

**Mobile Information Device Profile
(MIDP)**

Pablo César
E-mail: pcesar@tml.hut.fi
Web page: www.tml.hut.fi/~pcesar
 Helsinki University of Technology
 Telecommunications Software and
 Multimedia Laboratory

Tietoliikenne- ja multimediaalaitteiden laboratorio (TML) September 2004
 T-111,310 Interactive Technology
 T-121,310 User Interface Design Project

Outline

- Java Overview (Editions/Configurations/Profiles)
- Java 2 Micro Edition (J2ME)
 - Connected Device Configuration (CDC)
 - Connected, Limited Device configuration (CLDC)
- Mobile Information Device Profile (MIDP)
 - Restrictions
 - Architecture
 - Applications
 - User Interface
 - Event Handling
 - Multimedia
- MIDP Tools
- References

Tietoliikenne- ja multimediaalaitteiden laboratorio (TML) September 2004
 T-111,310 Interactive Technology
 T-121,310 User Interface Design Project

Java Overview

- Nowadays, trying to target all kind of computer devices
- Editions:
 - Java 2 Enterprise Edition (J2EE): for servers and enterprise computers
 - Java 2 Standard Edition (J2SE): for servers and personal computers
 - Java 2 Micro Edition (J2ME): for embedded devices, PDAs, mobile phones, and Digital television set-top boxes
 - Java Card: for smart cards
- Profile
 - Requirements for a specific vertical market of devices (set of APIs)
- Configuration
 - Minimum platform for a horizontal grouping of devices (VM + core APIs)

Tietoliikenne- ja multimediaalaitteiden laboratorio (TML) September 2004
 T-111,310 Interactive Technology
 T-121,310 User Interface Design Project

Java Overview

Servers	Personal Computers	TV STBs High End PDAs	Mobile Phones Low end PDAs	Smart Cards
Optional Packages	Optional Packages	Optional Packages	Optional Packages	
Java 2 Enterprise Edition (J2EE)	Java 2 Standard Edition (J2SE)	Personal Profile Foundation	MIDP	
		CDC	CLDC	Java Card
Java Virtual Machine			KVM	Card VM
Java 2 Micro Edition (J2ME)				

Tietoliikenne- ja multimediaalaitteiden laboratorio (TML) September 2004
 T-111,310 Interactive Technology
 T-121,310 User Interface Design Project

J2ME

- Defines two Configurations:
 - CDC: High end consumer devices
 - Processor: 32 bits
 - RAM Java Memory: around 2MB
 - ROM Java Memory: around 2.5MB
 - CLDC: Low end consumer devices
 - Processor: 16 bit/16 MHz or higher
 - Java total memory: 160-512 KB
 - Power: Limited power
- CDC (Connected Device)
 - Personal Profile
 - Adds support for lightweight AWT
 - Foundation Profile
 - Basic application APIs (no GUI)
- CLDC (Connected Limited Device)
 - Mobile Information Device Profile (MIDP)
 - Application APIs + GUI APIs

TV STBs High End PDAs	Mobile Phones Low end PDAs
Optional Packages	Optional Packages
Personal Profile	MIDP
Foundation	CLDC
CDC	KVM
JVM	

Tietoliikenne- ja multimediatalvikien laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project

MIDP Restrictions

- No floating point data types (until CLDC 1.1):
 - Float and Double are not valid types
 - No hardware floating point unit
- No finalization:
 - Object.finalize class not support, developers have to clean up
- No Java Native Interface (JNI)
 - It uses a lot of memory
 - It is a security hole
- No user defined class loaders
 - Security: one could override a system class loader
- No reflection
 - Security and heavy operation: Program's structures and metadata not visible for VM
- Limited error handling
- No weak reference
- No thread groups or daemon threads

Tietoliikenne- ja multimediatalvikien laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project

MIDP Architecture

Game
User Interface
Media

Application Management

End-to-End Security

Local Data Storage
Push Registry
Connectivity
OTA provisioning

Mobile Phones
Low end PDAs

Optional Packages

MIDP

CLDC

KVM

Tietoliikenne- ja multimediatalvikien laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project

MIDP Architecture

- Basic Layer
 - Local data storage
 - Persistent storage of data
 - Push Registry
 - Allows MIDlets to be launched in response to incoming network connections (e.g., alerts)
 - Connectivity
 - Connection for datagrams, sockets, and server sockets
 - OTA provisioning
 - Simplifies the way applications are delivered to consumers
 - Ability to dynamically deploy and update applications over-the-air (OTA). How applications are discovered, installed, updated...
- Second Layer
 - End-to-End security
 - MIDP provides a robust security model: http and https connections, and public key management

Tietoliikenne- ja multimediatalvikien laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP Architecture

- **Third Layer**
 - Application Management
 - Applications are called MIDlets, manager in charge of controlling their state
- **Higher Layer**
 - Game
 - Specific game API for developers
 - User Interface
 - Both High Level (ready made widgets), and high level API (developer can paint on the screen)
 - Media
 - Audio utilities API

Tietotieteiden ja multimediaalaitteiden laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP Architecture: packages

- **Core packages**
 - java.lang: subset of J2SE (does not include Float, for example)
 - java.util: subset of J2SE
 - java.io: classes for input and output through data streams
- **Networking package**
 - javax.microedition.io: introduces the concept of Connector (where there is not explicit protocol)
- **Application Lifecycle package**
 - javax.microedition.midlet: defines MIDP applications and their interaction with the environment
- **Persistence package**
 - javax.microedition.rms: provides mechanisms for store data and later retrieve it

Tietotieteiden ja multimediaalaitteiden laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP Architecture: packages

- **User Interface packages**
 - javax.microedition.lcdui: provides a set of features for implementation user interfaces
 - javax.microedition.game (since MIDP 2.0): provides a serie of classes to easiness game development
- **Public Key package (since MIDP 2.0)**
 - javax.microedition.pki: provides mechanisms for secure connections
- **Audio packages (since MIDP 2.0)**
 - javax.microedition.media: part of the Mobile Media API (MMAPI), defines Player, Manager...
 - javax.microedition.media.control: defines the specific control types that can be used with a player

Tietotieteiden ja multimediaalaitteiden laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP Architecture

- **Bluetooth API**
 - Permitting bluetooth transmission
- **Web Services API**
- **Wireless Messaging API**
- **Mobile Media API**
 - Similar to JMF
- **SIP API**
- **Mobile 3D Graphics API**
- **Location API**

Mobile Phones
Low end PDAs

Optional
Packages

MIDP

CLDC

KVM

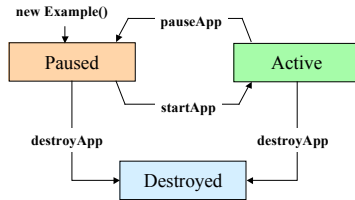
Tietotieteiden ja multimediaalaitteiden laboratorio (TML) September 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP Applications: MIDlets

- MIDP applications are called MIDlets
- Similar to web applets or MHP Xlets
- Three states: Paused, Active, Destroyed



Tietotieteiden ja multimediaalabiorin (TML) Syyskuu 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDlets Example

```

import javax.microedition.midlet.*;
public class Example extends MIDlet{

    /* Start midlet: to allocate desired system resources and initialise */
    public void startApp(){
    }

    /* When midlet is not active: suspend background activities and release
    resources */
    public void pauseApp(){
    }

    /* When midlet is in the end of life cycle */
    public void destroyApp(boolean unconditional){
    }
}
  
```

Tietotieteiden ja multimediaalabiorin (TML) Syyskuu 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP User Interface

- User interface for handheld devices are different from PCs
 - Smaller display size
 - Input device not always include pointing device
- MIDP is not a subset of AWT!!!!!!
 - AWT is designed for PCs
 - AWT assumes certain interaction models (e.g., mouse)
 - AWT assumes the use of Windows (drag, move, resize)

Tietotieteiden ja multimediaalabiorin (TML) Syyskuu 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project

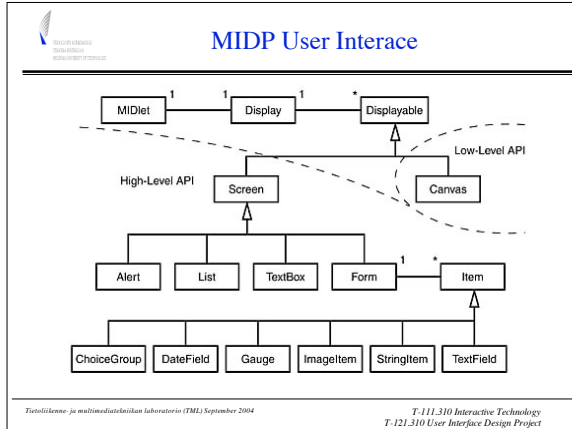


MIDP User Interface

- Basic Class (Display): is the actual output device of the mobile phone, one display has multiple Screens
 - 1 Application -> 1 Display
- Basic interface (Displayable): each screen of the services
 - 1 Application -> multiple Displayable objects
- Two kind of Displayable Objects (Cannot be mixed):
 - Screen: High Level API, each MIDP application has a Display in which a single screen is shown (title, multiple commands, ticker)
 - Canvas: Low Level API, it is extended for drawing
- High Level API
 - Intended for applications where portability is important
 - High Level widgets, developer has no control on their look (appearance) and feel (interaction)
- Low Level API
 - Intended for applications where portability is not as important as control over the graphics
 - Developer has full control over what is drawn, where, and how

Tietotieteiden ja multimediaalabiorin (TML) Syyskuu 2004

T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP User Interface: High Level API

- **TextBox:**
 - Screen that allows the user to enter and edit text
- **List**
 - Screen that contains a list of choices
 - Implicit: like a menu
 - Exclusive: select one element (radio buttons)
 - Multiple choice: select many elements (check boxes)
- **Alert**
 - Screen that shows a message and an optional image to the user

Tietotieteiden ja multimediaalaitteiden laboratorio (TML) September 2004
T-111,310 Interactive Technology
T-121,310 User Interface Design Project

MIDP User Interface: High Level API

- **Forms**
 - Screen that contains a combination of items
- **Items:**
 - Components of a Form
 - ImageItem, StringItem, TextField, ChoiceGroup, DateField, Gauge

Tietotieteiden ja multimediaalaitteiden laboratorio (TML) September 2004
T-111,310 Interactive Technology
T-121,310 User Interface Design Project

MIDP User Interface: High Level API Example

```

// The actual display, where to place the widgets
private Display midletDisplay;

// Retrieve the display from the static display object
midletDisplay = Display.getDisplay(this);

// Create a TextBox containing the "Hello Midlet World!!"
// message (TextBox is a subclass of Screen)
TextBox textBox = new TextBox("Hello Midlet", "Hello
Midlet World!!", 256, 0);

// Set the current display of the midlet to the textBox screen
midletDisplay.setCurrent(textBox);

```

Tietotieteiden ja multimediaalaitteiden laboratorio (TML) September 2004
T-111,310 Interactive Technology
T-121,310 User Interface Design Project



MIDP User Interface: Low Level API

- Developer extends Canvas class and override the paint method to create her own widgets
- Allows developers to:
 - Control what is drawn on the display
 - Handle primitive events (e.g., Key Released)
 - Access concrete keys and other input devices
- Similar to AWT's Graphics:
 - Drawing model: there is not composition of images, the canvas is visible in the display or not visible
 - Double buffer: canvas can be stored as a off-screen image buffer
 - Coordinate system: origin is the upper-left corner of the display
 - Translation: the coordinate system can be translated over X or Y axis
 - Clipping: clipping is possible (so, no modifications are done over constant pixel values)
 - Color model: both gray scale (0 to 255) or color (24 bits)
 - Fonts: requested to the device (never created)



MIDP User Interface: Low Level API

- Drawing primitives:
 - Lines, Arcs, Rectangles, Rounded Rectangles
 - Can be dotted or solid lines
- Images
 - Immutable: images loaded from resource bundles (jar file), files, or network
 - Mutable: images created as blank images (only white pixels), it has same characteristics as Display
- Char, chars, string, substring
 - drawString(string, x, y, anchor);
 - Anchor point: to minimise computation defined as a horizontal constant (LEFT, HCENTER, RIGHT) and a vertical one (TOP, BASELINE, BOTTOM)



MIDP User Interface: Low Level API Example

```
// import the correct packages
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class ExampleMidlet extends MIDlet {
    // Constructor
    public ExampleMidlet () {
    }

    public void startApp() {
        // My defined class
        Canvas canvas = new ExampleCanvas();
        // The output device of the mobile phone
        Display display = Display.getDisplay(this);
        display.setCurrent(canvas);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
}
```



MIDP User Interface: Low Level API Example

```
// my own canvas extends Canvas class
public class ExampleCanvas extends Canvas {

    // it must override the paint method
    public void paint(Graphics g) {

        // we can use fonts
        g.setFont ( Font.getFont ( Font.FACE_MONOSPACE,
                                   Font.STYLE_SMALL, Font.SIZE_PLAIN ) );

        // can set colors
        g.setColor(255, 0, 0);
        // can use drawing primitives
        g.fillRect(0, 0, getWidth(), getHeight( ));
        // we can draw strings
        g.drawString("Hello World!", 0, 0, g.TOP | g.LEFT);
    }
}
```



MIDP User Interface: Low Level API Example

```
public ImageDemoCanvas () {
    // mutable and immutable images
    Image mutableImage = Image.createImage (30,30);
    Image immutableImage = Image.createImage("pablo.png");

    // size of immutable image
    int image_width = immutableImage.getWidth();
    int image_height = immutableImage.getHeight();

    // create screen for mutable image
    Graphics g = mutableImage.getGraphics();
    g.drawImage(immutableImage, 20, 20, TOPLEFT);

    // borders of image
    g.setColor(0,0,128);
    g.drawRect(18,18, image_width+1, image_height+1);
    g.drawRect(18,18, image_width+2, image_height+2);
}
```



MIDP Event Handling

- Scrolling a list: performed by the device, application is not aware of it
- Commands:
 - Construct that encapsulates the semantic information of an action
 - BACK, CANCEL, EXIT, HELP, ITEM, OK, SCREEN, and STOP
 - Commands can be added to each screen
 - No positioning or layout is possible
 - Device decide its look (e.g., soft buttons)
 - Three parameters:
 - Label: shown to the user as a hint
 - CommandType: meaning of the command
 - Priority: to define the order of access
 - Command listener: one per displayable object
- Input events (Low Level API):
 - Methods of the Canvas such as keyPressed() can be overridden
 - Key Events: when a key is pressed, released, or repeated
 - Action Events: game actions such as left, right...
 - Pointer Events: pressed, released, or dragged



MIDP Event Handling: Example

```
public class Example extends MIDlet implements CommandListener{
    // define the commands
    private Command backCommand = new Command("Back", Command.BACK, 0);
    private Command mainMenuCommand = new Command("Main", Command.SCREEN, 1);

    // add a command listener to one of the Displayable objects
    public void startApp() {
        textBox.addCommand(backCommand);
        textBox.setCommandListener( (CommandListener) this);
    }

    /* to handle the action */
    public void commandAction(Command command, Displayable screen){
        String label = command.getLabel();

        // when back soft button is pressed
        if (label.equals("Back")){
            // go back to the previous screen
        }
    }
}
```



MIDP Multimedia

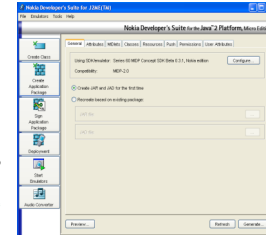
- Game API (MIDP 2.0)
 - GameCanvas: subclass of Canvas with specific game functionalities
 - Layer: visual element of the game (abstract class)
 - Sprite: animated layer that can display several graphical frames
 - TiledLayer: enables the creation of large areas of content, but at a low resource cost
 - LayerManager: to control the layers and the user's view
- 3D Graphics API (optional package)
 - Two APIs for displaying 3D content
 - Immediate mode API: create and manipulate 3D elements directly
 - Retained mode API (scene graph): load and display entire 3D scenes
 - Also defines a file format for scene graphs, .m3g.
- Mobile Media API (MMAPI) (optimal package)
 - Extends MIDP functionality by providing audio, video and other time-based multimedia support
 - It is a thin Java layer completely platform dependent
 - It is not IMF
 - MIDP 2.0 includes the audio-only subset

MIDP Tools: Emacs



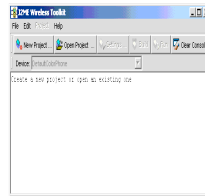
MIDP Tools: Nokia Developer's suite

- Provides developers with utilities for creating and deploying applications
- SDK for MIDP 1.0 and 2.0
- Can be installed in PC and provides an emulator
- Can be used as:
 - Standalone
 - Integrated plug-in to Borland JBuilder and SUN ONE Studio
- Web page: <http://www.forum.nokia.com/main/1,6566,034-2,00.html>

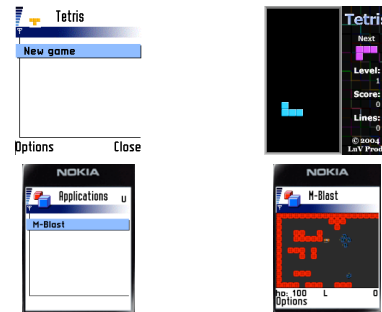


MIDP Tools: J2ME Wireless Toolkit

- The (J2ME) Wireless Toolkit is intended for developing wireless applications based on MIDP
- Two versions:
 - J2ME Toolkit 2.1
 - J2ME Toolkit 2.2
- Toolkit includes:
 - Emulation environment
 - Performance optimization
 - Tuning features
 - Examples for developers
- Web page:
 - <http://java.sun.com/products/j2mewtoolkit/download.html>



MIDP Examples: Tetris & MBlast





References: Basic

- Starting points:
 - Sun: <http://java.sun.com/j2me/index.jsp>
 - Nokia forum: <http://www.forum.nokia.com/main.html>
- Code examples:
 - <http://developers.sun.com/techtopics/mobility/midp/samples/>
 - http://www.forum.nokia.com/main/0..1_0_15.00.html
 - <http://developers.sun.com/techtopics/mobility/midp/articles/game/>
 - <http://developers.sun.com/techtopics/mobility/midp/articles/ui/>
- FAQs
 - <http://java.sun.com/j2me/reference/faqs/>
- APIs
 - MIDP 1.0: <http://j2medevices.com/documentation/midp1.0>
 - MIDP 2.0: <http://www.wmlscript.it/j2me/api20/overview-summary.html>



References: User Interface

- MIDP programming with J2M2
 - http://www.developer.com/java/j2me/article.php/10934_1561591_1
- MIDP GUI programming
 - http://www.onjava.com/pub/a/onjava/excerpt/wirelessjava_ch5/index1.html
- MIDP User Interface Libraries
 - <http://developers.sun.com/techtopics/mobility/midp/chapters/pwdevices/ch09.pdf>
- Device programming with MIDP
 - <http://www.javaworld.com/javaworld/jw-01-2001/jw-0105-midp.html>
- Low Level API
 - <http://developers.sun.com/techtopics/mobility/midp/tips/llapi/>
- High Level API
 - <http://www.samsublishing.com/articles/article.asp?p=27657>
- GUI Design in MIDP
 - http://www.csie.mcu.edu.tw/~hsong/mobile/05_midp_gui.pdf
- MIDP Event Handling
 - <http://developers.sun.com/techtopics/mobility/midp/articles/event/>



References: Specifications

- <http://java.sun.com/j2me/docs/index.html>
 - MIDP 2.0, (JSR 118):
 - <http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html>
 - MIDP 1.0, (JSR 037):
 - <http://jcp.org/aboutJava/communityprocess/final/jsr037/index.html>
 - CLDC, V1.0 (JSR-030):
 - <http://jcp.org/aboutJava/communityprocess/final/jsr030/index.html>
 - CLDC, V1.1 (JSR-139):
 - <http://jcp.org/aboutJava/communityprocess/final/jsr139/index.html>



References: Tools and Documentation

- J2ME Toolkit:
 - Download:
 - http://java.sun.com/products/j2mewtoolkit/download-2_1.html
 - Documentation:
 - http://java.sun.com/j2me/docs/wtk2.1/user_html/index.html
- Nokia developer's suite:
 - Download:
 - <http://www.forum.nokia.com/main/0..034-2.00.html>



Thank you!

Questions?
Comments?