

EDV & Multimedia Interaktionsdesign

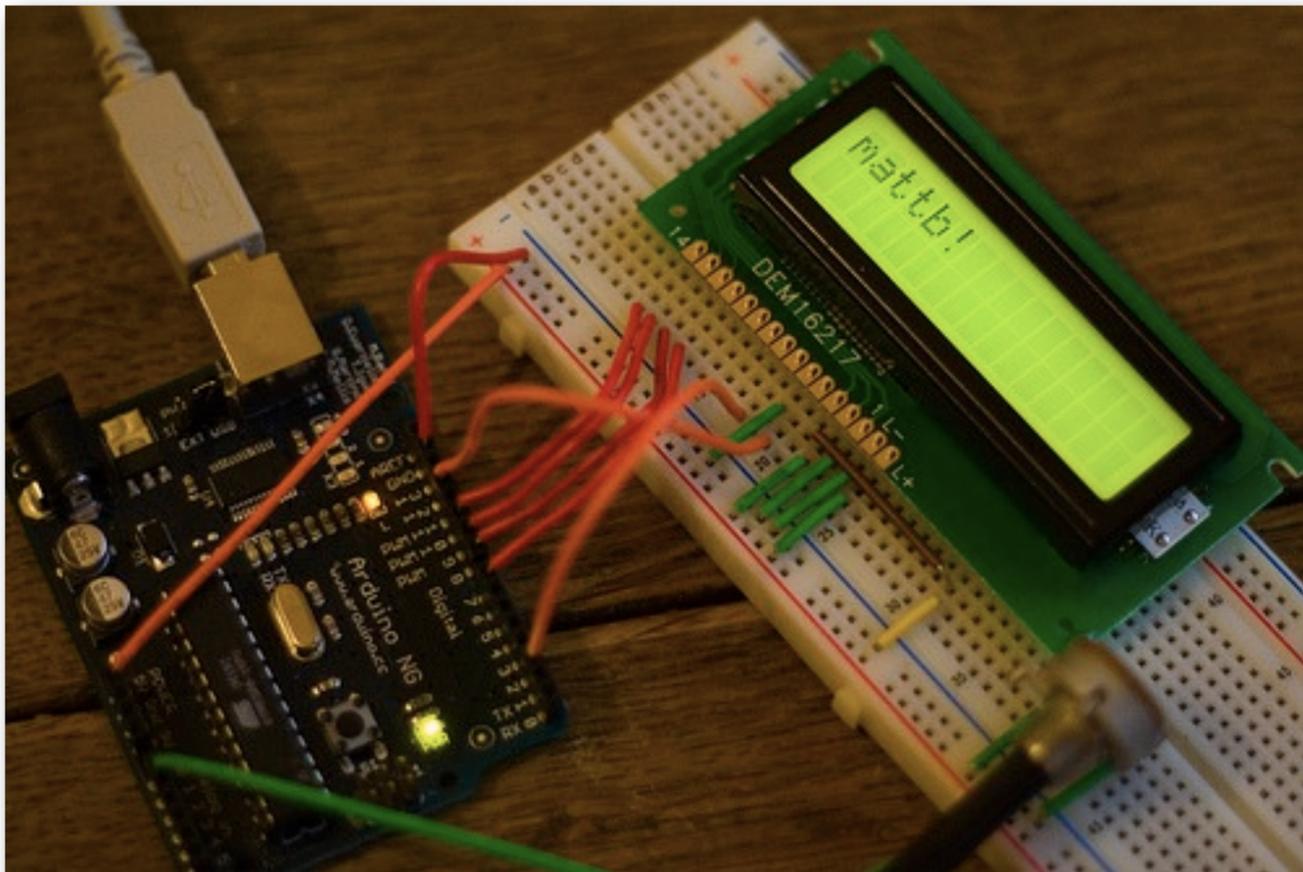
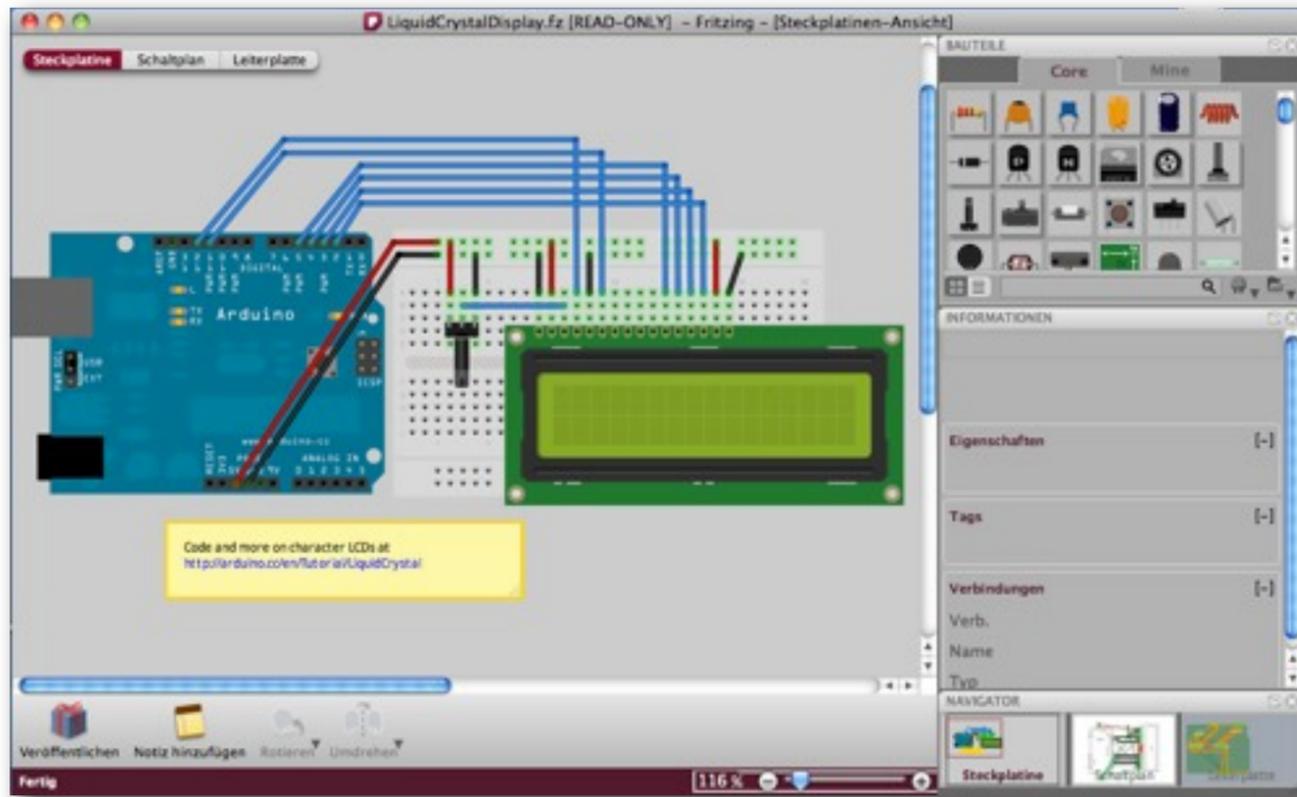
13 – Prototypen

Prof. Dr. Jochen Koubek

30. Januar 2012



Prototyp



Ein **Prototyp** soll dem Nutzer einen Eindruck des fertigen Systems ermöglichen.

Ziele im Prototyping



Ein Lo-Fi-Performance Papier Prototyp

Explorativ

Anforderungen ermitteln

Experimentell

Grundfunktionalität erkundbar machen

Performance

Einsetzbarkeit in der anvisierten Umgebung testen

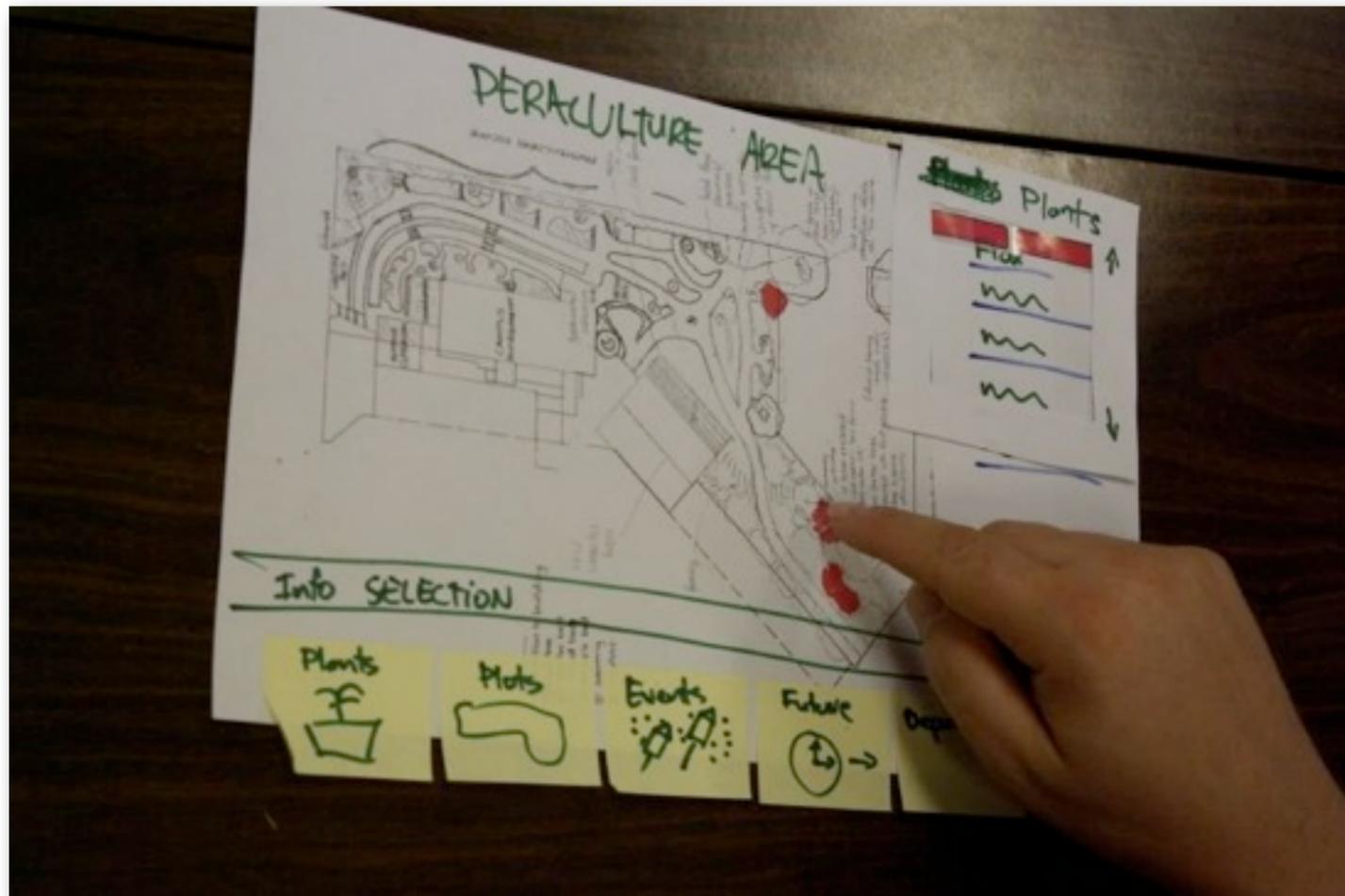
Evolutionär

Frühe Versionen erstellen

Didaktisch

Systemverständnis unterstützen

Merkmale von Prototypen



Ein breiter aber flacher Prototyp

Breite (Horizontal)

Eine Ebene (häufig die Oberfläche) wird umgesetzt.
Welcher Anteil der Produktfunktionalität wird repräsentiert?

Tiefe (Vertikal)

Ein Teilaspekt wird durch alle Ebenen umgesetzt
Wie detailliert wird die Funktionalität implementiert?

Aussehen

Die sinnliche Anmutung (häufig Aussehen) wird umgesetzt
Welche Ähnlichkeit gibt es mit dem fertigen Produkt?

Interaktion

Das Verhalten wird umgesetzt
Wie genau wird die Transformation von Eingabe zu Ausgabe simuliert?

Arten von Prototypen

Lo-Fi

Skizze

Storyboards

Wireframes

Papier

Mi-Fi

Mock-Ups

Wizard-of-Oz

Experience

Prototyp

Protoyp-Theater

Hi-Fi

Computereinsatz

Rasche

Implementierung

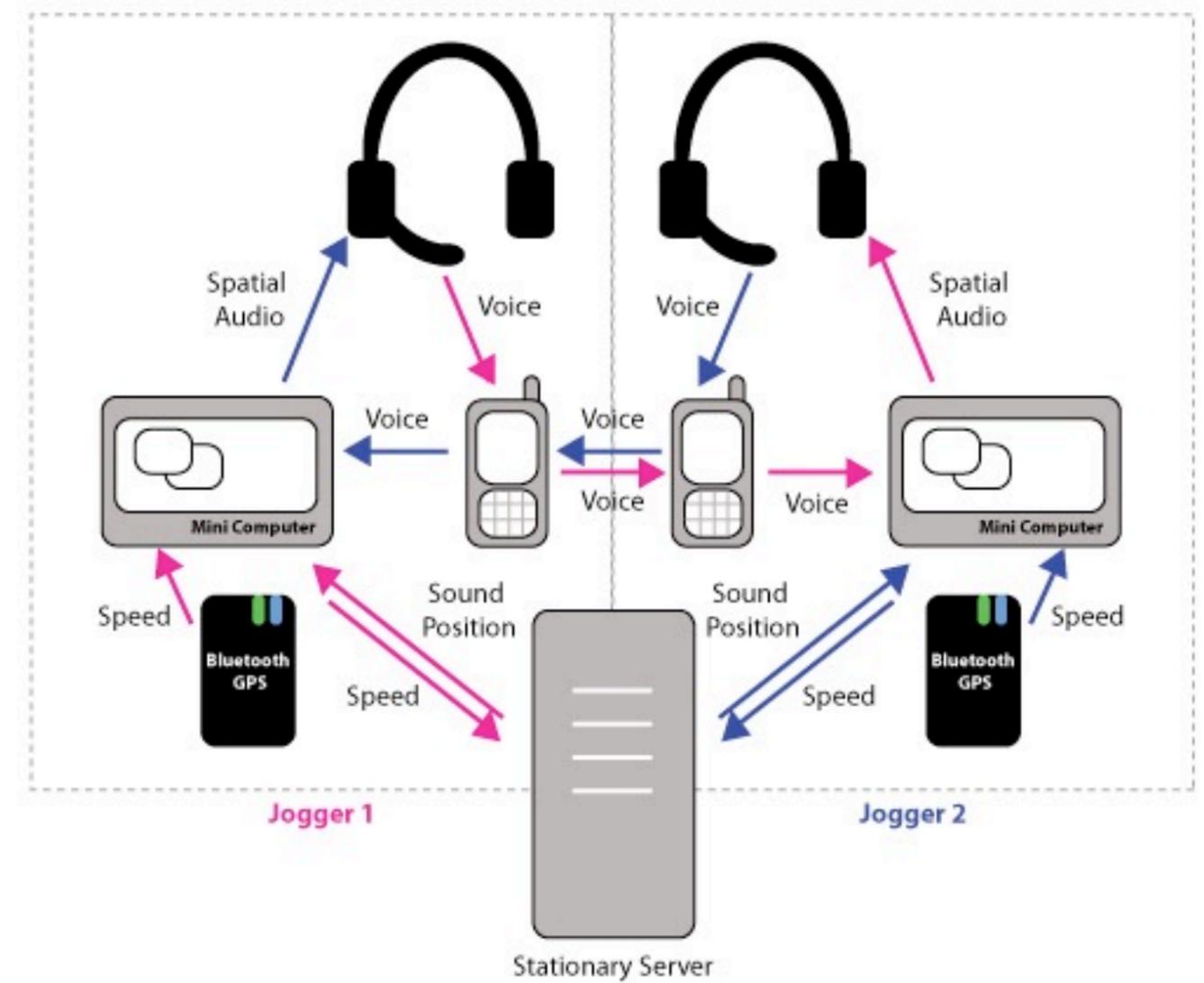
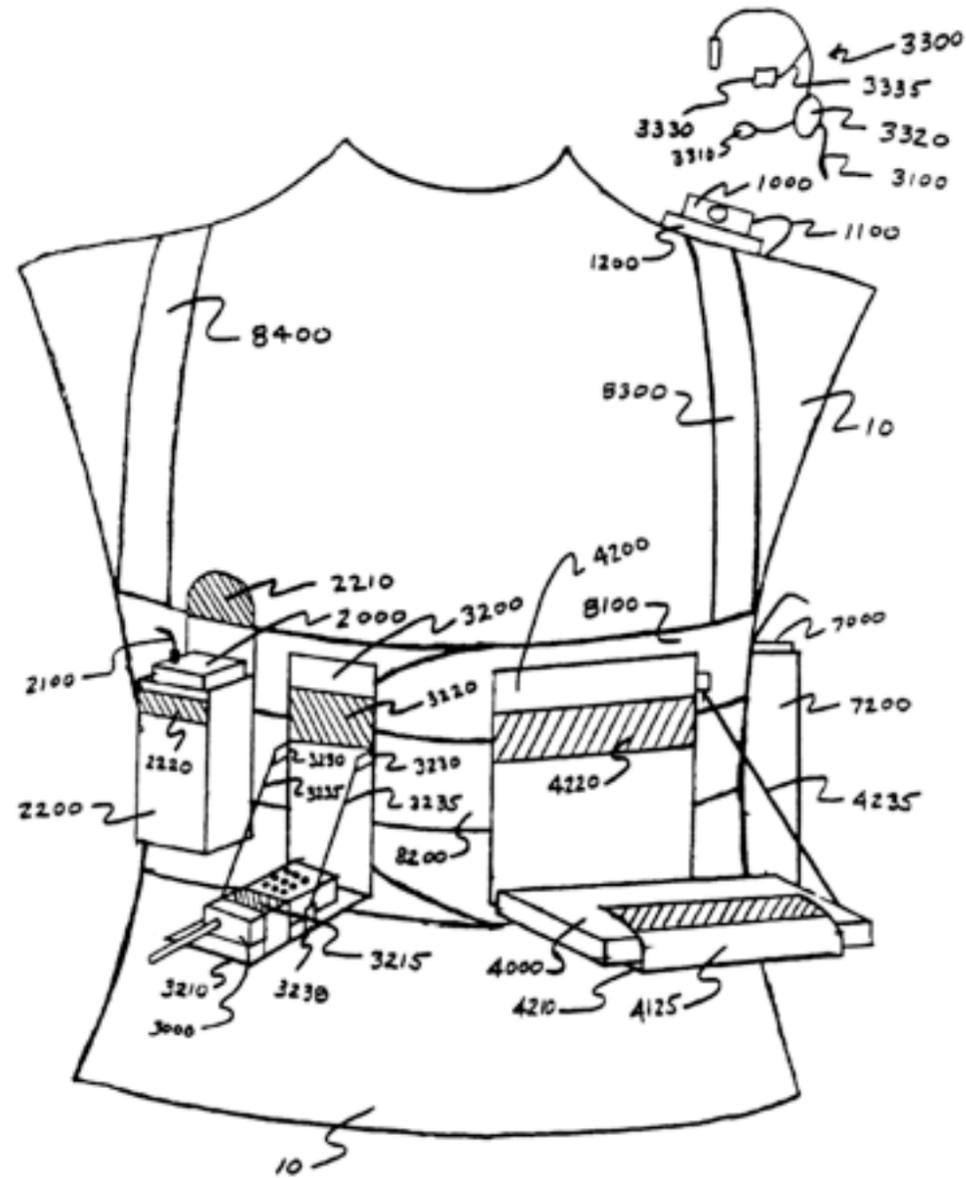
Lo-Fi-Prototypen

Ein Lo-Fi-Prototyp erfüllt sein Ziel ohne digitaltechnische Unterstützung.



Für Ideen, Abläufe, Kontexte

Skizze

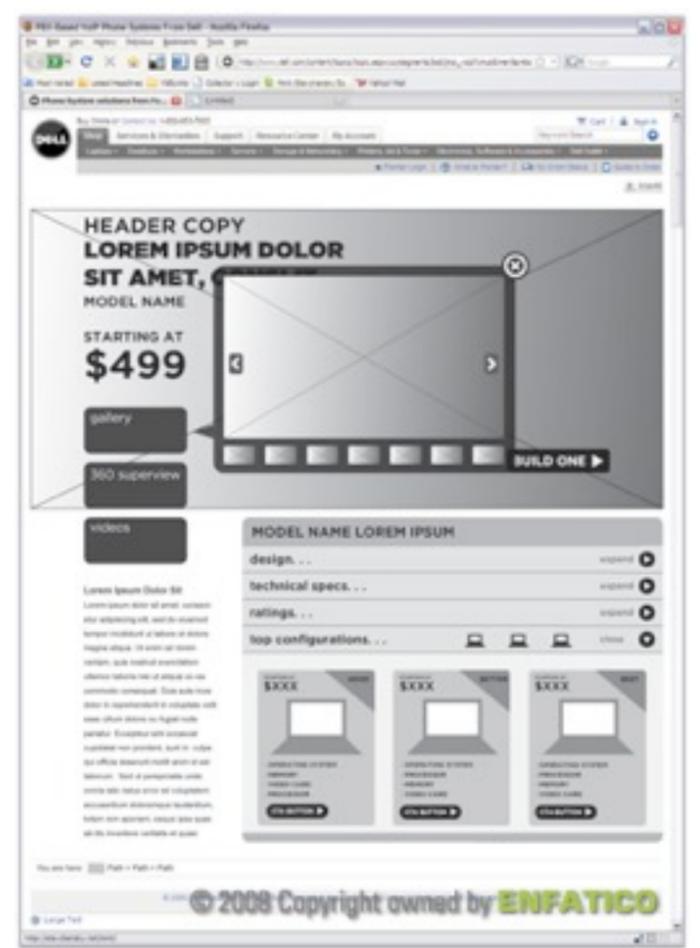


Wireframes

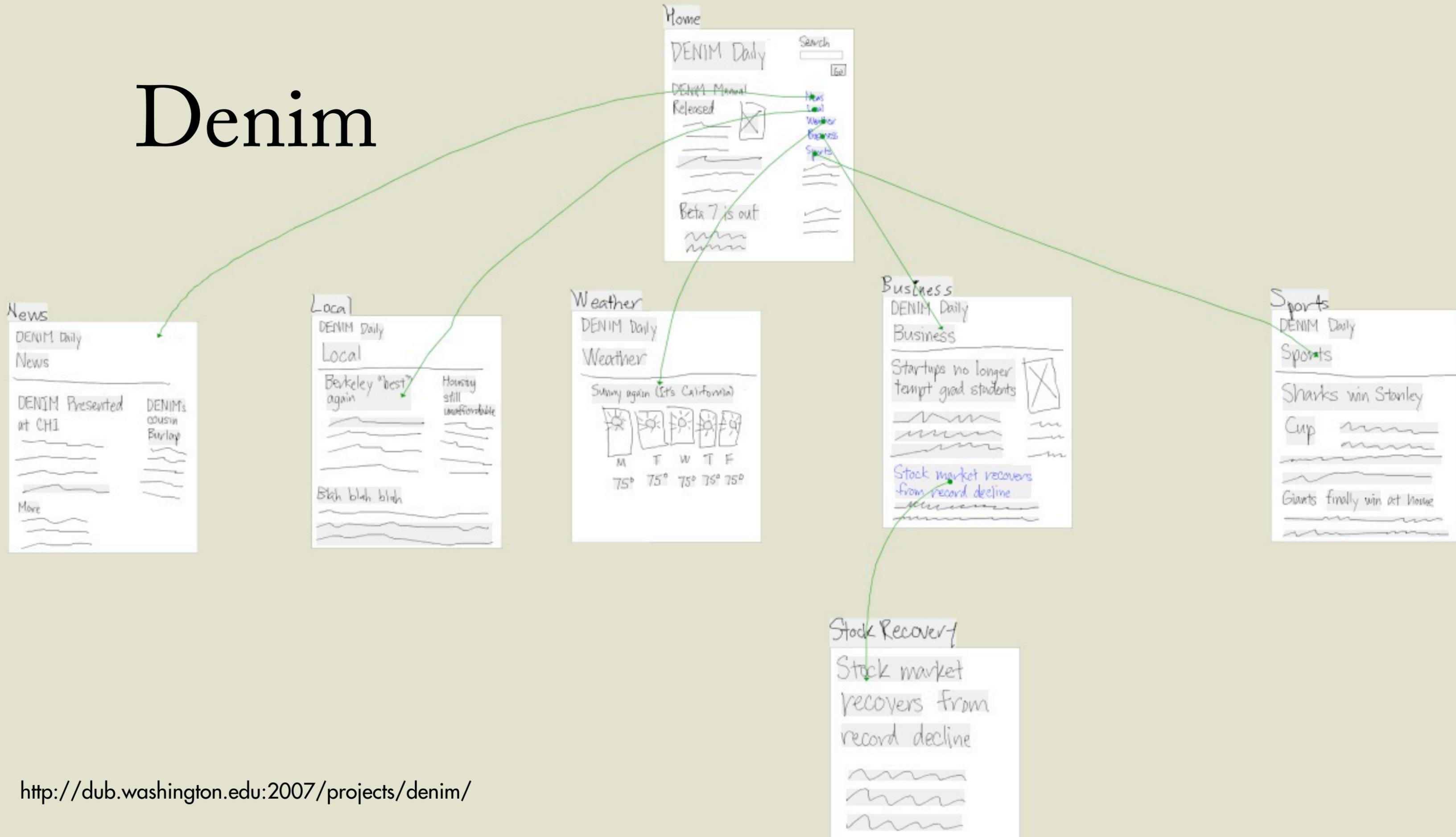


© 2008 Copyright owned by **ENFATICO**

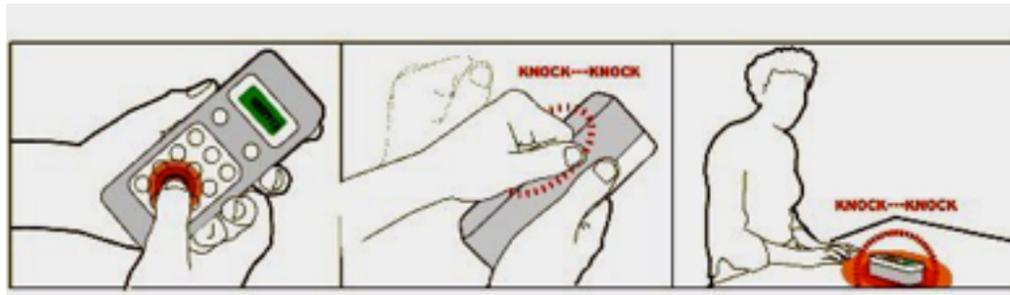
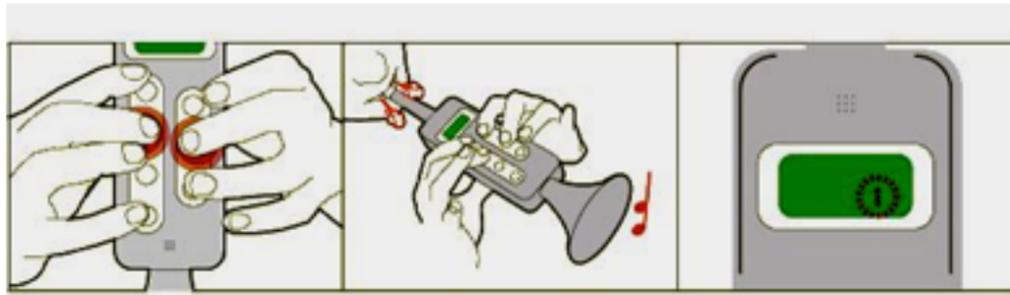
<http://stavchansky.net/work.php?wID=49&cat=3>



Denim



Storyboards

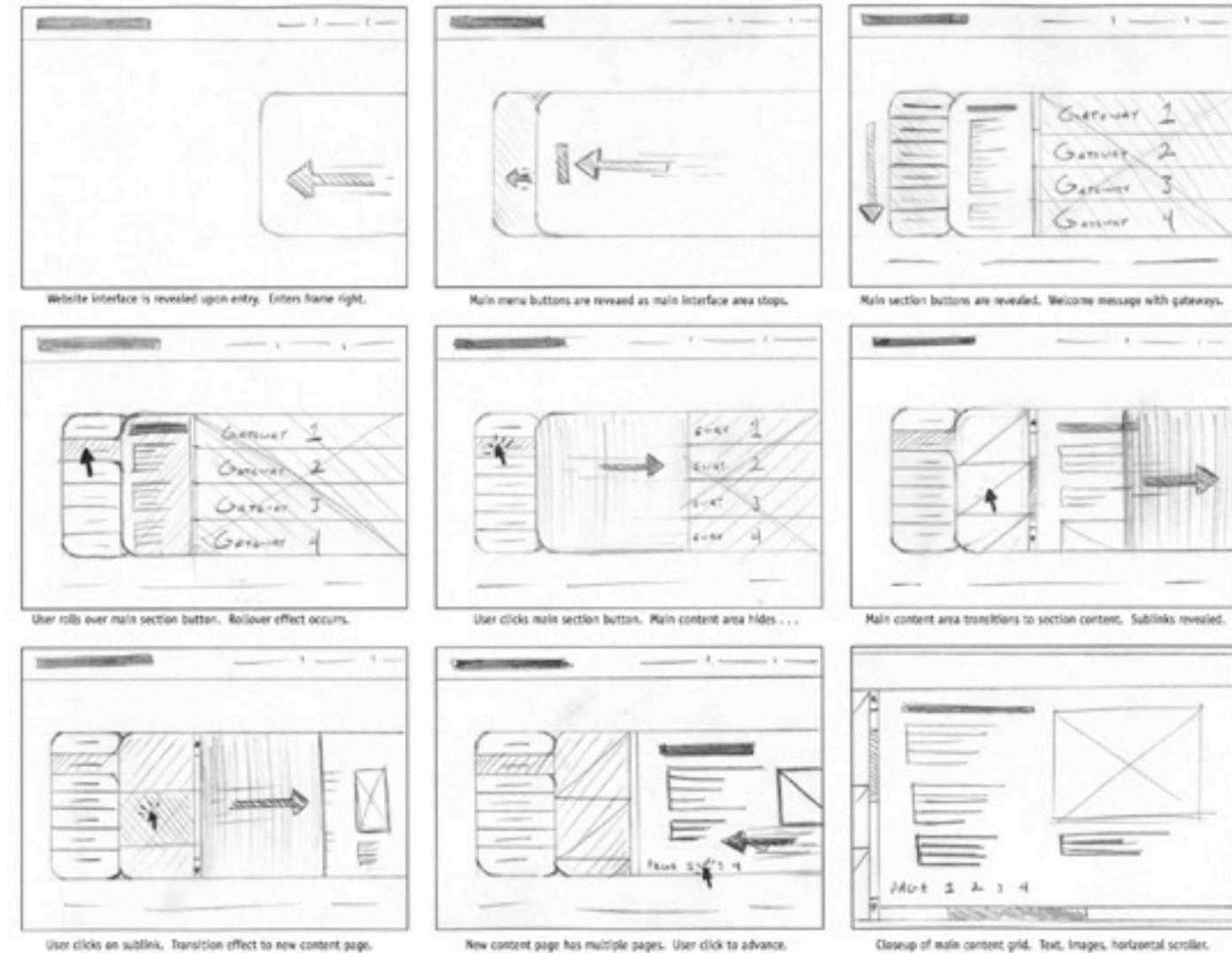


Number 10 Storyboards v1.0 July 6, 2009

Number 10 Storyboards v1.0 July 6, 2009



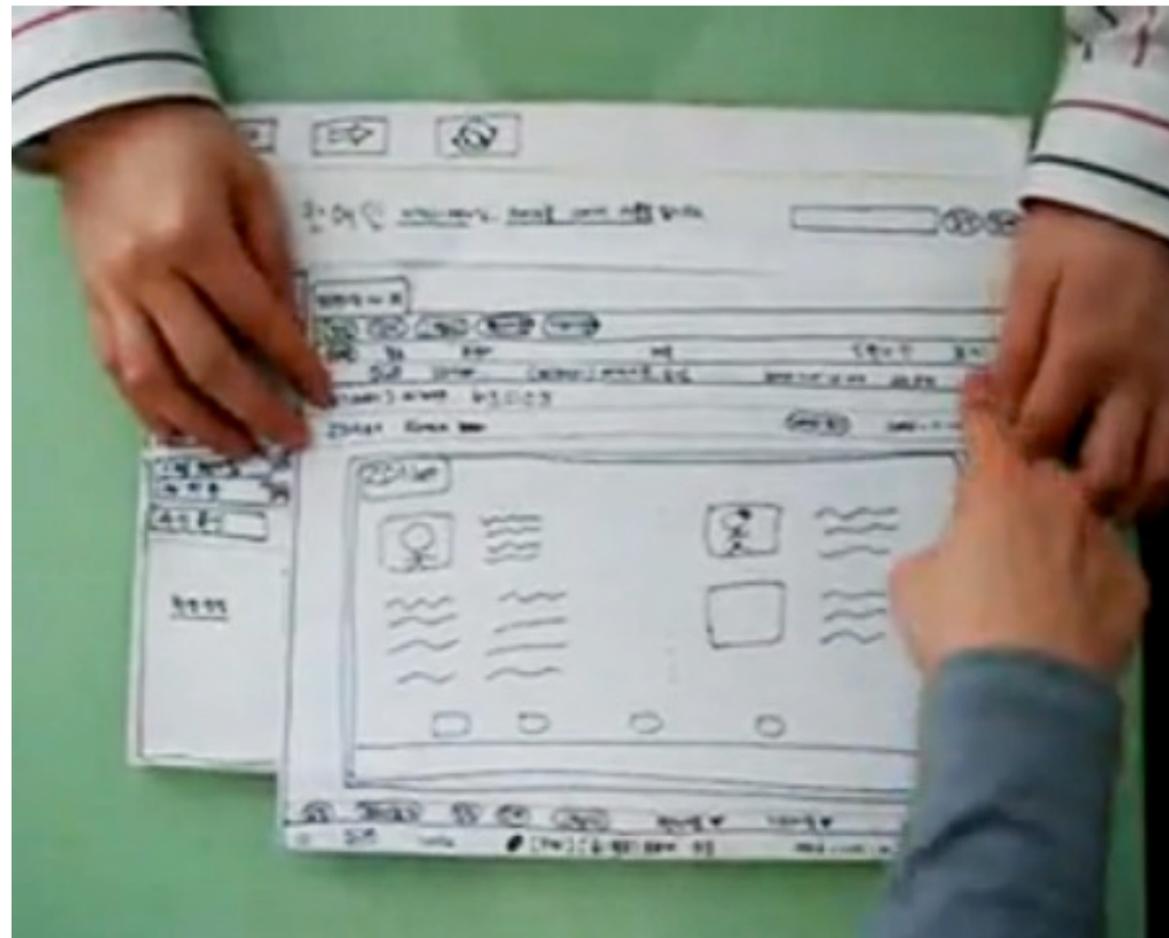
Number 10 Storyboards v1.0 July 6, 2009



<http://stavchansky.net/work.php?wID=42&cat=3>

Storyboards basieren auf den Szenarios der Anforderungsanalyse

Papier-Prototypen



Es muss nicht immer digital sein: Ob Spiele oder Grafische Benutzeroberflächen – Tests mit Papierprototypen sind zielführend, einfach in der Herstellung und machen Spaß.

Paper-Prototyping mit Karteikarten



Ausstattung

Weißes Papier, unliniert, 11-17", stabil
5-8"-Karten
Klebebänder, Post-It-Kleber
Farbige Stifte, Marker
Sticky Notes
Overhead-Folien
Scheren, Cutter, Lineale, Pflaster

Zeitgrenze setzen

Modelle bauen, keine Illustrationen

Test vorbereiten

1. Nutzer auswählen
2. Testszenarien vorbereiten
3. Üben

Test auswerten

Vor und Nachteile von Lo-Fi

Vorteile

Schnell, billig erstellt

Schnell veränderbar

Kann Spaß machen

Nachteile

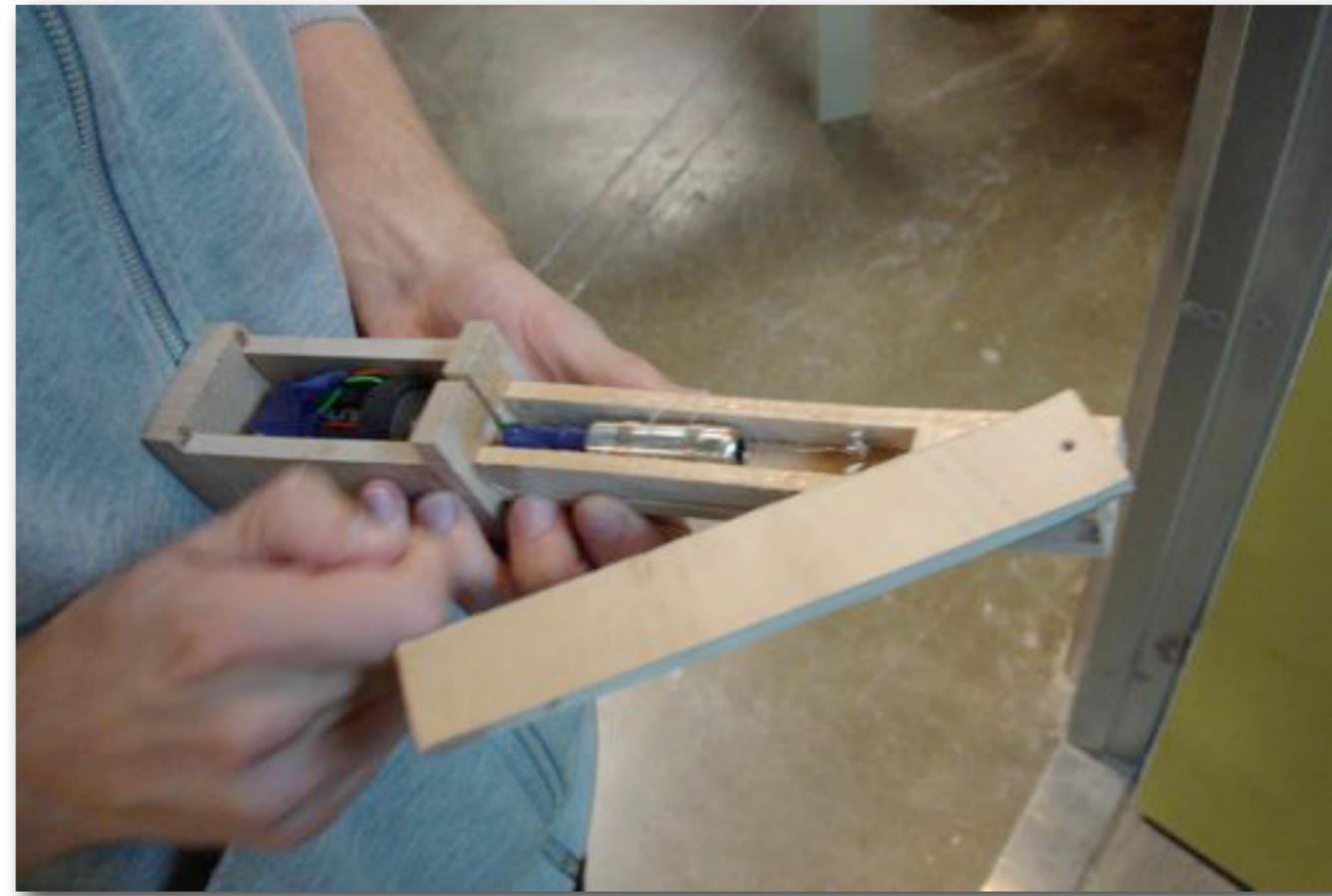
Abstrakt für Designer und Nutzer

Sehr abhängig vom Moderator

Evtl. technisch nicht umsetzbar

Mi-Fi-Prototypen

Breite, Tiefe, Interaktion und Visualität sind eingeschränkt

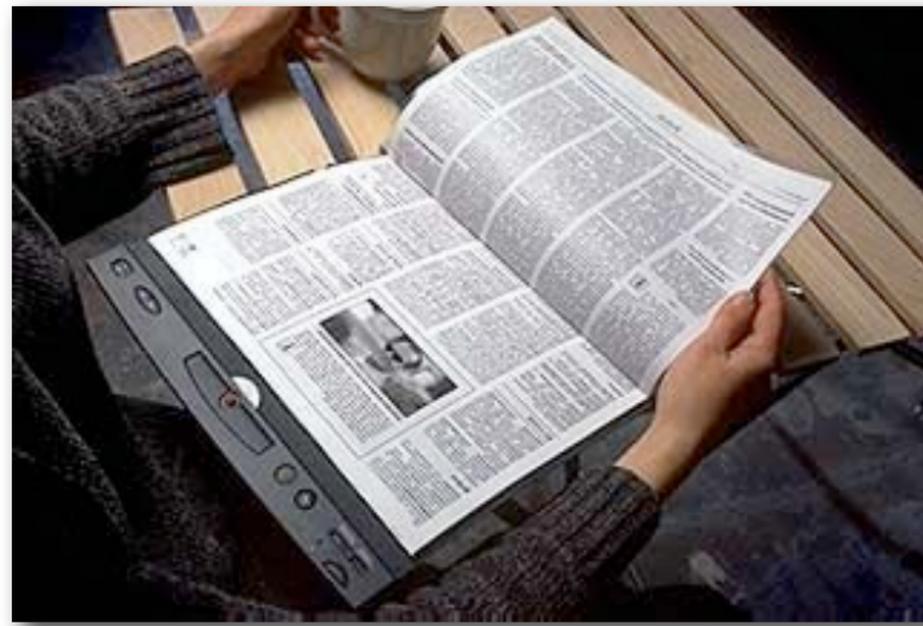


Designstudie

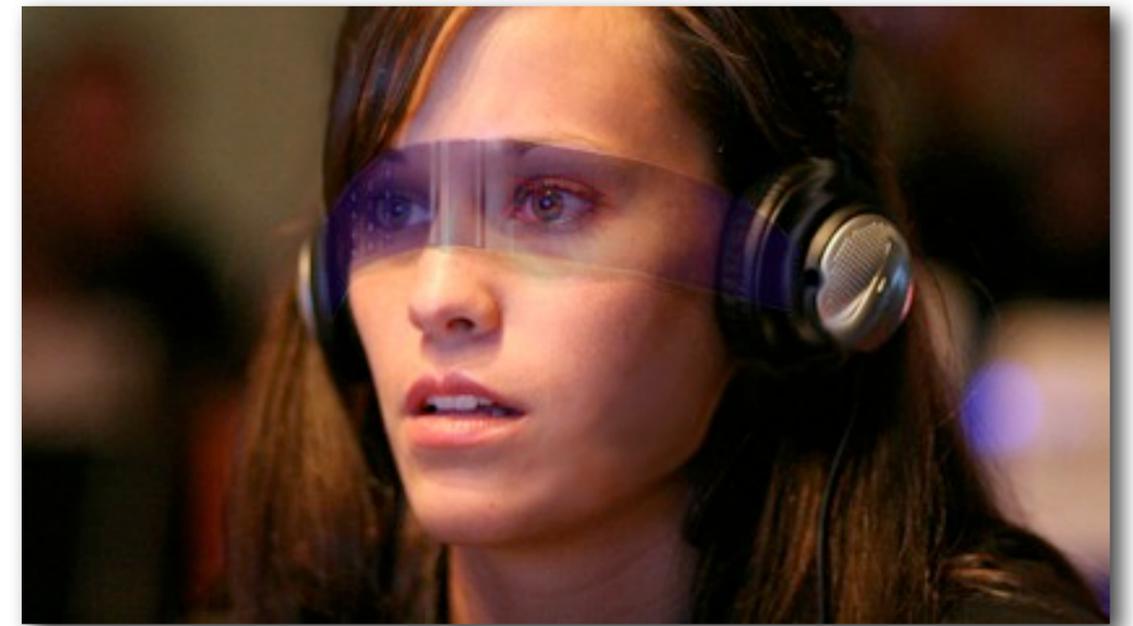
Eine Designstudie zeigt, wie das Produkt aussehen könnte, ist aber nur eine aufwändige Skizze



100 \$ Laptop

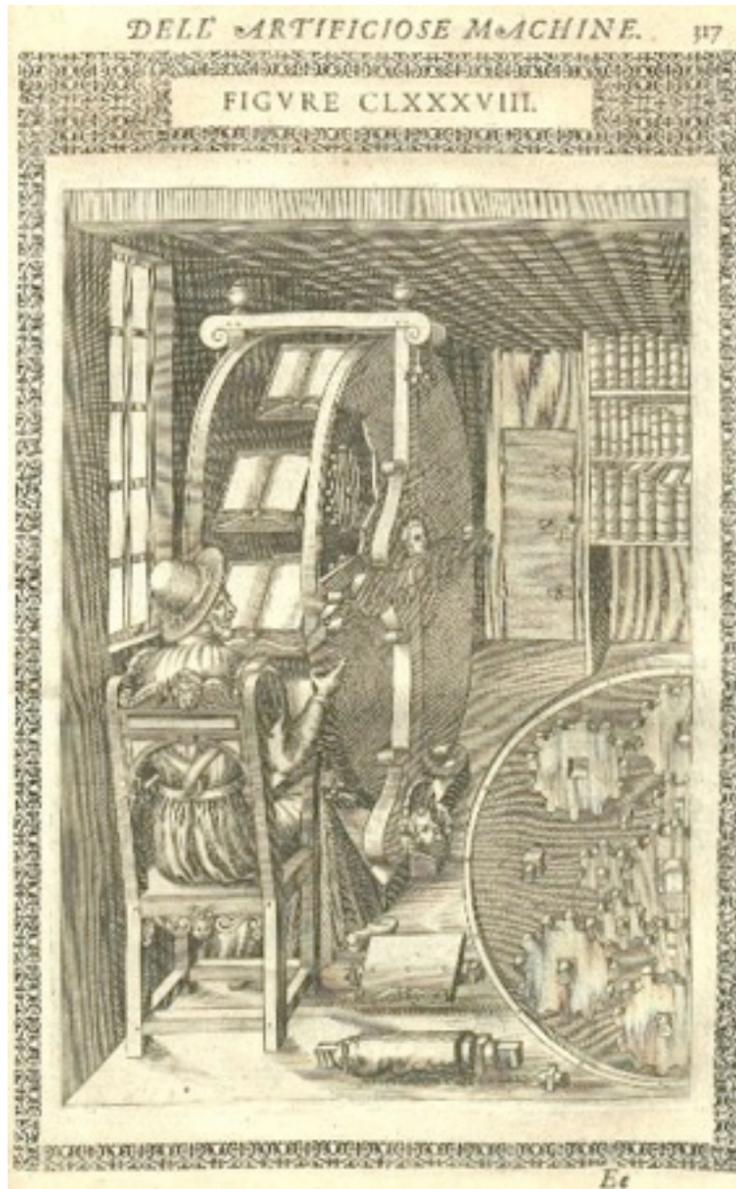


e-Newspaper

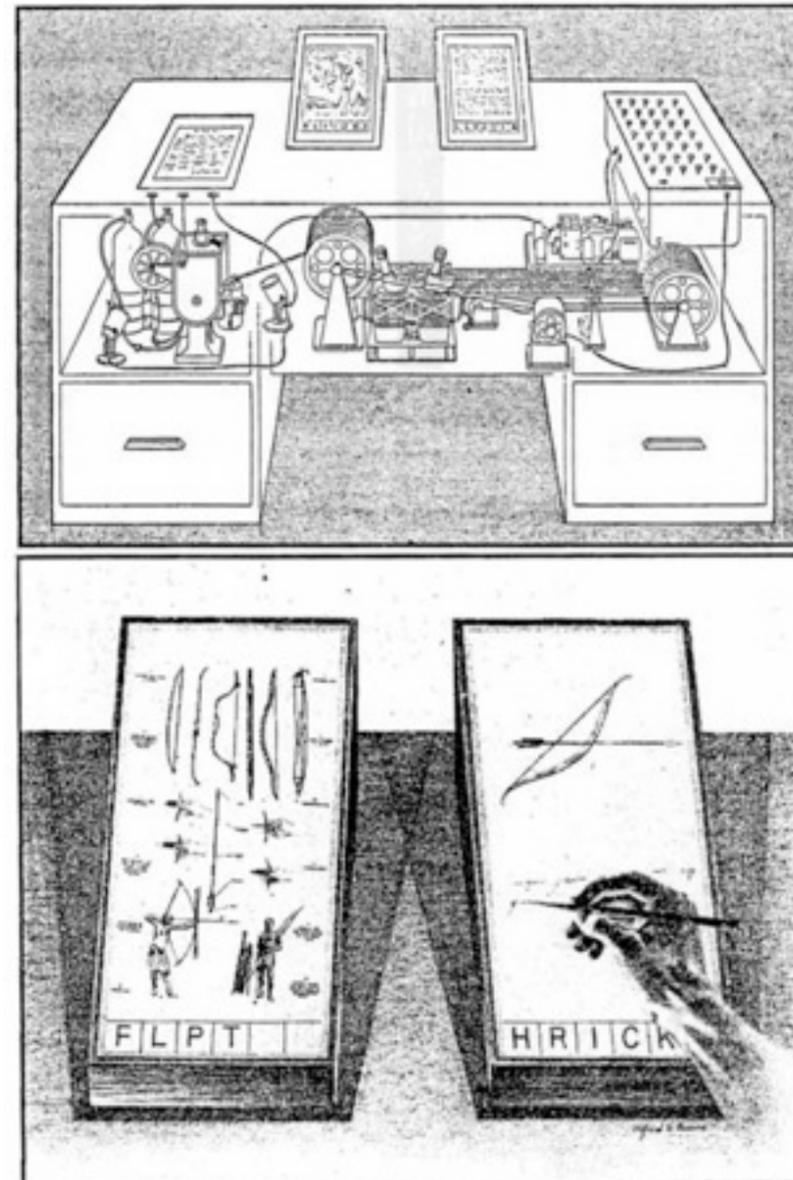


Wearable Computing

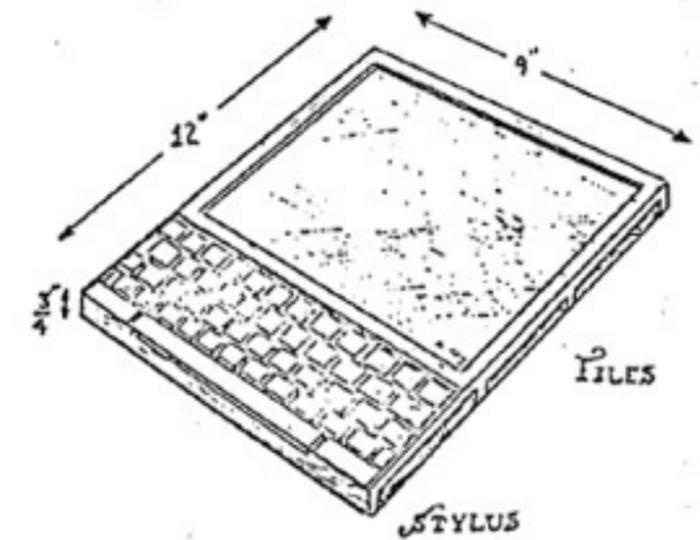
Berühmte Designstudien



Ramelli: Bücherrad, 16. Jh.



Bush: Memex, 1945



Kay: Dynabook, ca. 1972

Usage Video

Pomegranate  | NS08

Share Release Date



Projector

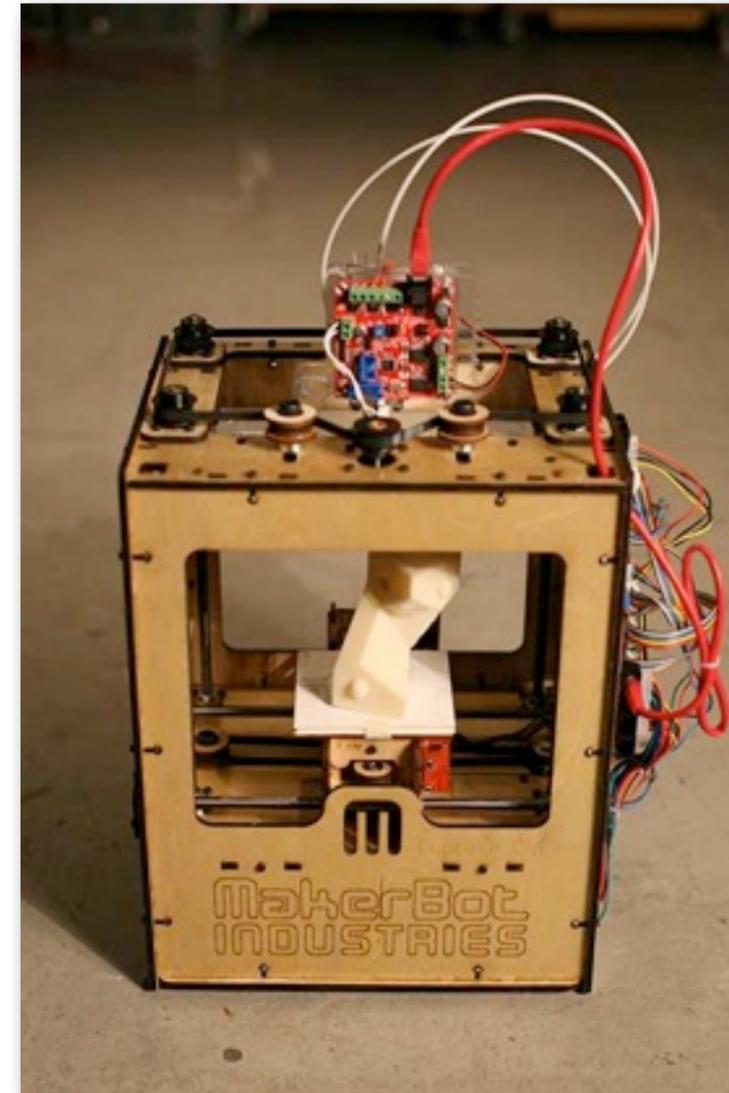
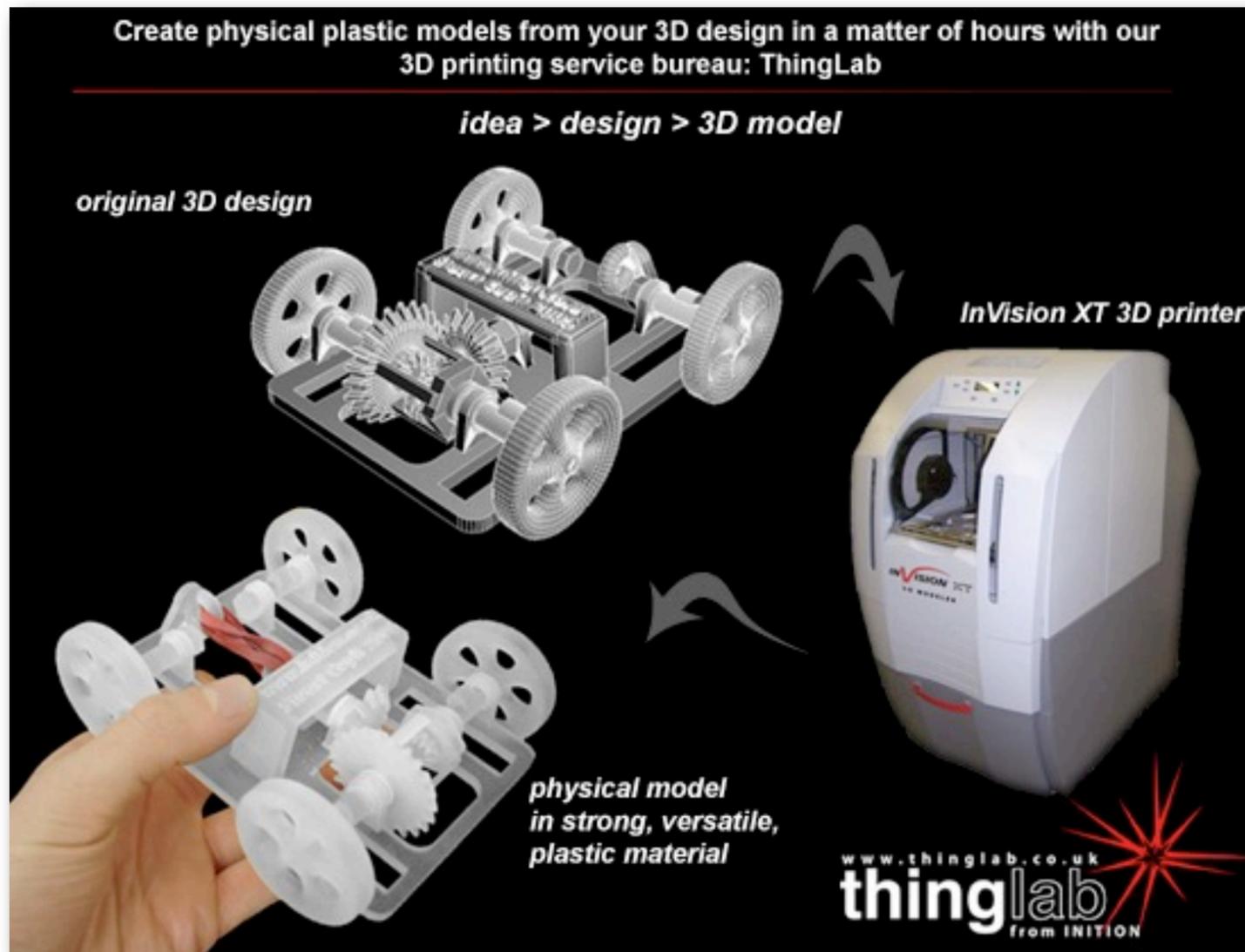
http://www.youtube.com/watch?v=tY_7HpgDrdQ&feature=related



<http://www.youtube.com/watch?v=NwVBzx0LMNQ>

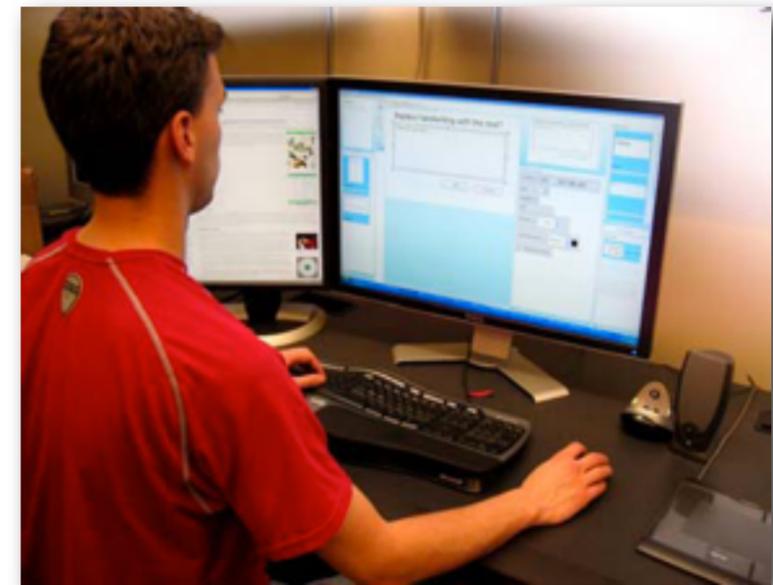
Mockups

Mockups repräsentieren die Erscheinung des Systems, ohne notwendigerweise Funktionalitäten zu implementieren.

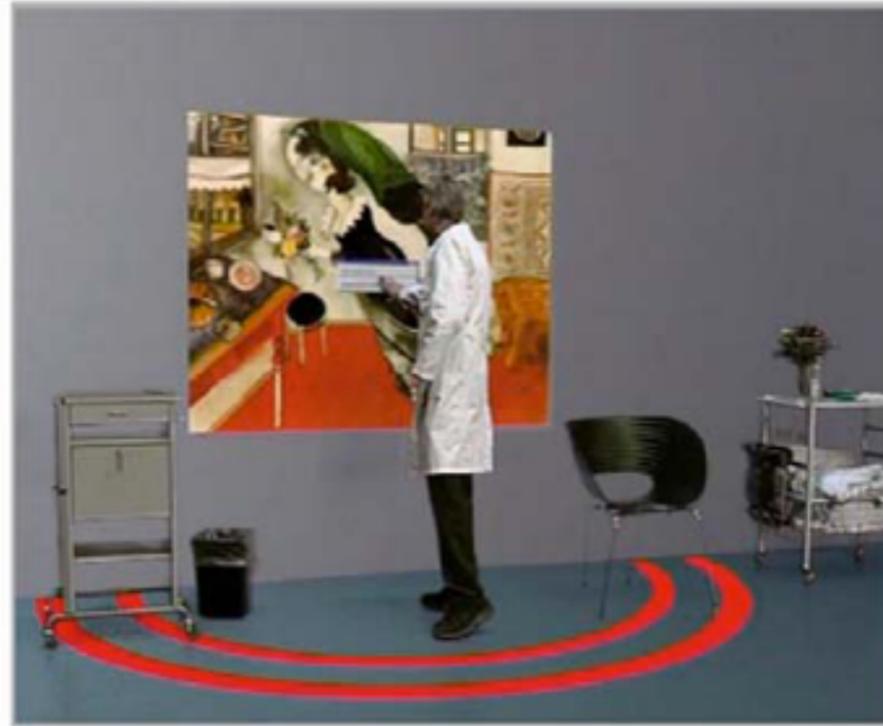


3D-Drucker ermöglichen u.a. das Erstellen von realistischen Mockups

Interaktion: Wizard-of-Oz-Prototypen



Prototyp-Theater



Ein Szenario wird vor einem Bluescreen durchgespielt und in der Postproduktion zu einem vollständigen Usage-Video ausgebaut.

Figure 4. Diana enters in a post office and has to take a number for the cue. There are several buttons to push for the numbers according to the service. Beside each button there is an explanation in Finnish, Diana scans the words with her magic thing, which are translated into English.

Experience Prototyping

Experience Prototype is any kind of representation, in any medium, that is designed to understand, explore or communicate what it might be like to engage with the product, space or system we are designing.



Figure 2: Experiencing a train journey.

The team combined objective passenger research with subjective discovery as they played out roles they assigned each other.



Figure 4: *Bodystorming layouts for an airplane interior.* Ideas were generated and evaluated rapidly by the team as they directly experienced physical and social issues in this full-scale environment.

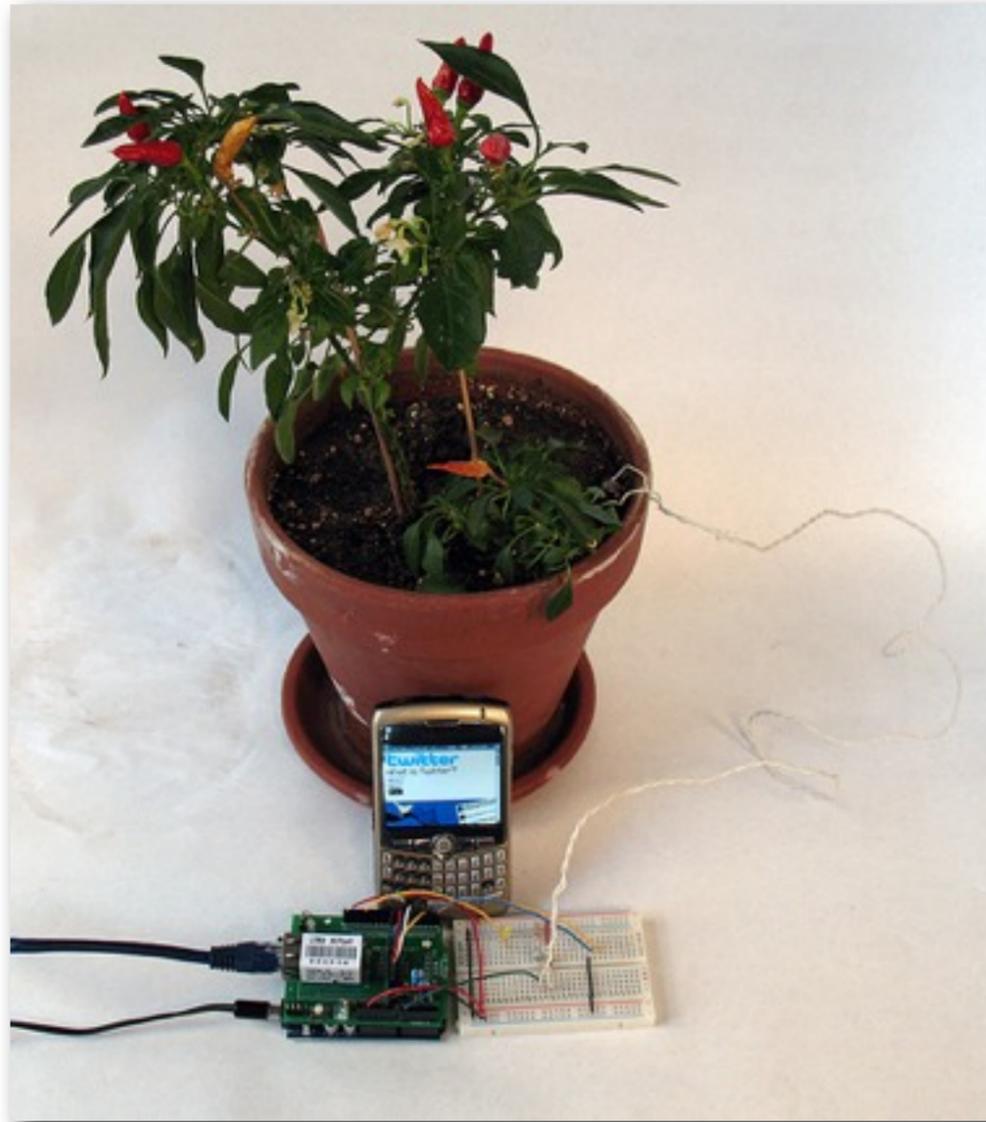
Hi-Fi-Prototypen



Figure 6: *Digital camera interaction architecture prototype.* The prototype used a desk-top computer's processing power to manipulate the dynamic qualities of the control system and screen behavior.

Für Gestaltung,
Aussehen, Zeitverhalten,
genaue Funktionen

Funktionsfähige Prototypen

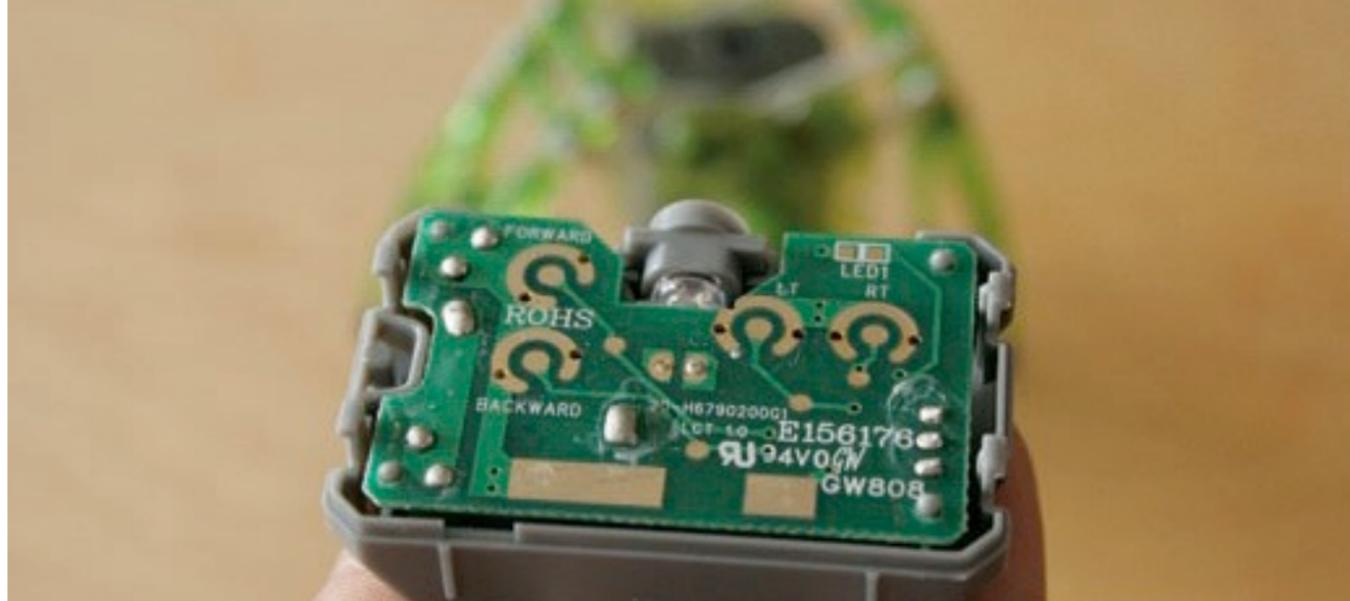


<http://shebytes.posterous.com/inspiring-arduino-project-botani-calls-twitte>

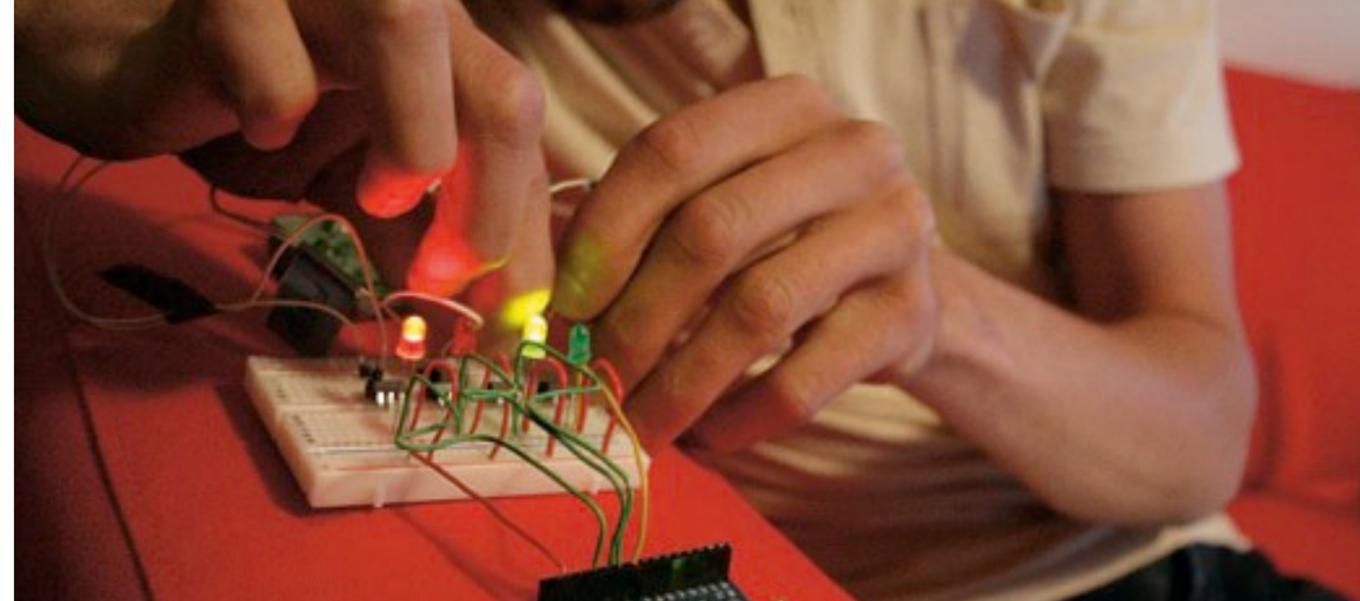
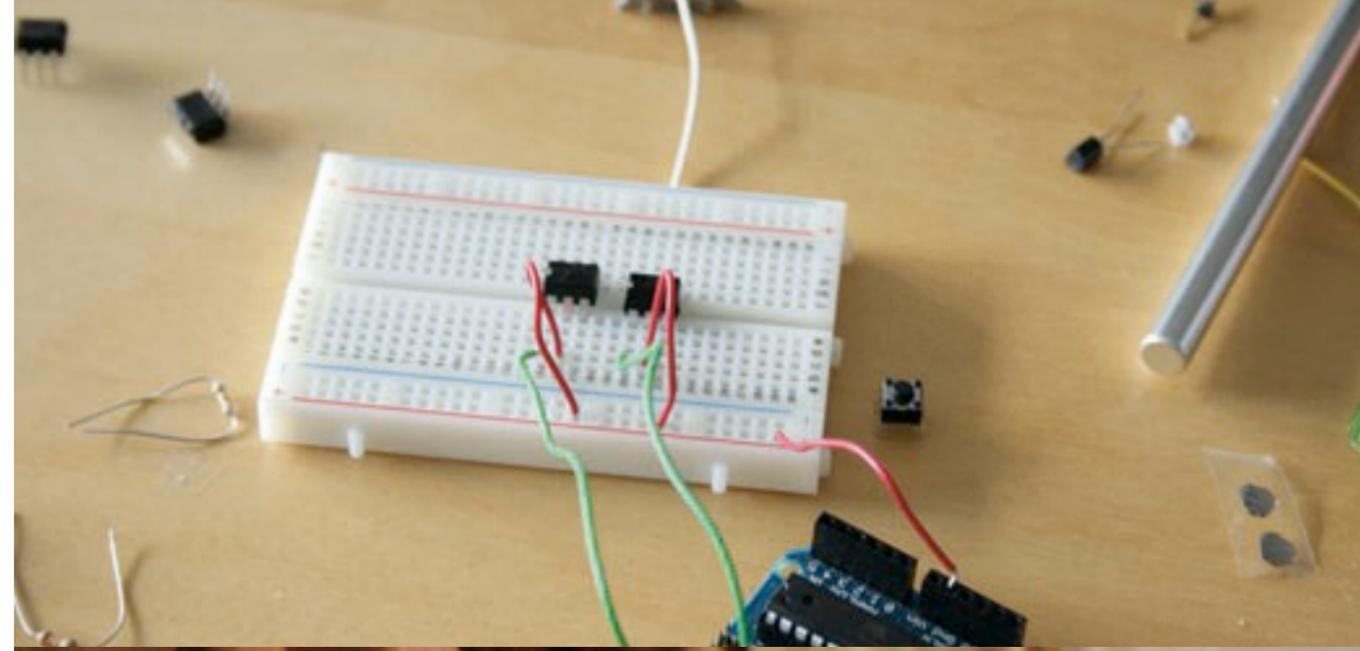
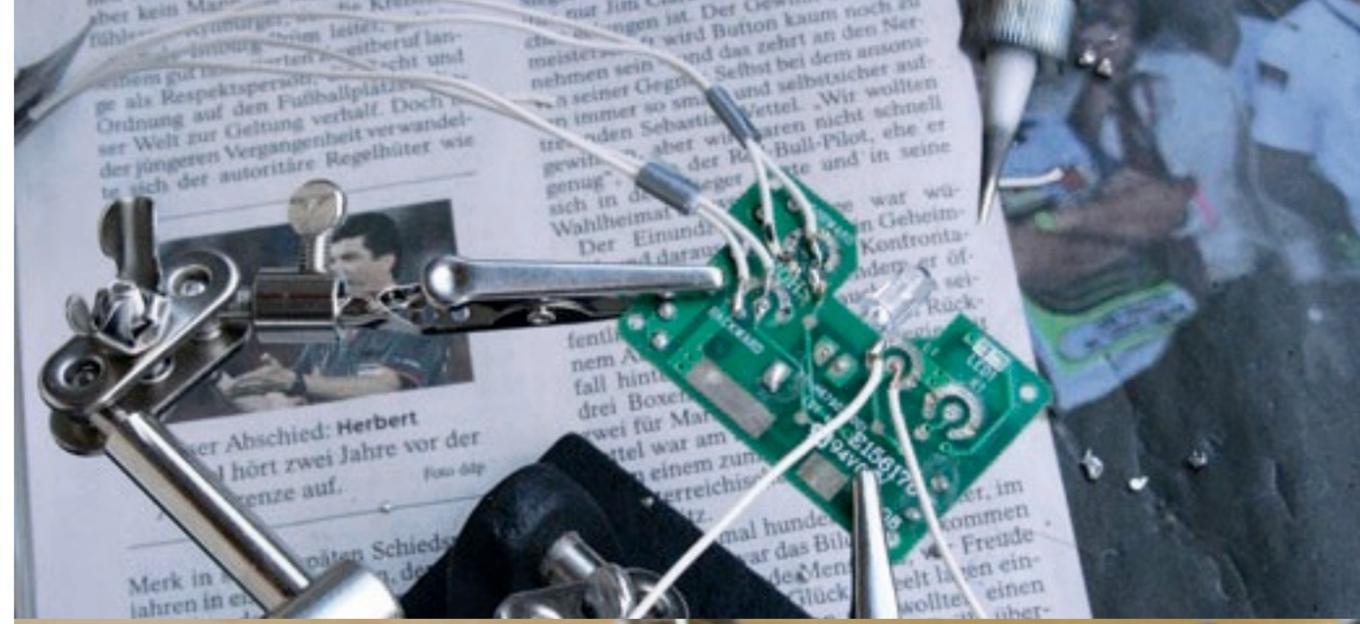


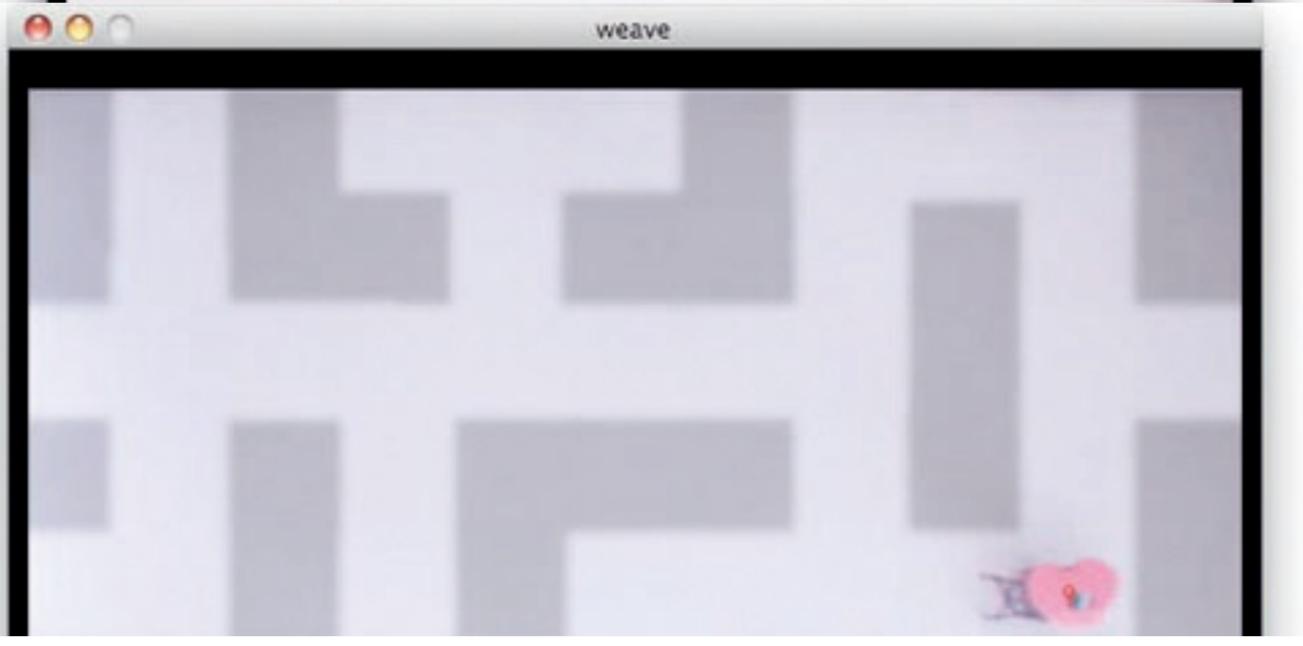
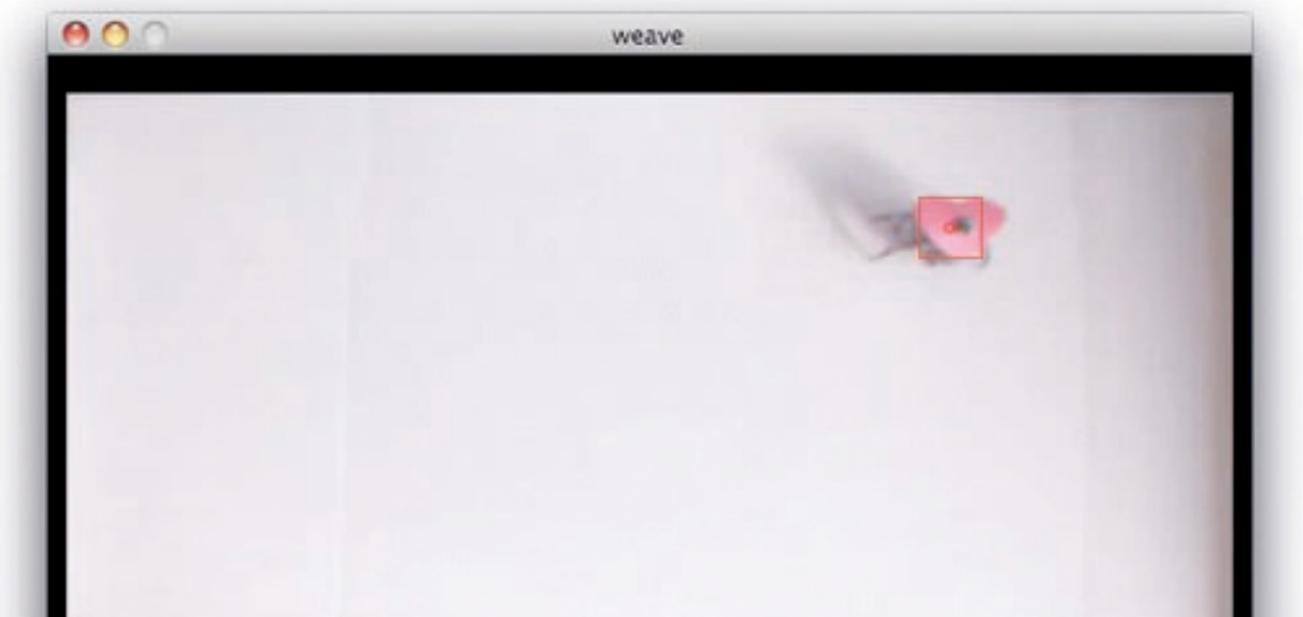
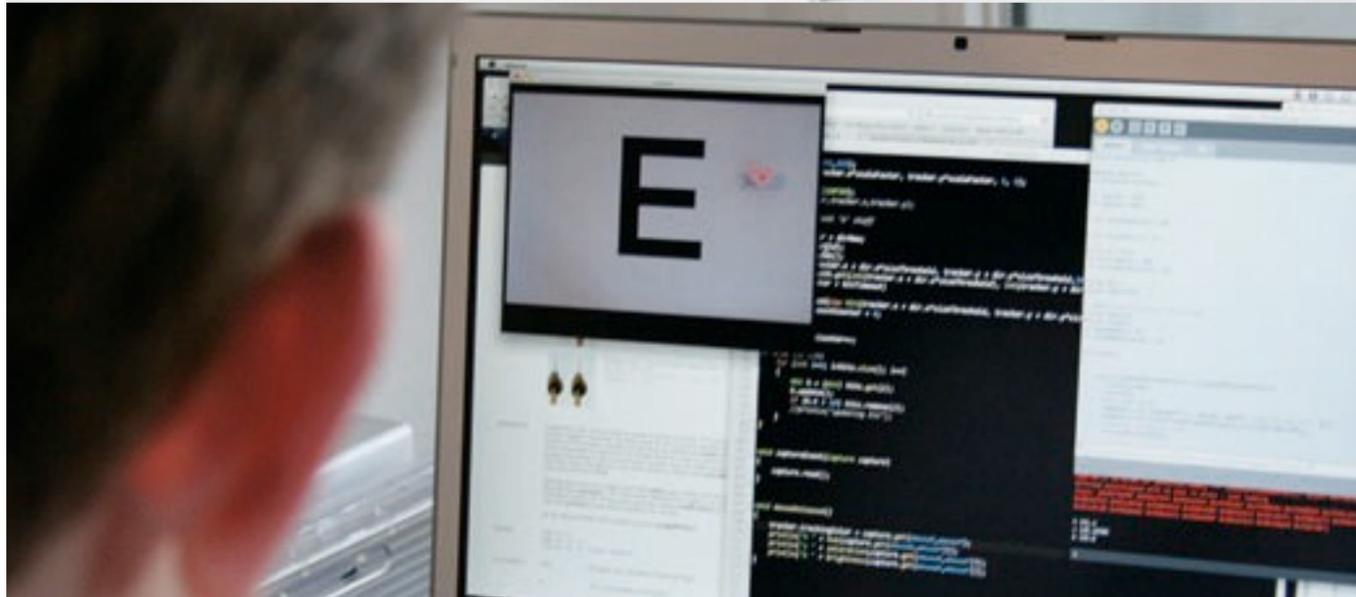
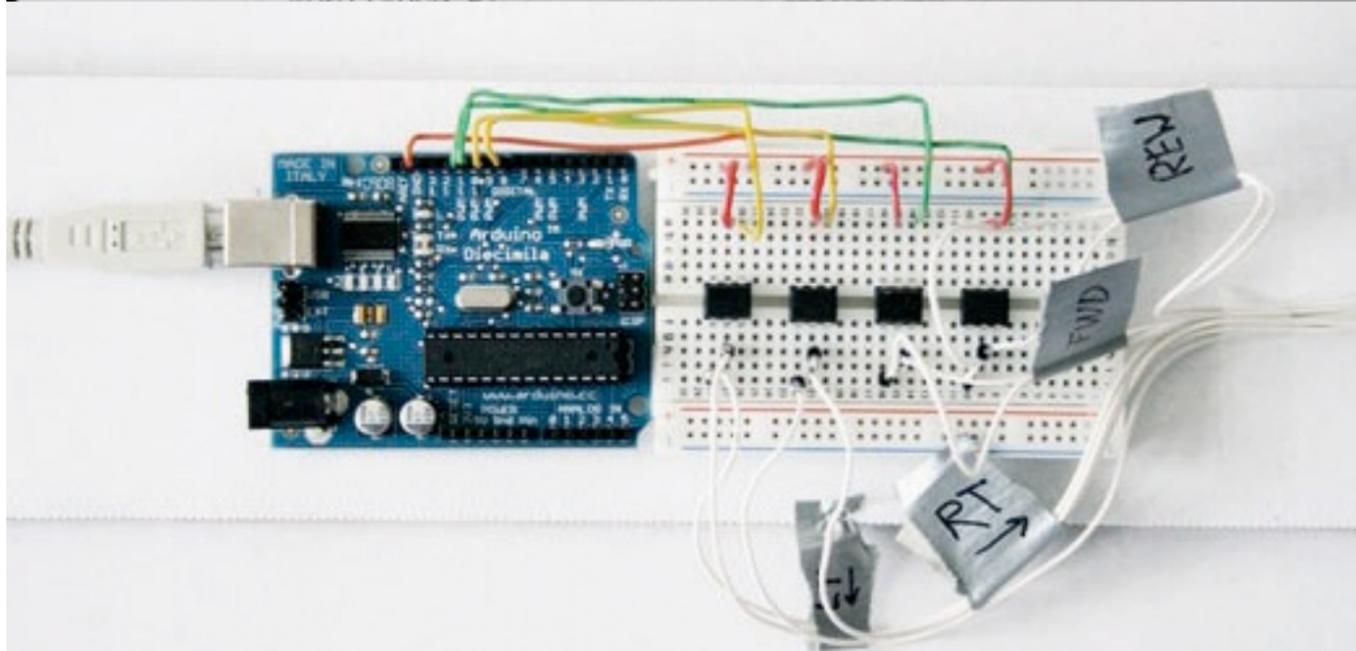
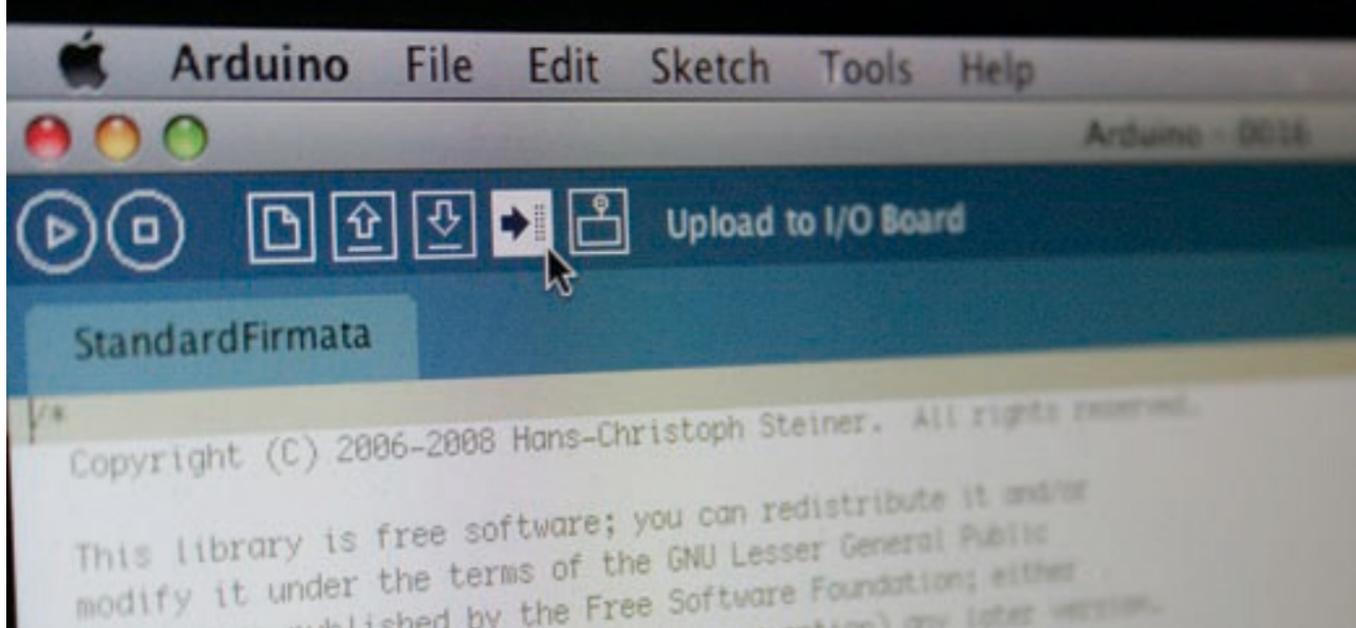
Energy Race, ein Projekt im iLab.

Tentakel-Roboter



<http://www.spiegel.de/netzwelt/games/0,1518,647596,00.html>





Wearable Computing

Steve Mann's "wearable computer" and "reality mediator" inventions of the 1970s have evolved into what looks like ordinary eyeglasses.



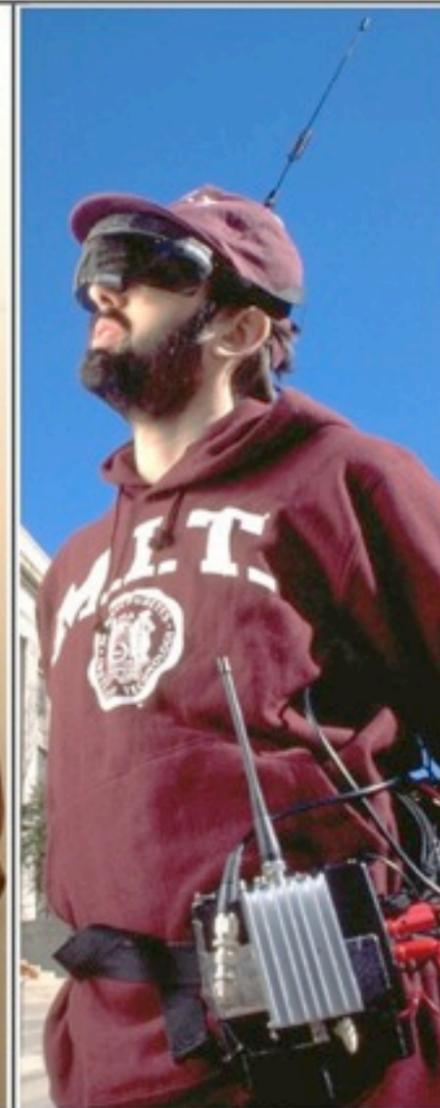
(a)
1980



(b)
Mid 1980s



(c)
Early 1990s



(d)
Mid 1990s



(e)
Late 1990s

Ubiquitous Computing



Pranav Mistry: Sixth Sense

<http://www.youtube.com/watch?v=mUdDhWfpqyg>

Werkzeuge

Die im iLab betreut werden



Exhibition

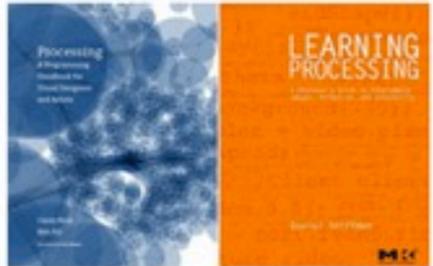


[Branching Morphogenesis](#)
by Sabin+Jones LabStudio

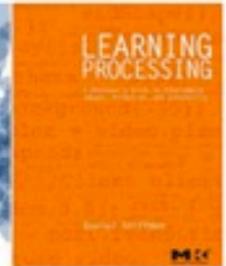


[Telekom Realtime Information Graphics](#)
by Zum Kuckuck

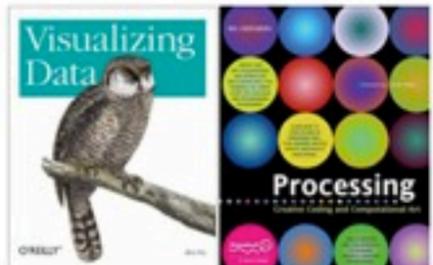
Books



[Processing: A Programming Handbook for Visual Designers and Artists](#)
By Casey Reas and Ben Fry



[Learning Processing](#)
By Daniel Shiffman



» Download Processing

Processing is an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool.

Processing is free to [download](#) and available for GNU/Linux, Mac OS X, and Windows. To contribute to the project's development, please visit <http://dev.processing.org/>, which includes bug tracking and instructions for building the code, downloading the source, and creating libraries and tools.

Processing is an open project initiated by [Ben Fry](#) and [Casey Reas](#). It evolved from ideas explored in the Aesthetics and Computation Group at the MIT Media Lab.

Please check out recent Processing activity on the Web:

[OpenProcessing](#)
[Processing @ Vimeo](#)
[Processing @ del.icio.us](#)
[Processing @ Flickr](#)
[Processing @ YouTube](#)

Updates

18 Oct 2009
Processing 1.0.8 released.

18 Oct 2009
Software from Sabin+Jones LabStudio added to the [exhibition](#).

2 Oct 2009
Software from Zum Kuckuck added to the [exhibition](#).

4 Sep 2009
Processing 1.0.7 released.

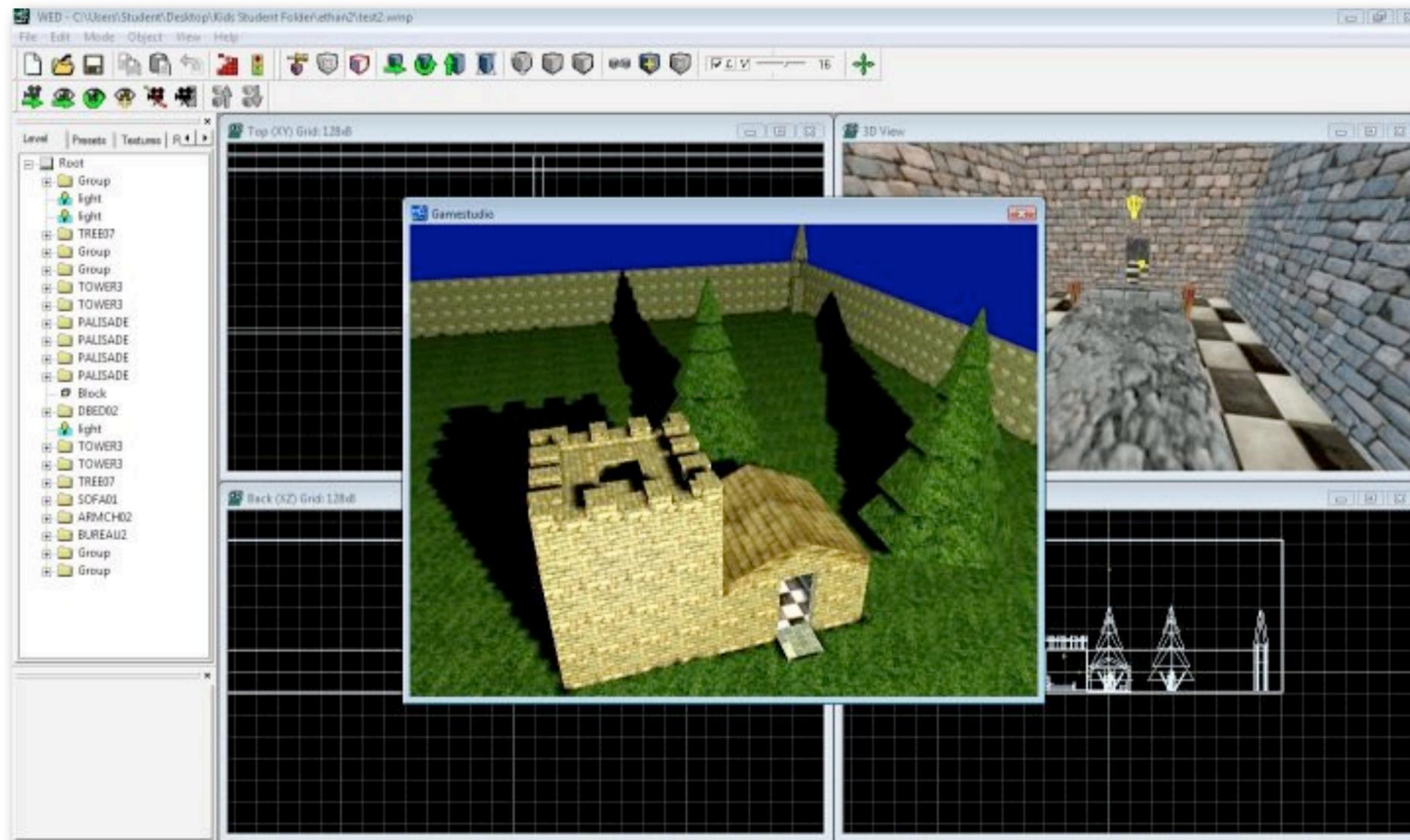
3 Sep 2009
Software from Ben Hemmendinger added to the [exhibition](#).

24 Aug 2009
Nine new libraries added to [Contributions](#): ttslib by Nikolaus Gradwohl, MRI3DS and ObjImport by Victor Martins, integralhistogram by Giovanni Tarducci and

Processing

processing.org

Game Studio A8



www.conitec.net/german/gstudio/

MAX / MSP / Jitter

<http://cycling74.com/products/>

The screenshot shows a Max/MSP patch window titled "color tracking". The patch is designed to track a specific color in a video stream. It starts with a "metro 2" object connected to an "open" object, which feeds into a "jit.qt.grab 320 240" object labeled "LIVE INPUT". The video data then goes through a "jit.brcosa @saturation 3" object. A "t b b l" (tab) object is used to switch between two video outputs. The first output is the original video, and the second output is the result of color tracking. The color tracking process involves a "pack write RED!" object, a "pack moveto 0 0" object, and a "jit.lcd 4 char 320 240" object. The tracked color is displayed on a video frame of a person holding a red circle with the word "RED!" written on it. The patch also includes a "COLOR BEING TRACKED" section with a color picker and a "POSITION OF COLOR BEING TRACKED" section with numerical outputs and scaling objects. The window title bar shows "Max/MSP File Edit View Object Font Options Trace Extras Window Help" and the system status bar shows "Sat 2:38 (Charged)".

<http://www.maxmspjitter.com/>

Pure Data

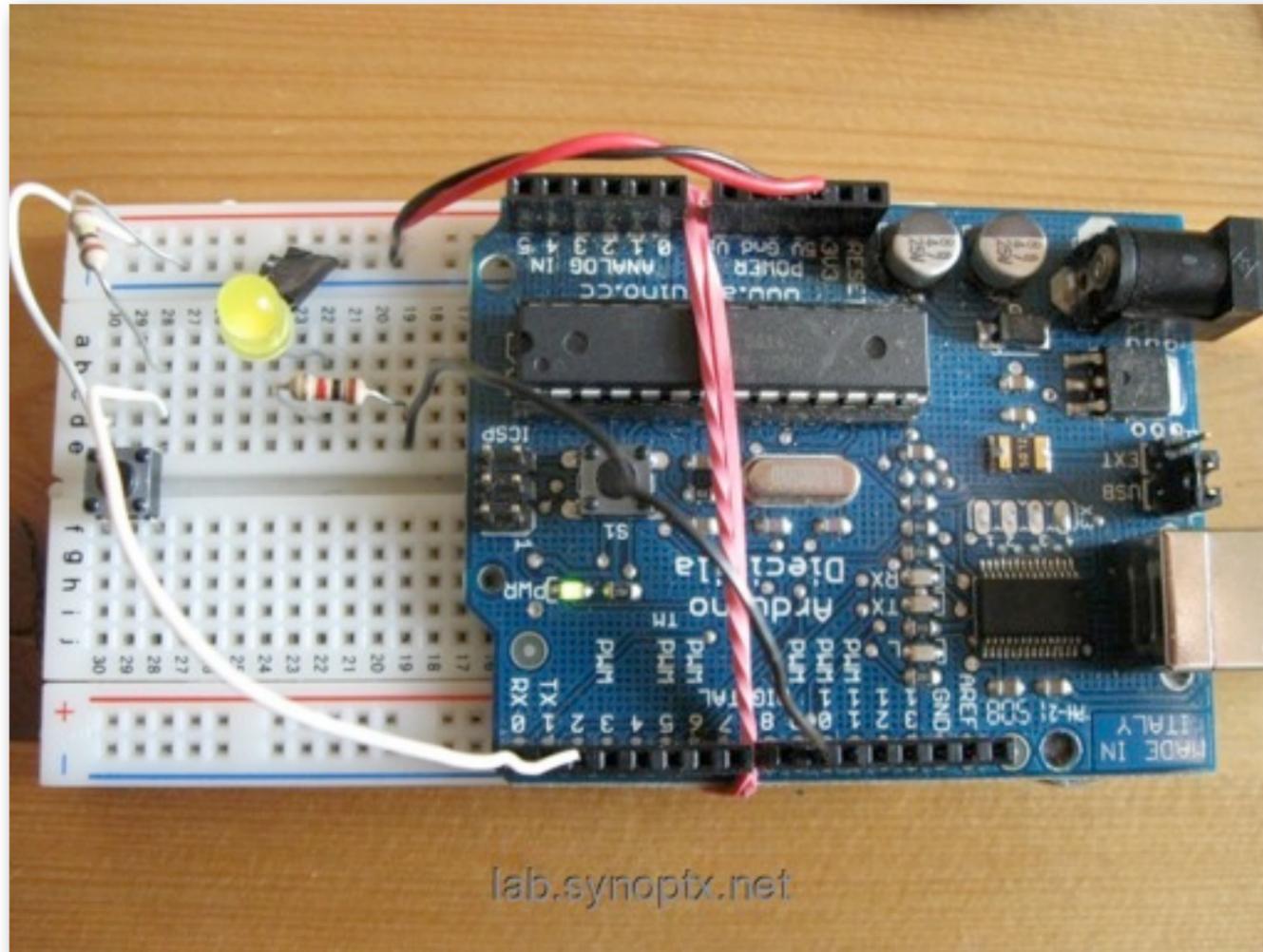
puredata.info

The image displays two windows from a Pure Data environment. The left window, titled 'reactIVision - 25 FPS', shows a video feed of a hand interacting with a game controller. A 'Command Prompt' window is overlaid on the left, displaying a list of 'set obj' commands with numerical data. The right window, titled 'TUIO_PureData-redux-fm.pd', shows a Pure Data patch. The patch starts with a 'Tuioclient' object connected to a 'route' object with arguments 'addObject', 'removeObject', and 'updateObject'. The 'route' object is connected to three 'unpack' objects: 'unpack \$1 \$2', 'unpack \$1 \$2', and 'unpack \$1 \$2 \$3 \$4 \$5'. The first two 'unpack' objects are connected to 'print added' and 'print removed' objects respectively. The third 'unpack' object is connected to three multiplier objects ('* 640', '* 480', '* 360'), which are then connected to '311.0', '317.7', and '32.17' objects. These three objects are connected to 'x-axis', 'y-axis', and 'angle' objects. The 'angle' object is connected to a 'mtof 180' object, which is then connected to a '* 360' object. The 'x-axis' and 'y-axis' objects are connected to '317.7' and '160.87' objects respectively. These two objects are connected to a '+' object, which is then connected to an 'osc~' object. The 'angle' object is also connected to an 'osc~' object. The two 'osc~' objects are connected to a '+' object, which is then connected to another 'osc~' object. Finally, the 'osc~' object is connected to an 'out~' object with arguments 'dB 36' and 'mute'.

<http://www.flickr.com/photos/redux/933366453/>

Arduino / Lillypad

arduino.cc



Flash

```
function  
  f() CardGame  
  f() addToHand  
  f() burn  
  f() deal  
  f() decideWinner  
  f() flop
```

```
480 // We don't know the dataType of the first argument (it's  
481 // likely a CardPlayer, but it could also be the "community"  
482 // String. Therefore, we'll enumerate the arguments  
483 // to set the _player_ and _cards_ variables passed to this  
484 routine  
485 var player = arguments[0];  
486 var given = arguments[1];  
  
if (String(player) == 'community') {  
  for
```

concat (Array):
concat method)
public concat([value:Object]) : Array
Concatenates the elements specified
the parameters with the elements in
array and creates a new array. If the
value parameters specify an array,



```
startRound  
turn
```

```
493 //var evaluation = pitBoss.evaluateHand  
494 (this[allPlayers[i]]);  
495 table[allPlayers[i].name+"_handDescription"].text =  
496 pitBoss.evaluateHand (allPlayers[i])[0];  
497 }  
498 } else {  
  if (given != undefined) {
```

Parameters
value: Object [optional] - Numbers,
elements, or strings to be
concatenated in a new array. If you
don't pass any values, a duplicate of

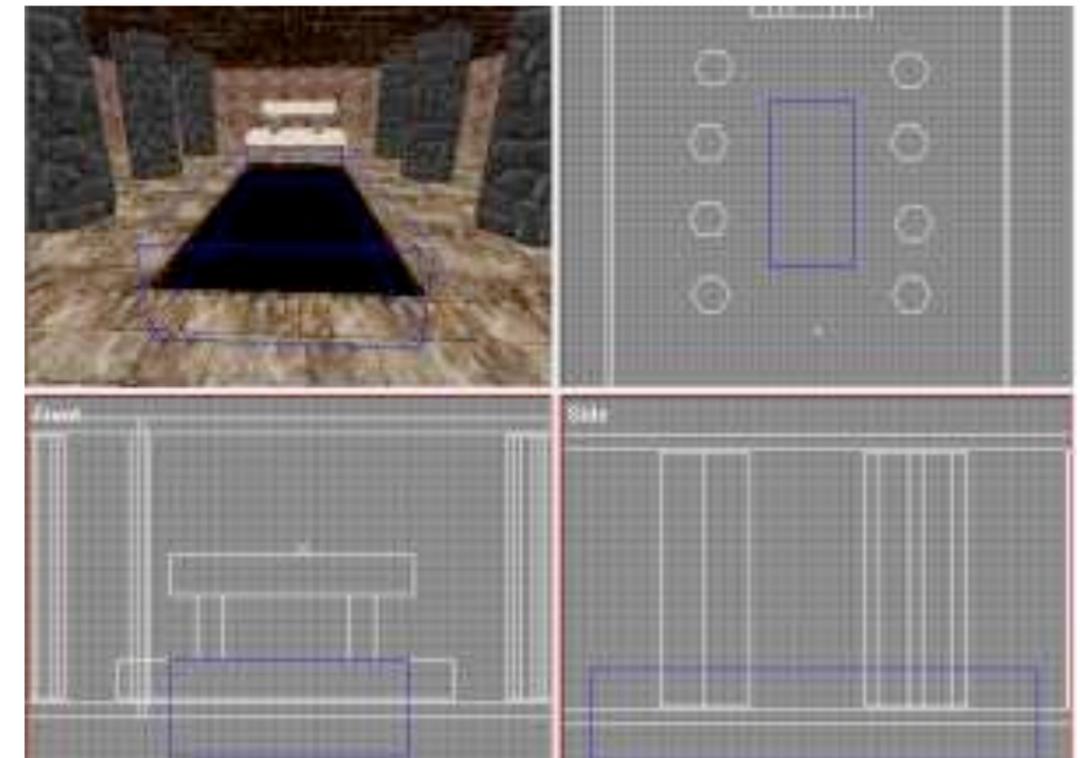
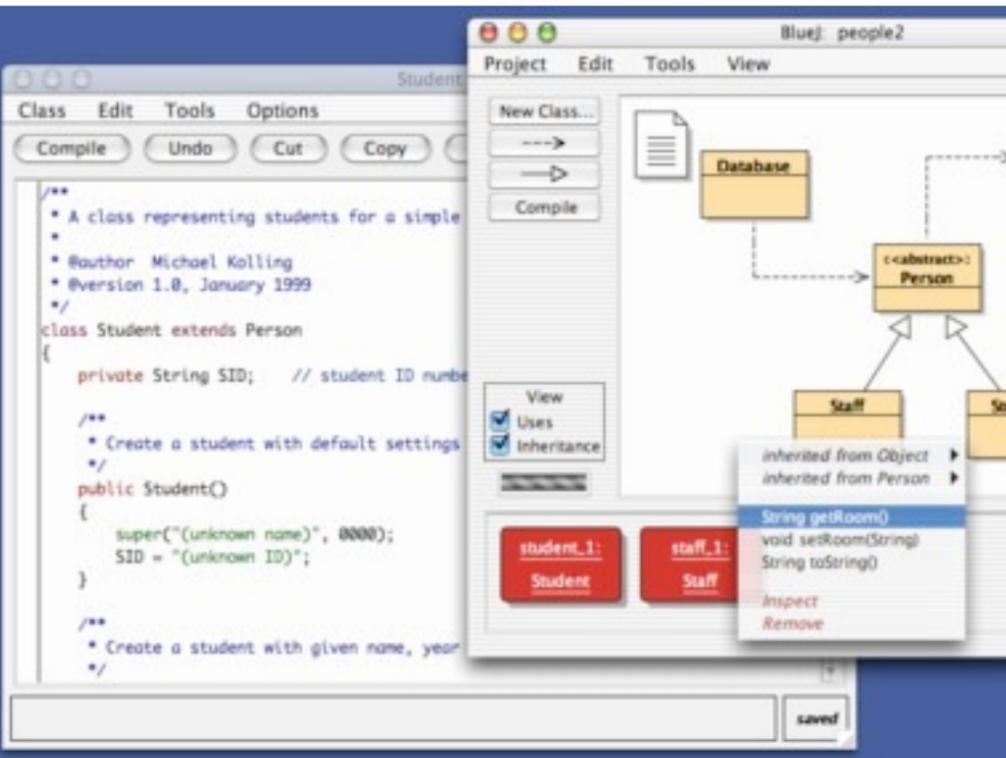
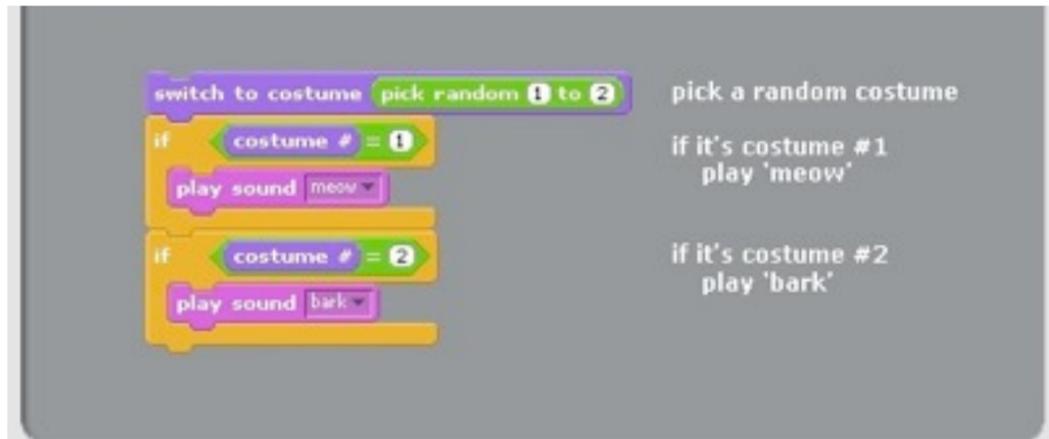
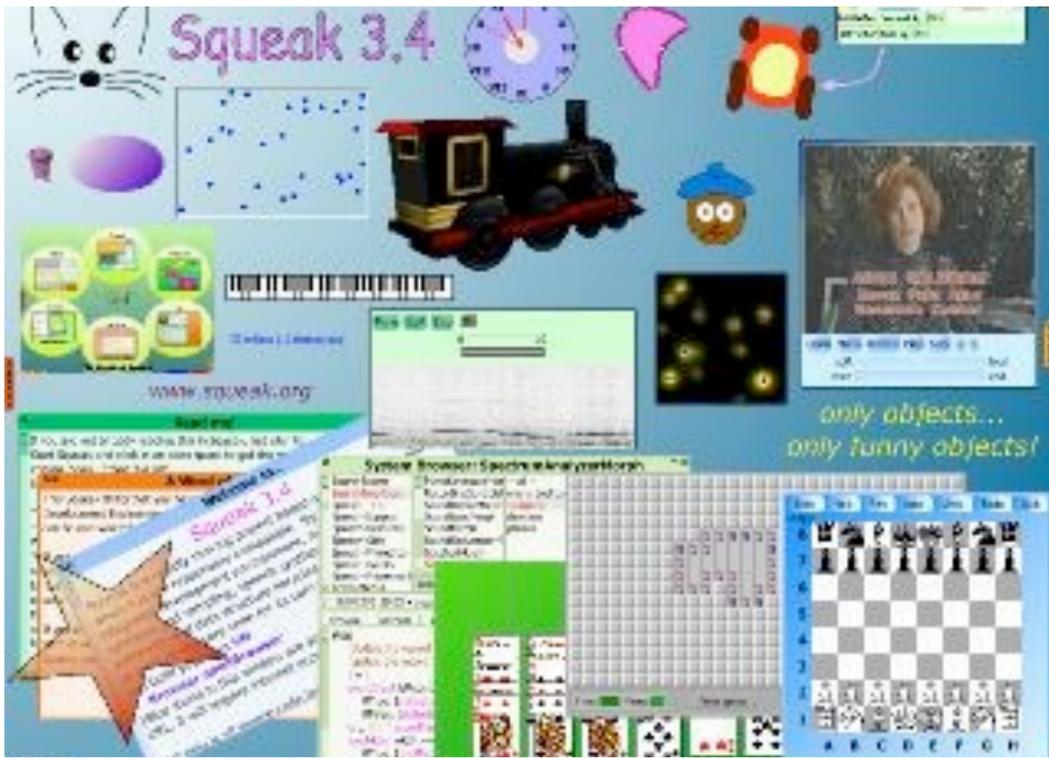


```
503 }  
504 }  
505 }  
506 }
```

Example
The following code concatenates two
arrays:
var alpha_array:Array = new Array
var numeric_array:Array = new
var alphaNumeric_array:Array =
trace(alphaNumeric_array);
// Creates array [a,b,c,1,2,3]

```
Line # Description  
168 TODO: Move the player definitions outside of the startRound() function to allow players to maintain stat...
```

Example
The following code concatenates two
arrays:
var alpha_array:Array = new Array
var numeric_array:Array = new
var alphaNumeric_array:Array =
trace(alphaNumeric_array);
// Creates array [a,b,c,1,2,3]



Squeak | Etoys | Hackety Hack | Scratch | BlueJ | Reality Factory