Essay Introduction Multimedia

My experiences with designing and implementing an interactive multimedia application.

Maurits Dijkstra (student 1583034, group 08310)

Abstract

In this essay I will explain the process I followed to create an interactive media application for the Introduction Multimedia course at the Vrije Universiteit. I will also elaborate on some of the technologies used to produce this application and possible changes made to these applications. To design the application we followed the standard process of first brainstorming, picking a concept and then working out the interactive aspect through the use of story graphs and story boards. To implement our design we used a cheap DV-camera mounted on a tripod to film our material, iMovie to edit the raw material and finally Adobe Flex Builder together with the XIMPEL framework turn it into an interactive movie.

Introduction

Structure and method This document consists of four distinct sections:

- The introduction. This sections explains the structure of this document, the approach used and the context of the subject, interactive media applications.
- A section describing the design phase of the project. In this section, I describe the steps taken while designing the application. This includes brainstorming, refining our concept, designing interactivity and orientation on our interactivity platform.
- A section describing the implementation phase of the project. In this section, I describe the steps taken while implementing the application. This includes filming, editing the video and building the application.
- Finally, a section describing my final experiences with the assignment and a short opinion about the 'usefulness' of this course.

The formal form that most accurately describes the tone and content of this document is the review/background form, although some elements of the technical analysis form also appear, especially in the implementation phase.

Background An interactive media application is, as the word describes, a multimedia application. This includes audio, video, text, touch, smell and pretty much any other form you can think of. The difference with a 'regular' multimedia application, however, is that the user is excepted to interact with the application in this case. You could say if traditional multimedia apps are static and one-way, from the content publisher to the viewer, then interactive multimedia apps allow the user to have some kind of influence on the application, giving a more involved and dynamic experience.

Design phase

Brainstorming Obviously, the first thing that had to be done was to think up a decent concept for our interactive movie. So me and Esra, my project partner, met in a computer room at the W&N-building to brainstorm about an idea. We just sat down and discussed the ideas that came up in our minds and eventually ended up with three distinct concepts:

- The first concept focused on targeting the stereotypical image a lot of people have of beta studies. The viewer would see a movie of a high school student sitting in a class room, daydreaming. Based on which subject the student is following, which would get to be picked by the viewer, a movie depicting the stereotypical view of that field would be shown. Mathematics would have people with long beards, chemistry would have people in lab coats doing nothing except looking at test tubes all day. After this stereotypical view the movie would try to present the reality of people who work in these fields. For chemistry we could show process engineers working at a big company like DSM and for mathematics we could show encryption code breakers. We could also tie in a quick quiz into this concept to add more interactivity.

- The second concept was based on the fact that although beta studies can be pretty boring at times, there are prospects of a very fun, rewarding career five or ten years down the road. The movie would involve a student of a viewer-pickable beta field during a normal day at college for him or her. The movie would show a pretty boring view of the day, with lots of classes and homework. After showing this fairly dreadful picture of life as a beta student, the movie would show the users the fun parts, such as practical assignments, trips abroad and solving difficult practical problems.

- The third and last concept was an attempt to make the viewer aware that several research groups at the VU are respected around the globe for their quality. Examples of this are the Artificial Intelligence faculty, the Distributed Systems group and several others. Another focus point of this scenario would be the willingness of international students to get a master's degree here. The movie itself would be about a fictional conversation between two students, one Dutch and one from abroad. The viewer

would get to select which student to ask a question and would get several questions to pick for each of the two students. Questions could include asking about their career prospects, the student life in amsterdam and for the exchange student, asking why he came to the VU. After a suggestion by the student assistant we decided to not just show the footage of the student answering, but also material that applies to the answer, such as pictures and video footage.

Picking and refining our concept After the gathering of ideas we did we needed to pick the best one. As I mentioned, there were several ideas to consider, partly because the assignment required it and partly because we did not feel like putting all our eggs in one basket. If one idea proved to be hard to implement we would always have several fallbacks to choose from.

In the end we decided to go with the third concept. The motivation for this is that I recently talked to a guy from an Eastern European country on the internet who really wanted to study at the Vrije Universiteit. In this case it was because of the MINIX3-project though, and not anything media-related. We still decided to focus on the international prestige enjoyed by some bachelor and master courses taught at the VU. It seemed like an original perspective compared to the basic "Beta studies aren't boring!" or "Look how cool Amsterdam is!" ideas that we also thought of. In addition to this the other two concepts had serious problems as well. Concept one was deemed by us to be too difficult to film, possibly requiring us to shoot at three or more locations with better equipment than what we had available to us We didn't choose the second concept because we didn't think it was interactive enough; it only had a single moment of viewer choice. The other concepts had many more choices and the project is about interactive media after all.

With the scenario down we turned to the more practical aspects of planning the assignment. Esra knew an exchange student who was currently studying for her masters degree in Artificial Intelligence. We asked her and a friend, also studying for her masters degree, if they would mind to be interviewed by us. They didn't mind, so we arranged a date to do the filming and moved on with the design of the application.

Designing interactivity With the general story down we were now up to the interactive part of the application. Our application had to be interactive, after all. We considered several forms of interactivity, from an online quiz to a small mini-game. Some of our interactivity ideas were difficult to implement because of restrictions imposed on us by external factors, others were too boring, time-consuming or difficult to do. We settled on our initial idea of interactivity: a simple 'model of viewer choice'. Our user would get to choose which questions would be asked and which student would get asked those questions. Our motivation for this was our uncertainty whether XIMPEL was up to the challenge of providing more functionality than this. We knew this would result in tree-shaped story graph and roughly which questions we were going to ask,

so it wasn't hard to build a story graph out of this with my trusty diagramming app[1]. With the story graph completed Esra made the storyboards by drawing them on paper and scanning them with her desktop scanner.

The platform Although we were mostly done designing the application we still had one major hurdle left, namely the platform. For the most part we didn't know what we could and couldn't do with XIMPEL, the platform we had to build our application on. We did do some basic orientation on its capabilities, but we knew could still be bitten by possible limitations or other problems with the platform. I knew XIMPEL was built on the Adobe Flex framework, which I use at work a lot for developing Rich Internet Applications (RIAs). I figured I So I downloaded the source code, copied the sources into a Flex Builder project so I could use its Eclipse-based IDE for development instead of having to run mxmlc by hand. Once my development environment was in working order I took a look at the sources. After that I viewed the XML schema file for the playlist files used by XIMPEL. This, together with the examples available on the internet, gave me a good understanding of the structure of the configuration files.

I was somewhat saddened by the fact that the actual 'workhorse' of XIMPEL, the XimpelPlayer component, was closed source and buried inside a Flex library file. The only code available was the ximpelApp.mxml, which had rather little customization potential. I had plans to add all kinds of neat features to XIMPEL, like vector overlays and a more flexible question system, but none of this was possible.

After orientating myself on the XIMPEL platform I decided that there were some things I could add, although it wasn't as flexible as I'd hoped. The things I considered adding were sound support and webcam support, especially recording. I thought it would be really cool to stream video from the webcams of users watching the movie to a flash media server[2], which would then store them. so we could see the reaction of people viewing the movie.

Implementation phase

Filming Having done all the hard work of designing the interactive application it was time to get to the fun parts of this project. First up was collecting all the assets we needed, such as the video material and the pictures to be shown. Esra browsed google images for decent photos and other assets to be shown. We also made an appointment with the two students featured in our movie to get the filming done.

The camera and tripod rented from the VU MediaXperience lab in the main campus building. The filming was done opposite of the hallway leading to the cafeteria in the W&N-building. In retrospect we probably should have filmed a different place, as it was very noisy at this spot. In our finished movie

[1] Omnigraffle Pro
[2] Red5, in this case.

the humming of a vending machine is clearly audible, as is the noise made by students walking through the hallway. I would ideally have removed this noise from the recordings, but I'm not very proficient with audio editing tools and we were on a tight schedule. Filming the scenes over also wasn't an option, as Natalie and Nataliya were both rather busy and didn't have a lot of free time to do it over again.

Editing the video After the filming was done we sat down a week later, again at the MediaXperience lab. I plugged the camera into my laptop and transferred all the video to my harddisk. The editing would be done with iMovie[3], a popular movie editor for Mac OS X. Even though iMovie is aimed at consumers making quick holiday movies, it is still a pretty intuitive and powerful tool. I chose to use this application instead of Adobe Premiere Elements installed on the media workstations on because I'm more familiar with it. Also, I wanted to be able to edit video from my home instead of having to visit the media lab every time a small change needed to be made.

The actual editing wasn't very eventful, although our editing software proved to be a bit too limited at times.[1] The learning curve was pretty gentle, hower. Still, if I were to do it over I probably would've gone with a more advanced piece of software because some of the limitations encountered were pretty annoying. The software, for example, did not support leaving title overlays visible at the end of a movie clip. This made adding titles to a video that would be looped in the XIMPEL application a real pain. I had to manually chop off the last second so the fade-out transition would be gone.

About six hours later and several problems conquered most of the footage was edited and ready. The only step left before we could start building the application was getting our video to the FLV format so the Flash player could read them. We had the Flash video encoder plugin installed which allows you to directly export to the FLV format from your editing application, but this did not work correctly. Consequently, we had to resort to using the manual FLV encoding app provided by Adobe. This provided fine results, but having to encode twice took quite a lot of time. One full run of exporting all the movies and converting them to FLV took about thirty minutes. Luckily there was more than enough coffee to drink during these times.

Building the player With all our media converted, edited and well we were ready to build the application. This entailed writing our own player config and playlist files, customizing the standard player application and adding any features to it we need. We chose to start off with the playlist, after a five minute edit of the player config to suit our needs. After starting up Flex Builder and building the player I made earlier I opened the video.xml file in my favorite text editor. There isn't a lot to say about the process of creating a playlist file, it's pretty 'XIMPLE'. Our movie utilized all the features XIMPEL had to offer except the true-or-false questions and the scoring system in general.

[3] iMovie ships with the iLife multimedia application suite.

I mentioned earlier that I wanted to add a few features to XIMPEL, namely sound and webcam support. Of these two features only one made the final cut; webcam support was deemed to be too time-consuming to implement. I added sound support by adding a few calls, variables and functions to the Script-component already defined in ximpelApp.mxml. Below are included the relevant pieces of my additions.

```
private var soundRequest:URLRequest;
private var soundFactory:Sound;
private var song:SoundChannel;

private function initSound():void {
    soundRequest=new URLRequest("audio/carol_of_the_kings.mp3");
    soundFactory=new Sound();
    soundFactory.addEventListener(Event.COMPLETE, completeHandler);
    soundFactory.addEventListener(Event.ID3, id3Handler);
    soundFactory.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
    soundFactory.addEventListener(ProgressEvent.PROGRESS, progressHandler);
    soundFactory.load(soundRequest);
    playSong();
}

private function playSong():void {
    song=soundFactory.play();
    song.addEventListener("soundComplete", soundCompleteHandler);
    song.soundTransform=new SoundTransform(0.2);
}

private function soundCompleteHandler(event: Event):void {
    trace("soundCompleteHandler: " + event);
    playSong();
}
```

Above code is pretty straightforward. The variables are defined outside of the functions because they need to be accessible globally. The first function initializes the sound engine, loads the file and plays it for the first time. The second function plays the sound, adds an event listener so it knows when the audio file is done playing and sets the volume to twenty percent so it doesn't get in the way of the dialogue. The third and last function of interest is the sound completion handler. This function gets called each time the music completes playing. The only thing it does is call the function to play the sound again, so the music loops.[2]

Conclusion

This project was quite nice to do and a refreshing distraction from the regular theoretical and programming courses. Still, in my opinion, the focus of

the classes is a little too broad and this distracts from the actual goal, which is building interactive media applications. Some classes on video editing and maybe even a little bit of Actionscript/Flex would be nice. However, I hear things are going to change significantly next year, so what I'm saying may not even be relevant anymore.

References

[1] Jim Heid. *The Macintosh iLife 08*. Peachpit Press, January 2008.

[2] Jeff Tapper, Michael Labriola, Matthew Boles, and James Talbot. *Adobe Flex 3: Training from the Source*. Adobe Press, April 2008.