



JavaOneSM

Sun's 2002 Worldwide Java Developer Conference™

JSR-118 (MIDP v2.0)

Session 1 of 2

Dr. James E. Van Peursem

Spec Lead, JSR-118

Chief Architect, J2ME Platform

Motorola, Inc.

Mark Patel

Senior Architect, J2ME Platform

Motorola, Inc.

Overall Presentation Goal

Learn about new capabilities being introduced in the next generation of the Mobile Information Device Profile (v2.0) specification.



Learning Objectives

- As a result of this presentation, you will get an overview of the new features being introduced in MIDP 2.0
- In addition, you will gain in-depth knowledge about the following new MIDP 2.0 features:
 - UI additions
 - Game APIs
 - Sound APIs



Speaker's Qualifications

- Jim Van Peursem is the Specification Lead for JSR-118 and is also the Chief Architect of Motorola's technology for the J2ME™ Platform and a Distinguished Member of Technical Staff at Motorola, Inc.
- Mark Patel is a Senior Architect at Motorola, Inc. and is also a member of the JSR-118 Expert Group



Presentation Agenda

- MIDP 2.0 overview
- Application Delivery and Billing
- New UI capabilities
- New Game capabilities
- New Sound capabilities
- Q&A



JSR-118 (MIDP 2.0)

- Java™ Specification Request (JSR) filed April 2001
- Expert group consists of nearly 60 companies including network operators, manufacturers, content developers, technology providers and individuals
- Producing the specification for MIDP 2.0
- MIDP 2.0 is a complete specification that includes all of the functionality and is backward compatible with MIDP 1.0



MIDP 2.0 New Functionality

- Overview presented today, and detailed discussions contained in session TS 3006:
 - New Security Model through Trusted MIDlets
 - HTTPS
 - Enhanced Networking and Push
- Detailed discussions today:
 - Application Delivery
 - Enhanced UI
 - Game
 - Sound



Security

- MIDP 1.0 used the “sandbox” security model, similar to Applets, where MIDlets are only exposed to “safe” APIs
- MIDP 2.0 specifies how MIDlet Suites can be cryptographically signed so that their authenticity and origin can be validated
- A new security framework is also specified such that some “trusted” or “privileged” MIDlets, based on their signature, can get access to APIs outside the sandbox, which are not safe to expose to all MIDlets
 - For example, originating a phone call, using the Push networking features, as well as privileged features from future JSRs



HTTPS/SSL

- MIDP 1.0 included support for HTTP
- MIDP 2.0 adds support for HTTPS, which enables end-to-end transactional security
- This can be implemented either through SSL, TLS, or WTLS; The MIDlet can query the method used for establishing the connection at runtime
- In addition, in MIDP 2.0, it is possible to use SSL/TLS with raw sockets, not just HTTPS; This enables more flexibility



Enhanced Networking and Push

- MIDP 1.0 included support for HTTP
- For MIDP 2.0, these additional protocols are specified:
 - HTTPS
 - Serial port
 - Sockets
 - Server Sockets
 - Datagrams
- In addition, MIDP 2.0 also includes a new network push capability, such that MIDlets can request that the handset listen for certain connections or data on its behalf, and launch the MIDlet and give it the connection when this occurs



Detailed Topical Discussions

- Application Delivery
- Enhanced UI
- Game
- Sound



Application Delivery

- After the MIDP 1.0 specification was published, an Over The Air (OTA) Recommended Practice document was published that described:
 - How users can initiate the download of MIDlets over the air through the browser resident on the device
 - The details for protocol and cookie handling related to the downloading of MIDlets
 - Notifications to be sent to the server (best effort, but not reliable) when the MIDlet is successfully installed
 - How MIDlet upgrades are to be handled



Application Delivery Enhancements

- MIDP 2.0 formally incorporates the Over The Air (OTA) Recommended Practice document associated with MIDP 1.0
- Enhancements were made to this specification to enable reliable delivery of server notifications
- Notifications can also be sent for successful application installation as well as deletion
- Cookie support was removed, as some networks/gateways may not pass them to the client; URL-rewriting is the suggested alternative





JavaOneSM
Sun's 2002 Worldwide Java Developer Conference™

UI, Games, and Sound

Mark Patel

Senior Architect, J2ME Platform
Motorola, Inc.

Enhanced UI—Overview

- Backward compatible with MIDP 1.0
- Builds upon the features of LCDUI
 - Custom Item support
 - Layout control
 - Graphics enhancements
 - Miscellaneous ‘tweaks’



Enhanced UI—CustomItem

- Allows custom Item classes to be created and added to Forms
- Developer has control over look & feel
- Full support for input events and user interaction



Enhanced UI—Layout Control

- Dramatically improves the layout capabilities of LCDUI while maintaining portability
- Items are grouped into rows
- Items have a preferred width and height
- Items can be made to expand or shrink as needed to fit the screen
- Spacer Items can be added to control spacing



Enhanced UI—Graphics

- Transparent image support
- Enhanced drawImage
 - Specific region
 - Transforms
- get/draw regions of RGB values
- fillTriangle



Enhanced UI—Miscellaneous

- Additional modes for Gauge
- Command support in List
- ImageItems can be updated
- Etc.

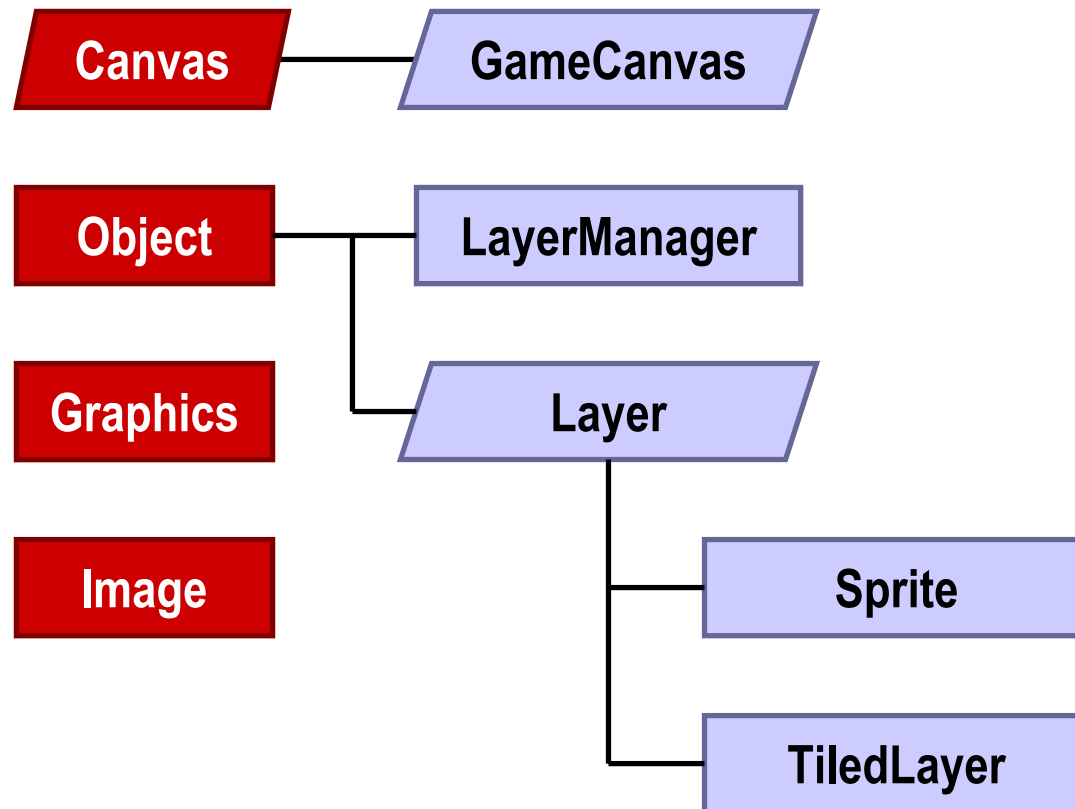


Game API—Overview

- Provides a rich set of features for developing 2D gaming content
 - Enables native implementation
 - Simplifies game development
- Flexible design
- Compatible with LCDUI Graphics classes



Game API—Class Hierarchy



Game API—Layer

- Abstract superclass of all visual elements
 - Size
 - Location
 - Visibility
- May be added to a LayerManager



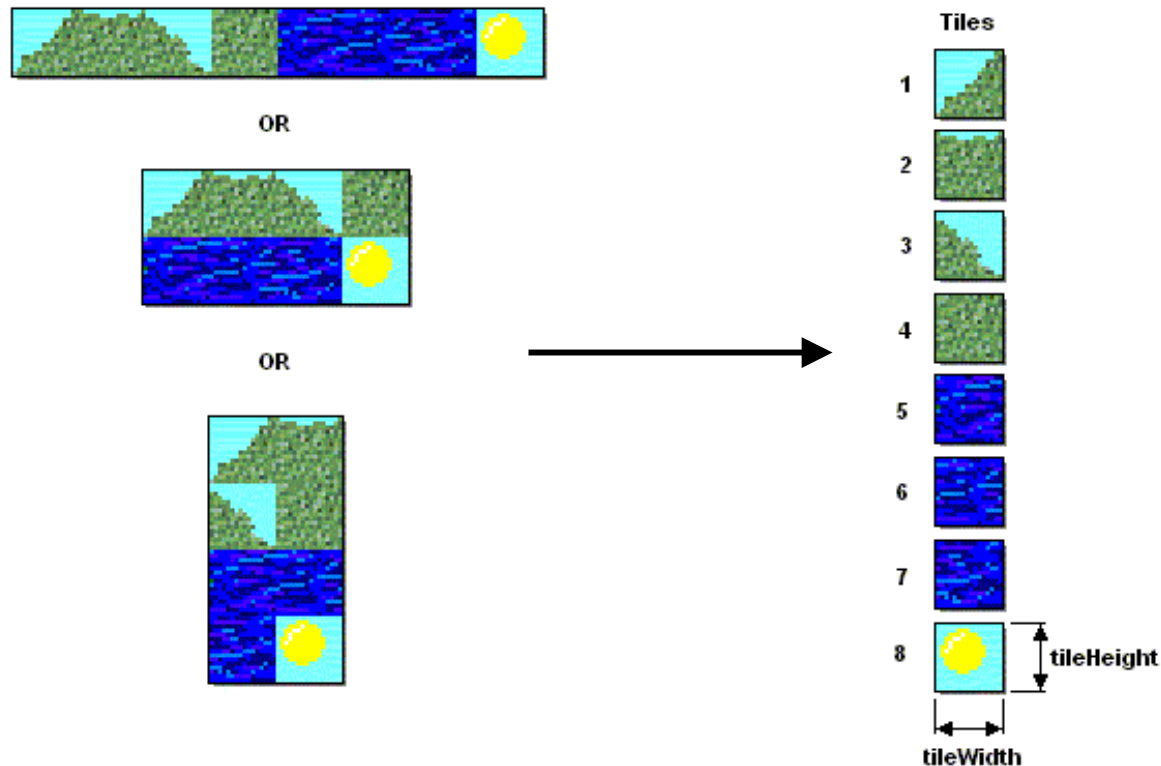
Game API—TiledLayer

- Visual element composed of a **grid** that can be filled using a set of **cells**
- Cells are provided in a single Image
- Animated cells can be defined
- Each element of the grid can contain a specific cell, or can be left empty
- Allows large visual elements to be created



Game API—TiledLayer

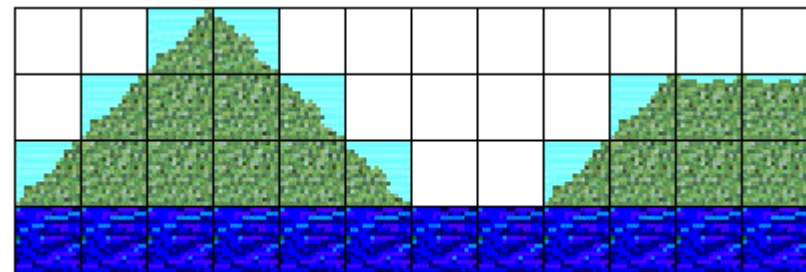
Cells are provided in a single Image



Game API—TiledLayer

Each element of the grid may contain a specific cell

0	0	1	3	0	0	0	0	0	0	0	0
0	1	4	4	3	0	0	0	0	1	2	2
1	4	4	4	4	3	0	0	1	4	4	4
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1



Game API—Sprite

- Visual element that can display one of several **frames**
 - Frames are provided in a single Image
 - Specific frame or animated sequence
- Transforms
- Collision detection
 - With a TiledLayer
 - With another Sprite

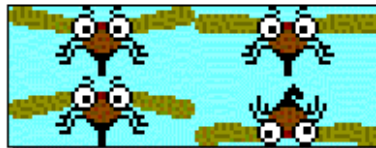


Game API—Sprite

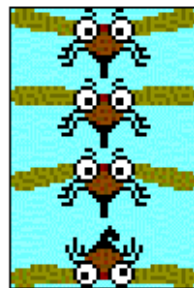
Frames are provided in a single Image



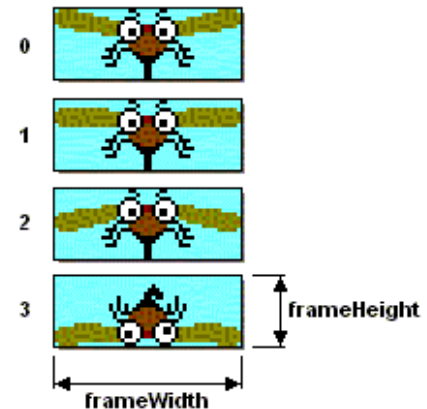
OR



OR



Frames



Game API—Sprite

Various transforms can be applied



TRANS_NONE



TRANS_MIRROR_ROT_180



TRANS_MIRROR



TRANS_ROT_180



TRANS_ROT_90



TRANS_MIRROR_ROT_90



TRANS_MIRROR_ROT_270



TRANS_ROT_270

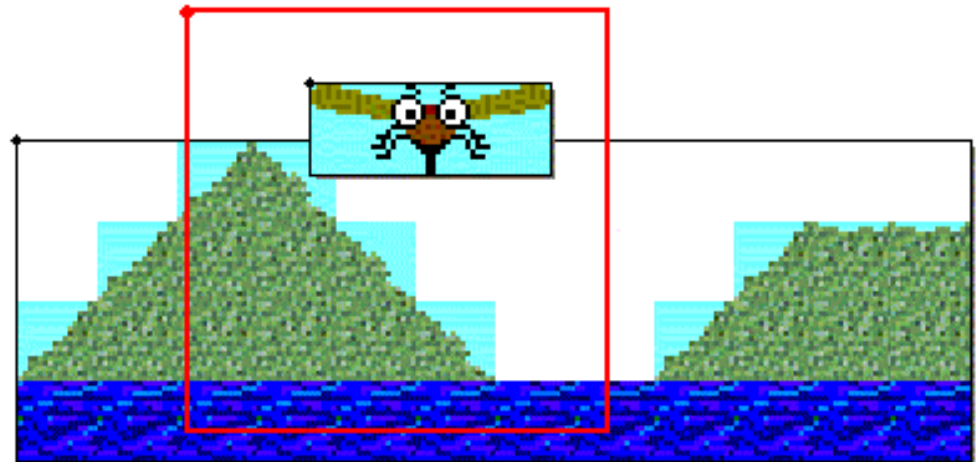
Game API—LayerManager

- Manages an ordered list of Layer objects
 - Layers may be added and removed as needed
 - List order implies Z-order
- Automates the rendering process
 - Definable **view window**
 - Scrolling support



Game API—LayerManager

Definable view window



Game API—GameCanvas

- Enhances Canvas with game-centric features
- Key input support
 - Query game key states directly
 - Simultaneous key presses
- Off-screen buffer
 - May be painted directly
 - Synchronous flushing



Sound Overview

- Provides rich audio capabilities to MIDlets:
 - Tone Generation (required)
 - Sampled sounds such as .wav (optional)
 - MIDI (optional)
- Structured as a subset of JSR-135 (Multimedia)
 - Full upward compatibility



Sound Interfaces

- **Control**
 - Used to control some media processing functions
- **Player**
 - Allows the MIDlet to control the rendering of time-based media data
- **PlayerListener**
 - Used for delivering asynchronous events generated by Players
- **Manager**
 - The access point for obtaining system dependent resources such as Players for multimedia processing



Sound Capabilities

- The interfaces allow the MIDlet to control audio, regardless of its format
- Tone Generation:
 - `Manager.playTone(note, duration, volume);`
- Sampled Sounds and MIDI:
 - `Manager.createPlayer(locator);`
 - The locator is a string in URL syntax that describes the media content
 - `Player.start()`, `Player.stop()`, etc.



Sound Requirements

- All MIDP 2.0 compliant devices must provide support for Tone Generation
- If a device supports sampled audio, it must at a minimum support 8-bit, 8 KHz, mono PCM wav format
- Additional sampled and MIDI formats may optionally be supported



Summary

- MIDP 2.0 is backward compatible with MIDP 1.0
- MIDlets developed for MIDP 1.0 will work on MIDP 2.0 devices
- Significant new functionality is being specified in MIDP 2.0
- Attend Session TS 3006 to learn more details about the new Security solution, HTTPS, and the enhanced networking support



If You Only Remember One Thing...

The new MIDP specification offers significant new capabilities that enables a wide range of new content. Take advantage of these new capabilities!

Review the details at:

<http://jcp.org/aboutJava/communityprocess/review/jsr118/>



Demo

Q&A



JavaOneSM

Sun's 2002 Worldwide Java Developer Conference™

BEYOND BOUNDARIES