COMP8780 Initial Project Plan: HCI Design and Implementation of a Data Centric Touchscreen Interface

Bodhi Philpot (u4004519)

August 2008

1 Background

There has been much research into methods of interaction with computers. Many different methods of interaction have been investigated, such as desktop metaphors and interaction via keyboard and mouse; speech-recognition techniques; and face-tracking, along with many other techniques. Recently, research into tangible interaction devices – such as touch-sensitive screens capable of detecting multiple touch inputs – has increased, in part encouraged by Jeff Han's demonstration[1] of a low-cost large-scale multi-touch capable input device. However, to a lay person much of the research appears to be focused on the device itself, rather than *how* this device can be used to improve interaction with computers. Demonstrations often consist of 'light-table' equivalents, where collections of photographs are moved and resized in a goal-less manner, or 'smoke painting' where users are seemingly able to draw with their fingertips directly on the screen. These systems are impressive, however my interest lies in the applied use of multi-touch devices on real-world problems. But given the time constraints on this project, the goals are limited to adapting simple existing programs to use a multi-touch capable device.

2 Task Description

The task for this project is to construct a multi-touch interface and to modify existing software to make effective use of a multi-touch capable device for input.

2.1 Hardware Construction

The hardware will be constructed using inexpensive available components. The basic design is as outlined in Jeff Han's paper[2]. Due to time and resource constraints, the illumination method will not use the FTIR technique outlined in Han's paper, but will use a simpler *Diffused Illumination* technique as described in [3]. To provide co-location of input and display, an LCD screen will be adapted for use as described in [4]. The bottom of the LCD screen will be flooded by an infra-red light source, which will illuminate a user's hands when they contact or hover over the screen. This scene will be captured by a digital video camera located underneath the screen. The captured video will be processed by one of the libraries below to produce appropriate information about the user's gesturing so that the user can interact with the software.

2.2 Library Evaluation

There are several free libraries available to make use of the hardware device. Due to cost and time concerns, only free libraries will not be evaluated.

`libavg`[5] a "high-level multimedia platform with a focus on interactive installations" has shown promising results in initial testing.

reacTIVision[6] a software library initially written to support the reactable[7], has implemented touch tracking required to support a multi-touch device as of version 1.4.

Touchlib[8] "a library for creating multi-touch interaction surfaces" is popular amongst the hobbyist and DIY community, however as it does not run correctly under Mac OS X, I have been unable to test it.

As some of the libraries are platform dependent, there will have to be an evaluation of which operating system platform to target for the project. There may be other libraries available that are suitable for use, so some time will be spent on research and evaluation.

2.3 Software Integration

It is planned to convert one or two programs from their current methods of input to use a the constructed touch screen. Given the time constraints on the project, it is planned to only adapt the second program if there is sufficient time towards the end of the project.

The first program is a tool that allows users to direct AI agents to move objects to goal positions in a virtual world.[9] This application is currently driven via a gaze-tracking interface, where the user stares first at the object to be moved until it is registered by the system, and then stares at the desired final location of the object. Once this is performed, the AI agents create and execute a plan to move the object appropriately. Converting this application to use a multi-touch interface will involve replacing the gaze input method with a system to allow the user to move the object using appropriate touch and drag gesturing on the touch device.

The second program is one that allows for the creation and modifying of *Mondrian*-like images.[10] The current software allows the user to create and edit these images through the use of a mouse. The mouse interface will be adapted to allow the use of the touch device to create and edit images.

2.4 User Testing

If possible, it is desired to perform some form of user testing to evaluate how the differing interaction method affects the use of the software. However this will not be a comprehensive usability study as it is expected that the design and implementation will use up most of the time available for this project.

3 Initial Schedule

Task	**Time**
Component Acquisition & Hardware Construction	Weeks 4-5
Library Evaluation	Week 6
Software Integration & Modification	Week 7-10
Possible User Testing	*Week 10*
Preparation of Report & Presentation	Weeks 11-12

References

[1] *Jeff Han: Unveiling the genius of multi-touch interface design* <http://www.ted.com/index.php/talks/jeff_han_demos_his_breakthrough_touchscreen.html> Last accessed 15th August 2008.

[2] Han, J. Y. 2005. Low-Cost Multi-Touch Sensing Through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology* (Seattle, WA, USA, October 23 - 26, 2005). UIST '05. ACM, New York, NY, 115-118

[3] Diffused Illumination <http://nuigroup.com/wiki/Diffused_Illumination_Plans/>

[4] Motamedi, N. 2008. HD touch: Multi-Touch and Object Sensing on a High Definition LCD TV. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 3069-3074.

[5] `libavg <http://www.libavg.de/>` Last accessed 15th August 2008.

[6] ReacTIVision `<http://reactable.iua.upf.edu/?software>` Last accessed 15th August 2008.

[7] reactable `<http://reactable.iua.upf.edu/>` Last accessed 15th August 2008.

[8] Touchlib `<http://www.nuigroup.com/touchlib/>` Last accessed 15th August 2008.

[9] Gedeon, T.D., Zhu, D. and Mendis, B.S.U., "Eye Gaze Assistance for a Game-Like Interactive Task," International Journal of Computer Games Technology, vol. 2008, Article ID 623725, 10 pages, 2008. doi:10.1155/2008/623725.

[10] MondrianDrawer: software written by Kerryn Boorman at ANU as a project supervised by Professor Tom Gedeon.