

2 Mobile and Ubiquitous User Interfaces

2.1 Mobile Computing

2.2 Design Guidelines for Mobile Devices

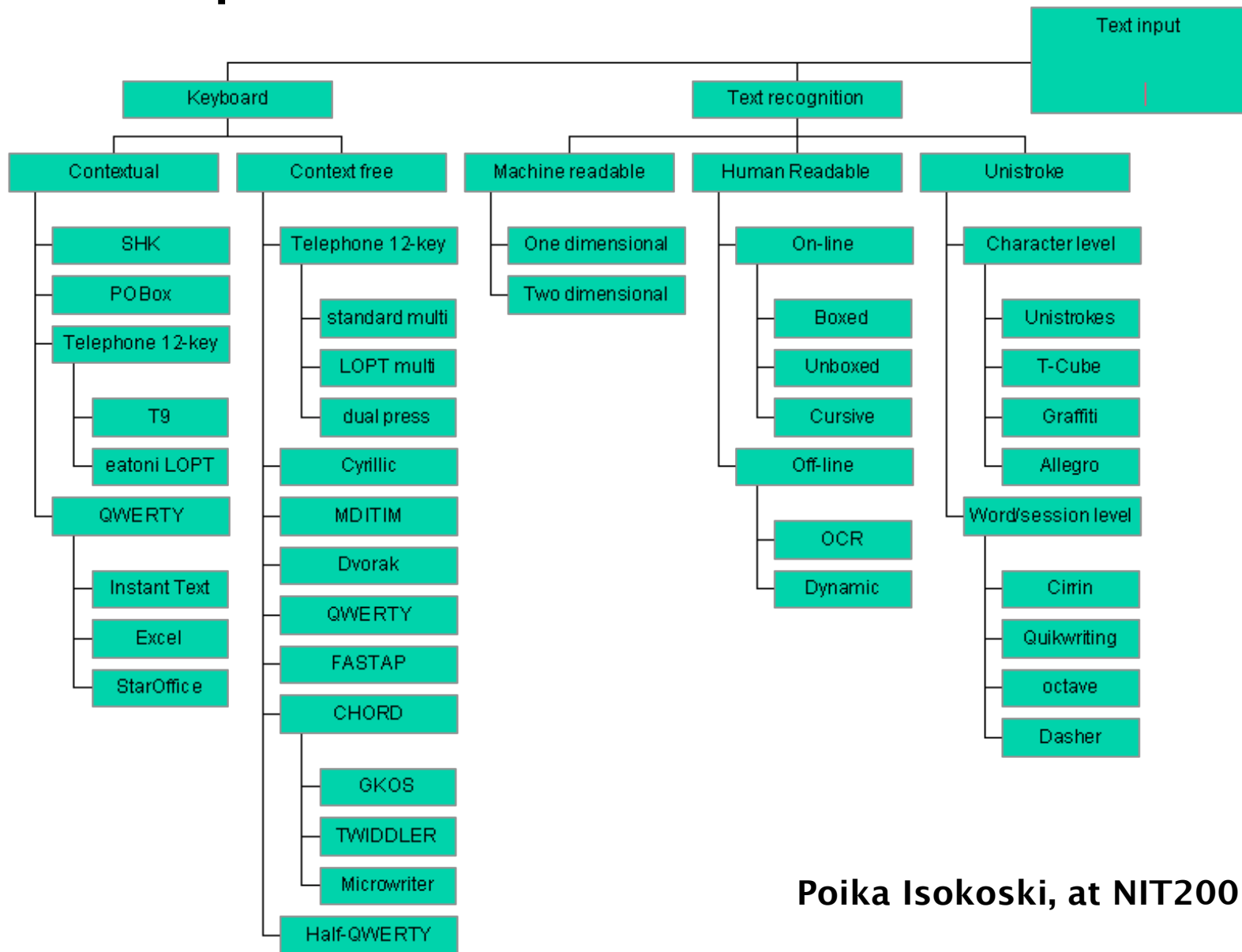
2.3 Input and Output on Mobile Devices (continued)

2.4 System Architectures for Mobile Devices

2.5 Example Applications

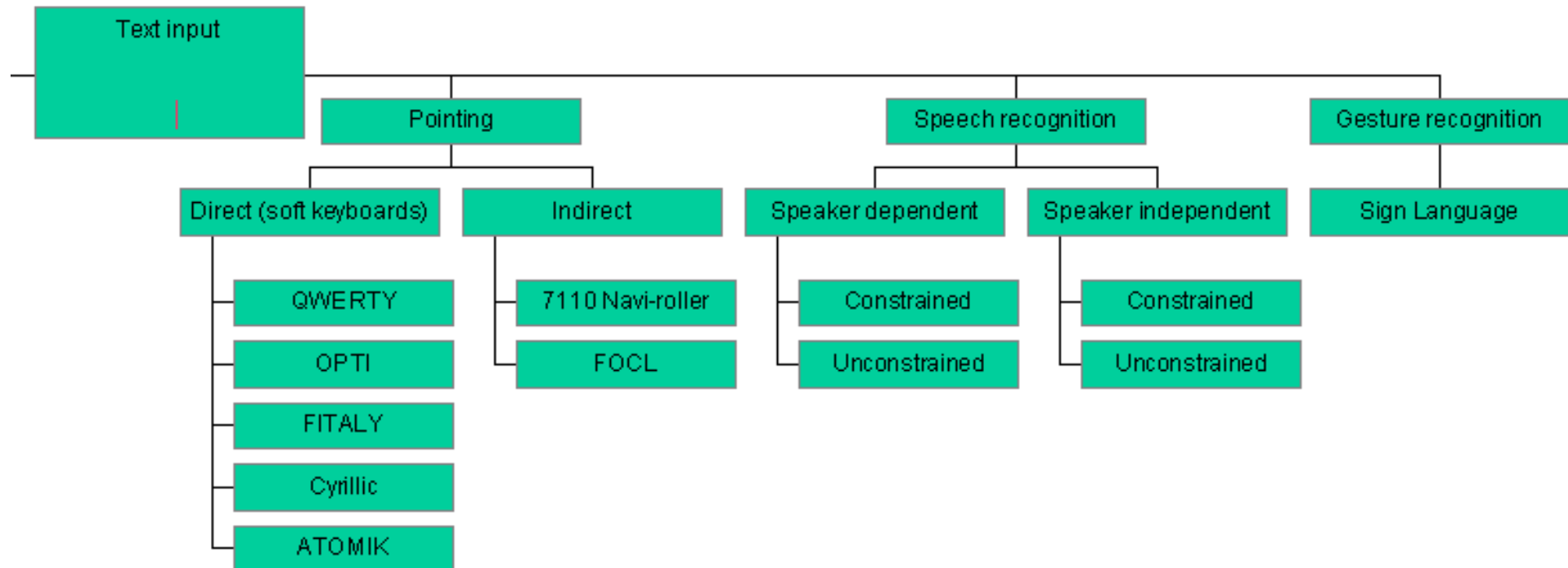
2.6 HCI and Ubiquitous Computing

Text Input Methods



Poika Isokoski, at NIT2001, 12.2.2001

Text Input Methods



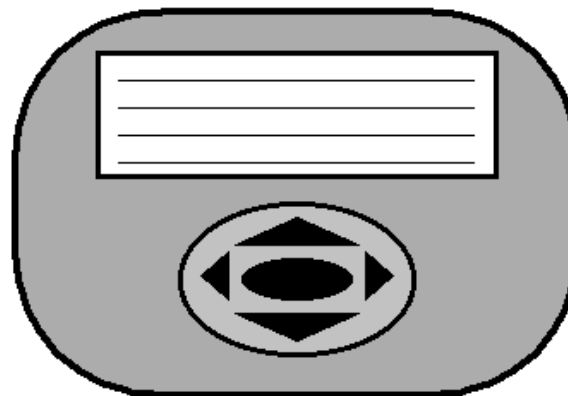
Poika Isokoski, at NIT2001, 12.2.2001

Example: FOCL

A Probabilistic Character Layout Strategy for Mobile Text Entry

Only 5 buttons for text entry

Character layout changes with letter frequency



A	NRTSLDIYCVMBGCPUFWXZEJABOQ
B	EUGALRIJSBYTMVWVDFGKHPQXZ
C	HAOEKTLRIUCYSNQDFGJMPVWXZ
D	EIOASRUVDLGVFTNWBJMCHKPQXZ
E	RNDASELTMYCVIWFPPXGOKHQBZJ
F	OIREATFLUSYMBDCDGHJKNPQVWXZ
G	EHORIAULSNGTYBMWDFJKPQVWXZ
H	EAIOTRUYSWLPHMBCDFGJKNPQVWXZ
I	NTSLCRDMGEVOKFPAZXBUQIHJWY
J	UOEIABCDPFGHJKLMPQRSTVWXYZ
K	EINSLYOUAMPBCDFGJKQRTVWXZ
L	ELIOADYSUTFKPMWVRGCBNEJQXZ
M	EAOIUPSBNYFLNTCDGHJKQVWXZ
N	DGETOSIYACLKNUJFBQVBRXMPZ
O	URNMTWOLSPVKDIFBACYEGZJXHQ
P	ELOARIUPSTHYWCFBDGJKMNQVWXZ
Q	UABCDEFGHIJKLMPQRSTVWXYZ
R	EOSIATYDNULRMGKCVHFBPWJQXZ
S	TEHIOAUSPKLCYWMNBQGFJRVXZ
T	HEOAI SRUYWLCBGNDFMZJKPQVX
U	TRSNLGCMEIBPDAFYQZQXKJUVW
V	EIOAYUBCDFGHIJKLMPQRSTVWXZ
W	AHIEONRSLDTBUKCFGJMPQVWXZ
X	TPAICEUOBDPFGHJKLMPQRSTVWXZ
Y	OESIPABLTWGHMUCDFJKNPQRVWXZ
Z	EELASYIOBCDFGHIJKLMPQRSTUVWX
-	TANSHFBMCOLPDRNGYEIKUVJQXZ

Figure 2. Alphabet orderings in order of likelihood to follow each letter or the space character

See also the Crawler system for 1 button entry

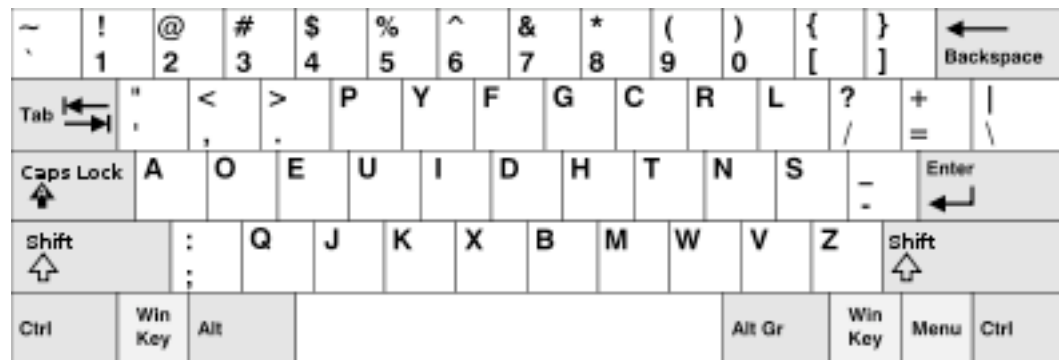
<http://www.chiark.greenend.org.uk/~naath/>

<http://www.yorku.ca/mack/GI98.html>

Example: Dvorak Keyboard

E.g. http://en.wikipedia.org/wiki/Dvorak_Simplified_Keyboard

Different key layout based on key frequencies and ergonomics
Faster but due to learning involved never replaced QUERTY



Example: Speedscript

Pen-based text entry

Part of the layout changes to reflect letter frequencies

F	G	W	B
L	C	AU	P
N	T	R	M
D	S	H	XZ
Q	K	V	J

F			
L	a		u
N	e	i	o
D	S	H	XZ
Q	K	V	J

F	G	W	B
L	C	AU	P
N	T	R	M
D	S	H	XZ
Q	K	V	J

F	G		B
L			
N	a	r	u
D	e	i	o
Q	K	V	J

x	z	
a	y	u
e	i	o

<http://www.speedscript.biz/?p=functionality>

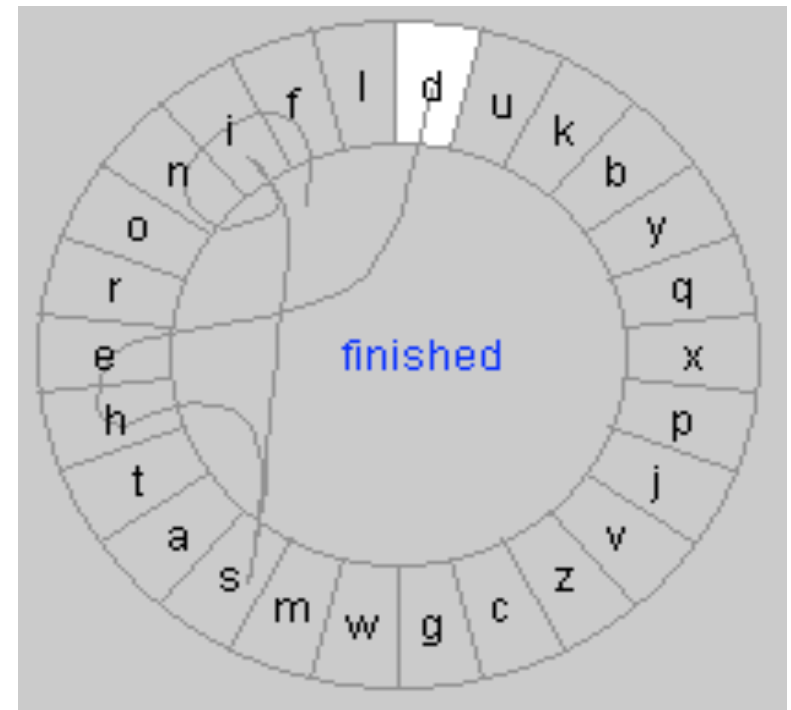
Example: Quikwriting

- <http://mrl.nyu.edu/projects/quikwriting/>
- Authors claim “Quikwriting is significantly faster and less stressful to use than Graffiti”

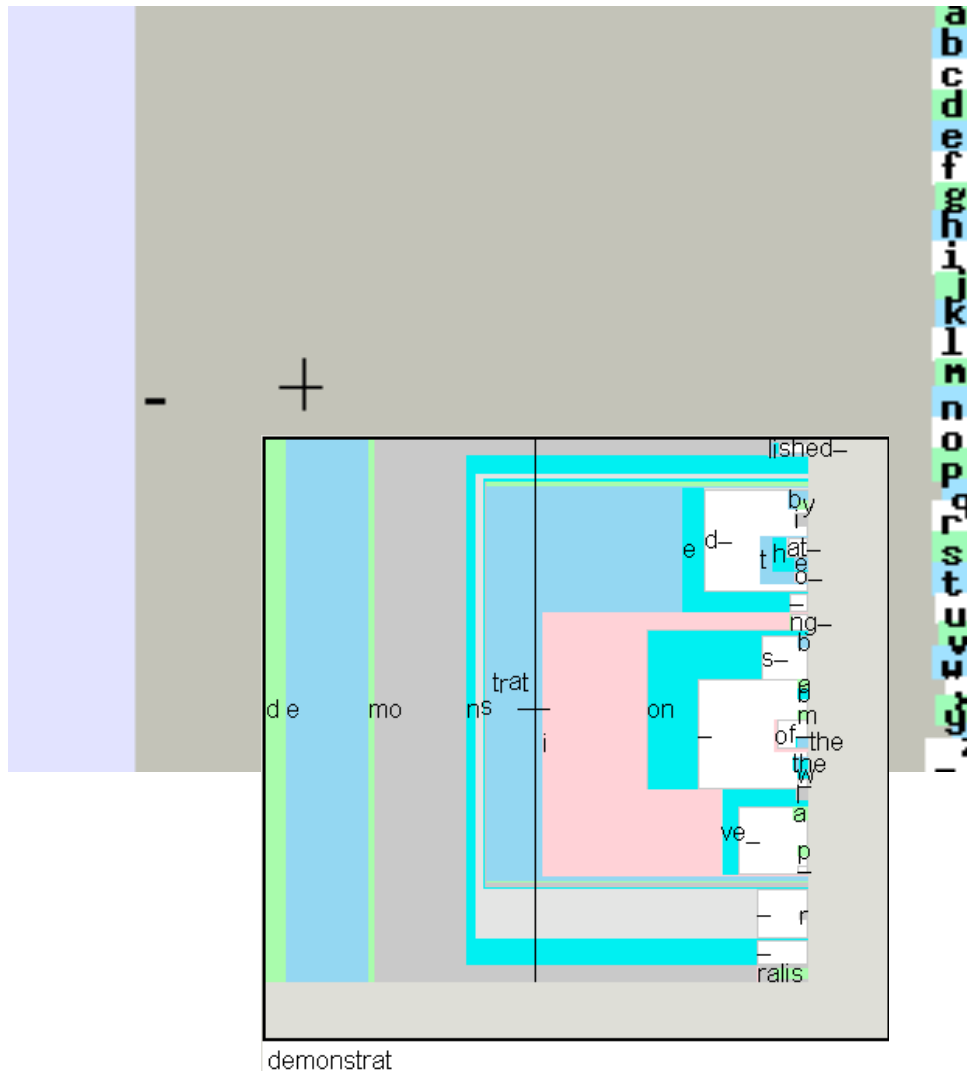


Example: Cirrin - (the CIRcular INput device)

- A word-level unistroke keyboard is a soft keyboard allowing a user to go from any key to any other key without lifting the pen or entering unwanted keys
- Jennifer Mankoff and Gregory D. Abowd.
Cirrin: A word-level unistroke keyboard for pen input.
In *Proceedings of UIST '98*. Technical note. pp. 213-214



Example: Dasher



• Dasher is a data entry interface incorporating language modelling and driven by continuous two-dimensional gestures.

• “Tests have shown that, after an hour of practice, novice users reach a writing speed of about 20 words per minute while taking dictation. Experienced users achieve writing speeds of about 34 words per minute, compared with typical ten-finger keyboard typing of 40-60 words per minute.”

• <http://www.inference.phy.cam.ac.uk/dasher/>

Yoyo Input Device designed for arctic environments

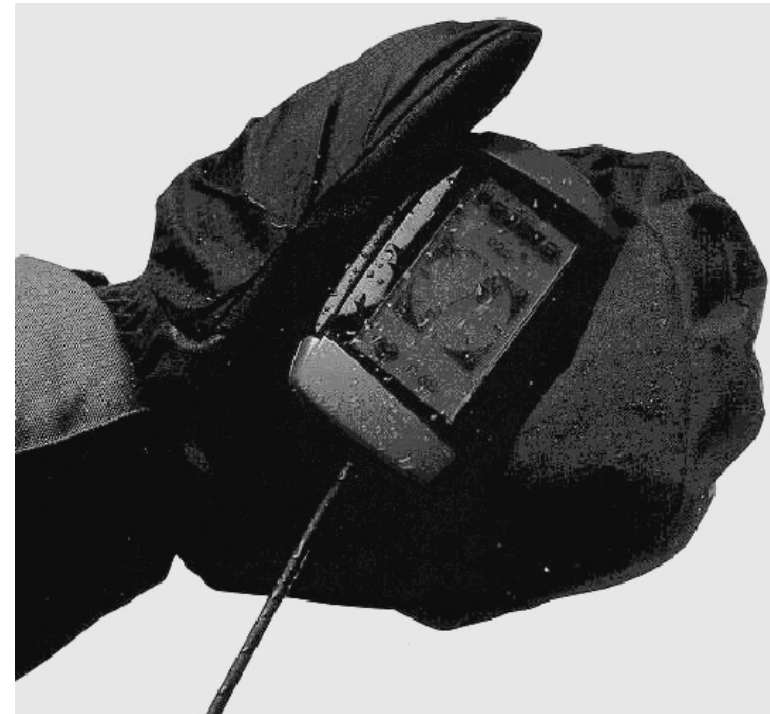


Figure 5. The Yo-Yo user interface.

- Smart Clothing for the Arctic Environment by J. Rantanen et al. in proceedings of the int. Symposium on Wearable Computing 2000 (ISWC2000)

Output – What and How to Present?

- Text
- Non-speech Audio
- Music
- Speech
- Images
- Video

- Tactile feedback (e.g. vibration alarm)

Head-up Displays

- Images in front of the eye
- Appears like free floating
- See through

<http://www.microopticalcorp.com>



Haptic feedback

Application in Pedestrian guidance

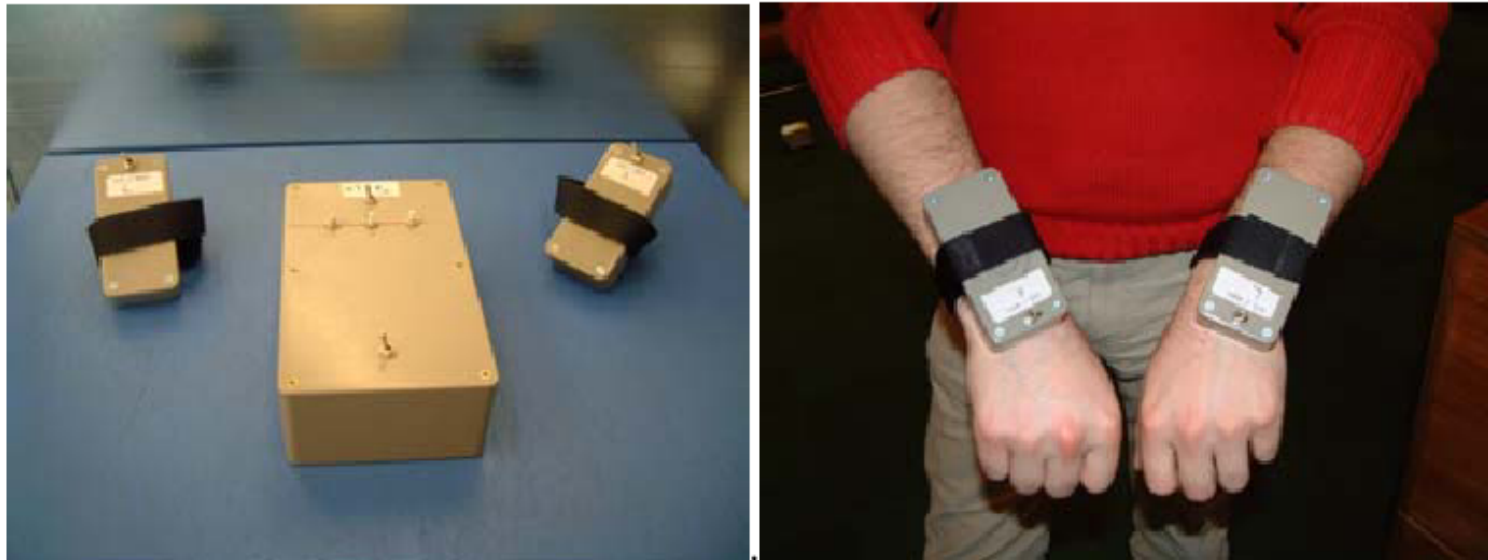
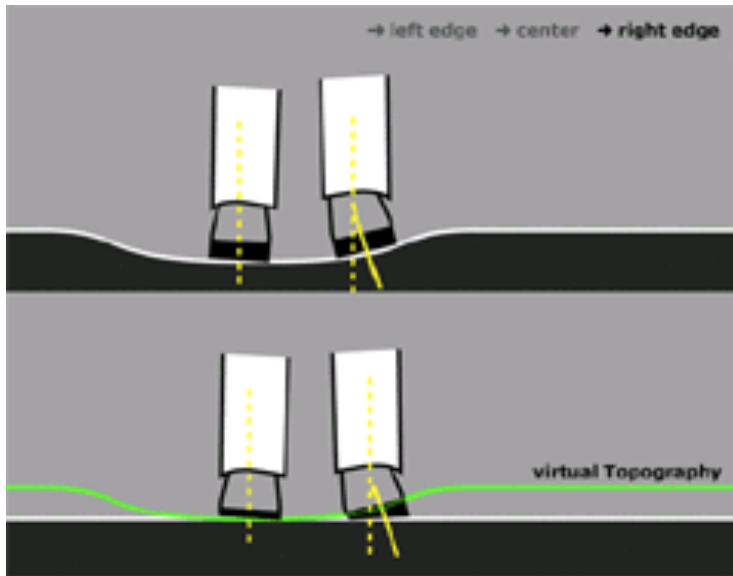


Fig. 1. (a) GentleGuide control unit and wrist devices (b) GentleGuide worn by a participant

- **GentleGuide: An exploration of haptic output for indoors pedestrian guidance** S.Bosman, B.Groenendaal, J.W.Findlater, T.Visser, M.de Graaf & P.Markopoulos. Mobile HCI 2003. Udine

Navigation Shoes

- Unconscious clues



<http://www.digital.udk-berlin.de/en/projects/summer05/main/freeproject/cabboots.html>

1. Mobile and Ubiquitous User Interfaces

2.1 Mobile Computing

2.2 Design Guidelines for Mobile Devices

2.3 Input and Output on Mobile Devices

2.4 System Architectures for Mobile Devices

2.5 Example Applications

2.6 HCI and Ubiquitous Computing

Literature:

- Scott Weiss: Handheld Usability, Wiley 2002
- Ch. Lindholm/ T. Keinonen/ H. Kiljander: Mobile Usability, McGraw-Hill 2003
- Nokia Series 60 Usability Guidelines,
<http://forum.nokia.com>

Application Types for Mobile Devices

- Supporting functions for telephony
 - Phone book, texting, service setup
- Personal information storage
 - Contacts, calendar, notes, ...
- Multimedia players
- Generic information services
 - Internet access, WAP, i-mode, ...
- Enterprise applications
 - E-Commerce
 - Supporting mobile workforce
- Games
 - Stand-alone/networked, virtual/physical
- Utilities and productivity applications
 - Calculator, alarm clock etc.
 - Data transfer, synchronization

WAP (Wireless Application Protocol)

- Set of protocols and languages to support Internet access on devices with limited computing & display abilities and low access speed
 - Versions 1.x and 2.0
 - WAP Forum now consolidated into Open Mobile Alliance (OMA)
 - » <http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html>
- Communication Client - WAP gateway in compiled (compressed) format
 - Over various carrier technologies, e.g. GPRS or HSDPA
- Wireless Markup Language (WML)
 - XML-based page description language
 - Reduced version of XHTML
 - Card deck metaphor: site = deck, page = card

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="no1" title="Card 1">
<p>Hello World!</p>
</card>
</wml>
```

```
----- Card 1 -----
Hello World!
```

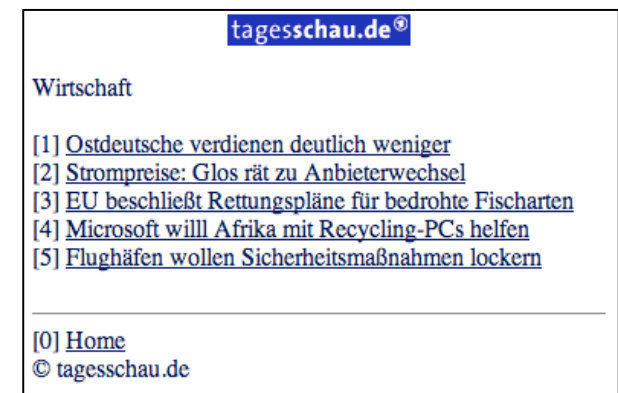
CHTML (Compact HTML)

- Reduced version of HTML targeted at mobile devices
 - Defined in W3C Note (1998)
- Not based on XML, but HTML-like syntax
 - Compatibility with HTML Web sites
- Does not support e.g.:
 - JPEG files, tables, frames, image maps, multiple fonts, style sheets
 - But supports GIF
- Easily operated by four buttons:
 - Cursor forward, cursor backward, select, back/stop
- Current trend:
 - Modern mobile browsers tend to render large part of HTML

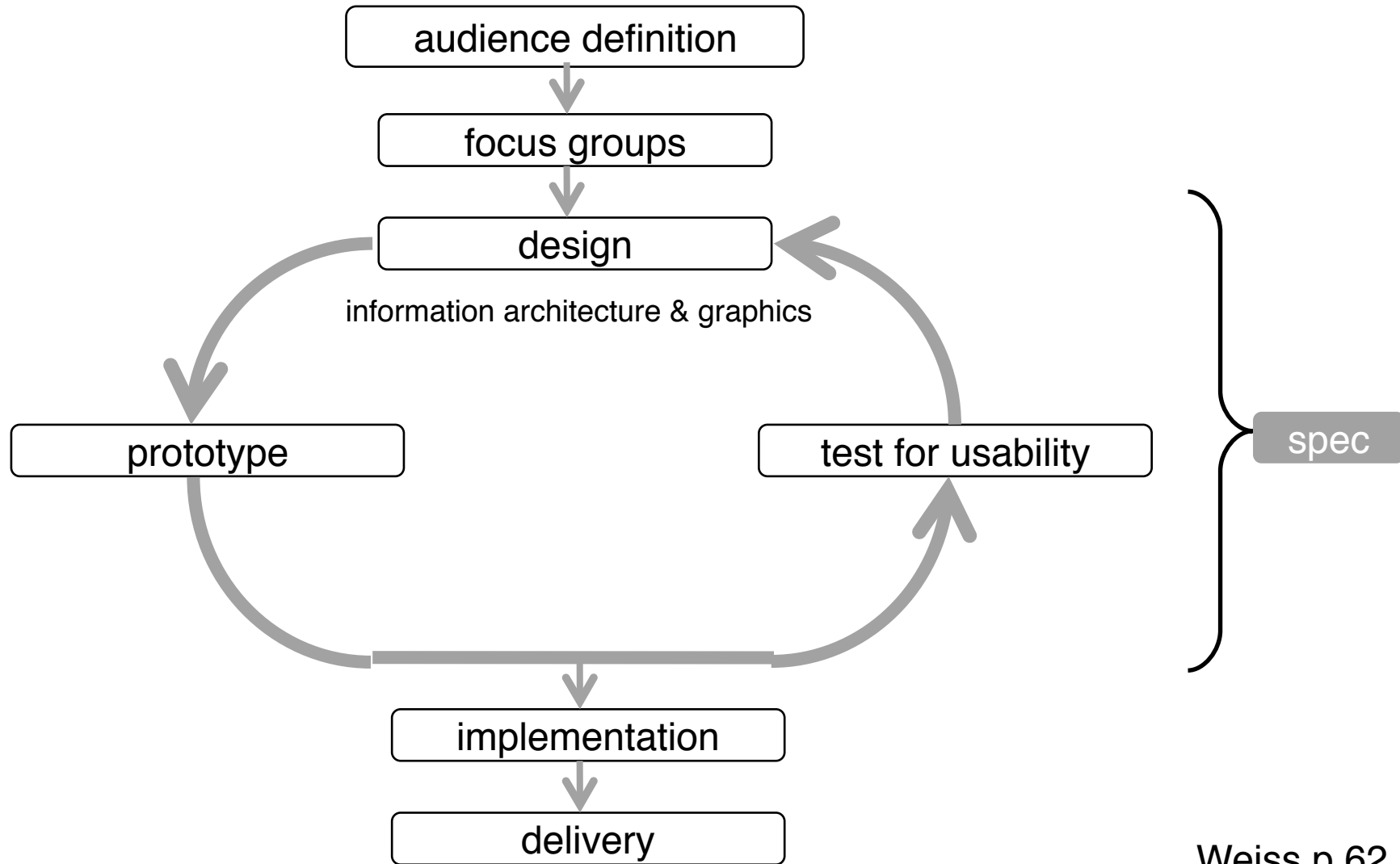


i-mode

- Portal service for mobile phones
 - Pages defined in CHTML (more precisely, iHTML)
 - Colour graphics, video, sound: Standard formats
 - Requires specific i-mode compatible handset
 - Packet-switched data transfer, “always on”
- Developed by NTT DoCoMo, Japan (1999)
 - Most successful mobile data service worldwide
 - » 2006: 47 million users in Japan
 - Offered also in other countries (Germany: e-Plus, stopped April 2008), being phased out due to lack of customers



Typical Design Cycle



Weiss p 62

Audience Descriptions (1)

- Audiences are described descriptively and by ranges of traits
- *Early adopter*
 - Might acquire technology for the sake of trying something new
- *Novice and savvy*
 - Novice: either little time of usage or low level of proficiency
 - Border novice/savvy may change with version updates
- *Internet savvy*
 - Time spent on the Web, ability to define specific terms
- *Teenager or senior*
- *Phone, PDA, and/or pager savvy*
- *Everyday consumer*
 - Amount spent online/in shops, # of catalogs browsed, income, ...
- *Physically challenged*

Weiss p 72 ff

Audience Descriptions (2)

- *Complex application user*
 - E.g. experience from home banking, spreadsheet usage
- *Online transaction user*
 - E.g. number of items purchased online per month
- *Credit user*
- *Enthusiast*
 - ...of anything
- *Commuter*
 - Transport means, distance, frequency, number of travellers, ...
- *High-income and high net worth*
 - “Einkommen” und “Vermögen”
- *Geo-specific*
 - Specific location
 - Multi-unit house vs. single-family home, ...

Weiss p 72 ff

Key Design Guidelines for Handheld Devices

- **Design for users on the go**
 - Handhelds are likely to be used in distracting situations
 - Wireless users typically have immediate goals
- Examples:

Desktop	Handheld (wireless)
Comparing prices of flights, making reservations	Checking status of particular flight
Gathering background on a company	Getting drive directions to specific office of the company - on the road
Researching a medical condition and its consequences	Monitoring a medical condition
Reading a movie/theatre review	Purchasing a cinema ticket to avoid the line

Weiss

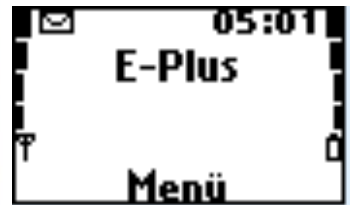
Key Design Guidelines for Handheld Devices

- Select vs. Type
 - Avoid text entry as far as possible
 - Support all available text input methods
 - Exceptions:
 - » Text input may be still adequate in specific cases!
- Be aware it is a phone
(of course only if it is, in fact)!
 - Any activity may at any point in time be interrupted
 - Main source for interruption = incoming call
 - Applications have to be able to suspend and resume

Example: Nokia Interface Evolution



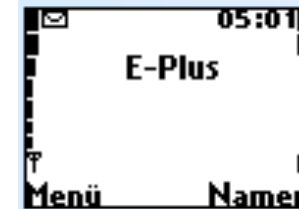
3330



Series 20
"NaviKey"



6610



Series 30



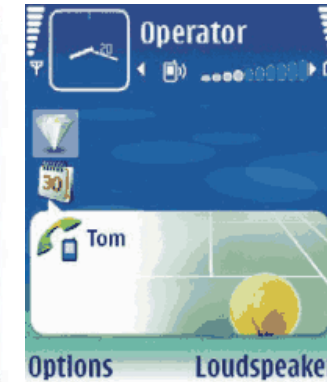
6230i



Series 40



3250

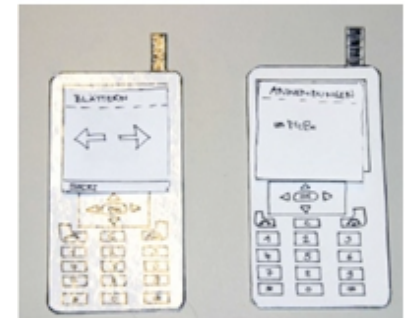
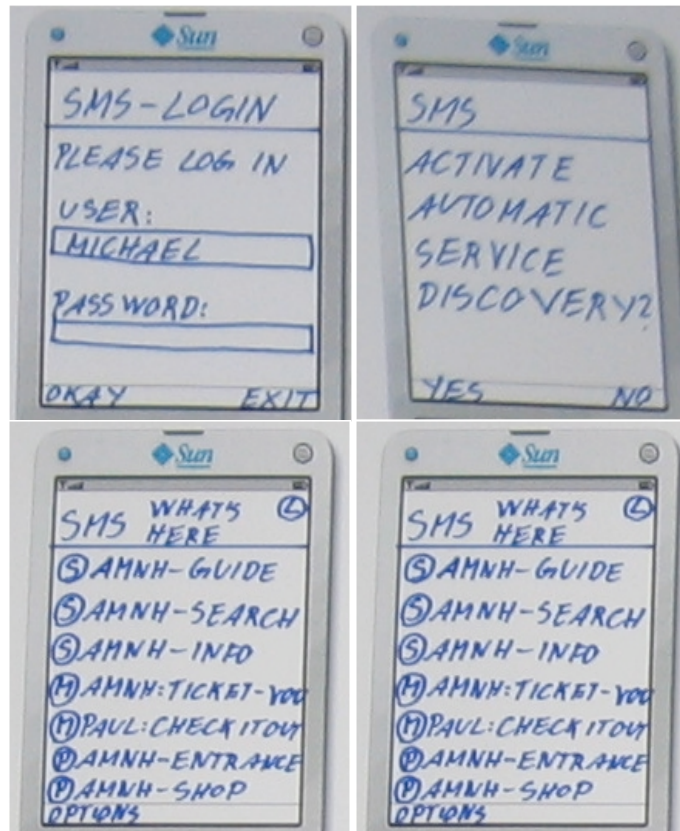


Series 60

nokiaport.de

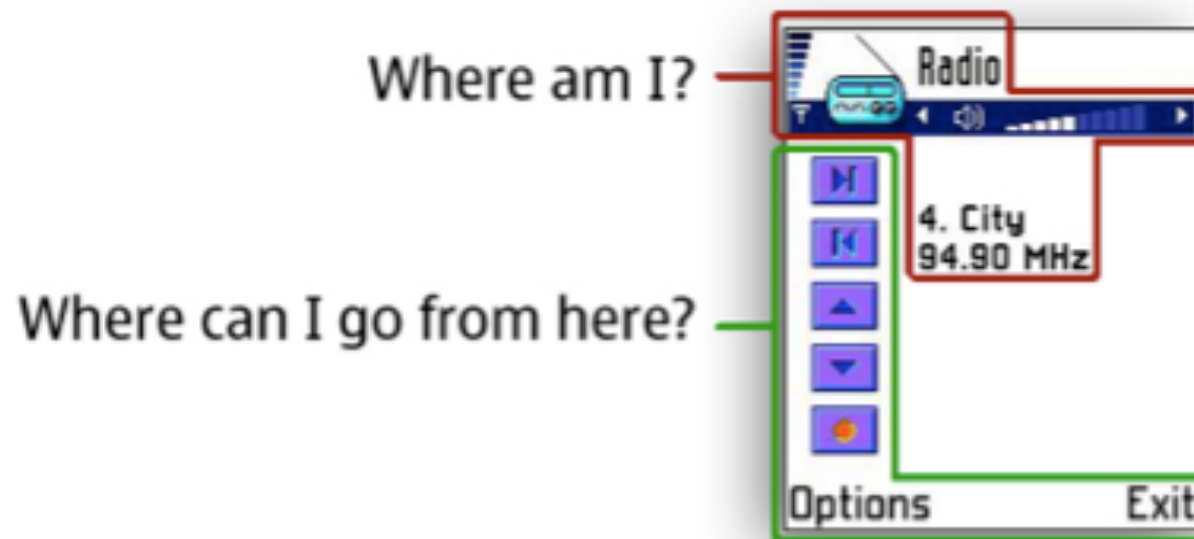
Application (Interface) Design

- Paper prototyping
 - Based on scenarios or more detailed maps
 - Using “blinders” (template resembling shape of device)



Excerpts from Series 60 Guidelines

- The key questions to be answered in navigation



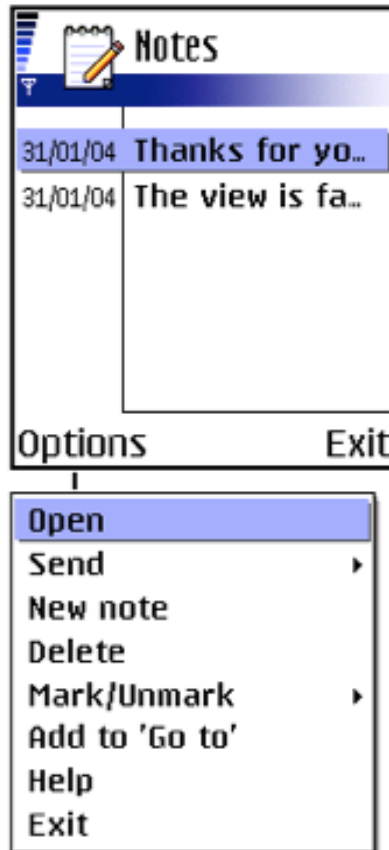
Series 60 Symbian phones = industry standard going beyond just Nokia

Excerpts from Series 60 Guidelines

- Main menu styles



No focus



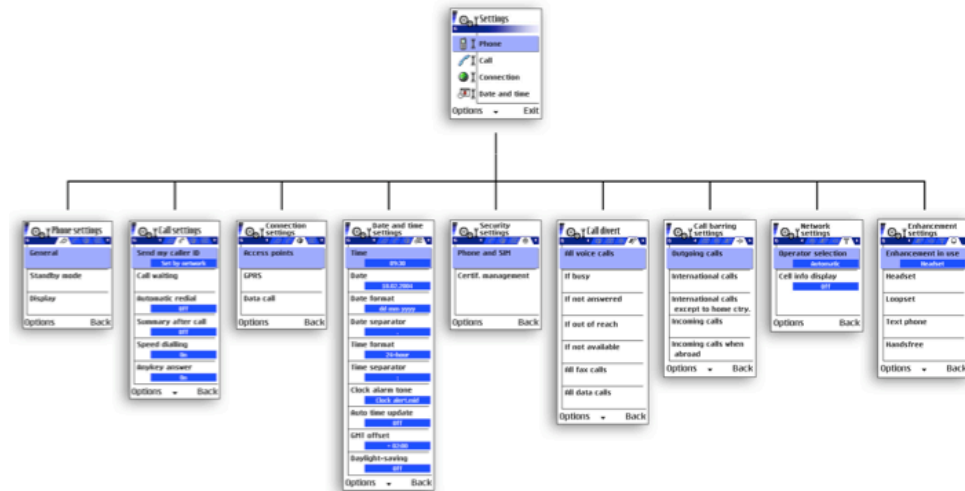
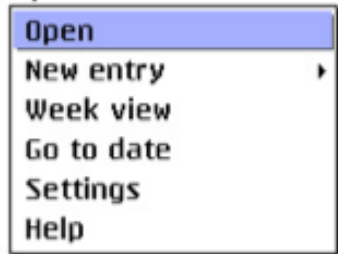
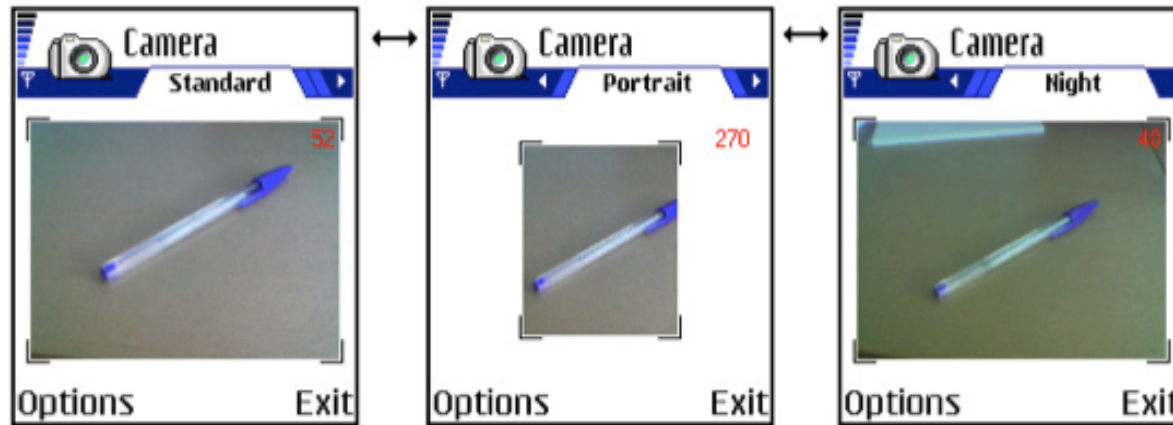
Using focus



Custom main menu (games)

Excerpts from Series 60 Guidelines

- Using tabs



Usability and Mobile Gaming

- Fun is a main factor for game usability
- Mobile games are typically played for brief time periods, so there is no extra time to learn how to navigate inside the game.
- Playing should be as intuitive as possible and the challenge should be in the game play, not in the interaction with the game user interface.
- Usability provides the framework and tools for playability
- The interface is the essential factor for a game's success
- If usability problems get in the way of intense game playing, the game probably will not be played again.

From: Tip Of The Month: Usability And Fun – Best Friends In Mobile Games (Nokia Forum)

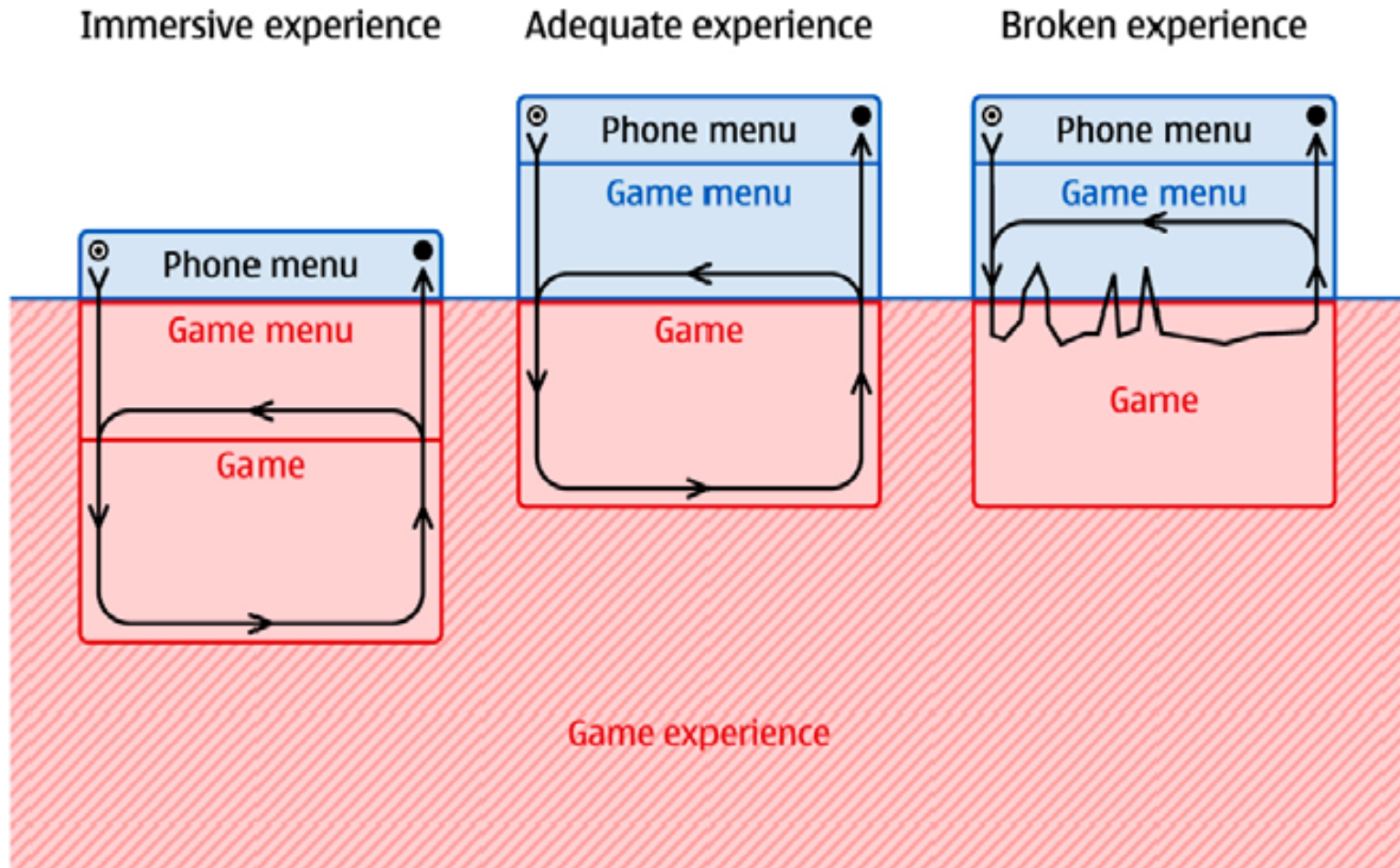
Basic Design Issues for Games

- When playing a game, users should experience the game world
- The game navigation structure should support the experience
- Use of high-level UI components should be avoided
- Game menus should look and feel like the game.

- Mobile games are played in a context where interruptions often occur:
 - somebody might call or send an SMS message,
 - the player might need to pause the game to buy a bus ticket.
 - Therefore, the game design should support saving and pausing.

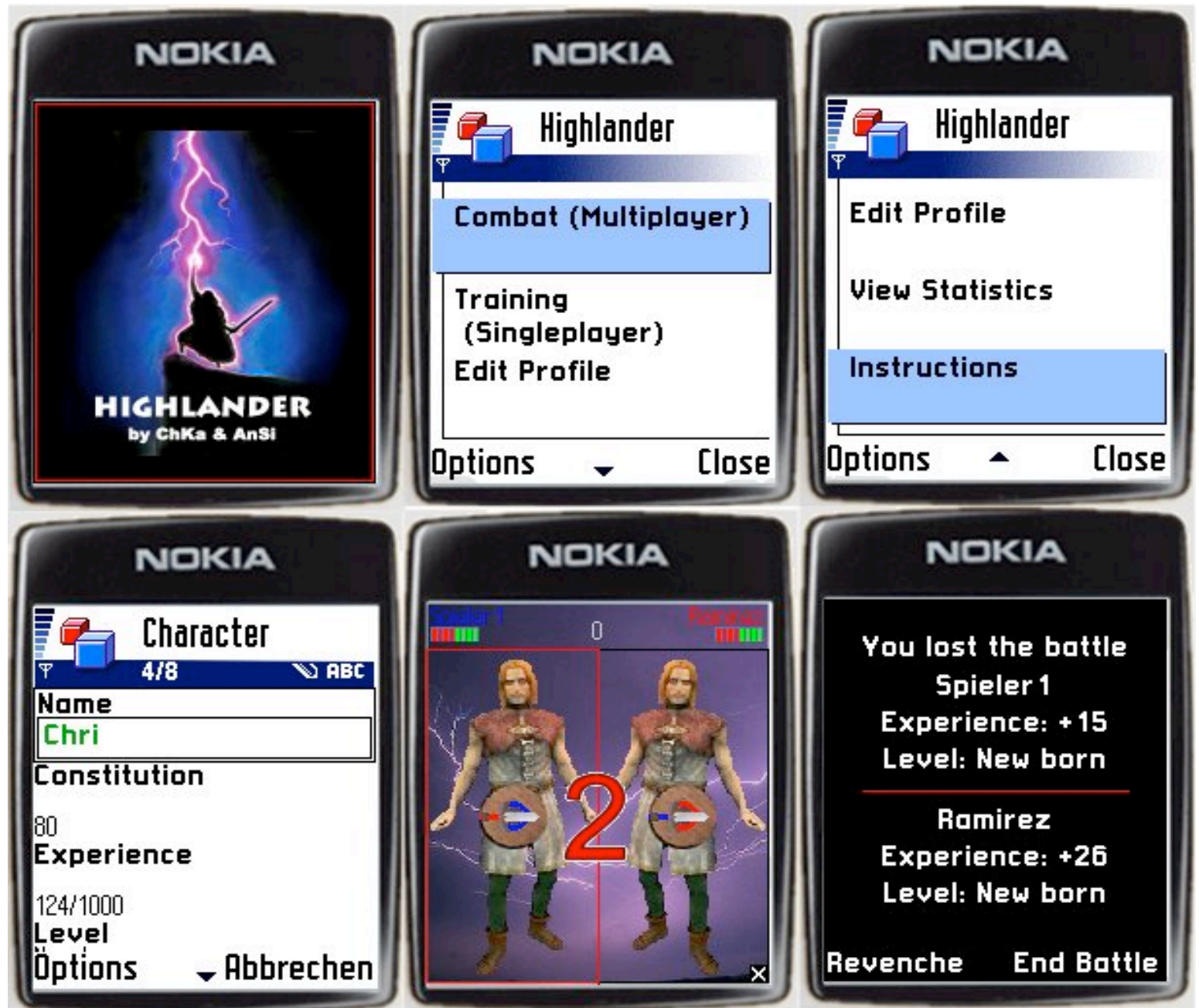
From: Tip Of The Month: Usability And Fun – Best Friends In Mobile Games (Nokia Forum)

Experience



From: Tip Of The Month: Usability And Fun – Best Friends In Mobile Games (Nokia Forum)

- Example
- Game vs. Menus



2 Mobile and Ubiquitous User Interfaces

2.1 Mobile Computing

2.2 Design Guidelines for Mobile Devices

2.3 Input and Output on Mobile Devices

2.4 System Architectures for Mobile Devices

2.5 Example Applications

2.6 HCI and Ubiquitous Computing

Developing Applications for Mobile Devices

Devices: Basic Phone, Extended Phone, Smartphone, PDA, Notebook

Platforms (Mobile Phone, Smartphone)

- Platform specific:
 - Symbian OS (C++, OPL)
 - Palm OS (C++)
 - Pocket PC
 - Vendor-specific
- Platform independent: J2ME (Java 2 Platform, Micro Edition)
 - Supported by Motorola, Nokia, Panasonic, Samsung, Sharp, SonyEricsson, Toshiba, etc.
- Android (Google, Open Handset Alliance)
 - technically Java, but not called Java for legal reasons
 - HTC, NTT DoCoMo, LG Electronics, Sprint, Motorola, T-Mobile, Samsung; eBay, Intel, Nvidia etc.

Java on mobile devices: History

1990: Java started as an internal project at Sun Microsystems

1995: Initial release of JDK 1.0 (applets / servlets)

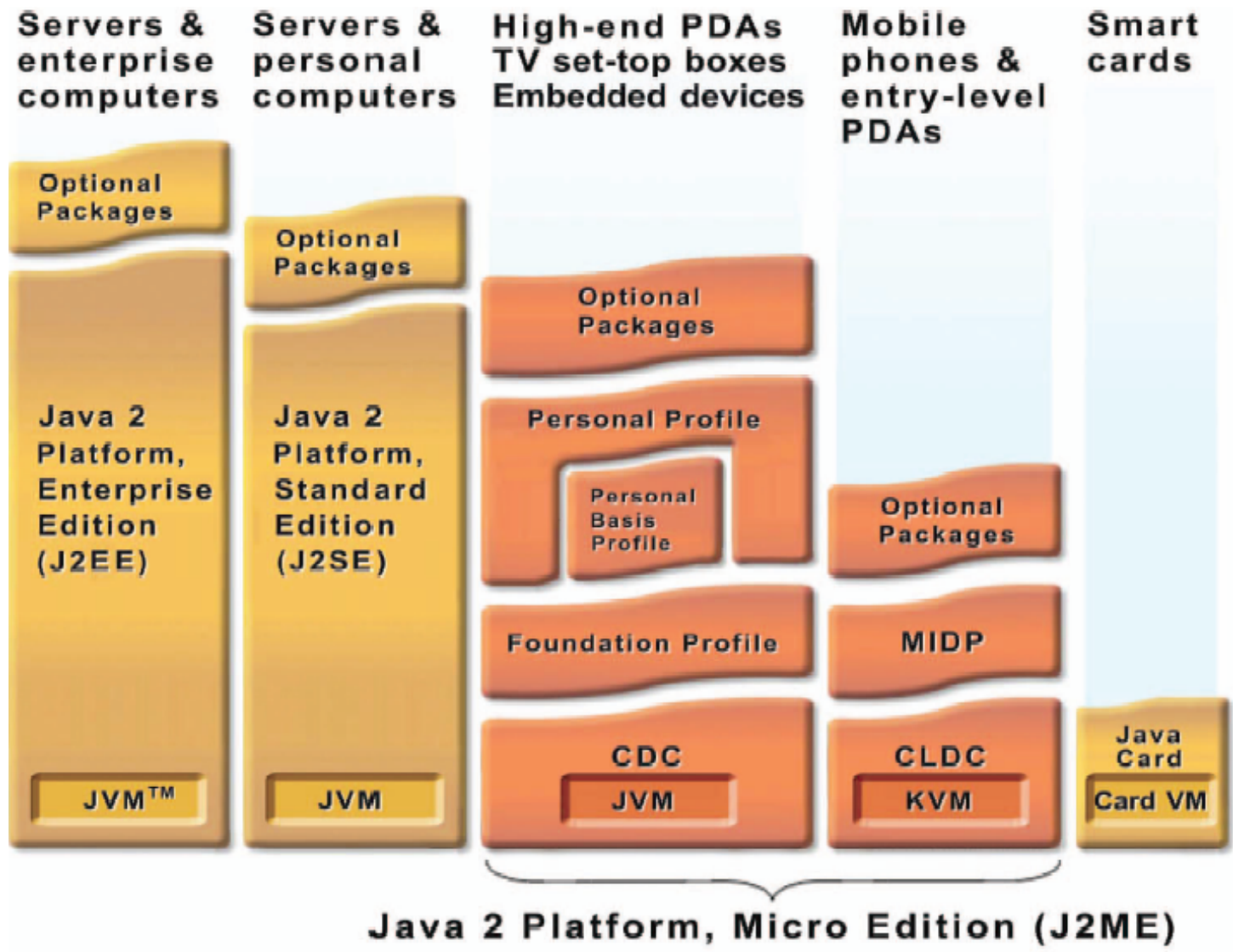
1999: JavaOne conference

– Subdivision of Java in

- Java 2 Enterprise Edition (J2EE)
- Java 2 Standard Edition (J2SE)
- Java 2 Micro Edition (J2ME, now called Java ME)
(successor of Personal Java and Embedded Java)

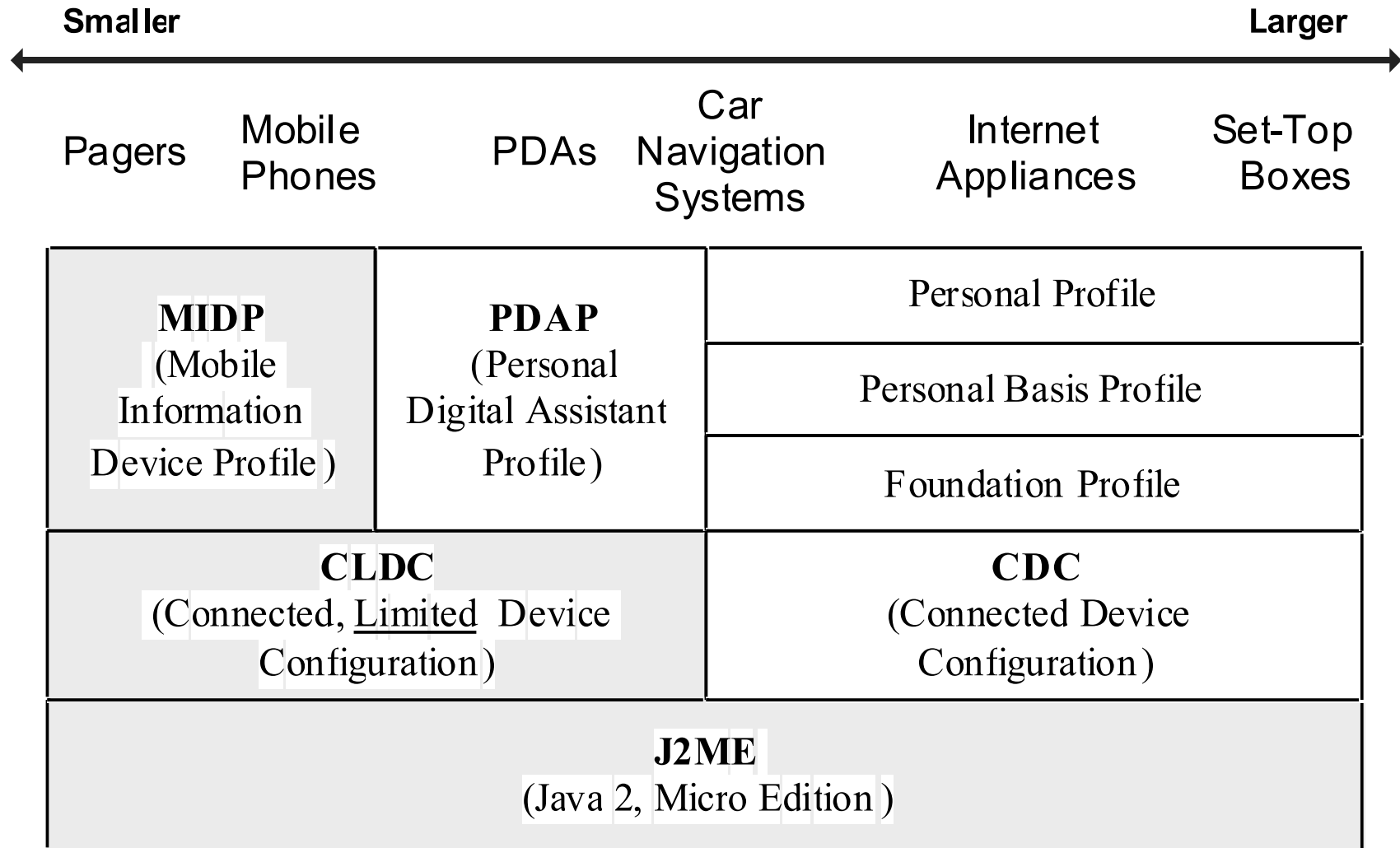
2000/01 First mobile phones with support for J2ME

The Java universe



KVM =
Kilobyte
VM

The J2ME Universe



MIDlet

MIDP applications are called MIDlets

- Several MIDlets can be combined into MIDlet suite

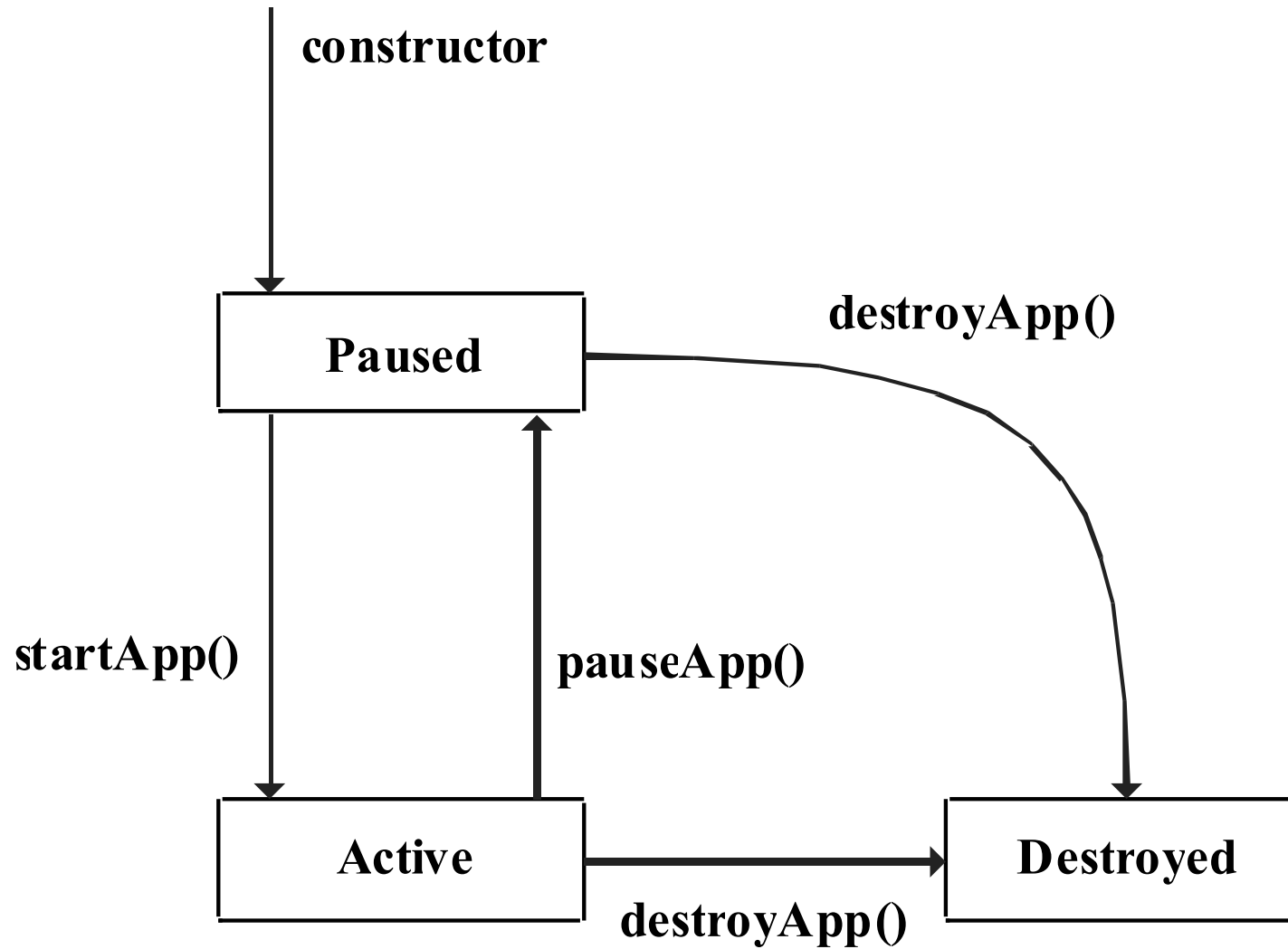
Every MIDlet is an instance of `javax.microedition.midlet.MIDlet`

- No argument constructor
- Implements life-cycle methods

Conceptually similar to Applets

- Can be downloaded
- Executed in host environment

MIDlet (MIDP Application): Life Cycle



MIDP User Interface: Methodology

Abstraction (-> Preferred Method)

- Specifying a user interface in abstract terms
- (Not:) “Display the word ‘Next’ on the screen above the soft button.”
- Rather: “Give me a Next command somewhere in this interface”

Discovery (-> Games)

- Application learns about the device + tailors the user interface programmatically
- Screen size -> Scaling

<http://developers.sun.com/mobility/midp/chapters/j2meknudsen2/Chap5.pdf>

MIDP User Interface: Simple Example

```
public class Commander extends MIDlet {  
  
    public void startApp() {  
        Displayable d = new TextBox("TextBox", "Commander", 20, TextField.ANY);  
        Command c = new Command("Exit", Command.EXIT, 0);  
        d.addCommand(c);  
        d.setCommandListener(new CommandListener() {  
            public void commandAction(Command c, Displayable s) {  
                notifyDestroyed();  
            }  
        });  
  
        Display.getDisplay(this).setCurrent(d);  
    }  
  
    public void pauseApp() {}  
  
    public void destroyApp(boolean unconditional) {}  
}
```



Connecting to the World

Generic Connection Framework

Extremely flexible API for network connections

Contained in `javax.microedition.io`

Classes based on Connection interface

- `URLConnection` (Get / Post) / `HttpsURLConnection`
- `SocketConnection`
- `ServerSocketConnection` (Responding to incoming connections)
- `SecureConnection` (TLS or SSL socket)
- `CommConnection` (SerialPort)
- `DatagramConnection` (UDP `DatagramConnection`)

MMAPI (Sound, Music, Video)

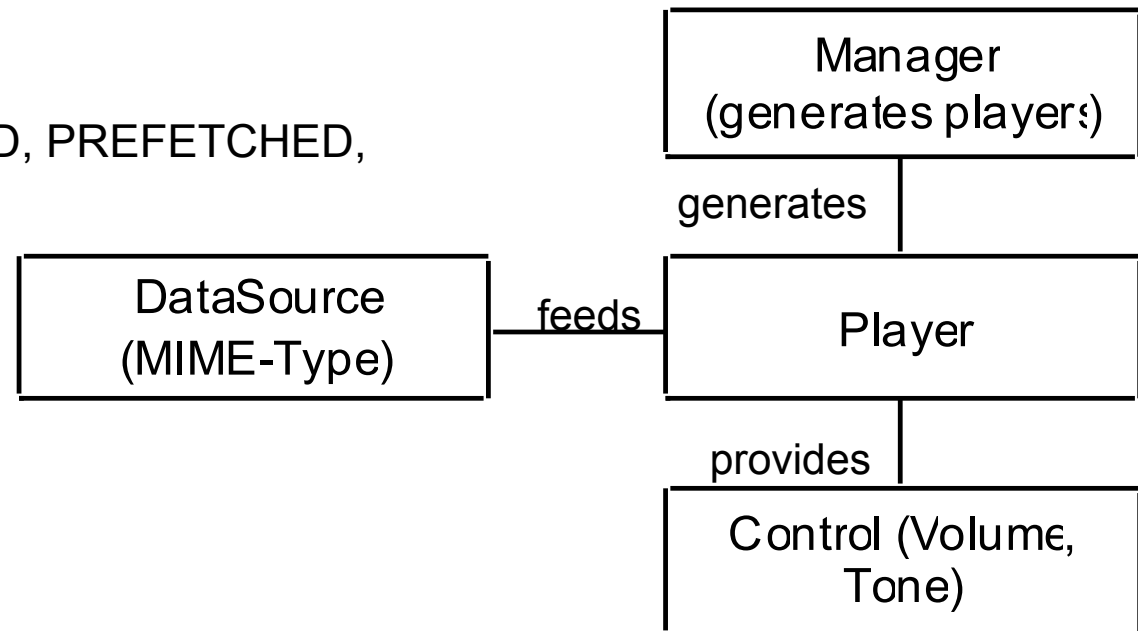
Mobile Media API – similar to JMF (Java Media Framework)

General API for multimedia rendering and recording

ABB (Audio Building Block) – plays simple tones (MIDI - note, duration, volume) and sampled audio (wav, mp3)

Player lifecycle:

- States
UNREALIZED, REALIZED, PREFETCHED,
STARTED, CLOSED
- Methods
realize(),
prefetch(),
start(),
stop(),
deallocate(),
close()



<http://developers.sun.com/mobility/midp/articles/mmapioverview/>

Further APIs (Examples)

Wireless Messaging API (JSR-120)

Mobile Media API (JSR-135)

Bluetooth API (JSR-82 no OBEX)

FileConnection and PIM API (JSR-75)

Mobile 3D Graphics API (JSR-184)

Location API (JSR-179)

Web Services API (JSR-172)

Advanced Multimedia Supplements (JSR-234)

Further APIs (not JSRs): kXML, kSOAP, Parsing of GPS data, etc.

Symbian Series 60 Phones



Symbian:

- Operating system for mobile devices
- Derivative of the Psion operating system EPOC
- 32-bit multitasking OS, mostly written in C++
- Dealing with calls and messages coming in during application runtime

Symbian Series 60 Phones

- Smartphone standard platform
- LG, Lenovo, Nokia, Panasonic, Samsung, ...

Software development for Series 60 phones, examples of languages:

- OPL (similar to BASIC)
- Visual Basic
- Java
- C++
- Python

http://www.onlamp.com/pub/a/onlamp/2004/09/16/symbian_programming.html

Python for Series 60 Phones

Python:

- Open Source programming language (Guido von Rossum)
- Interpreted, interactive, object-oriented

Python for Series 60 phones

- Python interpreter for Series 60 phones
- Large parts of Python standard library
- Smartphone-specific modules, e.g. GUI widgets, Bluetooth, GSM Location, SMS messaging, camera access, ...

Example:

- `import appuifw`
- `appuifw.note(u"Hello World!","info")`

<http://www.forum.nokia.com/python>

<http://www.heise.de/mobil/artikel/74083>



Adobe Flash Lite (1)

Player Flash Lite 2

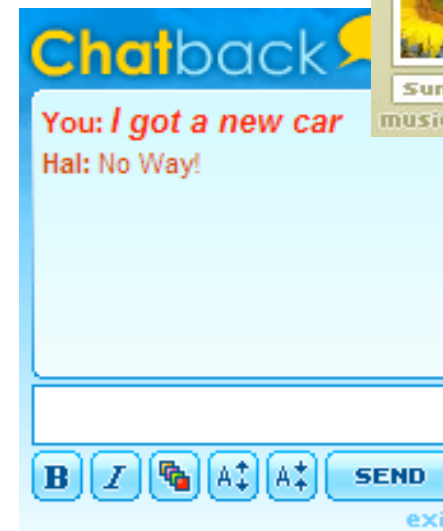
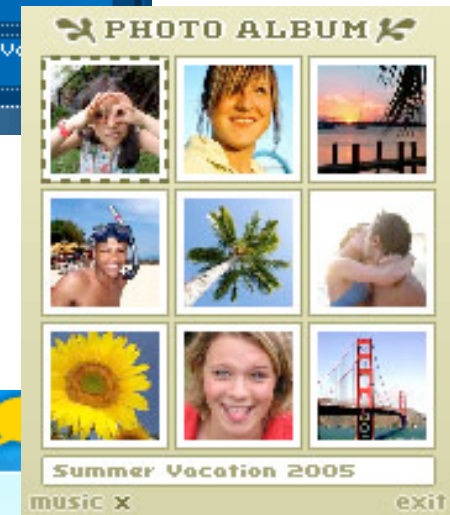
- Flash technology specifically developed for mobile phones and consumer electronic devices
- Based on Flash Player 7
- Pre-installed (Asia, Flash for i-mode)

Authoring tool:

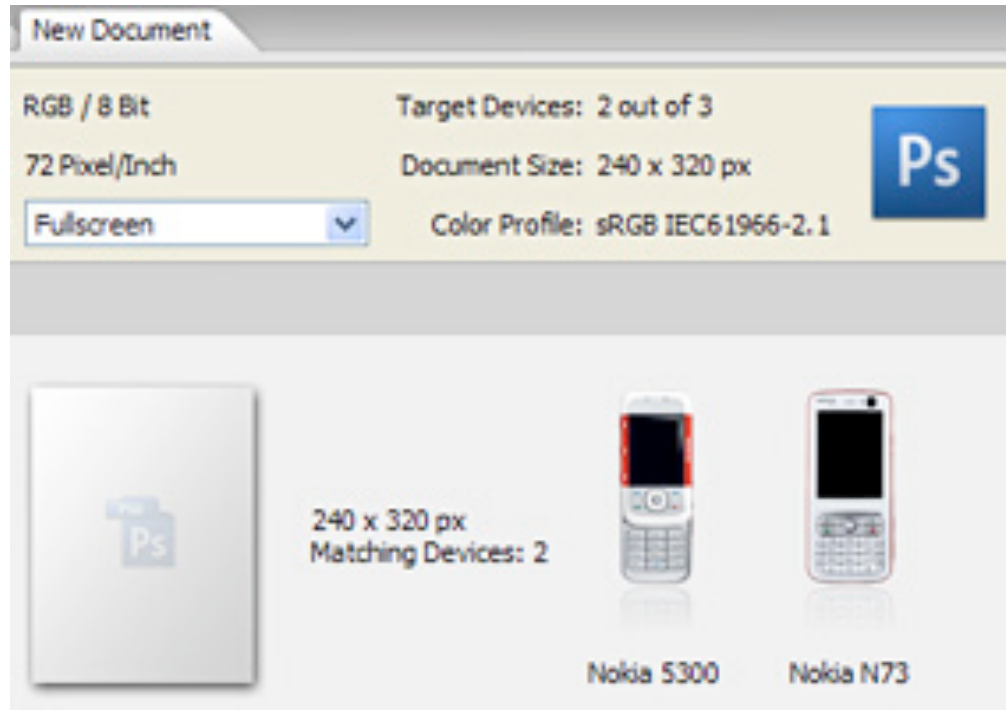
Flash Professional 8 / CS3

Example features

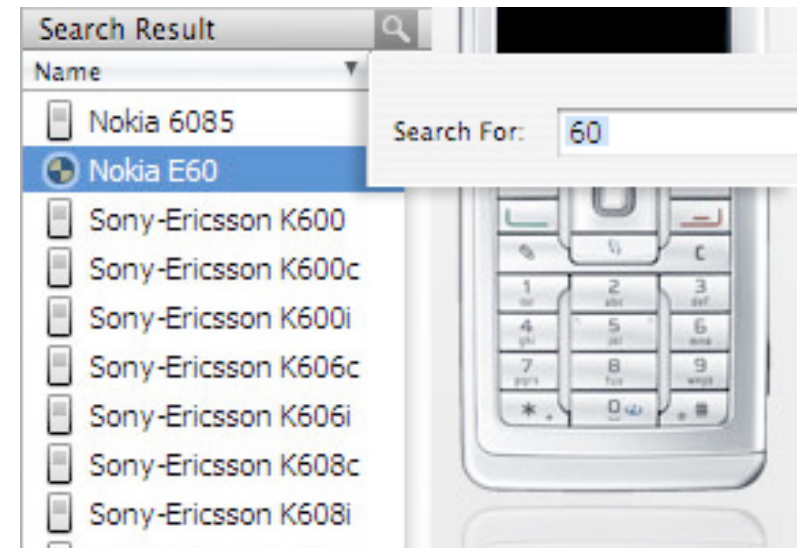
- Dynamic XML data
 - As in Flash player 7
- Dynamic multimedia
 - Loading of images, sound, video
- Text enhancement
 - User modifies text properties



Adobe Flash Lite (2)

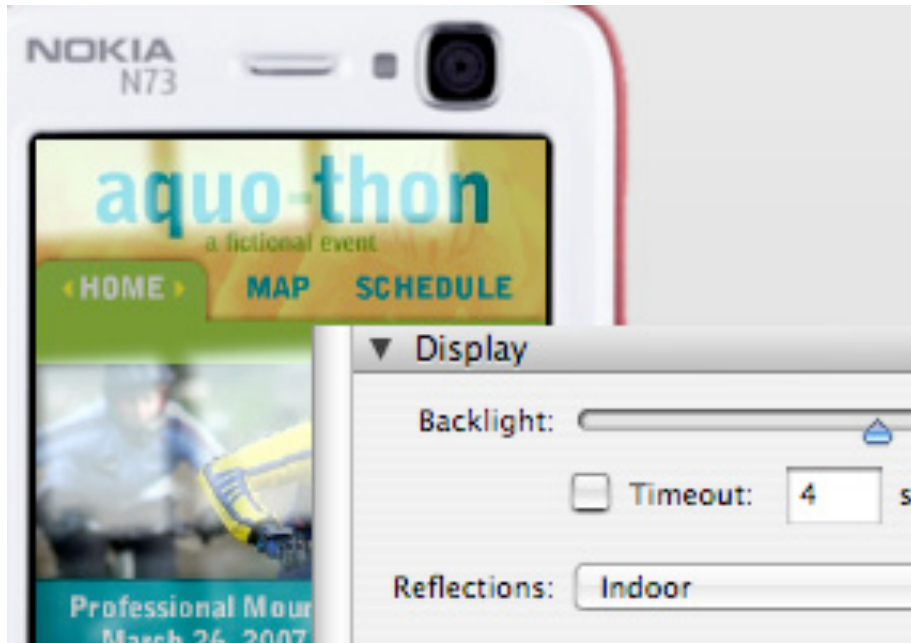


Authoring with Flash tools,
Customized for mobile devices

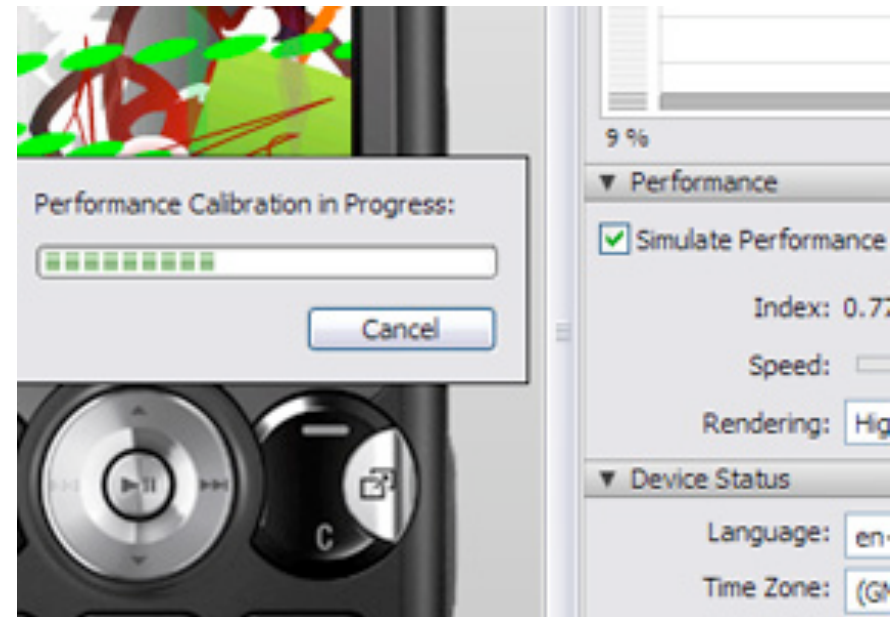


Searching a device profile

Adobe Flash Lite (3)



Emulators



Android



Created by Open Handset Alliance

- More than 30 companies, led by Google
- Includes mobile operators, chip makers, handset makers, software companies
- Overall (official) goal: Improve mobile user experience

Free, open mobile platform

- Many parts open-source (Apache license)
- Both Free and commercial software encouraged
- Contrast to more closed, commercial J2ME world

Applications can access same features as software shipped with device

Stress on networked applications, sensor-equipped phones

Claims easier development of applications than J2ME

First version of SDK in 2007, current version (Dec. 2009): SDK 2.0.1

Hardware: e.g.

- Verizon Droid by Motorola
- T-Mobile Cliq by Motorola
- Sprint Hero by HTC

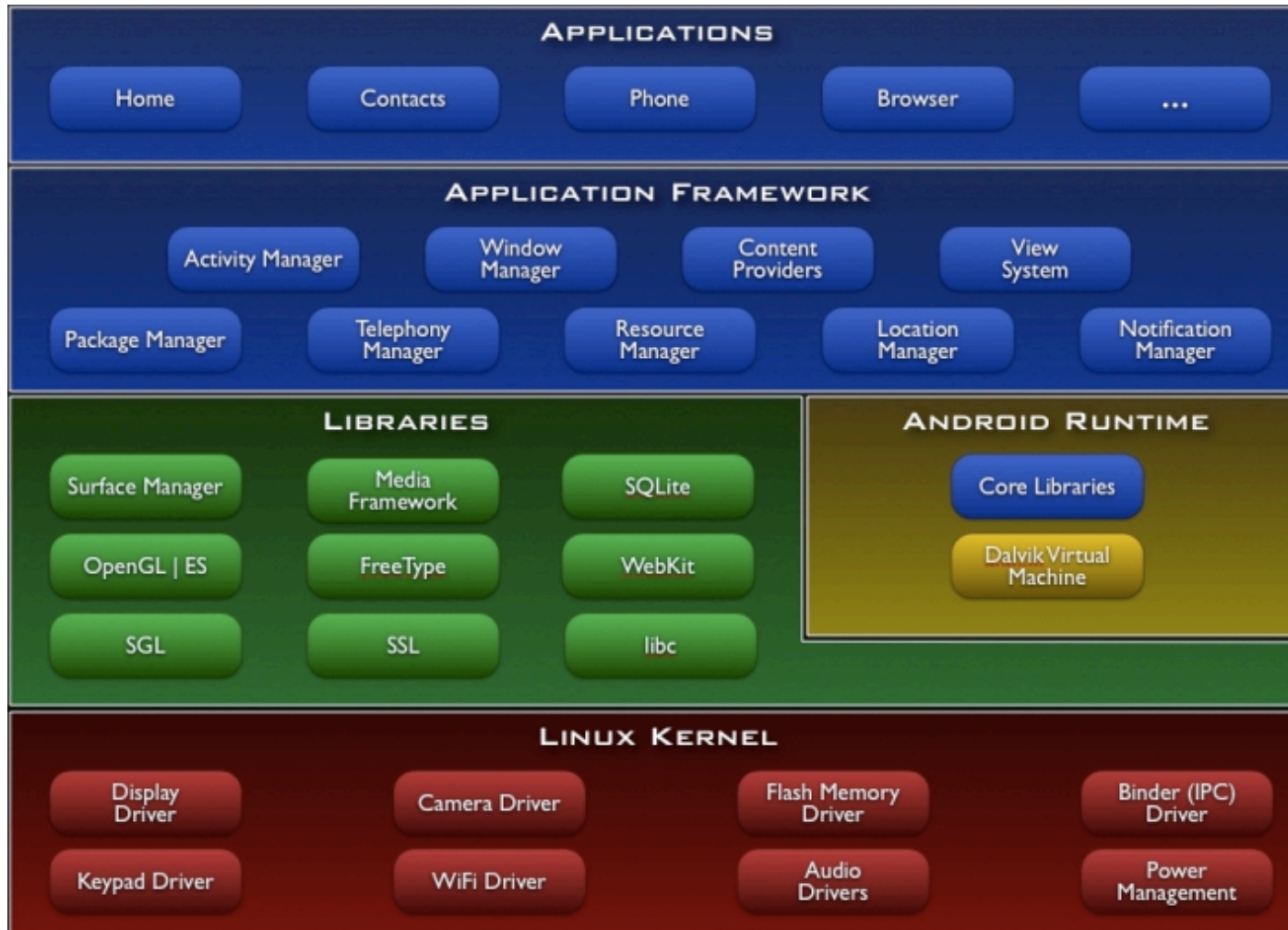
<http://code.google.com/android/>

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

<http://www.heise.de/newsticker/Ausblick-auf-Android--/meldung/108785>



Android Architecture



<http://developer.android.com/guide/basics/what-is-android.html>

Android – Hello World

```
package com.android.hello;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```



Android User Interface Development

Two possibilities:

- Programmatically (previous example)
- Based on XML descriptions

Example XML (equal to previous Hello World example):

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="@string/hello"/>
```

Element: TextView

Attributes:

- layout_width, layout_height
- text

2 Mobile and Ubiquitous User Interfaces

2.1 Mobile Computing

2.2 Design Guidelines for Mobile Devices

2.3 Input and Output on Mobile Devices

2.4 System Architectures for Mobile Devices

2.5 Example Applications

2.6 HCI and Ubiquitous Computing

Example: Mobile Game (Stand-alone, Virtual)

- One user
- Interaction only by local screen and keyboard
- Fully virtual setting
 - No relationship to surrounding real world



icemania.org

Example: Mobile Game (Stand-alone or two parties, Physical)

- Table tennis game “SymBall”
- Mobile phone as input device
 - Camera used for movement detection
- Mika Hakkarainen, Charles Woodward, VTT, Finland

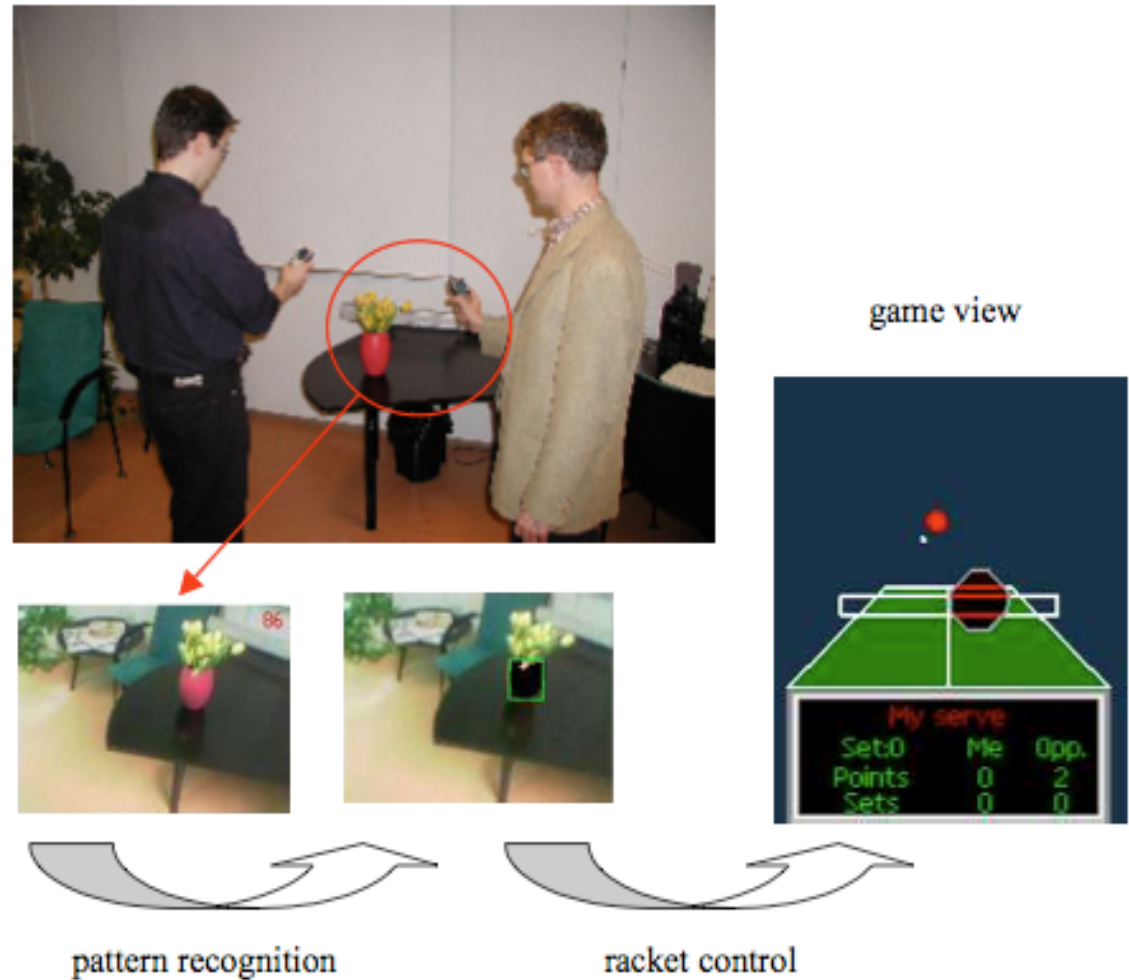


Figure 1. SymBall game concept.

Example: Mobile Game (Networked, Virtual)



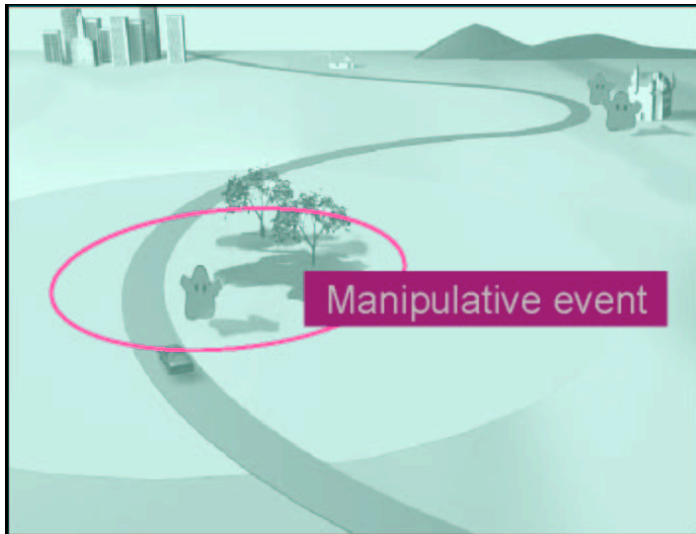
<http://www.3g.co.uk>

ASIA Hong Kong : Artificial Life announced the release of its first massive multi-user 3G mobile game called "Virtual Penguins."

The new game is based on the life and fate of the popular Emperor Penguins. The game simulates the difficult live cycles these penguins endure on the Antarctic ice and establishes a massive multi-user mobile virtual community. It is custom built for 3G phones. However, a single player version will also be released. The game is offered in high resolution 3D animation technology.

Players of the game take on the role of individual penguins. They have to go through several cycles of breeding and feeding their offspring. Players first have to find a mate (a second player) and then jointly the two players need to care for the egg that the female penguin will lay ...

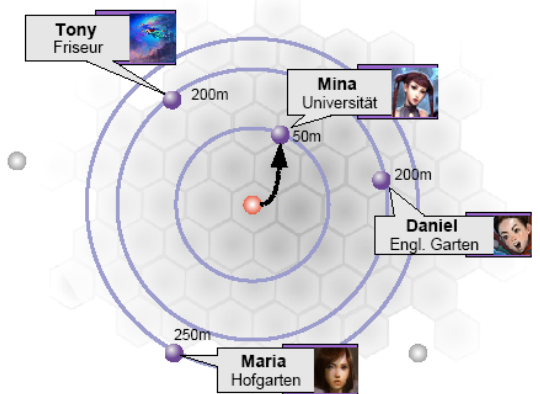
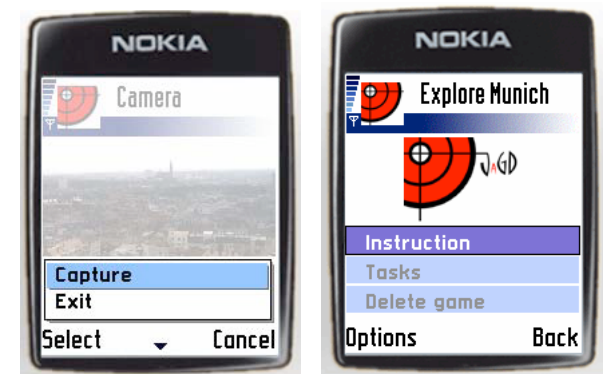
Example: Mobile Game (Networked, Physical)



- Backseat Gaming

Media Informatics Examples

- Practicle Course WS 04/05
 - 3 applications:
 - » JaGD
 - » Traffic Warden Support
 - » Posters as Gateways
- Practicle Course Mobile Productivity WS 06/07
 - Entwicklung von mobilen Anwendungen für blue-collar workers
 - 3 applications:
 - » Mobile Inventory System
 - » Mobile Product Evaluation and Comparison of Prices
 - » Mobile Tagging Platform
- Practicle Course SS 07
 - Project “Beeep” Mobile Tagging
- Practicle Course SS 08
 - Android: Mobile Health



Mobile Reporter

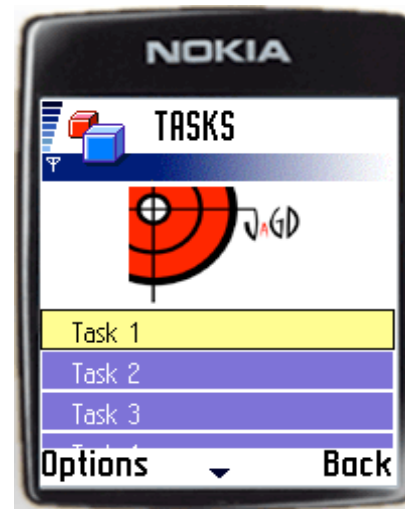
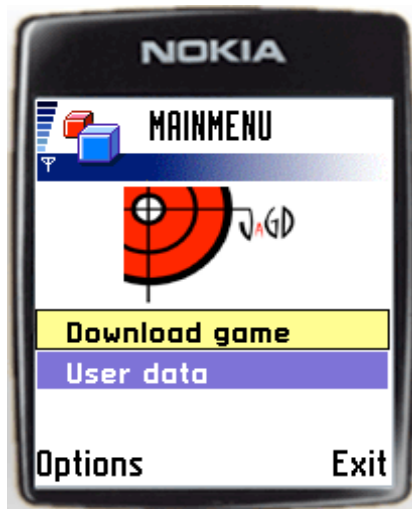
- Mobile Blogging Platform
- Submit via SMS, MMS, E-Mail, MIDlet and Webinterface
- Available at Sourceforge
 - <http://sourceforge.net/projects/mrep/>



Mobile Photo Treasure Hunt



- Mobile Learning and Gaming Platform
- Online game editor
- XML game format
- Mobile phone application



10 Usability Recommendations for Games (1)

- **Provide a Clear Menu Structure**

Use only one main menu, accessible with the left soft key. Keep the menu short. In general, use the left soft key for OK, select, and menu; use the right soft key for cancel and back.

- **Simplicity Is Key**

If two solutions are equally valid, use the simpler. Make sure each entity in the game is unique, and not easily confused with any other. Provide different game modes only if they are truly different and valuable.

- **Provide Help When Needed**

Keep help text short. If feasible, scroll text one screen at a time, not one line at a time. Display short text on the screen to explain new items, characters, and situations in the game. Provide a setting to disable in-game help. Provide a graphic representation of which keys are used for which functions. Do not expect players to read help text or force them to do so.

- **Be Relentlessly Consistent**

Use the mother tongue of the user. Be consistent with the phone's UI, with game industry conventions, and within the game itself. Use the left soft key for OK, select, and menu; use the right soft key for cancel and back.

- **Don't Waste the User's Time**

Allow her to skip the introduction. Do not require re-entry of data. Provide shortcuts and reasonable default values.

From: Series 40 Game Usability Study (Nokia Forum)

10 Usability Recommendations for Games (2)

- **Use Natural Controls**
Use the 2, 4, 6, and 8 keys for horizontal and vertical movement as well as the arrow keys; use the 1, 3, 7, and 9 keys for diagonal movement, if enabled. Use the 5 key as the action button. Design the game so that it does not lure the user into pressing two keys at once, since many mobile devices (and all Series 40 devices) do not support simultaneous key presses.
- **Enable Save and Pause**
Provide a simple save-game feature. Have the game auto-save when the user presses the red phone button - use the `destroyApp()` method to do this. Provide a pause mode (left soft key, which goes to the game menu); this can be done using the `hideNotify()` method. If the user quits the game from the pause mode, have the game auto-save.
- **Conform to Real-World Expectations**
For example, when jumping or throwing objects, the flight path should be predictable. There must be no invisible barriers that the player cannot pass or holes that he cannot reach. Do not end the game arbitrarily. Implement a realistic physics model if relevant (for example, racing games).
- **Go Easy on the Sound**
Provide sound for feedback, but ensure that the game is playable with the sound off, and provide an easy way to turn sound off within the game. No annoying sounds: not too loud, not too high-pitched. Avoid background music, if possible.
- **Implement a High Scores List**
Tell the user what score he reached before asking for a name; provide the previously entered name as the default. Do not force the user to enter a name; make it optional.

From: Series 40 Game Usability Study (Nokia Forum)

Six Mobile Messages

- (1) Mobile applications need to be redesigned for different terminals, because a **small user interface does not scale down**. Feature prioritization is critical.
- (2) Users are **cognitive, emotional, contextual and cultural** actors. It takes segmentation, personalization, and continuous evolution to fulfill their versatile changing needs.
- (3) The mobile industry faces a wireless **complexity** threshold. To overcome it we will have to provide a seamless user experience of terminals, applications and services.
- (4) For mobiles, **fancy visions** and real end-user **value** conflict. Taking the next step requires the resolve to make instead of dream and decide instead of speculate.
- (5) A user interface is as good as the workmanship behind it. Great UIs are born out of a passion for **detail**. The more you polish the better it gets.
- (6) The **challenge** of mobile UI development is **shared** among external software firms, industry consortiums, and service providers. Reasonable development stability is a must.

Keikonen/Lindholm