

Bachelor-Thesis Proposal

von Frederic Raber

1. *Abstract*

Im Rahmen meiner Bachelor-Arbeit schreibe ich ein modulares RAD-System, um typische Eingabeparameter von Multitouch-Interfaces wie Distanz, Winkel etc. zwischen verschiedenen Punkten zu berechnen und an die nächste Ebene weiterzugeben.

2. *Aufbau*

Das besondere an dem Projekt ist der modulare Aufbau.

Es gibt grundsätzlich 2 verschiedene Modulklassen: Inputmodule (IM) und Modifikationsmodule (MM).

Während Inputmodule ein Eingabegerät repräsentieren (z.B. einen der Finger auf einem Multitouch-Display oder eine Maus), verarbeiten Modifikationsmodule die Daten eines oder mehrerer Module und geben sie in einer anderen Form aus. Beispiele hierfür sind z.B. ein Distanz- oder Winkelberechner.

Durch diesen Aufbau kann mit einem Point&Click-Interface in wenigen Minuten ein System aufgebaut werden, das Rohdaten der Eingabegeräte verarbeitet und sie in der für das entsprechende Projekt benötigten Form ausgibt.

3. *Bedienung/GUI*

Die Erstellung eines solchen Systems erfolgt grafisch über ein Data-Flow Diagramm, das mit JGraph implementiert wird.

Ähnlich eines elektrischen Schaltplans können hier Module erstellt und die entsprechenden Eingänge und Ausgänge verschiedener Module verknüpft werden.

4. *Besondere Features*

4.1 *Echtzeitberechnung*

Ein weiterer Punkt ist die Berechnung der Daten in Echtzeit.

Da es bei Eingabegeräten enorm wichtig ist, die Daten so schnell wie möglich zu erhalten, muss die Datenverarbeitung in den einzelnen Modulen parallel vorgenommen werden, sodass der Datenstrom nie abreißt oder ins stocken kommt.

4.2 *Speicherfunktion*

Weiterhin soll der Aufbau abgespeichert und auch wieder geladen werden können. Die Files sollen auch "von Hand" lesbar sein.

Hierfür bietet sich möglicherweise XML an.

5. *Realisierung*

Die Realisierung dessen erfolgt durch ein nebenläufiges Design mit Multithreading. Jedes einzelne Modul wird durch einen eigenen Thread repräsentiert, der konstant die Daten auf der Eingabeseite annimmt und die verarbeiteten Daten auf der Ausgabeseite ausgibt. Eine funktionierende und effektive Synchronisierung vorausgesetzt, wird damit das Programm auf einem System mit entsprechend vielen CPU-Kernen dann (fast) in Echtzeit arbeiten.

Als Programmiersprache wird Java eingesetzt, da es einen sehr guten Umgang mit Multithreading besitzt und cross-platform kompatibel ist. Ebenso existieren für das FTIR-Display Steuermodule von NUI für Java.

6. *Hardware*

Der zweite Teil beschäftigt sich mit dem Betrieb des experimentellen, großen FTIR-Multitouch-Displays mit beschriebenem System. Die Aufgabe besteht darin, es zum laufen zu bringen und als Eingabegerät in das Programm einzubringen.

Dazu müssen theoretisch nur verschiedene, bereits vorhandene Module von NUI miteinander verknüpft werden.

6.1 *Steuerungssoftware*

Zur Aufnahme wird tBeta benötigt, das die Kommunikation zum Display übernimmt und die existierenden Blobs ausgibt. Diese werden von TUIO (Tangible User Interface Objects) weiterverarbeitet und schließlich an einen Java client weitergegeben, bei dem dann die Daten als Objekte inklusive id und Position vorliegen. Diese können dann fast direkt in ein Inputmodul gegossen werden.

7. Roadmap

April:

- Einarbeitung in die zu verwendenden Materialien: NUI Software, Multitouch-Display, JGraph
- Verbinden der NUI-Software
- Grob- und Feinentwurf des Projekts incl. UML-Diagramm

Mai/Juni:

- Implementierung der Module
- Implementierung der GUI
- Debugging

Juli, evtl. August:

- Zusammenfassung der gewonnenen Erkenntnisse und Niederschrift

Die erste Phase kann evtl. auch 2 anstatt einen Monat dauern, die dritte auch einen statt zwei.