Touring the VU

Jeroen Rietveld

Introduction

In context of the first year course Multimedia at the Vrije Universiteit Amsterdam, an interactive video application was written using material recorded at the university. The end result is small interactive playground with which the user can walk through the exact sciences building of the university, check out interesting locations, hear talks by students and teachers and find information about the university. This essay will be a technical analysis of the work done. This report will explain where the idea of making a virtual tour came from, what problems occurred along the way of making it and how the end result was created. After reading this a student should be able to redo the steps or extend on the existing source code.

Background, the choices made

The assignment given was actually to make a viral video in order to attract students to go and do exact sciences studies like computer science, artificial intelligence, mathematics and what not. All this preferably at the VU of course. The technical requirements were to give the user a couple of choices during the video which influence what happens next. The assignment was to be done by two students taking the course together. In the beginning there were two ideas, one more fitting the technical requirements and the other being more compliant to the rather strict requirements on the video having to have something to do with the VU and exact sciences. Both ideas will be described in short next.

First there was the Chinese lesson. A student of the faculty of Chinese Studies of Leiden University was asked to help out and a short lesson of for example counting in Chinese would be recorded. The user would have the option to go back to hear some part again, turn on the English subtitles (the speaker would be Dutch), and finally he could do a test and score points.

The second idea involved a tour around VU. Photos and videos would be taken all around the building and the user would have a virtual sandbox and could go wherever he would want to.

It turned out that the first idea didn't comply to all requirements, and the second one was just way too much work and didn't fully comply to the requirements either. The idea of the tour around the VU was updated only to include the FEW (exact sciences) building and it would feature professors of different study programs who could be found at places distributed around the building and would tell something about the study.

The next steps involved choosing the paths through the building which could be taken, contacting and interviewing professors and students and taking the photos and videos.

Technical analysis

This section first features a description of the routes that can be taken in the video and why these were chosen. Then a detailed description of the scenarios written and the interviews taken. Finally a

description of how the final product came into being, how the actual program was written and some hints for people who would like to either do the same or extends on the source code of this project.

Locations and routes

As was noted before, the first idea was to include the whole of the VU in the video. If Google can do all of San Francisco [1] and probably the whole world, then why wouldn't a couple of students be able to do the whole university. Well, it turned out to be quite large and actually the FEW building alone with 6 stories, full of stairs and elevators, turned out to be a bit too much. Therefore a couple of important locations inside the building had to be chosen to be featured in the video.

Of course the user starts outside the building and wants to go inside. Once inside there are of course lecture rooms. Let us not forget that this video will be checked out mostly by high school students who have never seen a large lecture room. Or a cafeteria in their school, so a lecture room and the cantina had to be featured in the video. Also the society for computer science students, Storm, will be interesting for oncoming students to look at, as well as their book store. Then there are the computer rooms where a lot of time will be spent when studying at a university, and FEW has quite a few different ones of them. Not all universities have different rooms with Windows, Linux, Minix and Mac OS running on the computers. Actually those are about all locations you will find a usual student at the FEW faculty. There is the library, but at the moment of recording it was hard to find and being renovated. There are of course a lot more interesting locations at the main building of the university, like a multimedia lounge, a large restaurant, locations outside to sit and play sports and so on. But the decision to stick to the FEW building had been made, as the assignment concerns exact sciences mostly.

Then there are the professors and students. A nice idea was to actually take an interview on a professor in a lecture room, in order to combine these two and at the same time make the users see what it is like to be in a lecture room and listen to a lecturer talking. Other professors could just be interviewed in their own offices and students just somewhere in a hallway or in a computer room for example.

These locations are spread all over the building. The entrance and exit are of course on the ground floor, as are Storm and the cafeteria. There are large lecture rooms accessible from the ground floor as well as the first floor. Computers rooms are virtually all over the place. In order to make connecting all locations easier in the program, it was decided to have all connections go to the elevator. This also means that it would be nice to have the locations spread over as many floors as possible. That is why this final setup was chosen, sorted by floor:

- Floor 0: Entrance, exit, Storm, book store, cafeteria
- Floor 1: Lecture room
- Floor 2: Professor computer science
- Floor 3: Professor artificial intelligence
- Floor 4: Computer rooms

This means all connections on floors 1 and higher are easy. Just film and implement into the video a connection between the elevator and the one location on that floor. If routes had to be filmed from all locations to all other locations, this would not only take a lot of time but also a lot of space on the server

hosting the video as well as bandwidth for the user downloading the video. In order to make the ground floor doable, use was made of connection points between the different locations. All routes go through these points. So for example in the large hallway somewhere in between the cafeteria and the entrance there is such a point. This also makes the user feel more in control because he can just stop there, have a look around and decide which way to go next. This requires a description of how the user will actually experience this whole tour.

Scenarios and interviews

The user will maneuver around the building by clicking on specific locations of photos taken. At all connections points panorama photos have been taken through which the user can look around. If there is a doorway leading to another location, this is clearly clickable. Once the user clicks one of these doorways a video starts which leads him to this next location. Once there, new photos are loaded and the user can look around again. If a professor is encountered the user will see him introducing himself and then be presented with a couple of subjects over which the professor can tell the user if he would want to hear this. This will seem the interviews seem shorter and at the same time provide the user with only the information he wants. It will also make the video seem (or be) more interactive.

The persons chosen to be interviewed are professors of the different studies within the faculty of exact sciences. Frank van Harmelen, professor of Artificial Intelligence; Natalia Silvis, study advisor of Computer Science and Huib Henrichs, professor of Physics and Astronomy. A small catastrophe prevented an interview of the latter to be taken. The whole course program of the physics faculty had been restructured, throwing out astronomy altogether. Also Tobias van Dijk, student of Information, Multimedia and Management would be interviewed.

As the most important goal of this video is to get people to come to the VU and to FEW and subscribe and study there, these people would have to tell something about the course they represent, why people should study this, particularly why at the VU. What makes the VU special over other universities offering a course of the same name? What kind of jobs can people expect to get once they complete the study? Are there opportunities to go abroad during the bachelor, during the master? Questions like these would have to be answered by the interviewees, preferably in a way that could convince people who were not considering studying at the VU to do so now. That is why those professors were chosen. From the experience of the author they would be more than others able to do so.

Putting it all together, the making of the virtual tour

For making the interactive video a package had been provided by the VU, under the name Ximpel. This package allows programmers to make playlists of videos. It is possible to have questions displayed during a video which can be answered by the user. Depending on the answer selected by the user, he gets points or not. It's also possible to have branch points during the video, where depending on what a user selects (one out of a maximum of three choices), another video from the playlist is jumped to.

The framework of Ximpel is quite strict. The idea of a tour through the VU does not really use questions or points in the sense Ximpel does. Also having only three choices is rather limiting. The extra fact that it seemed hard or impossible to display a lot of clickable links to sites containing important information

about studying at the VU made the author choose not to use Ximpel at all for this assignment. Use was made of Adobe Flex instead. Using Adobe Flex Builder a Flash program can be written using MXML and ActionScript as languages. If you have done some programming and webpage building but have never heard of these languages, don't be warned, as they are quite easy to learn. Just keep the api documentation ready at hand.

At first a simple framwork similar to the one provided by Ximpel was set up. A video screen in the middle, with a menu in the upper left corner which allows users to select where they want to go. As extensive use is made of photos on the static locations, a couple of arrows for moving through these photos are placed at the left and right corner of the screen for the standard locations, and at the top and bottom for moving in the elevator. These objects are placed on the page using mxml or by just moving them onto the page using the visual editor. Their attributes can be altered using functions in the ActionScript code which is also included in the page code. How to get started with a simple example can be found on many pages on the internet like [2]. An extensive video tutorial can be found at [3].

The objects have to take some kind of action when clicked. Take for example the right arrow. When at a static location, this arrow should load the next photo when clicked. This can be done by giving in an attribute:

```
click="function()"
```

Where function() should be the name of a function contained in the ActionScript part of the page. Variables defined in the ActionScript code may also be given as attributes to the function. So the right arrow should have as click attribute:

```
click="nextPhoto()"
```

The function itself is defined in the ActionScript code to find out what the link to the next photo is and load this. The loading is done by changing the source of the image from the framework. Calling an object on a page from the ActionScript code can be done by giving this object an id attribute:

```
id="displayedPhoto"
```

Now we can load a new photo by letting nextPhoto() call:

```
displayedPhoto.source = URL;
```

Where URL should contain the link to the photo which is to be loaded.

At first, when only the entrance was being programmed into the video, all functions and links to photos and videos were contained into one file named Rondleiding.mxml. Rondleiding is Dutch for tour, this is the file that will compile into the Flash program. ActionScript however is an object oriented language, so use can be made of classes just like with Java for example. Classes should more or less comply with the actual 'objects' you encounter during the tour, or extra machinery you need in order to make the tour reality. The classes used for this project are contained in the next table, which also has a short description of all:

Location	The most obvious class, a Location contains photos and the functions for determining which one is 'next'
Photo	A photo contains the link (URL) to the photo, as well as information about any clickable areas on the photo
Rectangle	This is a kind of silly name for a class containing the dimensions of a clickable area, as well as the name of the location which should be travelled to when clicked
Connection	A connection contains a playlist of videos which need to be played when travelling from one location to another
PlayList	This is just a list of URLs to videos and functions for adding videos and calling the next one
VideoFinder	This is a list containing connections which returns the right playlist when asked what videos need to be displayed when travelling from location A to location B
Initializer	This 'class' contains all the links to all the photos and videos, and initializes the connections, the video finder, the locations, the photos, the rectangles, everything

Working with objects makes sure you don't loose sight of where all the information is coming from. Especially when working with so many (mostly unreadable) links to photos as with a project like this, having a class that initializes objects that will just return you the photos in the right order is a must.

The program mainly does two things, loading locations and loading videos in order to get from one location to another. When at a location, the arrows need to be visible as well as the exit button. When watching a video, the user would like to have a skip button and the pause/play button gets displayed as well. Of course now the arrows need to be made invisible, and when we arrive at the next location the video screen needs to disappear together with all it's buttons. All functions that take care of this can be found in the ActionScript in the main class.

There are some shortcomings to the state the program is in at the moment. For some reason it is not possible to walk the same path twice, which can be really annoying to users. There are quite some ways to make sure a video stops playing like setting the autoRewind attribute to true, calling the stop() method or calling the close() method. It might also seem like calling load() will reload the source video. None of these seem to work however, so this is something that anyone who would like to do some further work on the tour would need to have a look at.

Also, the interviews are not working at the moment. There was a real time shortage near the end of the project resulting in the missing of this very important part of the video application. Classes need to be created in order to contain the interview parts. When the user enters the room of an interviewee, the introduction video should be shown immediately, followed by the showing of a menu of available videos from which the user can choose.

Conclusions

Probably the most important conclusion that can be extracted from this report is that it is not a good idea to try and work out an idea that's way over the top of the requirements that were set. That being said, do not forget that it is also important to look beyond the requirements and let your imagination and creativity loose on a project of whatever kind. You will find that you learn a lot along the way, things you might be able to do a related job in the future way more efficiently.

About the video and the program behind it, the best conclusion would be that the part that is finished works great. The interface is simple and clear, the users have enough places to go. The code has been nicely contained into classes. The interviews could have been fit in there by hard coding some links to videos into it, but this would have ruined the nice code. This leaves two things to do for any future extensions: adding more locations, which is quite easy (just enter them in the initializer) and creating the interviews.

References:

[1]: http://maps.google.com/?ie=UTF8&ll=37.806529,-122.4468&spn=0.006273,0.010042&t=h&z=17&layer=c&cbll=37.80518,-122.44731&panoid=YH3dJAhlMZIlxeMSv7YAAQ&cbp=1,357.96,,0,0.8466934523739619
Retrieved on May 15th 2008. Simply going to http://maps.google.com and finding San Francisco, then hitting 'Street View' will have about the same result.

[2]: http://blog.flexexamples.com/
Retrieved on May 16th 2008. This site contains many examples but can be a bit hard to move around at.

[3]: http://movielibrary.lynda.com/html/modListing.asp?pid=205
Retrieved on May 16th 2008.