



The Mobile Stage

ECE1778 – Final Report

April 11th, 2013

Authors:

Remi Dufour (1000078694)

Montgomery Martin (994791784)

Mike Dai Wang (993811951)

Word Count

Project Report: 1985

Apper Essay: 486

Table of Contents

Table of Contents	2
Introduction	3
Overall Design and Description of Each Part	4
Statement of Functionality & Screenshots from App	7
Lessons Learned and Reflections	11
Contribution by Group Members	12
Apper Context	13
Future Work	15
Business School Entrepreneurship Class Takeover	16
Resources Used	17

1. Introduction

"The Mobile Stage" is an iOS application for audience use during interactive, mixed-media theatrical productions and performance art installations. It is designed to foster greater engagement between theatre audiences and performers by expanding the creative scope for live performance into augmented reality.

Audiences use the camera of their devices to see a user-selected avatar of a "digital performer" who is projected onstage via Augmented Reality. The digital performer interacts with physically present actors as if they were really there, telepresent onstage in mixed reality.

At key moments in the narrative, audiences will be given the opportunity to vote on storyline changes. The production crew will communicate the results to the cast, allowing the audience to dramatically influence the course of the play.

Audiences will be able to comment on the action and share their experience online using in-app social media integration. The production crew can also use social media to connect with the audience, providing commentary to supplement the narrative or clues for their future choices.

2. Overall Design and Description of Each Part

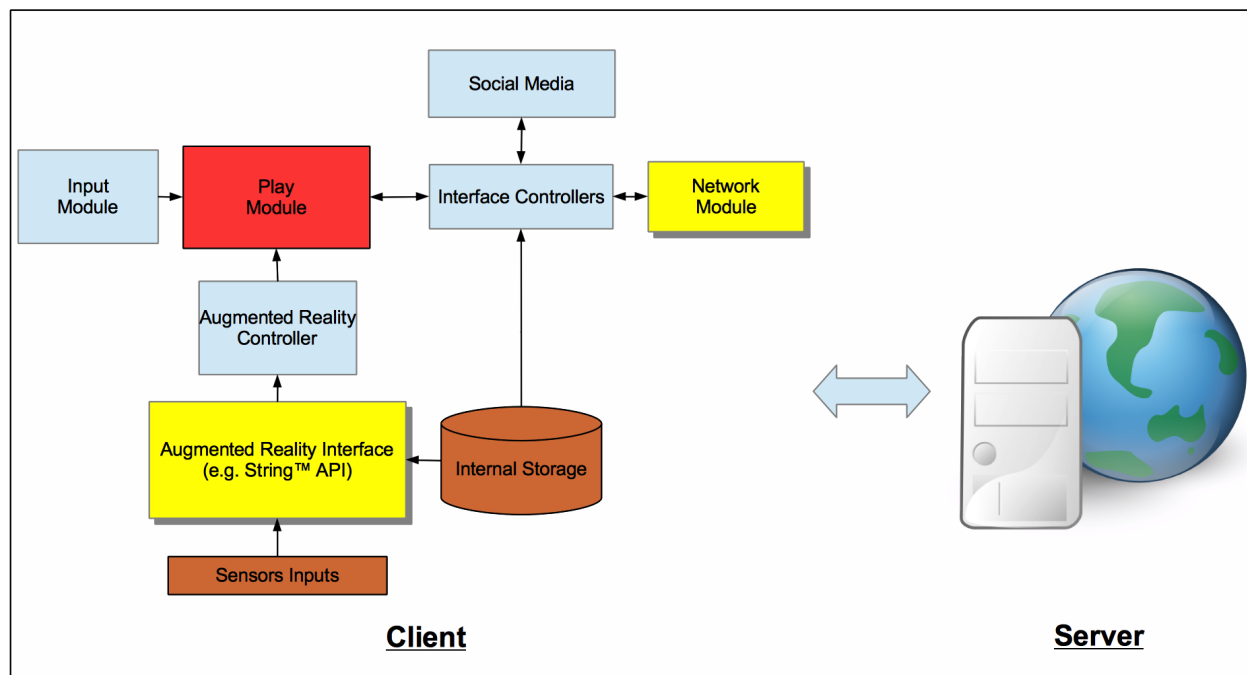


Figure 1 – Block Diagram

Client

Written in Objective-C and C++ using XCode and targets the iOS platform. While only officially supporting iPad, the app is targetable on iPhone.

Input Module

The Input Module reads the entire storyline and stores it into a local data structure. The storyline itself is represented in the form of a table, which contains event types (AR, Poll, etc.) and the time at which events are introduced. Event-specific details are also contained within this table. This module gets used only on program load as the events are stored in memory. For reusability, the storyline is presented in the form of an XML file, which can be reprogrammed by whoever is creating a theatrical play.

Play Module

The Play Module is the main component of our mobile device application. It acts as a central manager that commands the playback, flow and visibility of the relevant interfaces at various points of the storyline such as the character selection and the poll screens.

It is also responsible for invoking other modules, such as the Input Module and the Augmented Reality Controller, whenever they are needed. Interactions between the user-interface and the various blocks of our system will also be controlled within the Play Module.

Augmented Reality Controller

The Augmented Reality Controller is in charge of setting up the various parameters to launch an AR overlay on top of the camera input. It is also responsible for performing on stage marker recognition and video processing to realistically display the virtual character in the performance. The AR interface is mainly conveyed makes use of the String™ SDK, but was designed to be easily swappable.

Interface View Controllers

The interface view controllers are responsible for interactions between the audience members, or between the performers and the audience. For example, the performers or directors have the option of sending real-time notifications to the audience via the Network Module. Also, voting is an integral part of the storyline, where the audience is asked to interact on a situation and suggest a course of action. These poll results are forwarded to the performers, who may in-turn respond in a customized manner.

Social Media

Social media is another important part of the interactive experience as it can be accessed from any stage of the play at any time. In this app, Twitter is chosen due to its real-time nature. Rather than creating a simple timeline of Tweets, the social media interface is designed to mimic a chat program to encourage participation and discussion.

Audience members can Tweet their own thoughts and see discussions from all participants by using a common Twitter hashtag. In addition to discussions, Twitter also serves as a platform for delivering polling results and important storyline decisions.

Server – Connected with the Network Module

Written in Python with the Twisted networking engine and deployed on an Amazon EC2 Cloud.

The server infrastructure enables the interactivity and allows play production personnel to collect information from a large number of deployed mobile devices. They can also send commands to devices to tightly couple them to the storyline.

Our server follows a reactor event handling model where it is able to simultaneously handle connections from multiple devices. A custom TCP communication protocol was developed to allow simple and fast commands to be sent from the server and received from clients. These commands allow the apps to be remotely launched, triggered into different stages and send messages, alerts to the audience. The connections between the clients and the server will be maintained throughout the period of the play.

The aforementioned commands can be controlled from a simple and efficient web control interface. The stage control personnel have access to all commands with the click of a button and real time display of voting results. The voting results can also be tweeted directly from the server and shared with the audience through social media.

3. Statement of Functionality & Screenshots from App

The Mobile Stage can be launched from the iPad's home screen (Figure 2), which brings an audience member to the Standby view, as shown in Figure 3. It serves as a placeholder for downloading the play assets, feature which hasn't been implemented yet.



Figure 2 – App Icon

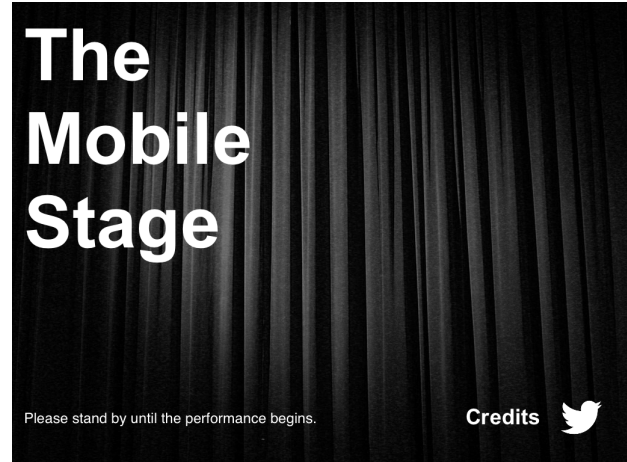


Figure 3 – Standby Screen

Before the play begins, the audience have the option to read the play programme by clicking “Credits”, which brings them to the Credits screen (Figure 4). They can alternatively use the social media interface and interact with other audience members until the producer launches the play. This brings everyone to the Home screen (Figure 5), where various interactive events will get triggered.

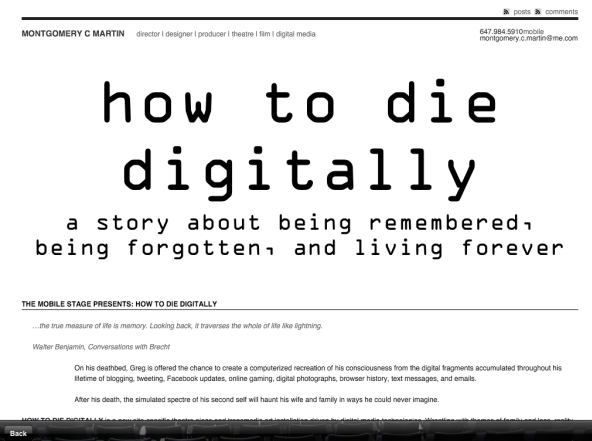


Figure 4 – Credits Screen



Figure 5 – Home Screen

Three event types that may happen are a Character Selection event, Voting events and Augmented Reality (AR) events. As shown in Figure 6, a Character Selection event let each audience member pick a particular AR character, which tailors to his personal preference. Throughout the performance, the audience is brought to the voting screen (Figure 7), where his decisions will influence outcome of the story.



Figure 6 – Character Selection Screen

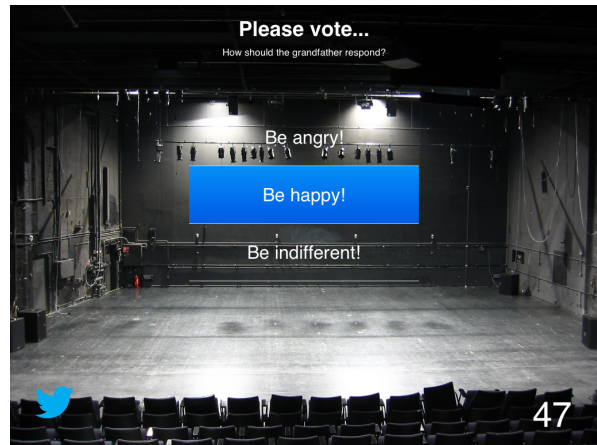


Figure 7 – Voting Screen

AR events are triggered multiple times throughout the play (Figure 8). As of now, only pre-recorded videos are supported, although multiple videos are played depending on poll results. Figure 9 shows a human-size AR video in a real life theatrical environment.



Figure 8 – AR in a Test Environment



Figure 9 – AR in a Theatrical Environment

At any time throughout the play, the audience may access the social media by pressing the Twitter button, where they can read other Tweets (Figure 9). In Figure 10, a person is composing a new Tweet, which gets displayed on everybody's mobile device within 15 seconds.

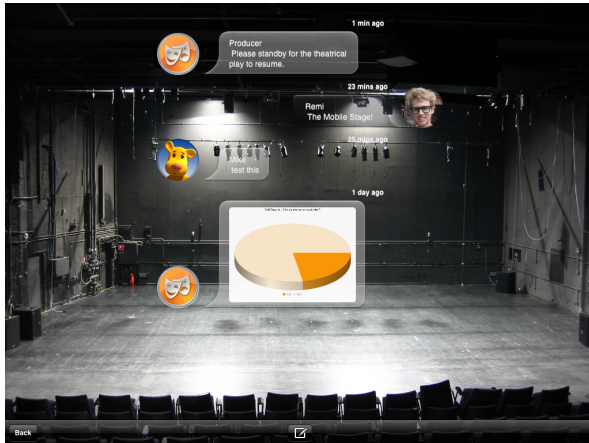


Figure 9 – Social Media Screen

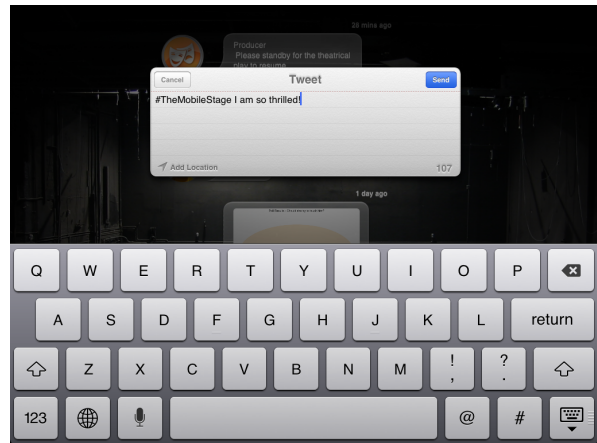


Figure 10 – Social Media – Compose

An audience member may use his personal Twitter account, although such decision could be revised in the future. Polling results are accessible throughout the Social Media interface (Figure 11), for which each graphs can be expanded through a simple finger press.

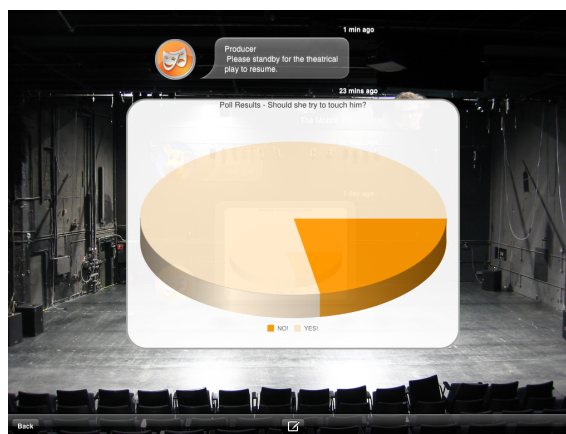


Figure 11 – Polling Results

The server interface is accessible using any web browser, as shown in Figure 12. Its location can be anywhere, ranging from a local host to a cloud server. It allows a producer to control the scenes of a play, using the playback buttons. Polling results are sent to that same server, where cast members can customize their act depending on the feedback.

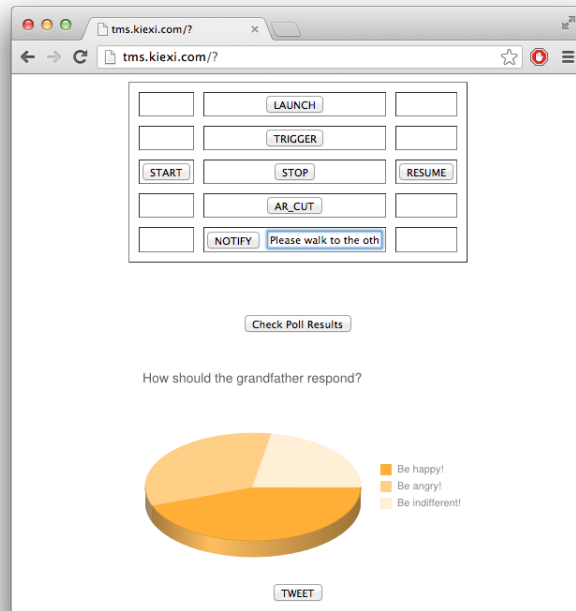


Figure 12 – Server Interface

4. Lessons Learned and Reflections

If this app had to be remade from scratch, we would involve the audience during the user interface design phase to generate feedback on the design. Furthermore, we would recruit actors sooner in the process and allocate more rehearsal time and pre-production work with the developing app. This would allow for more material for the programmers, and more polished and complete content to demonstrate the application itself.

We experienced some limitations with AR marker detection using the String AR SDK. It would be a worthwhile exercise to compare to other more open-minded solutions, such as Vuforia from Qualcomm.

Last but not least, social media forms an important part of the app. However, the way audience members engage with social network platforms needs to be carefully studied and examined. While we provide Twitter to encourage engagement and discussion, incorporating other services such as Facebook could open more channels for promotion and interactivity.

5. Contribution by Group Members

Remi Dufour – Programmer

- Designed and Implemented the Input Module and designed the XML which gets consumed by the app;
- Designed and Implemented the Play Module, which is responsible for managing the theatrical playback and interfaces;
- Implemented the video player which overlays a video stream on top of an OpenGL ES texture;
- Integrated the String SDK Augmented Reality libraries to our project;
- Created new shaders that overlays the video texture to the screen;
- Implemented the credits, character selection and voting features, including integration with a third-party 3D roulette.

Mike Dai Wang – Programmer

- Implemented the Social Media interface and integration with Twitter API and the server module;
- Implemented the Network Module, custom messaging protocol and remote control feature;
- Implemented the polling and voting feature with real time results collection and generation.
- Implemented the server side of our app, such as connection and communication;
- Performed investigations for Augmented Reality alternatives;

Montgomery Martin – Apper

- Came up with the original idea and described the high level behaviour of our app;
- Interface graphic design for the Launch, Standby, and Character Selection screens; designed AR marker, which had to respond to a specific set of criteria.
- Provided a storyline to be consumed in our demos;
- Filmed, composited, and edited video content seen in Augmented Reality
- Worked with actors and audiences for in-rehearsal testing and demonstrations of the app for a live audience

Both programmers and the Apper jointly performed planning, interface designs and testing.

6. Apper Context

Mobile technology and mixed-reality experiences raise agonizing questions about the transformative power of technology on perception, embodiment, and the boundaries of performance art. In my PhD research at the Centre for Drama, Theatre and Performance Studies and the Knowledge Media Design Institute, I am particularly concerned with how emergent digital technologies can foster a greater sense of interactivity, immersion, and creativity in theatrical performances. At the same time, I seek to address the anxieties and disconnects caused by using these technologies in performance by investigating strategies to deepen the perceived intimacy and immediacy between audiences, actors, and telepresent/ mediated (or otherwise technologically embodied) performers (as opposed to traditional theatrical modes of corporeal embodiment).

In my past work I have experimented with several means of creating a telepresent performer using projected images combined with traditional "smoke and mirrors" illusions. However, using augmented reality to create a telepresent performer is an especially troubling to established models of embodied performance due to the mixed-reality nature of the technology. My objective with the Mobile Stage was the gauge the potential applications and implementations of augmented reality and mobile device technologies in live performance as a potential entry point into resolving some of these issues.

Of course, the parallel development of a novel mobile application and a theatrical piece is a staggeringly difficult undertaking, but I have come out of this project with a clearer sense of what is possible (which turns out to be a great deal) and how to better implement it. The true testing ground for the Mobile Stage would be a fully realized production, which I hope to pursue for the 2013-2014 academic year.

The work conducted here has provided a clear proof-of-concept to launch future projects, and discussions and demonstrations with work with my colleagues indicated a strong interest in The Mobile Stage. Central to these discussions has been the efficacy of this kind of effect in traditional theatre venues versus nonstandard performance sites, such as gallery installations or site-specific performance (think *Nuit-Blanche!*). Given that app is called the Mobile Stage, and it is deployed on handheld devices: why should the audience remain seated? Site-specific works where the audience is able to freely roam and discover rich augmented reality content, combined with a compelling story, should set the stage for an immersive, interactive, and memorable experience.

Questions of presence remain, however: it is an open question as to what extent is a pre-recorded image of a real actor constitutes a genuine presence in the theatre space. In other words, when is the digital performer "really there"? Perhaps a sufficiently developed illusion within a well-staged performance is sufficient to achieve the semblance of presence, or perhaps more creative technologies, such as haptic feedback, motion sensors, and wearable AR glasses are needed before the illusion is tangible enough to be taken as "real" by an audience. What if, instead of a pre-recorded segment, it was a streamed broadcast?

7. Future Work

In sharing the app with actors and audiences, we discovered many possibilities for future usability improvements in the app:

- Audiences were interested in having more ways to interact with the Augmented Reality beyond voting on the story;
- Actors are receptive to the idea of performing with a pre-recorded augmented reality partner, and can convincingly play opposite a pre-recorded “digital performer” with some practice as long as they have the app in-hand;
- Originally we planned for the audio to play from the theatre speaker system instead from the device itself. However, difficulties with synchronization indicate that it would be best for audio to be integrated in-app; audiences will hear the audio through the built-in speakers or headphones;
- On the server side, more control freedom would be of considerable benefit in rehearsal and performance. A possible future project would be to make the Mobile Stage integrate with existing professional show control software.

Ambitious future improvements to the application would include:

- An improved interface for audiences, performers, and stage crew;
- Creating the augmented reality performer from a composite of multiple camera angles which uses the “Bullet Time” technique to produce a three dimensional image;
- Deploying the Mobile Stage in a fully realized theatrical production featuring rich interactive content, set in a site-specific venue where audience members can walk through the space freely;
- Developing the Mobile Stage for wearable devices such as Google Glasses.

8. Business School Entrepreneurship Class Takeover

The project will be open sourced, and we have no problem with having a Business School class on take the idea up as a collaborative project.

9. Resources Used

TinyXML – <http://www.grinninglizard.com/tinyxml/>

- Used in the Input Module to parse XML files.

iCarousel – <https://github.com/nicklockwood/iCarousel>

- Used in the Character Selection screen to pick a character using a 3D carousel.

String AR SDK – <http://www.poweredbystring.com/>

- Used in the Augmented Reality screen to track a marker image.

UIBubble – <http://alexbarinov.github.io/UIBubbleTableView/>

- Used as baseline to implement Social Media interface.

Twitter API – <https://dev.twitter.com/>

- For sending and receiving tweets.

Twisted Matrix Python networking library – <http://twistedmatrix.com/>

- Used in server implementation.