


Mise en Scène Project Report

Project Overview:

A Cinema research paper is full of references from small scenes of movies, audio clips and still images. This project is an attempt to make the reading of a cinema journal easier and exciting by the use of cellphones and Augmented Reality¹.

The intention is to provide the user the flexibility of watching any multimedia content associated with a particular research paper using an Apple cell phone. The multimedia content is provided by the author of the paper, and is usually copyrighted.

The app, shows some markers augmented on top of the a live camera view of a research paper. The marker images in itself tells about the kind of multimedia content. For example, if the content is a video, it shows image: [] to let the reader know that there is some video to be explored. The same goes for the audio and the image gallery.

Development Team and Roles:

The team consisted of 10 members in the beginning. Finally the project kicked off with 8 members. The Project Team followed the Agile development methodology, which was the part of the CS 5640 Software Development course. We decided for certain roles:

Tracker: I-Chi / Joshua

Acceptance Tester: Ben / Joshua

Designer: Ben / Jenny

Database: Nitin / Mahak

Tool Guy: KJ / Nitin

Code Reviewer: KJ / Mahak

Integrator: I-Chi / Ben

Presenter: Nitin

Documenter: I-Chi / Riley

Installer: Jenny / Riley

¹ **Augmented reality (AR)** is a live, direct or indirect, view of a physical, real-world environment whose elements are *augmented* (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data.

Customer Contact Information:

Ted Rippey

Phone Number: (419) 372-7137

Email: theodor@bgsu.edu

Chris Wagenheim

Email: cwagenh@bgsu.edu

Technologies Used:

1. Xcode 5: Xcode is an integrated development environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS using Objective C as the programming language. Xcode 5, the latest version provided by Apple has been used in the app development.

2. Metaio SDK 4.5: Metaio SDK(Software Development Kit) empowers the Augmented Reality(AR) applications with wearable device support, improved visualization, speed and tracking stability. The Metaio SDK allows flexible deployment and licensing for your stand-alone or cloud-based project. The name of the image that is to be augmented is placed in XML file in order to access it at runtime. Metaio SDK 4.5, the latest version has been used for the development for the project

3. iOS 7: iOS(previously known as iPhone OS) is the operating system provided by Apple that runs on iPad,iPhone and iPod. So, iOS 7, the latest version has been used in order to deploy or run the app on iPhone.

In order to run the app on iPhone there is a need of Developer Certificate and Provisioning Profile provided by Apple, in order to maintain security of iOS.

App Overview:

The overview of the app is as follows:

First screen shows the names of all the journals and when the journal name is clicked then the list of volume numbers are shown related to the selected journal. After clicking on Volume number the summary page opens that shows the Journal name, Article name , Author Name from the data that was collected during the previous screens. There is a button at bottom named as 'Start Augmenting Content' that is used to start the camera view of the app. When the camera is put over the image that has augmented content then a play buttons opens virtually showing that it has video as the content. When the play button is clicked then the video related to the image plays in the AV player of xcode. There is a exit button on the right corner of the camera view, which when clicked then we get back to our previous screen that shows the summary. Also, there is a back button on each screen in order to go back to the previous screen in a sequential manner.

Elaborative description of the app development and app's Limitations :

Since the waterfall model limitations were studied in great details, The Agile software development method was experimented for the development of the iOS app. User stories were decided in the group meetings. The user stories were picked up by individuals based on the previous knowledge and experience of the technology involved in developing those user stories. The entire project development followed the Pair programming techniques of the Extreme Software Engineering.

Client meetings were frequently arranged to keep track of the requirement.

The app is intended to be used by an end user. For an author, to upload multimedia content, at the server, a separate Web site is used. So the app can be used by any user, as a viewer of the multimedia content.

The Cinema Journals have some fixed attributes, like The Journal name, Journal number, the Issue number and the Author & Title of the Research paper. So to search a particular Research paper, the user should know the Journal name, Issue number etc. Usually all this information is given on the Research paper itself on the header or the footer etc.

Thus in our app, the front end involves the tabular selection screens to navigate to the camera view. This is done in order to avoid a random database search. This will help the user to find out if the research paper has some augmented content or not. Of course if there are very large number of journal names, then this process of scrolling the names manually becomes a bad user experience.

The app has 3 screens for selecting Journal name and the Issue number. Once the information is selected, a next screen showing the summary of the Article selected pops up with a button to open the camera view to start augmenting.

As the camera opens, if we place the cell phone at a certain distance from the Research Paper, we can see an augmented marker. (only if the hard paper copy is the paper with some augmented content).

Now here is a catch. The app recognizes a tracking image from its XML file against the image from the real paper. If the camera is open, we can use any Research paper (with augmented features saved in the apps XML) to start augmenting the content.

For the Augmentation, Metaio SDK is used. Once the app is in the camera mode, it constantly scans for the tracking images. Thus if two papers have augmented contents, the camera simply recognize the place where the augmented content is, irrespective of the previous selection of the Journal name and the volume number. That is one of the limitations of the Project.

Also the user story to play the video, was initially dealt with the video being stored on the local storage. But when we attempt to save the video on the agiledev1 server, things started to becoming a little complicated. The attempt to save the video on a remote location, and accessing it in our app, remained unfulfilled.

One possible solution that occurred to the group is to harness the capabilities of the Youtube to display the video. This would automatically use the capabilities of a web browser and the use of various Google tools like Youtube and Picasa (for picture gallery).

The idea occurred late to the group, but seems feasible to use the capabilities of Youtube, to compress the video and make it possible to play across devices.

For compression, the following link provides an easy tool.

<http://www.reely.com/darren/Guides/iMovie%20Export%20Guide/>

For displaying a picture Gallery, there was a requirement of deliberation on using picasa web albums, or displaying the gallery using the default iPhone app. The work remain unfinished because other technical challenges took most of the groups efforts.

The Agile software development requires test driven development (First test then code). This is yet another aspect which was not followed during the development because the group was overwhelmed by the new technology. XCode workbench was learnt easily by the group, but the challenges posed by the Objective C language, took considerable amount of time.

The technologies that we were familiar with took less time and effort to finish a user story. Setting up a webpage to upload the information about an article and its corresponding multimedia content took lesser amount of time, because one of the group members was familiar with the php scripting. Setting up the database and connecting the first page of the app with the database through JSON didn't pose any considerable challenge. But, the screen transition, with filtered data kept on remaining the blockers till the end of the project.

The Implemented User Stories and the technical challenges faced:

The project has 12 user stories with 10 must haves and 2 should haves, out of which 5 user stories (all must haves) are completed.

1. The user Story: As a reader, I want to scan an image, so that an augmented content appears.

The user story concerns the part of the project where an augmented marker is shown on the screen of an iPhone running in the camera mode within our app. The image marker that is shown, takes the help of an XML file to match the image saved within the app. if the image is configured in the xml, and is stored in a file within the app, the app will match the real research paper with that of the image in the file and show the augmented marker.

The real challenge faced during the implementation of this user story was to figure out the way to update the xml file from the database automatically. Unfortunately we could not figure out the

way to do it, and hence its a limitation of our project that the XML file is updated manually.

2. User Story: As a reader, I want to touch the augmented marker, so that multimedia content(video) appears.

This user was to link the marker, on click event, with an associated video and play it on the device. The story was completed using the video file saved locally. It was planned that once the database is up and running, the video can be saved remotely. Later when the database was made, it was agreed that the video will be saved on the file system on the server with the http link to the directory being saved in the database. This part was also completed. But when it comes to pulling the video from server to be played after the augmented marker was hit, the real challenges occurred.

Another challenge that was faced is of the XML file being updated every time the Database is updated. There is an xml file that keeps track of the tracking images, so that the app can recognize the image when the real research paper is shown. We could not figure out the way to do the XML updating automatically. The xml file updated manually every time a new augmented content is added.

3. User story: As a reader, I want to navigate the list of augmented articles, so that I can select the correct article.

This user story was implemented as the initial front end screens. The navigation is based on the user selection from the list of options available. The first Screen gives the names of the journals. Upon selection of it, the next screens gives the names of the Volumes number. The next screen, gives the summary of the Article chosen.

The user story was initially implemented using data places in an array inside the code. It was assumed that the statements would be replaced by the sql statements. But connection with the data database is a bit tricky with iOS. We had to settle down with the option of JSON and PHP. Instead of connecting our app directly with the database, we let a php script connect with the database, and put data in an json array. Then parsing the data in our app, we can show the first screen. The next screen was a real challenge, partially because we have all the data in an array, and searching the same data and sending filtered data using segue remained a challenge till the end.

By the way, once the camera view is on, the app can recognize any research paper whose information is saved on the XML file called tracking.xml.

4. User Story: As a Scholar, I would like to view, on a web page, which articles are compatible with the app.

This user story in its simplest form, is a web page displaying the contents of the database. The contents of the database itself says which articles are compatible with the app. The contents are accessible via a web browser, outside our app environment.

5. User Story: As a Scholar, I would like to upload augmented content for an article via the web. Another web page is added to let the author of an article upload any multimedia content into the

database, so that augmented capabilities can be enforced on that document.

Lessons Learnt throughout the project:

The SCRUM meetings was the best part of the project development. The timely meetings distributed the load within the team to focus more on troubled areas. Though the absolute paradigm of Extreme Programming was not used, but it was a good learning experience. Kanban board could not be updated on time because SCRUM meetings used to take place within the classrooms. Thus The power of Kanban was not utilized fully.

The same goes with 'First Test then Code' methodology. It was agreed during one meeting that i could be done, but the time constraints and mental inertia for learning a new tool to write test cases became a major drawback for not using this methodology.

Also the time boxed iterative method was not met gracefully. After the first iteration, it was While User Stories changed in the middle of the project, we figured out that there was no yardstick to set the priorities and Estimate the time and effort. When a user story was selected, most of the time was invested in performing an XP Spike. We looked for a suitable pairs for the Spike, but it remained the same throughout the duration of the project mostly because of the challenges posed my objective C.