

Endbericht zum Berufspraktikum

WALTER JENNER

BAKKALAUREATSARBEIT

Nr. 238-003-045-B

eingereicht am
Fachhochschul-Bakkalaureatsstudiengang

MEDIEN-TECHNIK UND -DESIGN

in Hagenberg

im Juni 2006

Praktikumsstelle:

FH JOANNEUM GmbH
Transferzentrum, Informationsdesign
8020 Graz, Alte Poststraße 149

++43/(0)316/5453-0
informationsdesign.fh-joanneum.at

Kontaktperson:

Dipl. Ing. Jochen MARTIN
Technischer Leiter des Transferzentrums

Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Hagenberg, am 14. Juni 2006

Walter Jenner

Inhaltsverzeichnis

Erklärung	iii
Kurzfassung	v
1 Das Unternehmen	1
2 Projekte und Tätigkeiten während des Praktikums	2
2.1 GMouse	2
2.2 PD Externals	3
3 Projektbeispiele	4
3.1 GMouse	4
3.1.1 Maus Filter Treiber	4
3.1.2 Implementierung des ersten Prototypes	5
3.1.3 Absolutes Zeigen	6
3.1.4 Logo und Icon Design	7
3.1.5 Sonstiges	7
3.1.6 Probleme	8
3.2 PD Externals	9
3.2.1 Anwendungen	9
4 Erfahrungen und Zusammenfassung	11

Kurzfassung

Ich habe mein Praktikum an der FH Joanneum in Graz, am Transferzentrum des Studiengangs „Informationsdesign“ absolviert. Das Transferzentrum ist eine kleine Forschungsabteilung mit 6 Mitarbeitern, darunter 3 Softwareentwickler, und führt zu einem guten Teil EU geförderte Projekte durch. Unter Anderem beschäftigen sie sich mit den Bereichen eLearning, Videobearbeitung und neuartigen Ausstellungsmöglichkeiten für Museen.

Da das Transferzentrum eine recht kleine Abteilung ist, werden oft Kooperationen mit anderen Einrichtungen eingegangen. Beliebter Partner sind die ebenfalls in Graz angesiedelten Institutionen TU-Graz und das Joanneum Research - aber auch das Ars Electronica Center in Linz arbeitet bereits mit dem Transferzentrum zusammen.

Studenten des Studiengangs werden auch immer wieder eingebunden, vor Allem im gestalterischen Bereich, wenn Webseiten, Plakate oder Logos für Projekte benötigt werden.

Ich selbst habe bereits im Sommer 2005 ein 5 wöchiges Ferialpraktikum dort absolviert, und habe mich danach auch recht schnell dazu entschieden, für mein Berufspraktikum dorthin zurückzukehren. In beiden Praktika war ich im Bereich der Software Entwicklung tätig. Ich konnte zum größten Teil selbstständig an meinen Projekten arbeiten - das erste und auch zeitintensivere hatte das Ziel mit einem magnetischen Tracker und einem Datenhandschuh einen Mausersatz u.A. für Präsentationen zu schaffen. Meine zweite Aufgabe war eine spezielle Erweiterung des schon abgeschlossenen Projektes Scalex, das mit Pocket PCs, Projektionen und Kiosksystemen Ausstellungsinhalte modern präsentieren will.

Kapitel 1

Das Unternehmen

Das Transferzentrum ist eine kleine Forschungseinrichtung des Studienganges „Informationsdesign“ an der FH Joanneum in Graz. Durch diese Institution wird man dem für Hochschulen üblichen Forschungsauftrag gerecht. Trotz der geringen Mitarbeiteranzahl werden immer wieder anspruchsvolle Projekte erfolgreich umgesetzt. Die Einbindung von Studenten wird, soweit möglich, forciert was für diese die Möglichkeit bietet in praktischen Projekten ihr Können unter Beweis zu stellen. Thematisch sind momentan Schwerpunkte im Bereich eLaerning, mit dem Projekt „Adele“, und in interaktiven Museumsanwendungen, mit dem Projekt „Scalex“, ¹ gesetzt. Neben diesen großen Projekten, an denen auch externe Partner beteiligt sind, werden auch kleinere Aufgaben umgesetzt. Ein Beispiel dafür ist ein zentrales FH Informationssystem, über das Flatscreens, die an verschiedenen Stellen der FH platziert sind, mit Inhalten gespeist werden können.

¹<http://www.scalex.info>

Kapitel 2

Projekte und Tätigkeiten während des Praktikums

2.1 GMouse

Die Idee dieses Projektes, das von mir den Titel „GMouse“¹ bekam, ist den Mauszeiger anders als durch die Computer - Maus zu steuern. Es sollte die gedankliche und konzeptionelle Metapher der Maus „zurück zu den Wurzeln“ gebracht werden und der Mauszeiger durch Zeigen mit der Hand bewegt werden. Das Artefakt der Computer - Maus, das eine Abbildung des menschlichen Zeiges ist, sollte eliminiert werden, um das Zeigen direkt ohne Umweg und somit intuitiv zu ermöglichen.

In Abb. 3.1 sieht man den Aufbau der Hardware, der einen ersten Überblick über die Projektidee schaffen soll.

Von der technischen Seite sollte dieses Ziel mit einem magnetischem Tracker und einem Datenhandschuh verwirklicht werden. Der Tracker, der auf der Zeige - Hand montiert wird, liefert die Position und deren Ausrichtung im dreidimensionalen Raum. Mit dem Datenhandschuh sollten Bewegungen der Finger Maus - Klicks auslösen.

Softwareseitig sollte dieses Vorhaben mit dem Programmieren eines Maus - Filter Treibers, der sich den SDKs² der beiden Geräte bedient, umgesetzt werden.

Ich konnte an diesem Projekt selbstständig arbeiten und zeitlich gab es keine konkreten Vorgaben.

¹Glove Mouse

²Software Development Kit

GMouse Setup

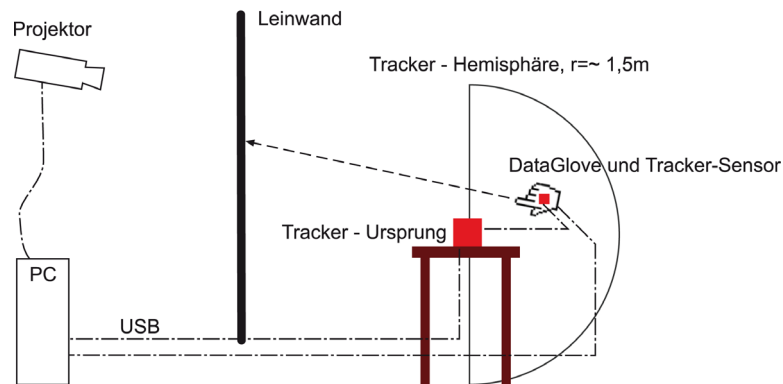


Abbildung 2.1: Darstellung des Aufbaus und der Verkabelung der Hardware

2.2 PD External

Als zweites und weniger umfangreiches Projekt habe ich zwei PD³ External programmiert. PD ist eine Open Source Software, die zur Echtzeit Signalverarbeitung dient. Dazu wird graphisch der Signalfluss programmiert. External sind in C/C++ geschriebene Erweiterungen, die dann in der PD Umgebung verwendet werden können.

Auch bei diesem Projekt arbeitete ich selbstständig und ohne Zeitdruck.

³Pure Data - www.puredata.info

Kapitel 3

Projektbeispiele

3.1 GMouse

3.1.1 Maus Filter Treiber

Bei der einführenden Projektbesprechung mit meinem Praktikumsbetreuer Jochen Martin wurde neben dem schon erwähnten Projektziel auch ein möglicher Weg dorthin besprochen. Und zwar sollte das Projekt als Maus Filter Treiber umgesetzt werden. Dieser Bereich - die Treiber Programmierung - war mir bis zu dem Zeitpunkt ein Buch mit sieben Siegeln¹. Deswegen bestand meine Arbeit in den ersten Wochen daraus, mich mit dieser Materie vertraut zu machen. Ich studierte Microsoft Driver Development Dokumente, installierte Development Kits und arbeitete mich durch Tutorials. Bald hatte ich zumindest ein wenig Ahnung und konnte schon ein Gerüst eines Maus Filter Treibers vorweisen, das sich in die Event Kette einhängte und die Maus Daten beliebig verändern konnte. Nun galt es die Pakete durch entsprechende Daten, basierend auf dem Tracker-, und Datenhandschuh - Input, zu modifizieren. Da offenbarte sich allerdings ein Problem. Die SDKs des Trackers und des Datenhandschuhs waren als DLLs im „User-Mode“ vorhanden. Treiber werden aber im „Kernel - Mode“ geschrieben, und man kann nicht aus einem Kernel Mode Programm aus, auf User - Mode Bibliotheken zugreifen. Eine Konstruktion, mit der man dieses Problem umgehen kann, ist das sogenannte „Inverted Call“ Modell, das auf der bekannten Treiber Development Webseite OSRonline² zu finden ist. Dieses Modell stellt eine Methode vor, mit der man über ein Hilfsservice - das die notwendigen Calls an die Bibliotheken tätigt - in einem Kernel-Mode Treiber mit dem User-Mode kommunizieren kann. Man braucht also neben dem Treiber noch ein Service.

An dieser Stelle wurde die Entscheidung getroffen, anstatt einem Treiber

¹Obwohl laut MTD Webseite Treiber Programmierung teil der Lehrveranstaltung AST ist.

²<http://www.osronline.com>

und einem Service lediglich eine gewöhnliche Applikation zu schreiben, die sich mit Hilfe von Windows Funktionen in die Event Kette einhängt. Damit diese Applikation möglichst im Hintergrund laufen kann, aber trotzdem immer bedient werden kann, fiel die Wahl auf eine Task-Bar Applikation. Als GUI Bibliothek wird im Transferzentrum „Qt“³ verwendet, eine sehr ausgereifete und benutzerfreundliche Bibliothek, mit der ich auch diese Applikation umsetzen sollte. Ich hatte aus 2 FH Projekten bereits Erfahrung mit C++ GUI Entwicklung, allerdings mit der Bibliothek „wxWidgets“⁴. Der Umstieg fiel mir nicht besonders schwer, und so konnte ich nach den ersten eher „theoretischen“ Wochen nun schnell mit der praktischen Arbeit beginnen.

3.1.2 Implementierung des ersten Prototypes

Grob kann man die notwendige Arbeit in zwei Teile teilen. Der eine Teil besteht daraus aus den Tracker Daten die aktuelle Position der Maus herauszufinden. Das Zweite ist das Detektieren von Maus Klicks auf Grund von Bewegungen der Finger, die über den Daten Handschuh geliefert werden. Ich entschloss mich mit dem Klicken zu beginnen.

Der verwendete Daten Handschuh⁵ liefert für jeden Finger einen Integer Wert zwischen 0 und 4096, der die aktuelle Beugung repräsentiert. In der Praxis sind die Werte ca. in einem Bereich von 800 und 2000, wobei kleine Werte wenig Beugung bedeuten.

Als erster Test wurde ein hart kodierter Schwellwert für Zeigefinger und Daumen angenommen, bei dessen Überschreitung ein Mausklick ausgelöst wird. Dabei war der Zeigefinger der primäre Mausknopf, also der linke, und der Daumen löste eine rechten Mausklick aus. Das funktionierte im Prinzip auch und somit wusste ich zumindest einmal, das ich mich am richtigen Weg befand. Das Ganze war allerdings bei Weitem noch nicht ausgereift, aber es war ein Anfang.

Kalibration des Handschuhs

Um das Klicken zu verbessern, war eine Kalibration des Handschuhs durch den Benutzer nötig. Es sollte der Wert der Finger bei neutraler Handhaltung und bei geschlossener Faust gemessen werden um den Dynamikbereich der Finger festzustellen. Weiters wurde die Möglichkeit implementiert, den Schwellwert, bei dem ein Klick ausgelöst wird, über das User Interface einzustellen. Somit funktionierte das Klicken für den Moment ausreichend, und ich konnte mich der Mausposition widmen.

³<http://www.trolltech.com>

⁴<http://www.wxwidgets.com>

⁵5DT Ultra Glove 5 , <http://www.5dt.com>



Abbildung 3.1: Mit dem ersten Prototyp angefertigte „Hello World“ Graphik

Mausposition

Ähnlich wie mit dem Feststellen der Klicks, näherte ich mich der gesuchten Lösung auch hier schrittweise. Der Tracker liefert die aktuelle Position und Orientierung des Sensors. Die Position ist in einem 3D Vektor verpackt und die Orientierung in wahlweise in einem Quaternion oder in einer 3 mal 3 Rotationsmatrix. Anfangs ignorierte ich die Orientierung noch und versuchte einen „relativen“ Mauszeiger zu implementieren. Damit meine ich, dass ein Heben bzw. Senken des Sensors die Maus nach oben bzw. unten bewegt und eine Bewegung der Hand nach rechts oder links den Mauszeiger in die entsprechende Richtung bewegt. Die absolute Position der Hand und die Richtung, in die sie zeigt, ist hier noch bedeutungslos, allerdings konnte so schon durch Bewegung der Hand der Mauszeiger gesteuert werden. Da auch das Klicken bereits einigermaßen funktionierte war ein erster Prototyp fertig. Wie es sich für einen ordentlichen Programmierer gehört, wurde auch gleich ein „Hello World“ Experiment (siehe Abb. 3.1) durchgeführt.

3.1.3 Absolutes Zeigen

Das ist der eigentlich Gedanke an dem Projekt; der Benutzer soll mit seiner Hand auf eine Stelle am Bildschirm oder der Leinwand zeigen und der Mauszeiger soll an dieser Stelle sein.

Um einen Schnittpunkt des Zeige - Strahls und der Leinwand ausrechnen zu können, musste die Bildschirmenebene mathematisch definiert werden. Der



Abbildung 3.2: Logo der GMouse

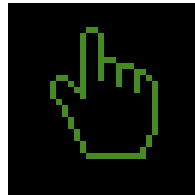


Abbildung 3.3: Icon der Applikation

einfachste Weg um das zu erreichen war, den Tracker so zu plazieren, dass die Bildelebene parallel zur Ebene, die durch die X und Y Achsen des Koordinatensystem aufgespannt wird, ist. Man braucht somit nur noch den Abstand der Bildelebene zum Tracker - Ursprung wissen und die Ebene kann mathematisch definiert werden. Bei einem Abstand von 3 m, würde die Ebene so aussehen: $X : z = -3$

Also muss der Benutzer bei der Kalibration die Entfernung des Tracker - Ursprungs zur Leinwand, bzw. zum Bildschirm wissen. Weiters muss auch noch das Bildschirmrechteck durch den Benutzer kalibriert werden. Dazu musste in die 4 Eckpunkte des Bildschirms gezeigt werden und mit diesen Vektoren wurde der Ausschnitt des Bildschirms auf der Ebene festgestellt.

Durch dieses, im Prinzip einfaches, Verfahren konnten die Tracker - Daten in Mauskoordinaten umgerechnet werden.

3.1.4 Logo und Icon Design

Nebenbei habe ich auch noch ein Logo und Icons (Abb. 3.3) entworfen.

Beim Logo (Abb. 3.2) war der Grundgedanke die altbekannte Zeiger - Hand einzubinden, da dieses Symbol der Idee der GMouse entspricht. Es verdeutlicht, dass das Zeigen mit der Maus von Anfang an ein Ersatz der Hand als Zeigeeinstrument war, der nun durch die GMouse hinfällig wird.

3.1.5 Sonstiges

Abrundend wurde noch ein Installer für die Applikation implementiert und zusätzlich eine deutschsprachige Version erstellt.

Eine alternative Einsatzmöglichkeit der GMouse wurde auch ausprobiert; und zwar wurde der Tracker - Sensor am Kopf montiert, um den

Mauszeiger mit dem Kopf steuern zu können. Diese Art die Maus zu steuern erwies sich als überraschend einfach, und stellt eine durchaus sinnvolle alternative Anwendung der GMouse dar.

3.1.6 Probleme

Klicken

Bei der Implementierung des Klickens ergaben sich mehr Probleme als erwartet. Das erste Unangenehme war, das sich beim Beugen der Finger unweigerlich auch die Hand, und somit die Mausposition, mitbewegte. Das hatte zur Folge, das der Benutzer teilweise Probleme hatte, auf die gewünschte Stelle zu klicken. Genauso wird das Doppelklicken erschwert, da ein Doppelklick nur dann vom Betriebssystem als Doppelklick erkannt wird, wenn sich die Mausposition zwischen 2 Klicks nicht ändert. Als Lösung für dieses Problem wurde ein zweiter Schwellwert eingeführt, der die Mausbewegung deaktiviert. Somit bewegte sich die Maus schon bei der Einleitung eines Klicks nicht mehr und der Benutzer traf leichter auf die richtige Stelle und konnte auch besser doppelklicken.

Es wurde auch versucht zusätzliche Funktionen über Fingerbewegungen aufzurufen. So wollte ich zum Beispiel das Scrollen über den Handschuh ermöglichen. Das scheiterte allerdings daran, das es nicht möglich war mehrere Gesten eindeutig und mit geringer Fehlerquote zu erkennen.

Mausposition

Ein weiteres Problem war ein Zittern des Mauszeigers, das durch große Längen des Zeige - Strahls hervorgerufen wurde, da dadurch schon kleine Bewegungen der Hand große Bewegungen des Mauszeigers hervorrufen. Dieses Zittern wurde durch das Mitteln von einer einstellbaren Anzahl von aufeinanderfolgenden Mauspositionen verringert.

Latenzzeiten

Da es sich bei der GMouse um eine Echtzeitanwendung handelt sind Latenzzeiten so klein wie möglich zu halten. Im Gegensatz zu Computerspielen, bremsten bei dieser Applikation nicht aufwendige Rechenaufgaben die Performance, sondern waren Verzögerungen durch die USB Verbindung ein Problem. Wie man in Abb. 2.1 sieht, befindet sich der Rechner in der Testumgebung hinter der Leinwand, und daraus ergab sich eine Entfernung vom Benutzer zum PC von über 10 m. Anfangs wurde ein USB-over-IP Gerät eingesetzt, das allerdings viel zu große Verzögerungen (>100 ms) hervorrief. Danach wurde mit aktiven USB Verlängerungen gearbeitet, wodurch die Latenzzeiten stark verringert wurden, und die ganze Anwendung ohne bemerkenswerte Verzögerung lief.

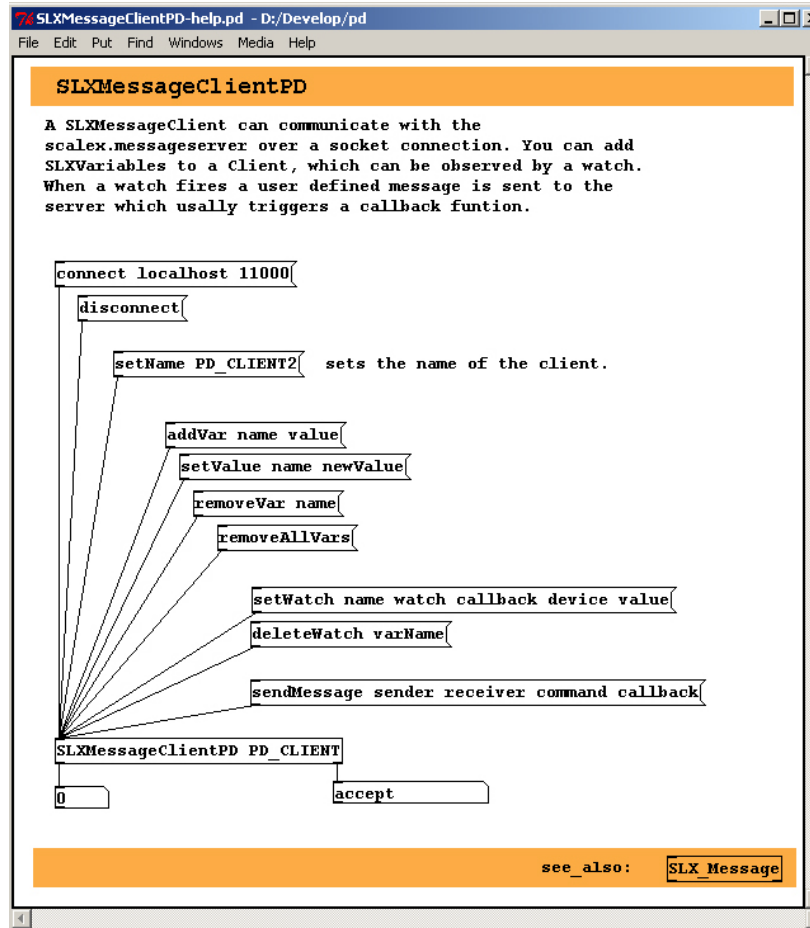


Abbildung 3.4: Eines der zwei PD Externals

3.2 PD Externals

Der Sinn dieses Projektes war es die Kommunikation zwischen der Open Source Applikation PD und der Transferzentrum Applikation „Scalex“ zu ermöglichen. Man wollte so die Funktionalität von Scalex mit minimalen Aufwand um die Möglichkeiten die PD bietet erweitern. Scalex basiert intern auf einer Server - Client Architektur, weshalb für PD nur noch ein neuer Client implementiert werden musste, der sich mit dem Scalex Server unterhalten kann.

3.2.1 Anwendungen

Ein Vorteil von PD ist, dass man sehr einfach mit MIDI Daten arbeiten kann. In Kombination mit MIDI - Geräten, an die man diverse Sensoren

anschließen kann, ist es möglich mit geringem Aufwand neuartige Interaktionsmöglichkeiten zu schaffen. Neben MIDI Sensoren können auch Kamerabilder mit PD einfach verarbeitet werden, und so können wiederum sehr einfach Kamerabasierende Interaktions - Ideen umgesetzt werden.

In Abb. 3.4 sieht man eines der zwei External, die von mir programmiert worden sind, im Einsatz.

Kapitel 4

Erfahrungen und Zusammenfassung

Das Praktikum am Transferzentrum der FH Joanneum hat mir viel Spaß gemacht und ich habe dank der recht anspruchsvollen Projekte viel gelernt. Die Projekte selbst waren sehr interessant und abwechslungsreich. Auch das Betriebsklima war sehr angenehm und der Kontakt zu Kollegen aus den verschiedenen Fachgebieten war wohl etwas das im Transferzentrum, bzw. an dem Studiengang besonders ist. So sind unter den Kollgen Telematiker, Software Entwickler, ein Psychologe, Medienkünstler, Leute aus der Werbebranche und schlussendlich auch Studenten mit denen die Zusammenarbeit größtenteils Spaß macht und auch lehrreich ist.

Natürlich ist Graz selbst auch ein schönes Plätzchen, das genug an interessanten Freizeitmöglichkeiten bietet.

Aus diesen Gründen werde ich dem Transferzentrum aller Voraussicht nach auch in Zukunft erhalten bleiben und dort ab September 2006 als wissenschaftlicher Mitarbeiter für ein Jahr arbeiten, um dann ab Oktober 2007 dort an der FH meinen Master im neuen Studiengang „Media and Interaction Design“ zu absolvieren.