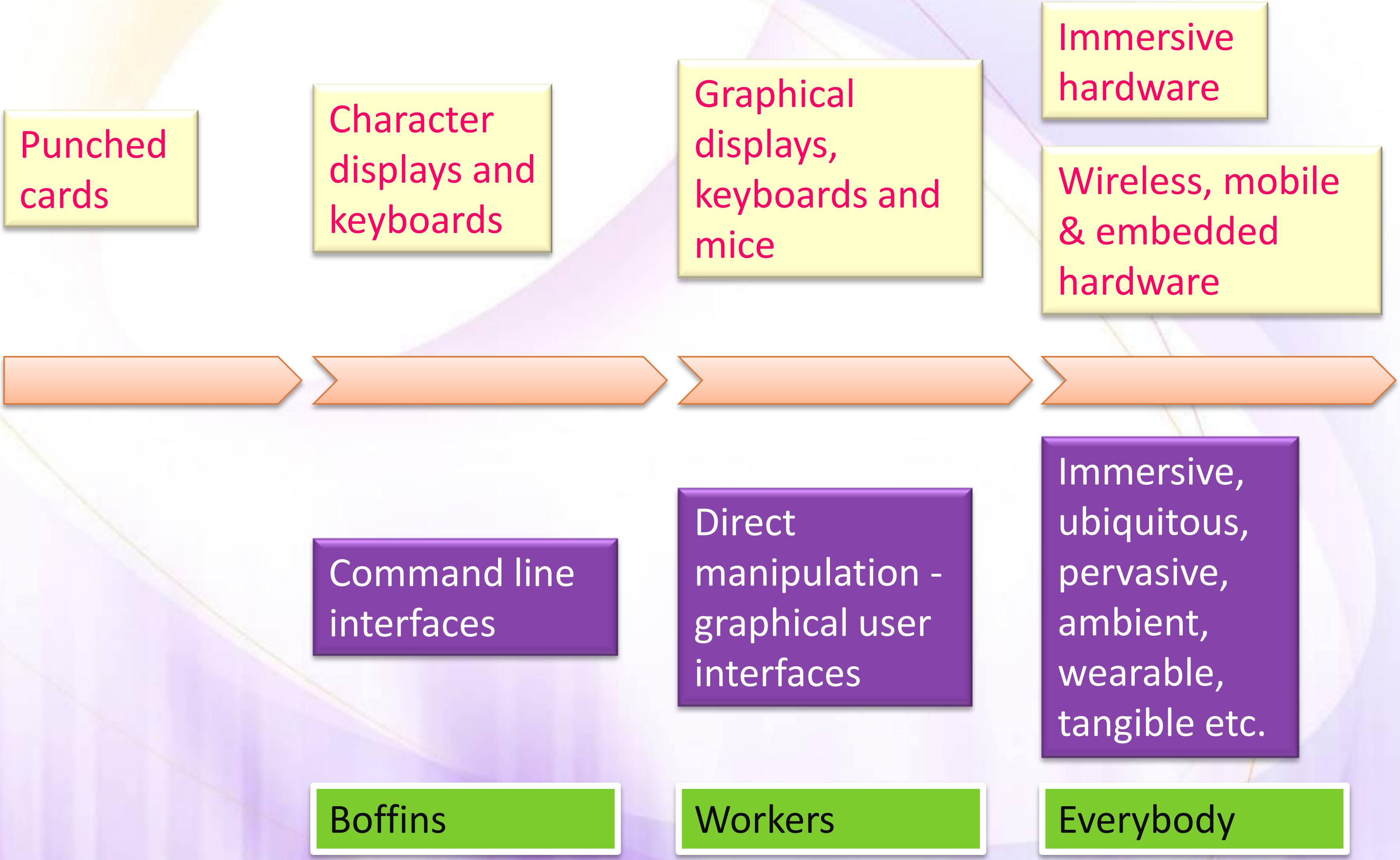
Ubiquitous Interface

Co-evolution of hardware, interface and users



Punched cards

Character displays and keyboards

Command line interfaces

Boffins

Graphical displays, keyboards and mice

Direct manipulation - graphical user interfaces

Workers

Immersive hardware

Wireless, mobile & embedded hardware

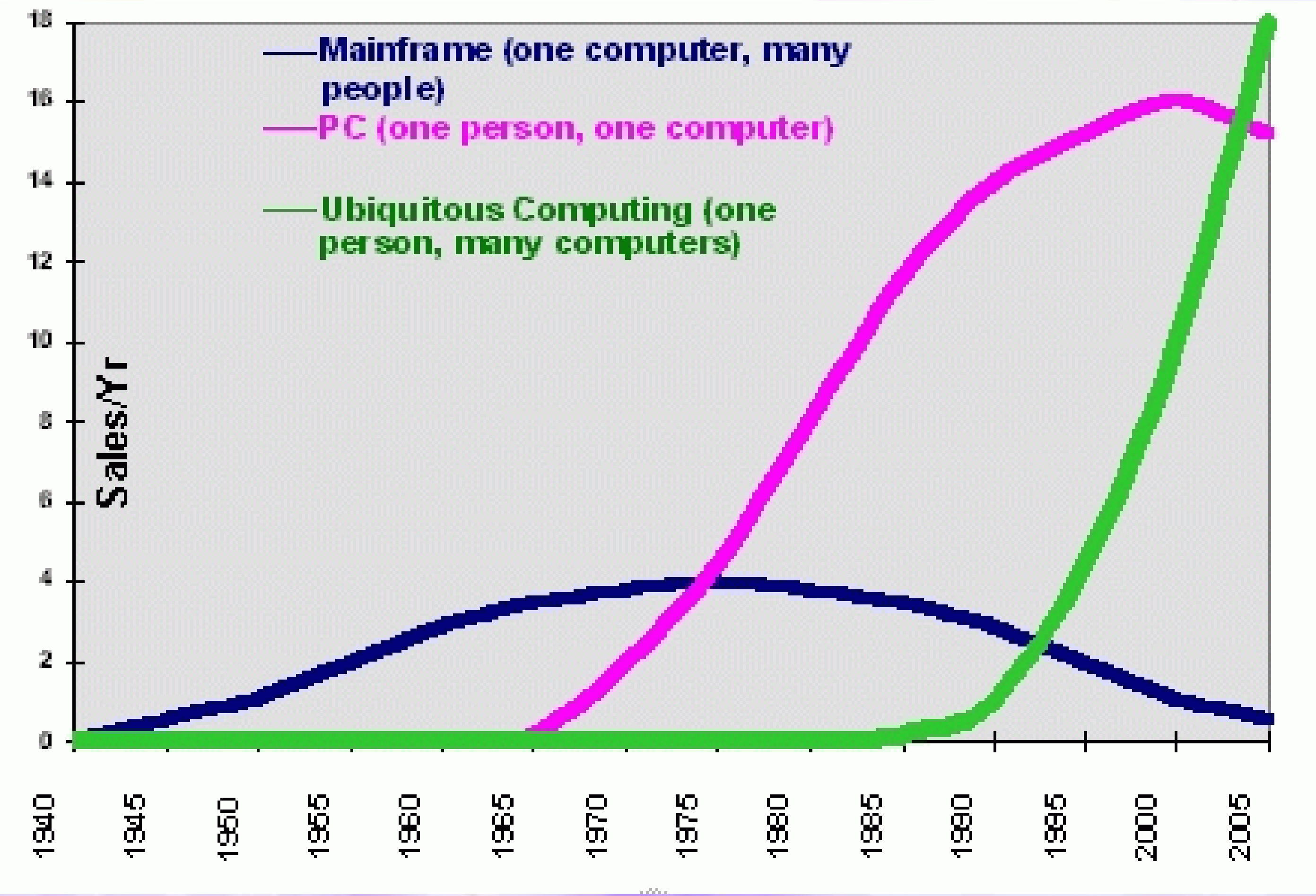
Immersive, ubiquitous, pervasive, ambient, wearable, tangible etc.

Everybody

Covered topics

- Ubiquitous Computing
- Wearable
- Mobile

Ubiquitous Computing



The third wave of computing?

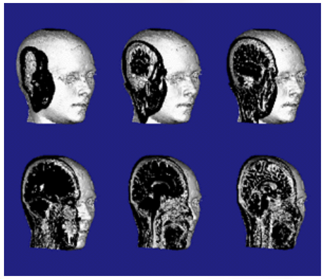
Ubiquitous computing

- What is the ubiquitous computing?
  - The method of enhancing computing use by making many devices (services) available throughout the physical environment, but making them effectively invisible to the user
- Ubiquitous means:
  - present everywhere
  - simultaneously encountered in numerous different instances
  - computers become a useful but invisible force, assisting the user in meeting his needs without getting lost in the way
- Ubiquitous computing is roughly the opposite of virtual reality.
  - Virtual reality: puts people inside a computer-generated world
  - Ubiquitous computing: forces the computer to live out here in the world with people.

Virtual Reality

- Applications

Scientific visualisation

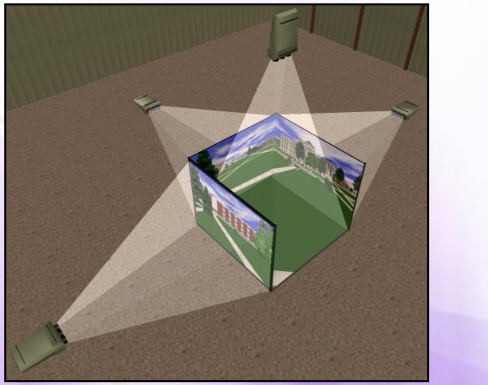


Simulation

Games

- Interfaces → challenges

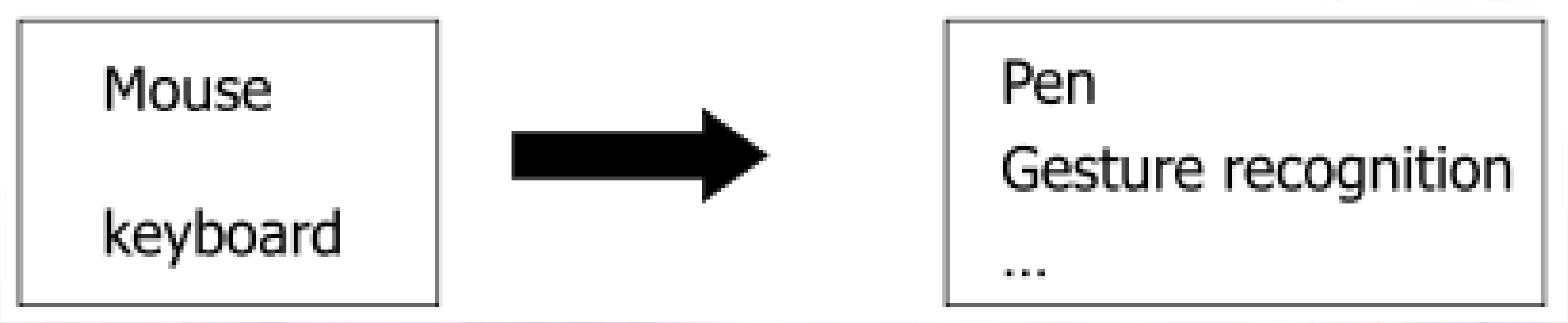
  - System performance
  - Movement
  - Health and safety
  - Haptics & small presence



PLAYSTATION 3
SONY
©2006 Sony Computer Entertainment Inc. All rights reserved.
Design and specifications are subject to change without notice.

Ubiquitous interfaces

- The multitude of different Ubicomp devices with their different sizes of displays and interaction capabilities represents another challenge



Mouse

keyboard

Pen

Gesture recognition

...

Ubiquitous interfaces

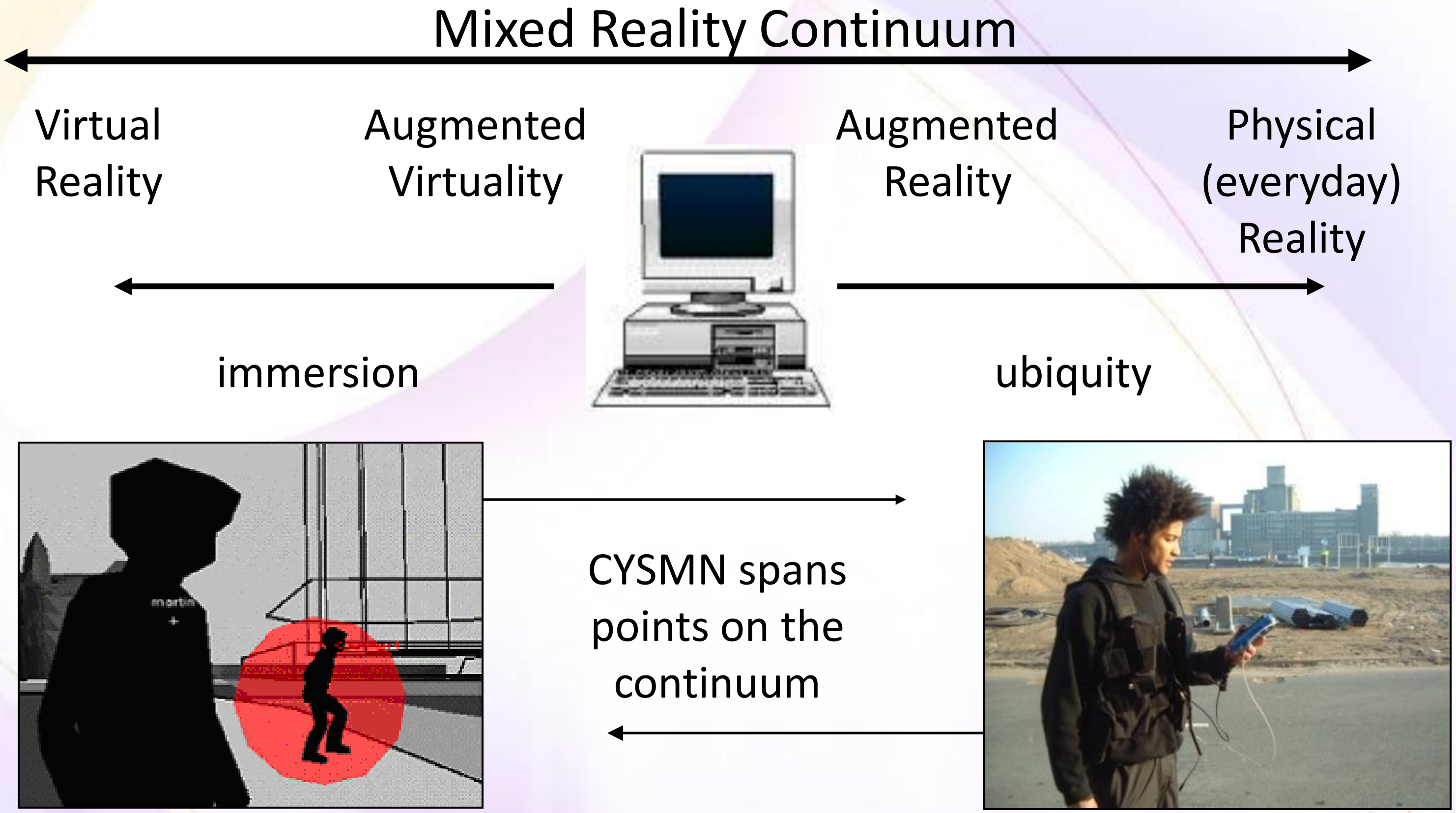
- Challenges
  - Interaction with invisible sensors
  - Context awareness
  - Interruptions and (dis)engagement
  - Privacy, security and accountability
  - Interaction in public settings
  - How to study and evaluate experiences?

Interacting with (invisible) sensing systems

AUTOMATIC FAUCETS
TO TURN ON/WALK UP TO SINK
TO TURN OFF/WALK AWAY FROM SINK
(BLACK CLOTHING DOES NOT OPERATE FAUCETS)

- How do I address one (or more) of many possible devices?
- How do I know the system is ready and attending to my actions?
- How do I effect a meaningful action, control its extent and possibly specify a target or targets for my action?
- How do I know the system is doing (has done) the right thing?
- How do I avoid mistakes?

A Continuum of Realities



Mixed Reality Continuum

Virtual Reality

Augmented Virtuality

Augmented Reality

Physical (everyday) Reality

immersion

ubiquity

CYSMN spans points on the continuum

- **Designing for seams:** { Hide them/Reveal them
  Manage them/Exploit them/Remove them

Covered topics

- Ubiquitous Computing
- Wearable
- Mobile

Context-aware and Wearable Computers

- People on the move need computing facility with a wide ranging capabilities
- Wearable computers provide this as a facility that is always available everywhere
- Capabilities range from
  - Simple stored-information retrieval; to
  - Synchronous or asynchronous collaboration to context-aware platforms with proactive assistants
- Context-awareness adds to their capabilities

Context-aware and Wearable Computers

- Application domains
  - Inspection and Maintenance
  - Manufacturing
  - Navigation to on-the-move collaboration
  - Position sensing
  - Real-time speech recognition and language translation
- Techniques used
  - User-centered design
  - Rapid prototyping
  - In-field evaluation
- Principles
  - Merge wearable computers with the user’s workspace
  - Blend seamlessly with the user’s existing environment
  - Provide as little distraction as possible

Characteristics of Wearable UIs

- Displaying information and changing state
- Additionally: Context information
  - Context-dependent presentation
  - Context includes input and output modes and devices available
  - Context change triggers information display / state change
- Idea:
  - Specify abstract UI using:
    - Hierarchical structure
    - Graphical Syntax
    - Many temporal operators
    - Focus on activities
  - Use context change triggers
  - Decide context-dependent presentation during runtime

Context-dependent presentation

- Example: a web browser with two presentation modes
  - Desktop mode: Like firefox
  - Mobile mode: like opera “small screen rendering”

Mode	UI Specification	UI “Rendering”
Desktop	html document, links	Compress graphics, change positions, use different fonts
Mobile	html document, links	Change interaction: no mouse click, but chose links via cursor keys

Abstract UI Specification

- Example: Develop an interface for writing an Aircraft Repair Report
  - Input:
    - Text of repair report
  - Output:
    - Indicate that the repair report entered is complete

→ Web browser equivalent: Form

- Text input field
- “submit” button

Abstract UI Implementation

- PDA: Java 1.2 (AWT)

```
Panel p = new Panel();

p.add ( new  Label
   ("Enter Report") ;
TextField  tf  = new
   TextField ("Your
   Report  Here " ,
   256) ;
p.add ( tf ) ;
Button  b = new  Button
   ("Save ");
p.add ( b ) ;
```

- Desktop: Java 5 (Swing)

```
Jpanel p = new
   Jpanel();
p.add (new  JLabel
   ("Enter Report") ;
 JTextField tf = new
   JTextField ("Your
   Report  Here " ,
   256) ;
p.add ( tf ) ;
JButton  b = new
   JButton("Save ");
p.add ( b ) ;
```

- QT 4

```
Qlabel *reportLabel =
   new  QLabel (tr( "
   Enter report " ) ) ;
QTextEdit *reportEdit =
   new  QTextEdit ;
QPushButton *saveButton
   =  new  QPushButton (
   tr ( " Save " ) ) ;
myLayout  =  new
   QHBoxLayout ;
myLayout->addWidget (
   reportLabel ) ;
myLayout->addWidget (
   reportEdit) ;
myLayout->addWidget (
   saveButton ) ;
```

Abstract to concrete

- How to get from abstract to concrete?
  - Use an expert programmer, give him the spec, let him program, use result
- How about different devices?
  - Use expert for every possible device, send to expert programmer, let them work together.
- How about different contexts?
  - Use domain expert to describe contexts, send to device expert to design context-dependent optimal display for specific device, send to programmer, program
  - Only viable for small number of devices and huge sales. i.e. mobile phone games

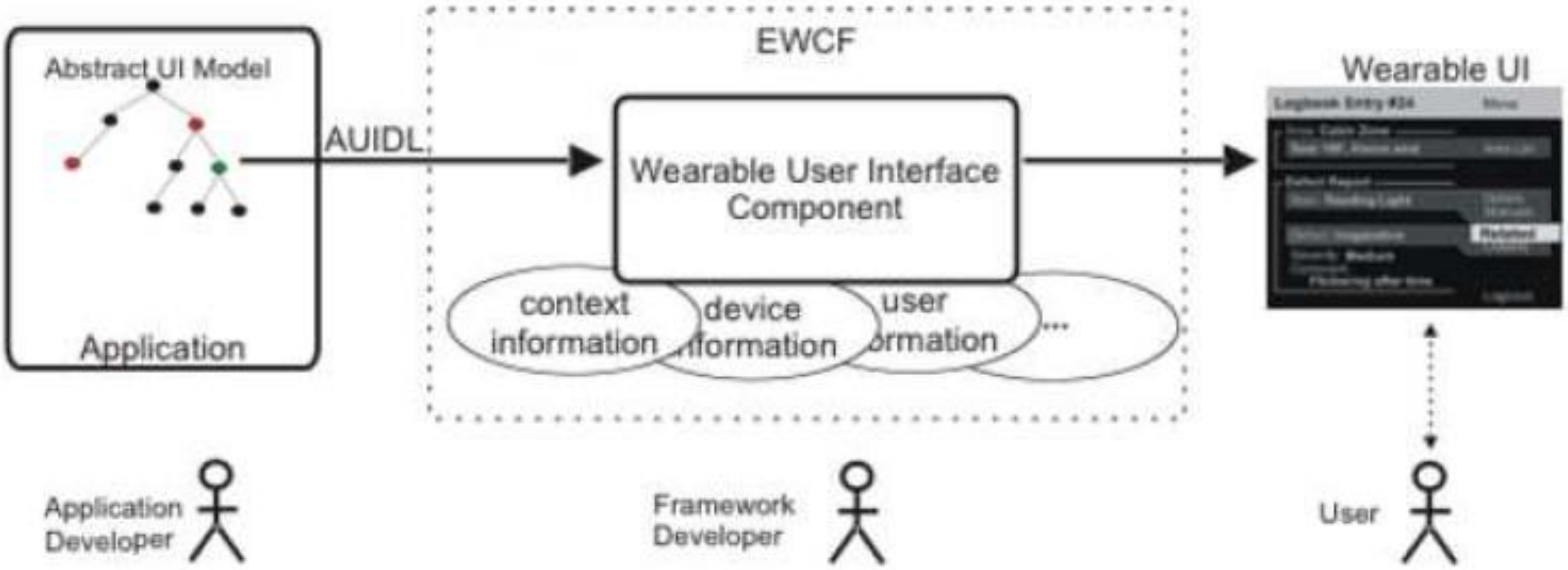
Abstract to concrete

- Can we do without all these experts?
  - Divide the application program in two parts:
    - The abstract UI
    - The renderer
- How about different devices?
  - The renderer can be device-specific: It knows best how to use UI elements of the target device
- How about different contexts?
  - The renderer itself can use context information in a device-specific way
  - The abstract UI can choose from a number of available renderers.
  - This choice can be based on:
    - device availability,
    - user preference,
    - context.

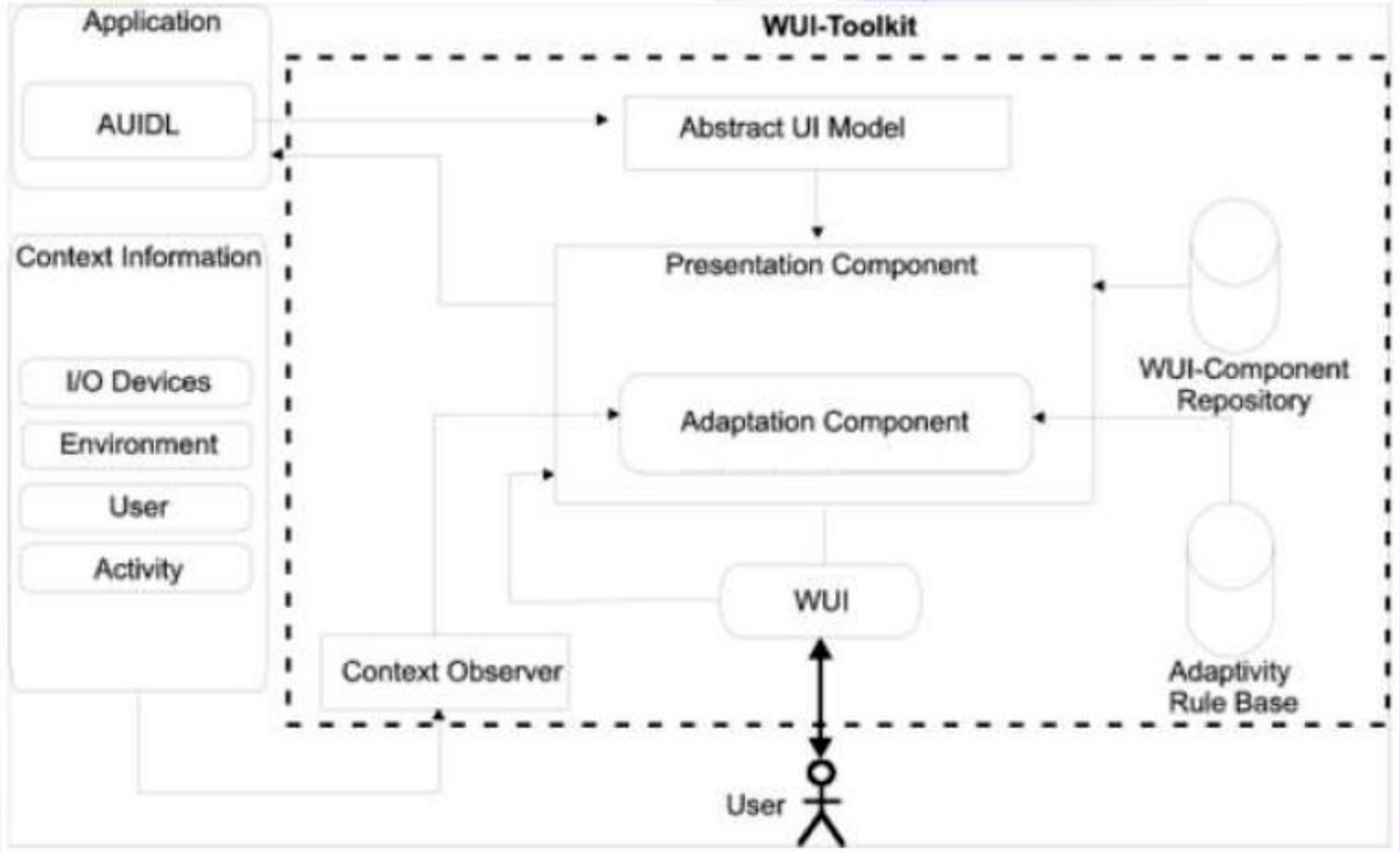
Wearable UI Methaphor

- Output Mechanism
  - Visual: HMD
  - Audio
- Input Mechanism
  - Keys: Keyboard, Twiddler
  - Hands: gestures, direct manipulation
  - Speech
- Interaction Methods
  - menu selection, direct manipulation, form fill-in
  - command language, natural Speech

WUI development



Structure of WUI



Wearable Evaluation

- How to measure the performance of a wearable system?
  - Remember: Supporting a primary task
  - Idea: measure the performance in the primary task Example: Wearable Maintenance support
    - Time
    - Quality
- Drawbacks:
  - Long time needed
  - Variation in users/Tasks: Even more time needed
  - System has to be built and integrated to be evaluated
  - What if evaluation outcome is negative?
- Real-world evaluations are rare
  - Idea: Implement parts of the system in a lab.
  - “Living Lab” approach

→ Question: How to simulate primary task in the lab?

Wearable Evaluation: Aspects of the primary task

Physical Task

- Simple tasks: Walking, running, biking
- Strenuous tasks: running fast, carrying loads
- Manipulative tasks: push buttons, operate machines, use tools, select tools
- Precision tasks: handle tools carefully, avoid damage and spills
- Also physical tasks: input (e.g. gesture input)
- Body has physical limits: accuracy, force, energy limits

Cognitive Task

- Simple tasks: Reading, Listening, Identify objects, following signs, "matching tasks"
- Complex tasks: calculations, translations, geometric tasks (see your favourite IQ test)
- Also cognitive tasks: input, understanding output
- Analog to physical limits: "cognitive load" limit
- Cognitive load varies with age, familiarity with task, between persons

Attention

- Both physical and cognitive tasks need attention
- Attention is limited
- e.g.: you can only memorize a small (5-11) Number of things at the same time in your short time memory
- Some brain functions have limits: Humans only have one motor cortex
- Degrading attention leads to degraded performance: Precision lowers, reaction time rises, task execution takes longer
- Divided attention: affected by task similarity, task difference, practice

Measuring performance

- Idea: Use this information to craft artificial tasks to measure performance
- Cognitive tasks: simple but measurable tasks, measure execution time and correctness
- Examples: Matching tasks, finding repetitions in letter sequences, . . .
- Physical tasks: Not too easy, but easy to measure
- Examples: Pushing buttons , “Hotwire experiment”
- Experiment:
  - Measure physical task w/o cognitive task
  - Measure cognitive task w/o physical task
  - Measure both together

Covered topics

- Ubiquitous Computing
- Wearable
- Mobile

Mobile computing

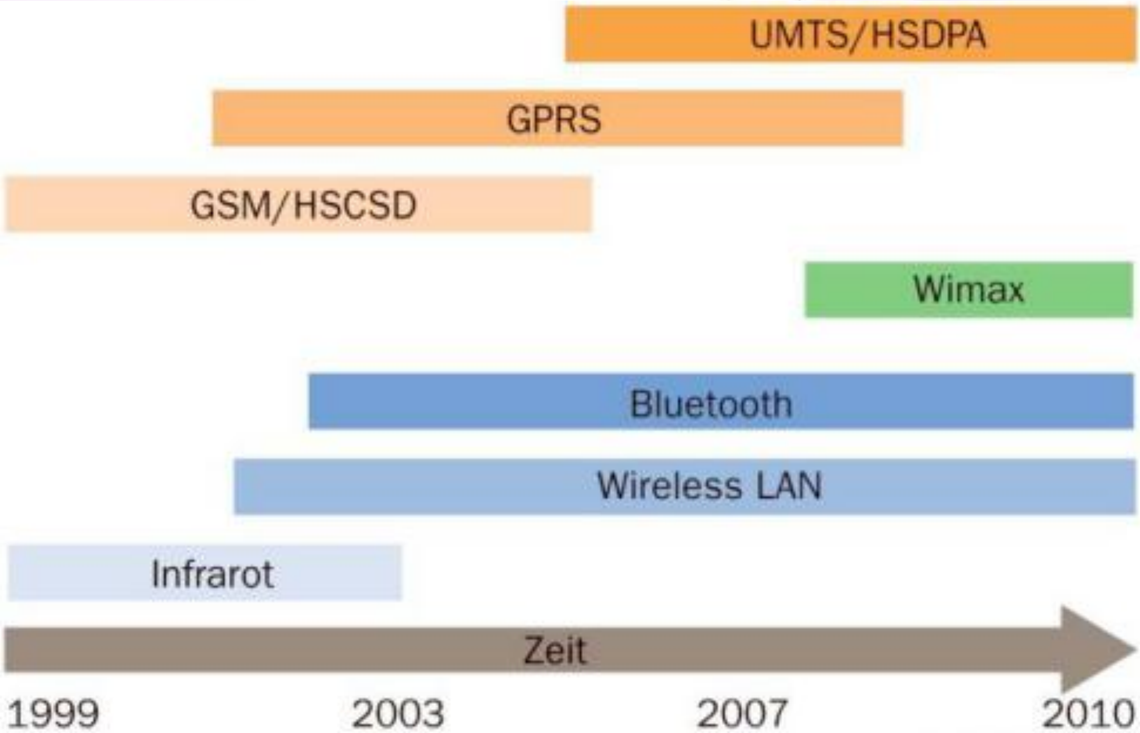
- Computing while on the move, using a computer (laptops, palmtops, PDAs, smart phones, pagers, sensors, etc.) that:
  - Is not continuously connected to the base or central network.
  - May communicate with a base location with or without a wireless connection.
- Example:
  - Wireless communication: using a modem-equipped PDA to receive text messages via satellite technology.
  - Non-wireless communication: sending data from a laptop to a central database or network server over a temporary dial-up connection

    → the laptop can still be used as a mobile device regardless of whether or not it ever connects to another computing device.

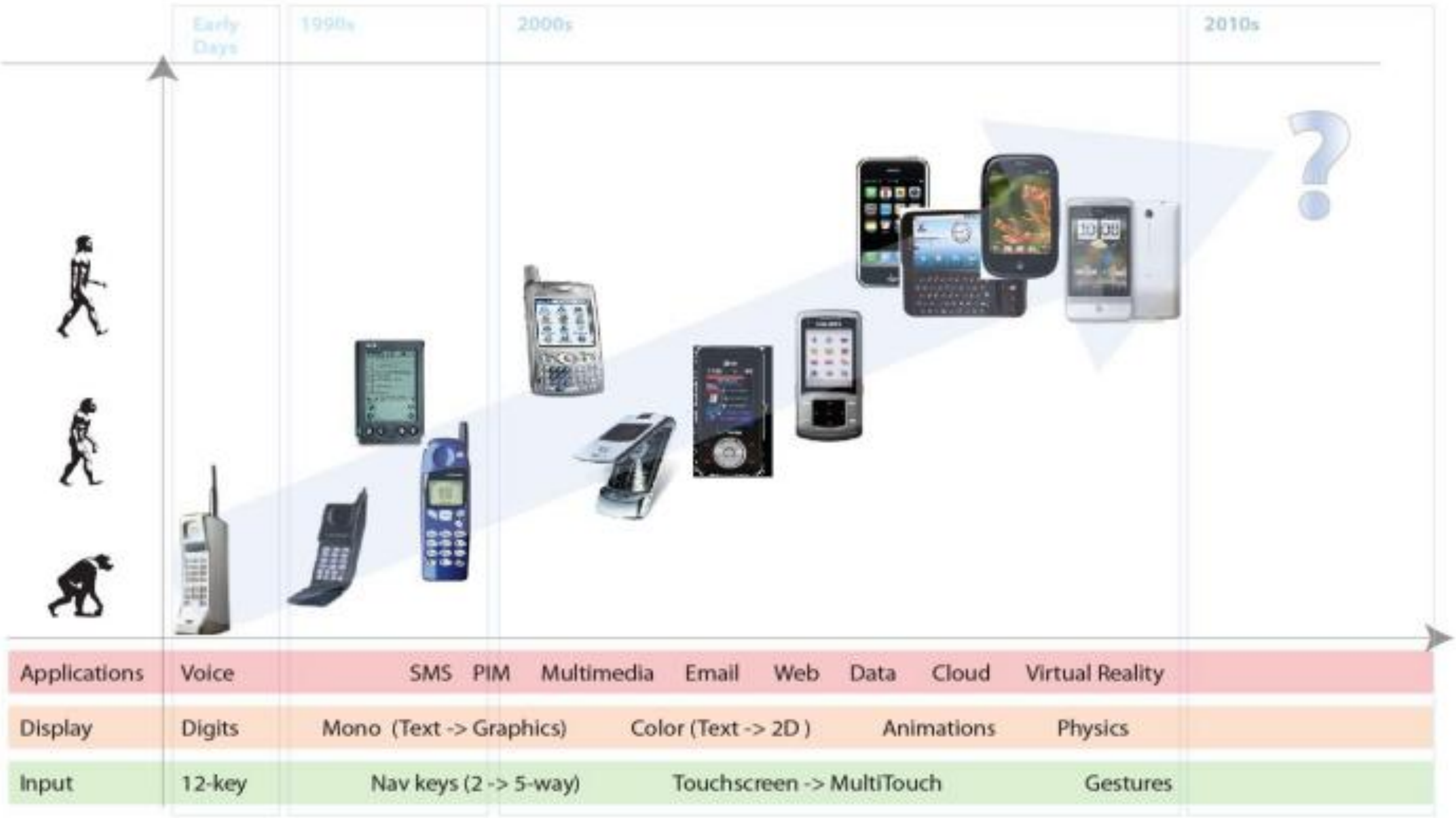
Mobility

- Made mobile computing possible through wireless communication technologies

  - Problem of disconnectivity !!!
  - This behavior is an inherent property of the ubiquitous computing concept and it should not be treated as a failure



Evolution of the mobile UI



Early Days | 1990s | 2000s | 2010s

Applications: Voice SMS PIM Multimedia Email Web Data Cloud Virtual Reality

Display: Digits Mono (Text -> Graphics) Color (Text -> 2D) Animations Physics

Input: 12-key Nav keys (2 -> 5-way) Touchscreen -> MultiTouch Gestures

Fixed system vs. Mobile system

	Fixed system	Mobile system
Purpose	Lengthy information processing tasks, Web browsing, email	On-the-go lookup an entry of information, quick communication
Form	Requires table, best used when seated	Less than DIN A4, often fits into shirt pocket or even invisible
Power	Requires power connection	Relies on battery life - has to deal economically with power
Connectivity	Fast and reliable connectivity	Slow and unreliable connectivity, but improving...
Input	keyboard and mouse	discreet, touch, backside, device
Display	Large	Small, compress, off-screen, extend, audio, tactile
Memory	Large working memory (GBs)	Small working memory (MBs)
Storage	Extensive storage options including large hard disks	Sometimes none, often limited to removable media

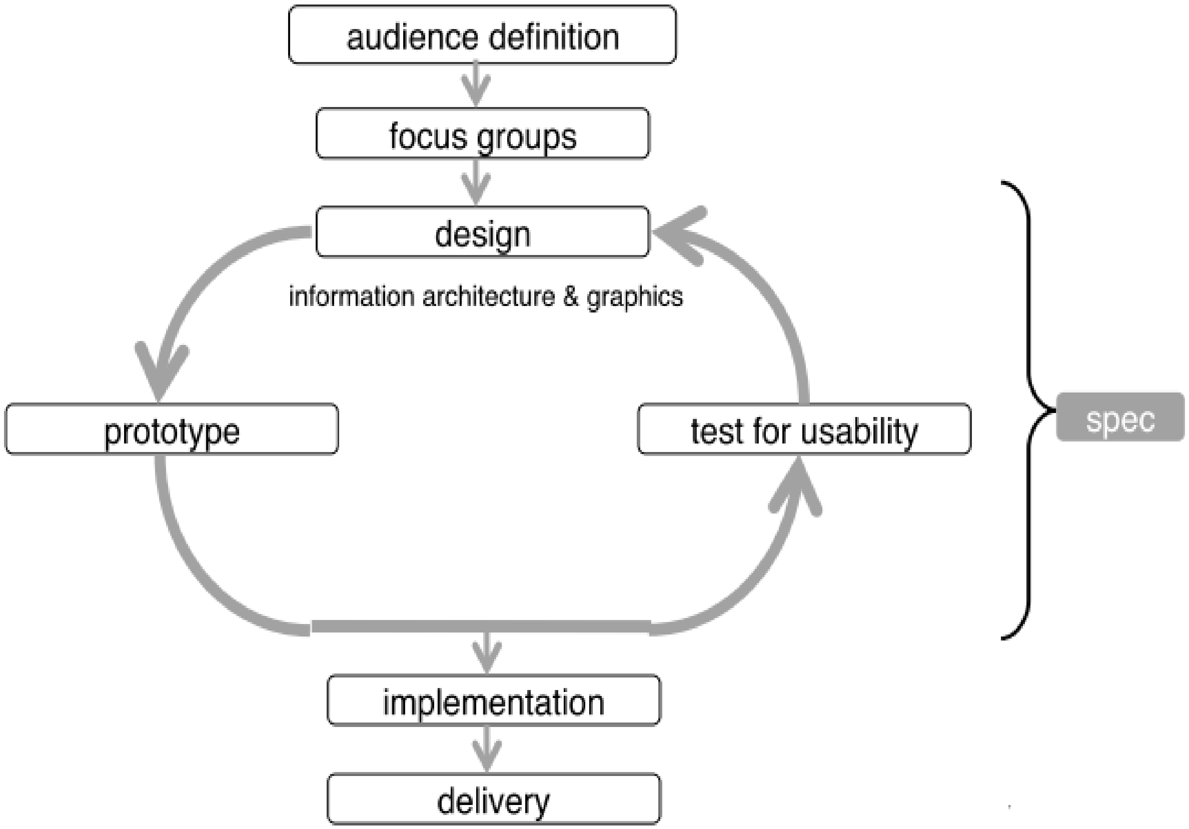
Example of mobile interface - input

- Discreet: for discreet tasks use discreet controls
  - Example: buttons for typing or launching apps

- Pointing/Touch: quite ergonomic compared to desktop

Mobile Design Principle

- Mobilize, don't miniaturize
- User context
- Handling device proliferation
- Emulators and simulators



Vu Thi Huong Giang, PhD

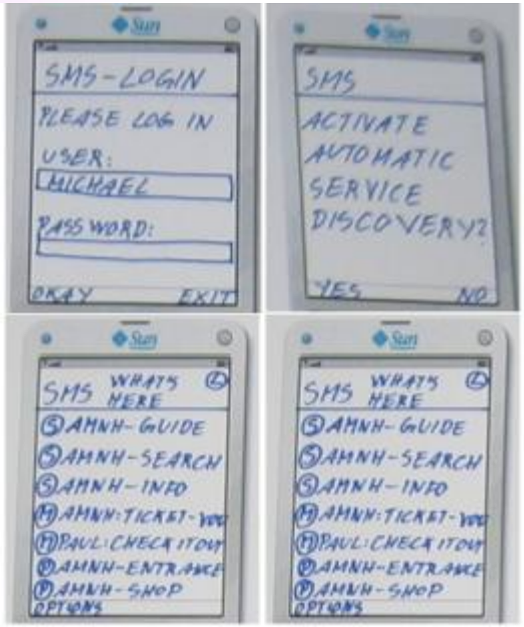
Application Types for Mobile Devices

- 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
  - Data transfer, synchronization
  - …

Application interface design

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

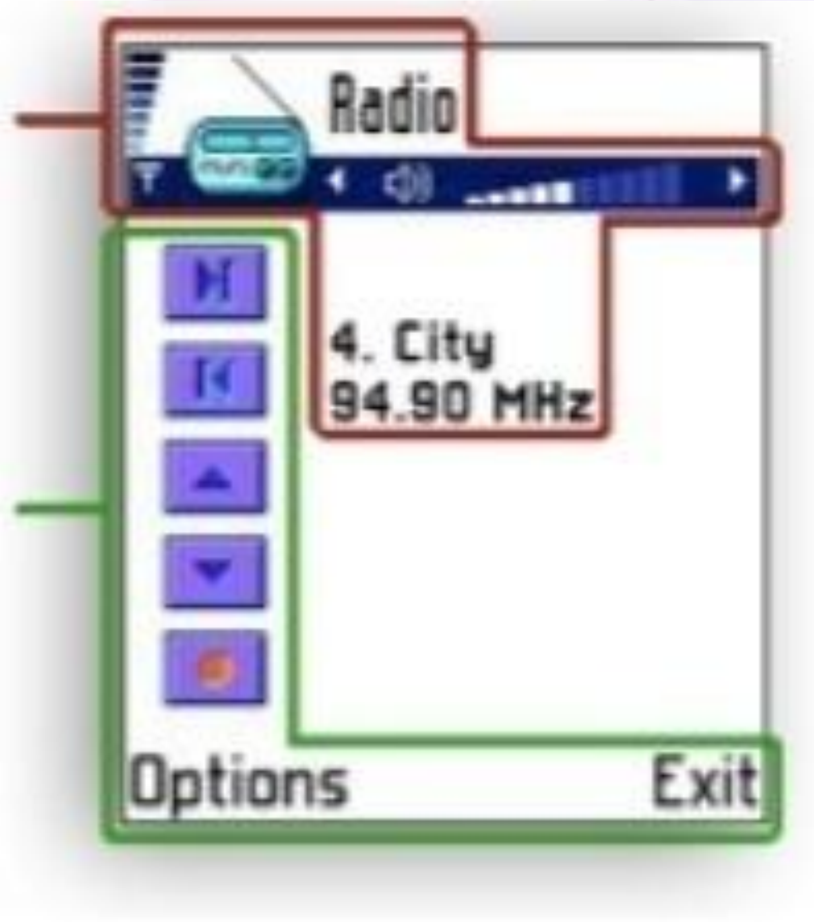


Mobile UI guidelines

- A small user interface does not scale down →
  - Redesign mobile applications for different terminals
  - Feature prioritization is critical.
- Wireless complexity threshold →
  - Provide a seamless user experience of terminals, applications and services
- Users are cognitive, emotional, contextual and cultural actors →
  - Fulfill their versatile changing needs: segmentation, personalization, continuous evolution
- Fancy visions and real end-user value conflict →
  - Solution, not dream: deciding instead of speculating
- Great UIs are born out of a passion for detail →
  - A user interface is as good as the workmanship behind it: the more you polish the better it gets.
- The challenge of mobile UI development is shared among external software firms, industry consortiums, and service providers →
  - Reasonable development stability is a must.

Mobile UI guideline's illustration

- The key questions to be answered in navigation

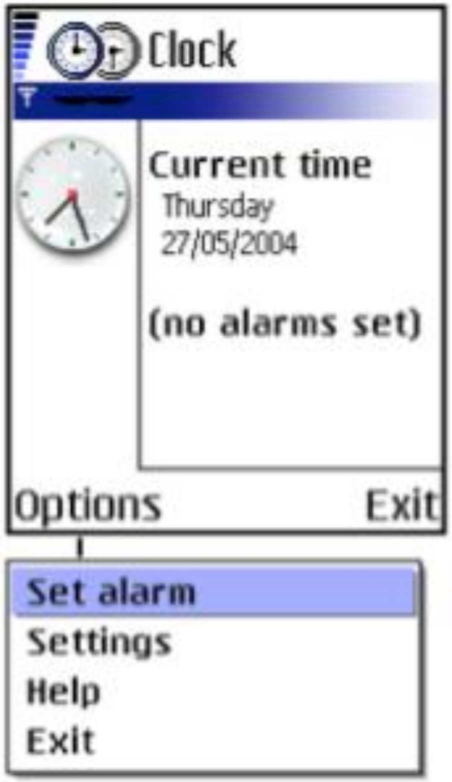


Where am I ?

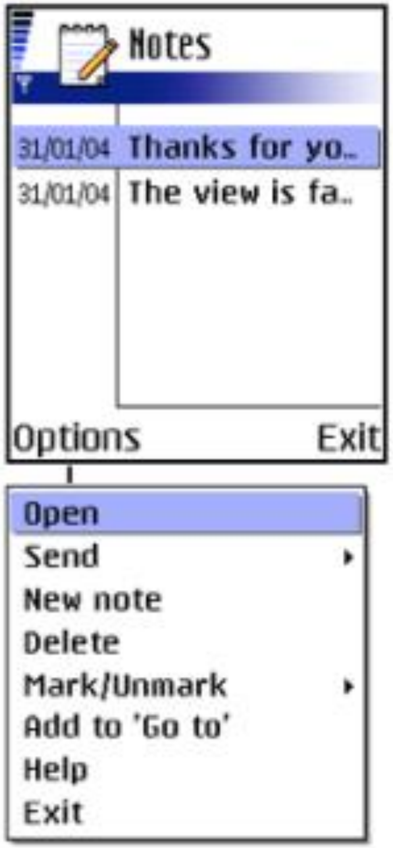
Where can I go from here ?

Mobile UI guideline’s illustration

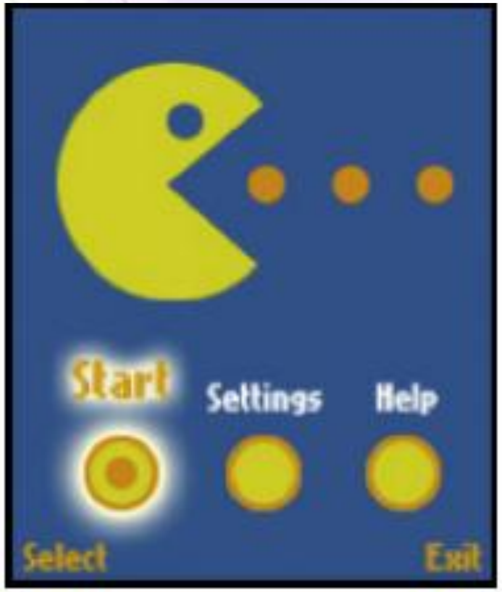
- Main menu styles



No focus



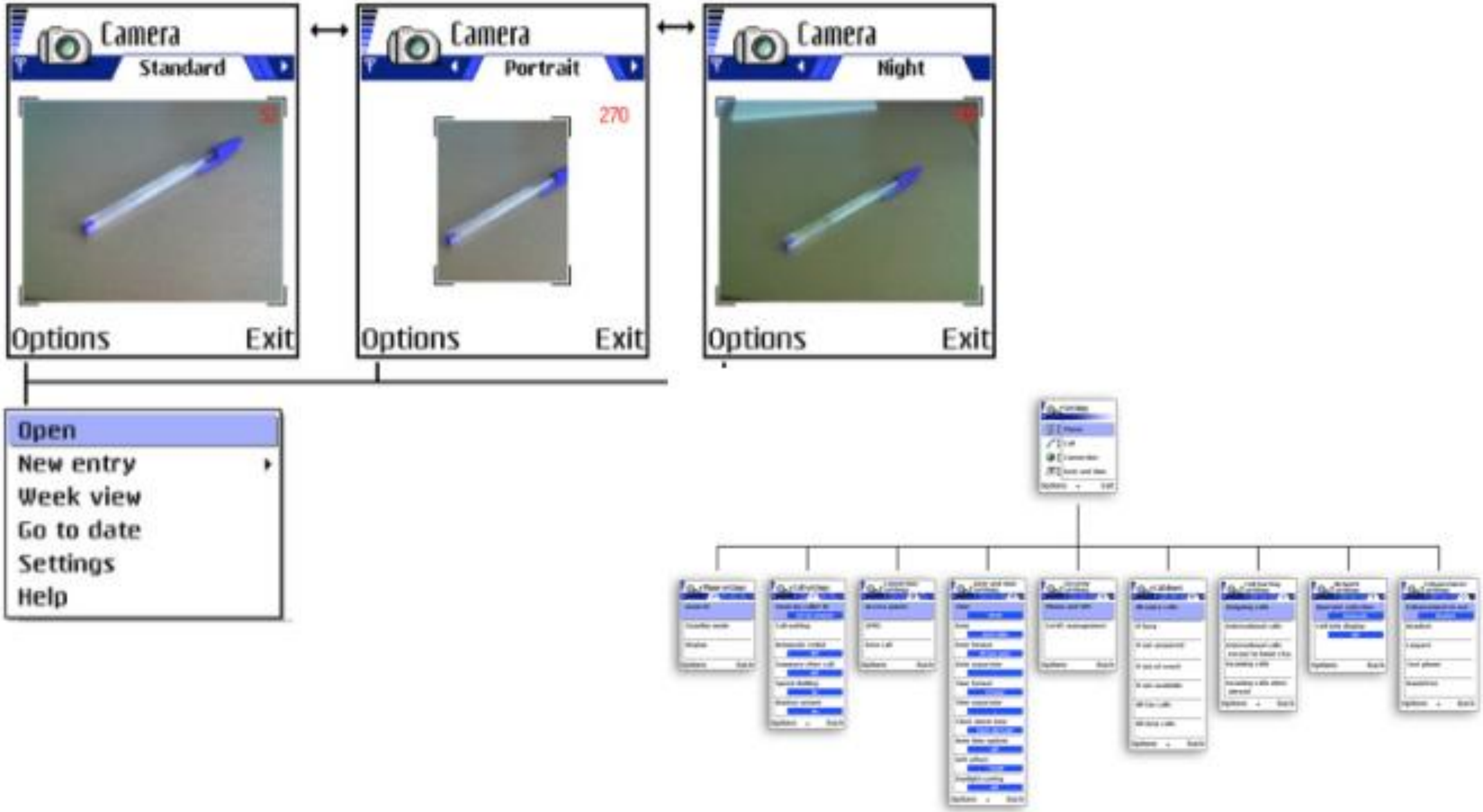
Using focus



Custom main menu (games)

Mobile UI guideline’s illustration

- Using tabs

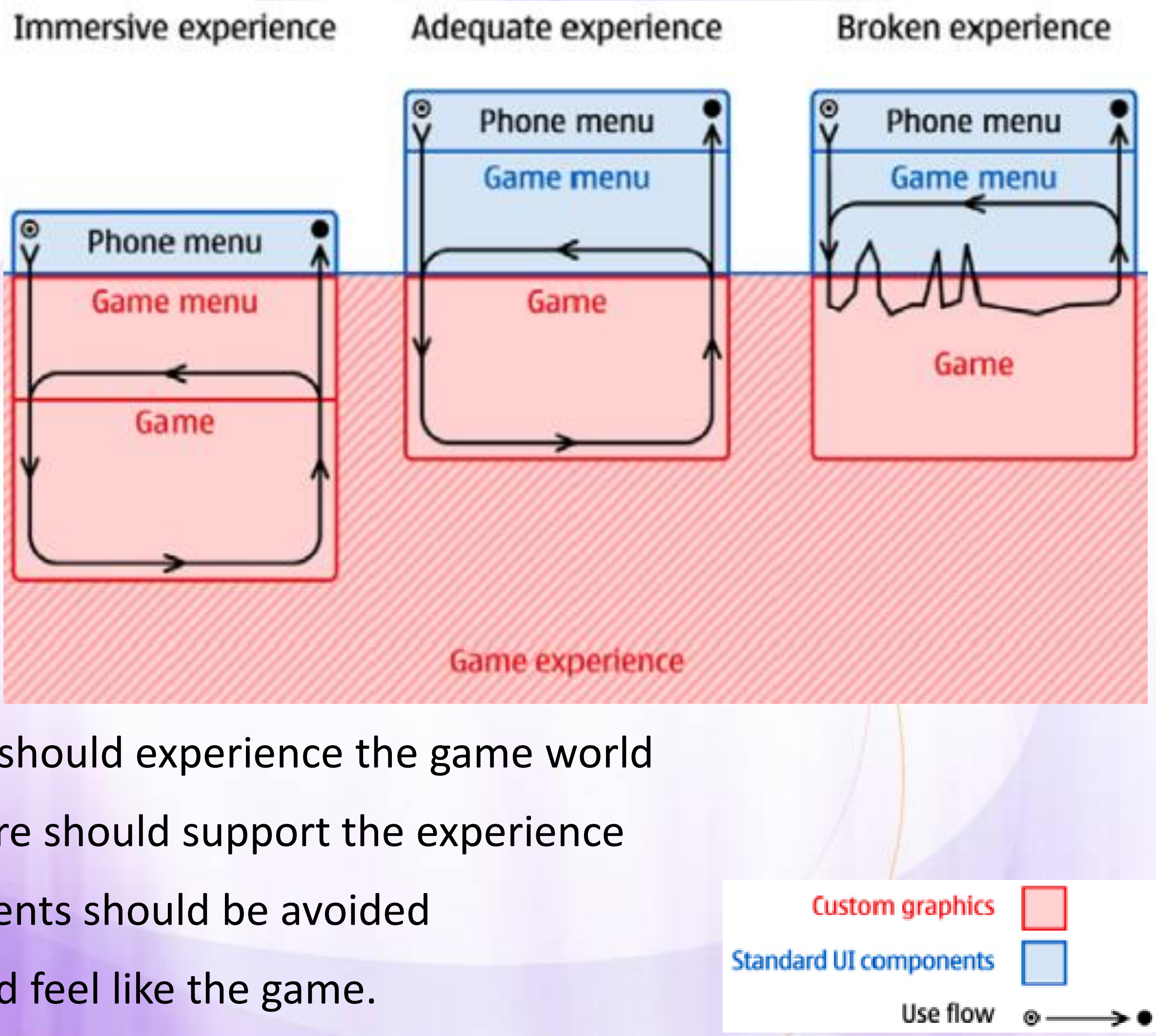


Usability and Mobile Gaming

- 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.

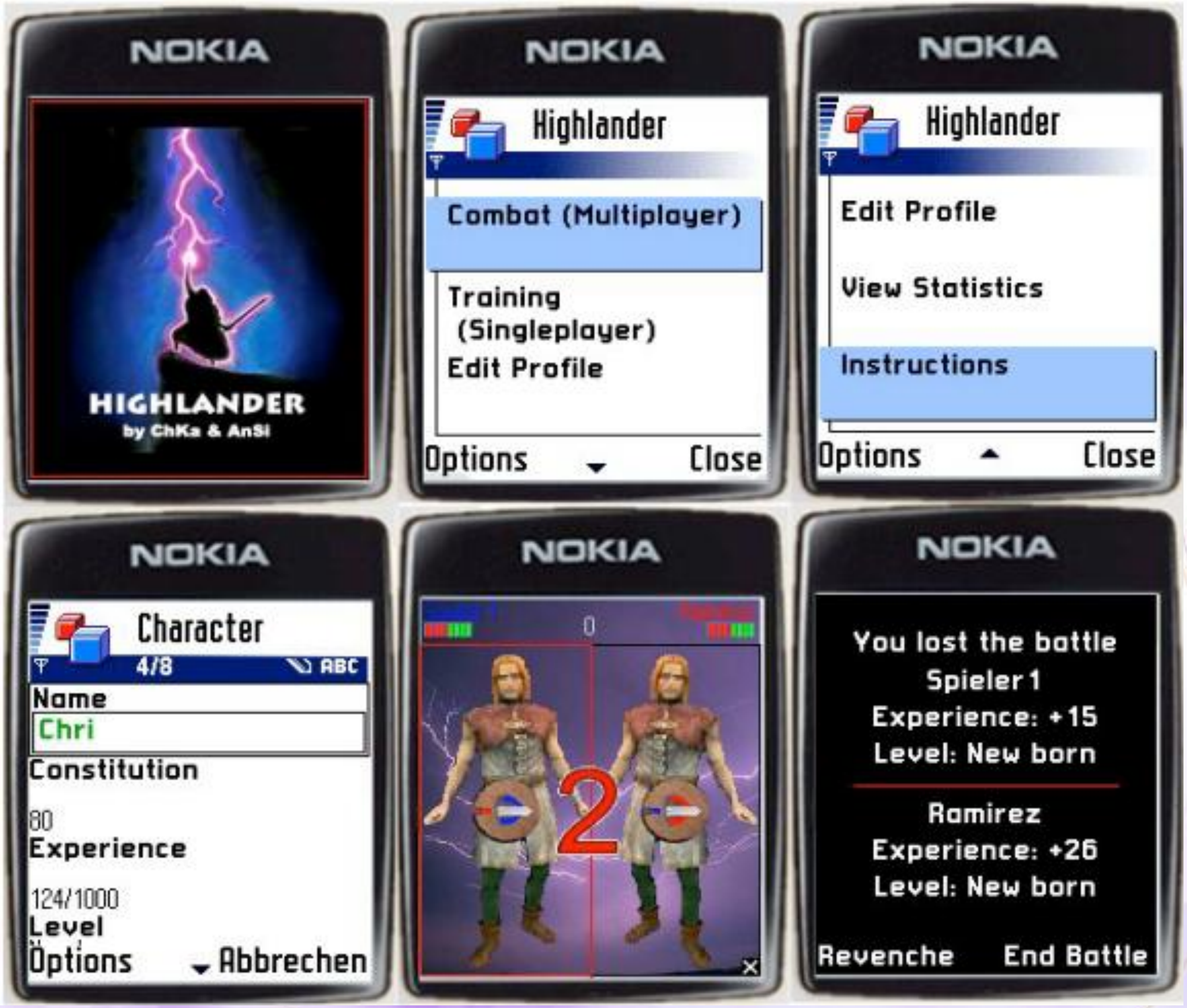
Basic Design Issues for Games

- 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
  - → Saving and pausing should be supported



- 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.

Example games vs. menu



Name
Chri
Constitution
80
Experience
124/1000
Level

You lost the battle
Spieler 1
Experience: +15
Level: New born

Ramirez
Experience: +26
Level: New born

Mobile Game Usability Recommendations

- Provide a Clear Menu Structure
  - Use only one main menu, accessible with the left soft key.
  - Keep the menu short: left soft key for OK, select, and menu; 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.
- 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.

Mobile Game Usability Recommendations

- 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.
- 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.
- Use Natural Controls
  - Horizontal and vertical movement: 2, 4, 6, 8 + arrow keys
  - Diagonal movement: 1, 3, 7, 9
  - Action button: 5
  - Design the game so that it does not lure the user into pressing two keys at once, since many mobile devices do not support simultaneous key presses.

Mobile Game Usability Recommendations

- Enable Save and Pause
  - Provide a simple save-game feature.
  - Have the game auto-save when the user presses the red phone button.
  - Provide a pause mode (left soft key, which goes to the game menu).
  - If the user quits the game from the pause mode, have the game auto-save.
- 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.

Mobile Game Usability Recommendations

- Conform to Real-World Expectations
  - Do not end the game arbitrarily.
  - Implement a realistic physics model if relevant (e.g., racing games).
  - 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.
- Go Easy on the Sound
  - Provide sound for feedback
  - Ensure that the game is playable with the sound off
  - Provide an easy way to turn sound off within the game.
  - No annoying sounds
  - Avoid background music, if possible.

How do we build websites that work for mobile users with disabilities?

- Main ingredients web standards:
  - HTML, CSS, JavaScript, XML...
- Flavoured with W3C guidelines:
  - Mobile Web Best Practices 1.0
  - Web Content Accessibility Guidelines 2.0
  - User Agent Accessibility Guidelines 1.0

Final Word

- Good interface design is fundamentally about understanding and designing for people
- Much of it seems like common sense - but the skill is in how you do it in practice
- Interface design is becoming ever more important as ‘users’ and interfaces diversify