the super class about the type of
// custom controller to be created.

class nuCheckBox extends Controller<nuCheckBox> {

  int current = 0xff0080FF;
  float a = 100;
  float na;
  float labelfill = 255;
  int y;
```

```
  boolean enterButton = false;

  // use the convenience constructor of super class Controller
  // nuCheckBox will automatically registered and move to the
  // default controlP5 tab.

  nuCheckBox(ControlP5 cp, String theName, int coloring) {
    super(cp, theName);
    current = coloring;

    // replace the default view with a custom view.
    setView(new ControllerView() {
      public void display(PApplet p, Object b) {
        // draw button background
        na += (a-na) * 0.1;
        p.fill(current, na);
        p.stroke(2);
        p.rect(0, 0, getWidth(), getHeight());

        // draw the custom label
        p.fill(labelfill);
        translate(18, 30);
        p.text(getName(), 0, 0);
      }
    }
    );
  }
void setAlpha() {
    a = 100;
    labelfill = 255;
    setValue(0);
  }

void setActive() {
    a = 200;
    labelfill = 0;
  }

  // override various input methods for mouse input control
  void onEnter() {
    enterButton = true;
    if (a == 100)
    {
      //cursor(HAND);
      //println("enter");
      a = 200;
      setValue(1);
      labelfill = 0;
    } else if (a == 200)
    {
      a = 100;
      setValue(0);
      labelfill= 255;
```

```
    }
  }

  void onRelease() {
    onEnter();
  }

  void onDrag() {
    if(getIsInside()){
    onLeave();
    }
  }

  void onReleaseOutside() {
    onLeave();
  }

  void onLeave() {
    enterButton = false;
  }
}
```

Class CheckBox2

```
class nuCheckBox2 extends Controller<nuCheckBox2> {

  int current = 0xffA0A0A0;
  float a = 100;
  float na;
  float labelfill = 155;
  int y;
  boolean enterButton = false;

  // use the convenience constructor of super class Controller
  // nuCheckBox2 will automatically registered and move to the
  // default controlP5 tab.

  nuCheckBox2(ControlP5 cp, String theName) {
    super(cp, theName);

    // replace the default view with a custom view.
    setView(new ControllerView() {
      public void display(PApplet p, Object b) {
        // draw button background
        na += (a-na) * 0.1;
        p.fill(current,na);
        p.stroke(2);
        p.rect(0, 0, getWidth(), getHeight());

        // draw the custom label
        p.fill(labelfill);
        translate(14,22);
        p.text(getName(),0,0);
```

```
      }
    }
    );
  }

  // override various input methods for mouse input control
  void onEnter() {
    enterButton = true;
      //cursor(HAND);
      //println("enter");
      a = 200;
      setValue(1);
      labelfill = 0;
  }

  void onRelease() {
    onEnter();
    //println("release");
  }


  void onReleaseOutside() {
    onLeave();
  }

  void onLeave() {
    enterButton = false;
    //println("leave");
    a = 100;
    labelfill = 155;
 }
}
```