CHAPTER 38
Software Prototyping
Prototyping Stages (DIS1)

Product Idea

- Low Fidelity
  - Paper / Post-It Prototype

- Medium Fidelity
  - Screenshot / Click-through Prototype

- High Fidelity
  - Non-linear Software / Hardware Prototype

Working System

Time
CHAPTER 39

Prototyping Standard Apps
• Wireframing tool

• Design mockups

• First released in 2008

• **WYSIWYG** editor
When to Use

- **Product Idea**
  - Paper / Post-It Prototype

- **Low Fidelity**
  - Paper / Post-It Prototype

- **Medium Fidelity**
  - Screenshot / Click-through Prototype

- **High Fidelity**
  - Non-linear Software / Hardware Prototype

- **Working System**

Time Flow: Paper → Screenshot → System
Mockups is Easy

Start by adding controls from the UI Library

Double-click to edit the controls you add

Want to see something really cool?

Hold R, T or Y and drag out a square with your mouse.

See what else you can build with Balsamiq
Demo: balsamiq®
Balsamiq: Features

• Focused on structure rather than colors and icons

• For rough wireframes

• Library of UI elements

• Can design widgets
CHAPTER 40

Sketching
Adobe Xd

• UI sketching and prototyping tool

• Create user interfaces for mobile and web applications

• First released in 2017

• Free
When to Use

Product Idea
- Paper / Post-It Prototype

Low Fidelity
- Screenshot / Click-through Prototype

Medium Fidelity
- Non-linear Software / Hardware Prototype

High Fidelity
- Working System

Time
Nonlinear Software Prototypes

- In-App preview
- Mobile preview app for iOS and Android
Sharing Prototypes

Designer
Creates and shares the design with the developer via a web link

Designer
Reviews and responds to the developer comments

Developer
Adds comments after reviewing the prototype (if required)
Demo: Adobe Xd
Adobe Xd: Features

- **Scrolling** by resizing canvas
- Repeat Grid
  - Tool that lets you replicate a group of objects
- Voice Design
- Supports applications in the Adobe Creative Suite (Illustrator, Photoshop, Photoshop Sketch)
- Developer file
Sketch Software

- First released in 2010
- Designing **UI** and **UX** of mobile **apps** and **websites**
When to Use

Low Fidelity
- Product Idea
  - Paper / Post-It
  - Prototype

Medium Fidelity
- Screenshot / Click-through Prototype

High Fidelity
- Non-linear Software / Hardware Prototype
  - Working System

Time
Interface

Toolbar

Layer List

Inspector

Canvas
Demo: Sketch
Sketch: Features

- Sketch vs. Illustrator and Photoshop
  - Combines Design and Prototyping
  - Focused on the needs of the UI and icon designer
  - Good for newbies

- **Scrolling** by resizing canvas

- **Symbols** allow reusing elements across different Artboards and Pages so you can save time and stay consistent

- **Libraries** with designed widgets, e.g., for iOS, provided on Apple website
Origami Studio

• Free design tool by Facebook

• Plug-in for Quartz Composer

• Visual programming environment

• Programming in Quartz Composer by connecting patches to each other

• Good companion to Sketch
When to Use

Low Fidelity
- Paper / Post-It Prototype
- Screenshot / Click-through Prototype

Medium Fidelity

High Fidelity
- Non-linear Software / Hardware Prototype

Working System
Interface

Profile Dr. Jan Borchers: Designing Interactive Systems 2

- Viewer
- Layer List
- Properties
- Add Components
Demo: Origami
Origami: Features

- Building small high fidelity prototypes
  - Rather complex in its functions

- Tweaking the interaction you have in mind

- Not suitable for quick wireframing

- **Scrolling** possible

- Pasting of Sketch UI files possible

- Good for logic of realistic interaction sequences — state changes, timeouts, …
### Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Tool</th>
<th>Balsamiq</th>
<th>Adobe Xd</th>
<th>Sketch</th>
<th>Origami</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plattform specific options</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yellow</td>
</tr>
<tr>
<td>Sharing Designs</td>
<td></td>
<td>Only PNG (not interactive)</td>
<td>Online Link</td>
<td>Online Link</td>
<td>Via Mail</td>
</tr>
<tr>
<td>Libraries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Red</td>
</tr>
<tr>
<td>Scrolling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Green</td>
</tr>
<tr>
<td>Developer file</td>
<td></td>
<td>PNGs without specifications</td>
<td>Sharing includes all assets, dimensions</td>
<td>objects are exported with specifications</td>
<td>Export code snippets</td>
</tr>
<tr>
<td>Real Testing</td>
<td></td>
<td></td>
<td>App for iOS and Android</td>
<td></td>
<td>Origami Live app on Mac</td>
</tr>
<tr>
<td>Advanced Interactions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Microphone &amp; Sound</td>
</tr>
</tbody>
</table>

**Prof. Dr. Jan Borchers: Designing Interactive Systems 2**
Which Tool to Use When?

• balsamiq:
  • Standard (productivity) apps
  • Fast wireframing by using widgets for medium fidelity

• Adobe Xd & Sketch:
  • Standard UIs via libraries, and custom UIs (immersive apps, games)
  • Medium to high fidelity

• Origami:
  • Small, high fidelity, model interaction sequence
  • Reuse Sketch files
CHAPTER 41
Multimedia Prototyping: Max/MSP
Intro to Max

- Visual **dataflow-based programming** inspired by patch panels in recording studios
- Tuned for working with time-based input streams
- Programs are called **patches**
- **Objects** contain functionalities and are visually represented by boxes
- **Patch cords** connect objects
HCI View on Max

- **Prototyping** UIs
  - For working with time-based media
  - Using time-based media as input/output devices
- No need to be a DSP guru
- **Liveness** allows fast development
  - Locked: use patch
  - Unlocked: edit patch
“If most MIDI apps are pizza, then Max is a kitchen.”

- Miller Puckette
History

- 1986: **Patcher** for Mac MIDI and control processing

- 1989: **MSP** to process digital audio

- 1989: Ircam Musical Workstation featuring Max containing
  - GUI on NeXTStep
  - „Faster Than Sound“ card

- 1995: **pd** (pure data) ability to share patches between multiple projects

- 2000s: **Jitter** real time processing of matrix data, e.g. video, 3D; Windows support

- 2010s: **Vizzie** visual processing module, **Ableton Live** integration, **Gen** integrates imperative programming blocks, **NodeJS** support
Objects

• Normal objects
  • Squares with object name in it
  • Object name can be followed by arguments
  • Inputs are at the top, outputs at the bottom

• UI objects
  • Sliders, dials, number fields, buttons, etc.

• Comments
Messages

- Sent through patch cords

- Various **types**
  - number: int, float
  - list of numbers, separated by spaces
  - word ("symbol")
  - bang
  - combinations

- **Message order**
  right to left, bottom to top
Math in Max

- Basic operations as objects like $\ast$, $+$, $-$, $/$
- Common functions like $\text{sin}$, $\text{abs}$, $\text{sqrt}$
- $\text{expr}$ object for more complex terms
- $\text{if}$ for conditions
Creating Patches

**DEMO 1**
Using a table to map slider input to slider output

**DEMO 2**
Adding integers
MIDI Handling

• Objects for input and output of notes, control messages and raw data, e.g. `ctlin`, `midiout`, `notein`

• Objects for translation between MIDI notes and frequencies

• MIDI messages are ordinary messages: MAX handles MIDI as sequence of numbers
MSP

- MSP object names end with ~
- MSP signal patch cords are curly
- Curly signals cannot be connected to regular objects
- DSP operations happen at sample timing precisions
MSP Features

- **adc~** and **dac~** for audio in-/output

- **fft~** and **ifft~** convert from time to frequency domain
  - phase vocoder (timestretching)
  - complex filters

- **buffer~** for storing sample data

- UI elements for visual analysis
Simple MSP Patch

DEMO 3
Performing ring modulation

- **cycle object** generates a periodic waveform
- **ctlin object** outputs MIDI control signals
Jitter

• 130 additional objects for Max/MSP

• processing of matrix data (rows and columns)
  • video, still images, 3d geometry, text, spreadsheets, particle systems, voxels or audio
Jitter Objects

• most names start with \texttt{jit}.

• attributes instead of dozens of arguments

• the green matrix patch cords cannot be connected with other Max/MSP objects
Jitter Objects

- **jit.qt.\*** for video
- **jit.gl.\*** for 3d/OpenGL
- **jit.p.\*** for particle effects
- **jit.la.\*** for linear algebra

- **jit.peek~** and **jit.poke~** for connection to MSP
Extending Max/MSP

• **Software**
  - write plugins in C that appear in Max as new objects
  - free SDK available

• **Hardware**
  - MIDI controllers and synthesizers
  - DMX Lighting
  - Arduino
Über diesen Macintosh

Macintosh LC

Systemsoftware 8.1.5
© Apple Computer, Inc. 1983-94

Gesamtspeicher: 4.096K
Größter freier Block: 1.523K

Systemsoftware: 2.543K
Topics today

$UI = \text{Form} + \text{Function} + \text{Sensors \& Actuators}$
CHAPTER 42

Prototyping Form: Mechanics
Why Hardware Prototyping? – Ubiquitous Computing
1. CAD Design

2. 3D Printing

3. Casting
CHAPTER 43

Prototyping Function: Embedded Programming
Adding function to hardware prototypes

–

Embedded programming & electronics
14 digital I/O (40mA)
incl. 6 PWM ‘analog’ out

ICSP

USB power

...or...

7–12V in

...or...

ATmega 328
@ 16MHz

6 analog in 3.3 & 5 V
mbed
Raspberry Pi
Particle
CHAPTER 44

Prototyping Peripherals: Electronics
Form
3D design, 3D print, casting

Form
2D design, laser cutter

Function:
Programmed Arduino Board

Peripherals:
LED, switch, colour sensor
CHAPTER 46

Electronics Prototyping: Fritzing
• Open-source initiative

• Release: 2008

• Electronic design automation (EDA) tool for non-engineers

• Key advance:
  • Drag & Drop realistic components
  • Software generated PCB design for manufacturing an Arduino shield
Interface

Canvas

Components

Properties
Demo Fritzing