Research and Development for Expanding Usability of a Multi-Touch Device

Fredrik Medeström
Research and development for expanding usability of a Multi-Touch device.

Fredrik Medeström
KarCuss@gmail.com
**Abstract**

As new ways of using computers are constantly being developed, other ways of interacting are made possible. One of the most desired types of interaction with computers is a touch-screen where the user can actually organize everything simply by using the fingertips. The purpose of this project is to enhance an existing framework with an application that works with a touch-screen device. The application is to be used as a tool for presentational usage.

The result is an application where the user by simple means can use a touch-screen to hold a professional presentation. This report will describe the making of this application and the experiences that I have gained from working with it.

**Sammanfattning**


Resultat är en applikation där användaren kan använda sig utav en tryckkännslig skärm och hålla en professionell presentation. Den här rapporten kommer att beskriva skapandet av denna applikation och mina erfarenheter som jag fått genom att arbeta med det.


**Preface**

This thesis project was done at Natural User Interface during a total period of ten weeks in the spring of 2008. The company has been developing software and expanding usability of Multi-touch devices.

We were four students who got an assignment to develop different applications for the multi-touch framework. This thesis project goal was to develop an application that could be used as a presentation tool and to develop the techniques needed.

I would like to thank Mikael Bauer who has been my mentor during this project and Harry Van Deer Veen (of natural UI ) and I would also like to thank my examiner Patrik Holmlund (LTU) for giving me the opportunity to work on this project.
# INDEX

Abstract................................................................................................................................. ii
Sammanfattning ........................................................................................................................ ii
Preface ...................................................................................................................................... iii

1. Introduction............................................................................................................................... 1
   1.1 Goal and Purpose .............................................................................................................. 1
   1.2 Background ...................................................................................................................... 1
      1.2.1 Natural UI Framework ............................................................................................... 1
      1.2.2 Research and prototyping for Flow ........................................................................... 2
      1.2.3 Designing the application ......................................................................................... 3

2. Technical Approach and Implementation ............................................................................... 5
   2.1 Desktop Handler: ............................................................................................................. 5
   2.2 Entity and Sorting of Entities ......................................................................................... 6
   2.3 Menu and Sound ............................................................................................................. 8
   2.4 Input ............................................................................................................................... 9
   2.5 Network .......................................................................................................................... 9

3. Side Project ........................................................................................................................... 10

4. Final Result ........................................................................................................................... 11

6. Discussion ............................................................................................................................. 12

6. References ............................................................................................................................ 13

7. Abbreviations and Terms ..................................................................................................... 14
1. Introduction

In this section the main purpose of the project will be described briefly and a description on the tools that were used in this project.

1.1 Goal and Purpose

The goal of this project was to develop a touch screen user interface application, were the user can structure and manage images, documents, text files and video. Mainly for use during presentational purposes where the user structures the files on the touch-screen device and can show a desired object to the public.

The purpose of this intends to make organizing and displaying your presentation material more intuitive, hence simplifying the process of doing a presentation, since doing a good presentation can be crucial in today’s marketplace.

1.2 Background

1.2.1 Natural UI Framework

The Natural UI framework utilizes C++ and OpenGL [4] it's a framework that can be used on various platforms. Therefore since I have been doing quite a lot of programming using C++ and OpenGL [4] I focused more on how to actually use the framework.

The framework uses a library for finger inputs called Touchlib [1] which is an input library that focuses on reading the inputs from the user on the screen. Each of them provides information on what the user did to the screen. It receives information about the fingers position and keeps track of its movement on the screen. This information can then be used to manipulate the objects, thus making them move, scale and rotate.

By using this, the workload of actually managing objects on the screen is much simplified.
1.2.2 Research and prototyping for Flow

Flow is the application name for the presentation application that was developed. At first we didn't have access to the framework, during that time we prototyped applications with the desired functionality. These prototypes were made in a way that would make them easy to use with the framework. This includes prototypes that could be used for

- Sorting
- Displaying objects over network
- Grouping objects
- “Desktop handler” for everything on the screen
- Loader for objects
- First draft of the object class.

Sorting:
The sorting of objects that were developed was made to handle different sorting criteria’s. These criteria’s are filename, file type, date changed and size of the object.

Network Displaying:
The prototyping included making an application that could receive a picture, movie or document and display it on the screen. That application would be able to display the wanted object on a projector or even at another computer. Another thing that the display server would be able to do was drawing on an object, so that for presentations if the person wants to draw a circle or mark something on an object this would be a possibility.

Grouping:
When we did develop grouping ability of objects we decided that two different types of grouping would be enough. The two types are either putting the objects into a stack or manage them in a grid which can be seen in the pictures. These can then be sorted in the desired criteria of the user.

Desktop Handler:
The Desktop handler is a top class which purpose is to handle what the different components of the application will be used when the user gives the input. For example when a user for wants to send a file to the display server or even sort the objects on screen by filename into a grid, the desktop handler know where to look.
Loader:
This is a loader for doing all loading of the objects that are supported in the application. For graphical files it uses a API named DevIL [3], which is also used in the framework. This API made it a lot easier to get pictures into the application. When we started looking into having text documents we wanted both PDF and Power point support but we found no information what so ever on how to make a Power point load in OpenGL [4] we excluded that. Therefore it only supports PDF’s for text documents.

Entity:
The entity class main priority is to contain all information about the objects in the application. This includes keeping track on position, scale, size, rotation and much more.

XML:
This is what we mainly use for adding descriptions to a object. And also some of the settings can be altered with ease because of this. For an example the network display server has the IP of the client stored in this file.

1.2.3 Designing the application
The graphical design was made very simple but yet useful Also we wanted to have leave application open for change to the graphical design if wanted. The only graphics in the application that we implemented for a start off at least, was some basic quads that we had in the back of the filename and so this is because none of us are actually educated in computer graphics.

This Screenshot shows one of the first versions of the application.
During our research and prototyping we got a perspective on how the actual application would look like. This was done by some help from Adithya Ananth who had made a flash application with some of the features that we were trying to achieve. This somewhat gave us a little vision on how we would make the interface look like.

This led us to give the application a better graphical design. With the help of Lasse Partanen we got graphical content that we could use in the application. He made some good looking glows and glass looking images which we could use for our menu, a drop of area for the display, buttons for PDF’s and frames for the description. This was a very large improvement for the graphical design of the application.

This is the final look of the application where it’s possible to see movies, images and PDF’s loaded into the application. The objects are just some random pictures to show a possible scenario. The drop off area for the network display server can be seen in the bottom right of the window. The arrow on right hand side conceals a menu, that when pressed slides out and shows different sorting types and stacking / grid possibilities. The actual implementation and a more technical approach will be described in the next chapter.
2. Technical Approach and Implementation

This will be a description on how things were done; this takes in consideration the files that I worked most on, even though we were manipulating a lot in the same files. This chapter will describe some of the components in a more technical approach and also how it benefit from each other.

2.1 Desktop Handler:

To the top layer of our application we have what we call the Desktop Handler, this is the link between the applications main program and the actual code that we developed. The desktop handler is a class that keeps track of the hierarchy of the other classes to get it organized and making it easier to manipulate the application. As for an example this class is the first to receive inputs from the user through the FingerDown(), FingerUpdate() functions which are passed through to the right class which then has its own method of dealing with these inputs.

Each of the actual parts of the code has a link to it through pointers so that each of the used components can be called for updates and drawing, therefore each of the components can be altered independently.

So the parts that goes through the Desktop Handler end up in the right component, this can be seen in this UML design, there every part shows as a member of the Desktop handler and then has its very own functions, which then can either add entities or call separate functions from other classes. For example the draw function in the desktop handler is the main draw for the application, this draw function call each component that is to be drawn by the program, were it simply calls their written draw function.
2.2 Entity and Sorting of Entities

The entity is basically our objects in the application. All of the images, videos and PDF documents are entities. Each of the entities have the basic functions for positioning, and also abilities for moving around so that the framework can interact with the FingerDown(), fingerUpdate() and fingerUp() functions. In the entity class there is also a function for making picking of objects. That function is done so that the framework knows which one is selected, and how to properly manipulate with the object.

The Entity class also has the ability to draw on entities this is done by making a input on the object that is to be drawn on. And after this the actual entity keeps a track of what the user draws and stores that information into a vector which then can be used for the network class to show what the user draws on the entity.

A UML generated image showing the main Entity class can be seen to the right.

The sorting is a class that is called through an Entity Group, this class is being on top of the sorting and grouping of the entities.

If a sort is to be done the user simply has to make an input into the function for the sort. By putting in a value between 1 through 4 the user can select if the entities are to be sorted by last date it was changed, file size, type of file and of course filename.

UML of the Entity Group class can be seen on the picture below.
The actual sorting class has two types which is first a node that has a link to the entity so it basically knows how it supposed to sort them. This is called QSNode, this function sets and gets values depending what the main sort needs.

The sorting class utilizes a sorting algorithm called quick sort, which is a fast and efficient method of sorting for what this application need. Also the algorithm is quite easy to understand and get to work in a decent manner.

This UML generated images show first the main Quick Sort class and how it utilize the QSNode class.

This is some pseudo code on how the sort works basically.

```plaintext
function quicksort(vector)
    var list less, greater
    if length(vector) ≤ 1
        return vector
    select a pivot value pivot from vector
    for each x in vector
        if x < pivot then append x to less
        if x > pivot then append x to greater
    return concatenate(quicksort(less), pivot, quicksort(greater))
```
2.3 Menu and Sound

The menu class is designed to have a menu that can be slid out from the side of the application through a arrow. In the menu the user of the touch-screen can choose what type of sorting the entities will have and also what type of grouping.

This is simply done by sliding out the menu and point on one of the buttons.

When the user selects the described sorting and grouping, the menu calls to the sorting function and grouping then it sets a value in which the entities will be sorted or grouped as.

The UML to the right shows how the Menu is built, in a simple manner.

The Audio Handler class is a thing that we did implement if we wanted to have sounds in the application. It works through the framework which utilizes FMOD [2] for sound playback. It only has some basic features for loading the sound, playing and so on. It stores the sound into a vector and then plays it when the user does the described input or what ever the sound may be linked to.

The UML image shows the Audio Handler class in its basics.
2.4 Input

This is done by a class named Input Handler, which is a class that handles most of the picking of the entities and such. It’s a quite basic class but nevertheless it’s is a necessary class for our application.

The main purpose of is to check when a user selects a image, video or PDF. Then send the data to what class that needs input for managing and everything related to manipulating objects.

2.5 Network

The network displaying is done by a class called Network Handler. It utilizes Winsock [6] which is a networking protocol for UDP, TCP/IP and such. Its main purpose is to send data to our display server that then draws the data sent to it. It has the ability to send documents, video, and images. Also the feature to draw on a image or document for presentational purposes are sent through the network using this class.

This UML shows the basics of it, since it uses other classes for separate manipulations. But as this is the main class for the network adding the others would just be a waste of space. This is though one of the most advanced things that we did implement in the application. And the one that did require most fixing before it worked as intended.
3. Side Project.

When we were developing the presentation application we got a task from another person in the NUI group. This resulted in both me and Johan to work on another application which was quite similar to the application we were developing. This application was developed with our mentor Mikael Bauer and Pawel Solyga. This was a fun task to work with and I hope that it will prove useful for its purpose.

As this report is not to focus on this application its still a thing that we did develop during our internship. Since I can’t tell more about the actual application itself the only thing I can show is screenshot of what it looks like.

This shows the application no further description will be added about it.

But the thing that I can say about our work progress on the application is that it was time consuming and in a short period of time we actually managed to make an application that we were satisfied with.
4. Final Result

So in the end we managed to make a presentation tool for use with the Natural User Interface [5] framework. So if we are to follow the project plan that we did create in the beginning of this project we did follow it from top to bottom. But as time progressed we did some changes along the road to improve the actual application.

The application itself can load various types of image files like, jpeg, png, bmp, tif, tga and so on. It also has the ability to load videos which can be very useful to have in a presentation. And as for documents it can load PDF, this is because most documents can be converted to PDF. Further down some pictures from the actual application can be seen and a description to explain what is done.

In this screenshot the main purpose is to show the way we can display documents, the one that is in the front is a PDF and it also has some drawing to it as drawing and marking might be a necessary tool.

These two screenshots shows the ability to stack and also how it looks when every entity is in the both of them is sorted. This is done by the menu that is to the right in the picture which displays selected sorting type and selected grouping type. The sorting is done by the user selecting the entities that he / she wants to by drawing a shape that is closed in around the entities.

And as a final screenshot of the application we have the a details tab open just to show where its placed and how it the xml description that a user can write is shown. Also in this picture it’s possible to see video, which is the QuickTime movie. Also to the right bottom the drop off zone for the display server is shown. When dragging an entity to this zone it will be sent over to the display server and finally shown and can be drawn on as well.
6. Discussion

I'm satisfied with how the work progress did proceed on how we estimated how long everything would take to implement.

Even though we did not follow the time plan accordingly because of the extra project we worked on with Pawel. So we had to make a few adjustments in the time plan to compensate for that project. The result of this is that we did develop two applications during this internship, I hope that these applications will be useful for a person who uses these touch-screens. One thing that I did learn during this project is that doing researching before and do small prototypes can be really helpful but also very painful to use. But as we did most of the prototypes before we actually had access to the Natural User Interface [5] framework, we could still use almost everything that we did with them. A few of them had to be altered to work better with the framework but I guess that is hard to know in the beginning.

Further work on this application is in my opinion to make a more decent loader which runs faster and is more memory efficient, because the application takes some time to start up. The second thing that might need some improvement is the network displaying, this works well for its purpose, but the thing that might improve it even more is to use an open source library for networking. This might be useful if the users are developing on another OS then windows, this could be done by not using Winsock [6] therefore the use of OSC is a much better choice.

But overall I'm satisfied with the end result and how the application turned out to be.
6. References

Http://www.whitenoiseaudio.com/touchlib/

Http://www.fmod.com

http://openil.sourceforge.net

http://www.opengl.com

[5] Natural User Interface
http://www.naturalui.com

7. Abbreviations and Terms

API – Application programming Interface.
DevIL – An API for loading textures
PDF – A format commonly used.
OpenGL – An API for graphics