Java is a trademark of Sun Microsystems, Inc.

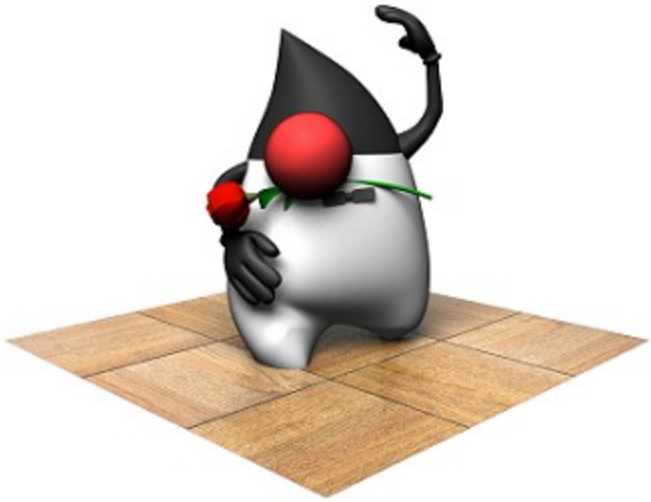
JavaOne℠

Duke's Dancing Partner: Connecting Handheld Game Consoles with Java™ Technology

Max MU
Liang XU
LEE Chuk Munn
Sun Microsystems

Agenda

- Brief introduction of phoneME project
- Make JavaME applications run on PSP and NDS by phoneME
- Get your JavaME applications connected via Project Darkstar



Brief Introduction to

PhoneME project

http://phoneme.dev.java.net/

PhoneME Project

- Open source implementation of JavaME platform
  - PhoneMEFeatrue - CLDC, target to “feature phone” devices
  - PhoneMEAdvanced – CDC, target to high-end smart phone devices, set-top box etc.
- Have been ported to many mobile platforms since it's open sourced in 2006

PhoneME Project

- Advanced
  - High performance
  - Low footprint
  - Optimized for various CPU architecture
- Proved to be stable
- Full-Feature
  - MSA full set (JSR248)
  - 2D/3D graphics, messaging, multimedia, files and directories etc.

PhoneME Porting Layer

- Javacall Porting Interface
  - A well-defined porting interface
  - Can be compiled and tested as separated library without Java VM involved
  - All platform related code should be in Javacall implementation

PhoneME Porting Layer (Cont.)

- Basic functions:
  - Time/Timer/Logging/Events
  - Frame Buffer/File System/Keypad
- Advanced functions:
  - Networking
    - DNS query and ClientSocket is mandatory
    - Datagram and ServerSocket: not important
  - Multimedia
    - MIDI and Wave supporting is essential for games

PhoneME Porting Layer (Cont.)

- More features:
  - TrueType font
  - GPS
  - More media types to support, streaming media
  - Annunciator (Vibrate, Back light, etc.)
  - Messaging (SMS/MMS/CBS)
  - More file system features (for File Connection)
  - Personal Information Management (PIM)

Depending on device ability and your demand

PSPKVM

http://www.pspkvm.com/

Purpose of PSPKVM

- Porting phoneMEFeature to PSP
- Compatible with commercial games/apps
- Enable homebrew developers to develop PSP games/apps in Java
- Unique PSP features
  - 480 x 272, 16:9 wide screen
  - High speed Wifi connection
  - Large memory:
    - 32 MB – PSP 1000, 64 MB – PSP 2000
  - VFPU – Vector Floating Point Unit (http://wiki.fx-world.org)

Hardware Specifications

- CPU
  - MIPS R4000 core, up to 333MHz
  - FPU, VFPU (vector unit) @ 2.6 Gflops
- Memory
  - 32 MB main memory (64MB – PSP2000)
  - 4 MB embedded DRAM
- Display
  - 4.3”, 16:9 widescreen TFT LCD screen
  - 480 x 272 pixel
  - 16.77 million colors

Tools

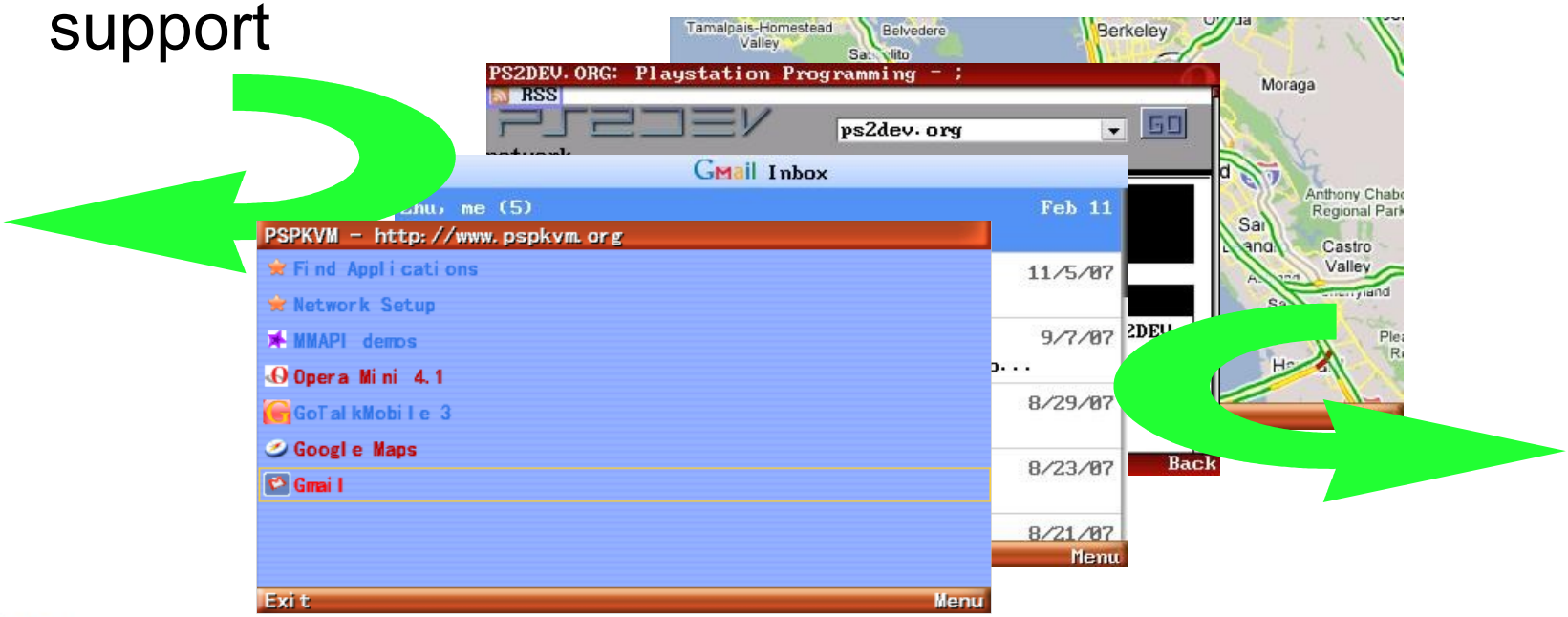
- PSPSDK toolchain (pspdev.org) + libs
  - FreeType - True Type font rendering
  - SDLMixer + libvorbis + libogg (MIDI and WAV playback)
- PSPLink – logging and debugging

Implemented Features

- CLDC/MIDP
- Networking (via PSP's Wifi connection)
- Basic MMAPI (MIDI and WAV playback)
- File Connection
- FreeType2 font rendering
- SVG (JSR226)
- GPS Support (JSR179 via Sony's GPS290 receiver)

Implemented Features (Cont.)

- Virtual keyboard + Chinese Input Method
- Emulate different device models (screen size, keycodes...)
- Full-feature AMS with MVM (Multi-tasking VM) support



Upcoming Features

- OpenGL ES Binding (JSR239)
- M3G (JSR184)
- More MMAPI support
  - Video playback, streaming media

Use Java to Develop PSP Games

- Working on unique device features
  - 480x272, 16:9 screen mode
  - Free, high speed network connection
- Extension APIs for PSP programming
  - Get PSP platform information
  - Get PSP raw buttons status
  - Adhoc mode support (JSR259 might be a better choice, but it's Inactive now)

Limitations

- Unable to run on Sony official firmware
- Lack JIT (Just-In-Time) dynamic compiler for MIPS architecture

Demo

DoubleVision

http://doublevision.sourceforge.net

Purpose of DoubleVision

- To port CLDC/MIDP to Nintendo DS
  - Based on phoneMEFeature open source project
- Map unique NDS features to pMEF
  - Dual screen
  - Touch screen
- Allow developers to write NDS games in Java
  - Use standard MIDP libraries
  - Use proprietary Java libraries
    - Work in planning stages

Tools - Hardware

Nintendo DS

MicroSD

Flash cart

Slot 2 memory expansion

Hardware Specification

- Two TFT backlit screens at 256x192
- 2 ARM processor at 67MHz and 33MHz
- 4MB main memory, Wifi, sound, 12 input buttons, mic
- 2D graphics – 656KB VRAM in 9 banks
  - Assignable to 2 2D graphics core
    - Main – 512KB max, sub – 128KB max
    - Sprite, tile engine, layered background, bitmap, rotation
  - 3D graphics
    - Supported only in main core
    - Renders approximately 6100 vertex

Tools – Software

- devkitPro/devkitARM toolchain with
  - libnds – NDS core libraries
  - libfat – FAT filesystem to access SD card
  - libwifi – networking
  - libram – for accessing slot 2 memory expansion
- Programming options
  - C / C++
  - ARM assembly / thumb

Features – Networking

- Networking – IEEE 802.11
  - Uses settings store in firmware
  - Need to use another application to configure it first
    - Eg. Opera browser
- Network is enable on demand
- Currently do not support server side socket
  - ServerSocketConnection

Features – Multi Screen / Touch Support

- Touch support on sub/bottom screen
- Multi screen support
  - 4 modes
    - 1 normal, 3 switchable modes
    - 1 programmable, non switchable, 2 independent screens
  - Mode3GameCanvas class for mode 3

Screen Config	Main Screen (Top)	Sub Screen (Bottom)
Mode 0	User screen	Log console
Mode 1	Log console	User screen
Mode 2	User screen	
Mode 3	User screen	User screen

Screen Modes

Mode 0

Mode 2

Mode 1

Mode 3

Features – Virtual Keyboard

- Full QWERTY keyboard
  - Native support from libnds
  - MIDP constrains not implemented
  - Caret position not supported

Limitations – Memory

- Internal memory is only 4MB
  - Limit the number of features we could implement
- Tried to use slot2 memory (s2m)
  - JVM™ heap from s2m, JVM runs on main memory
  - s2m and libram is awkward
    - GBA bus only supports 16bit writes, garbage otherwise
      - eg `ram[0] = 0xdead;`
    - libram conflicts with libFAT, bound with lock/unlock
  - Restrain memory writes to 2 bytes
    - `#define memcpy ds_memcpy`
- JVM main loop is written in ARM !!

Limitations – Screen Rendering

- Could not utilize the powerful 2D engine
  - Nice 1 to 1 mapping between MIDP games concepts and NDS tile based graphics
  - Very specific
    - 8x8 tile size, palette, color depth, char and map base
    - Homebrewers use grit to generate images
- Have to use bitmap – not that powerful
  - Consumes lots of VRAM
    - 128K per bitmap – pMEF only supports 16bpp
    - 128K for back buffer
    - Tricky to support sub screen

] main screen

Demo

Project Darkstar

http://www.projectdarkstar.com

What is a Game Server?

- Server-side logic for online games
  - Connected to player/client-side logic via Internet
- Main responsibilities
  - Maintaining critical player data and shared game state
  - Coordinating player communications and interactions
  - Cheat detection and correction
- Deployed across more or physical servers
  - May serve thousands or even millions of players

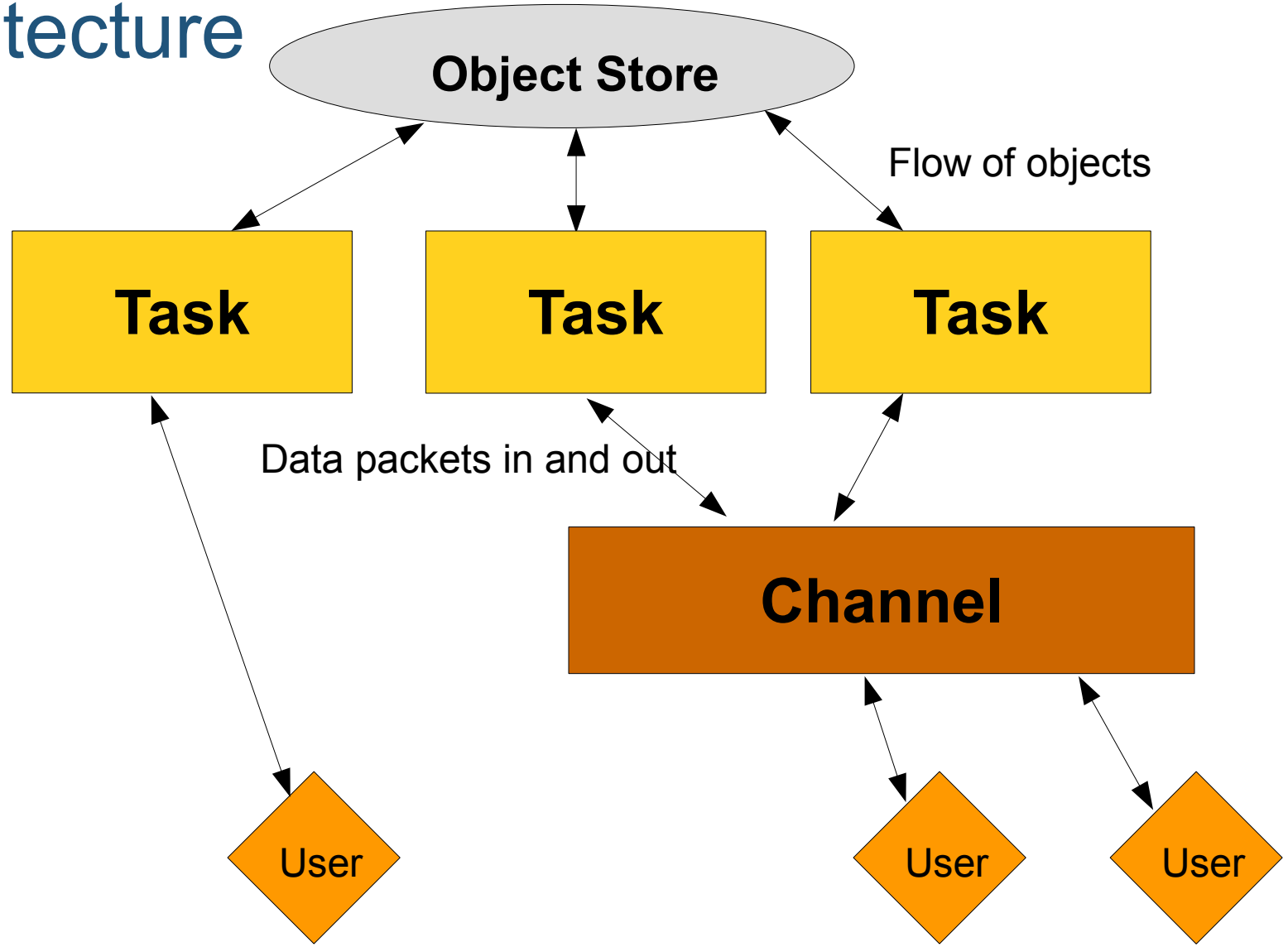
What is Project Darkstar?

- Easy to user server side software infrastructure
  - For online games, virtual worlds, and social networking applications
  - Open source, created by Sun
- Unique architecture
  - Flexible and efficient scaling model
  - Robust data model
  - Simple programming model

Architecture

- Tier 1 – Communication layer
  - Publish and subscribe, direct client/server
  - Order/unordered, reliable/unreliable messages
- Tier 2 – Execution kernel
  - Executes task in response to event
  - Appears mono threaded
- Tier 3 – Object store
  - Persistence store for game objects
  - Transactional to maintain data integrity

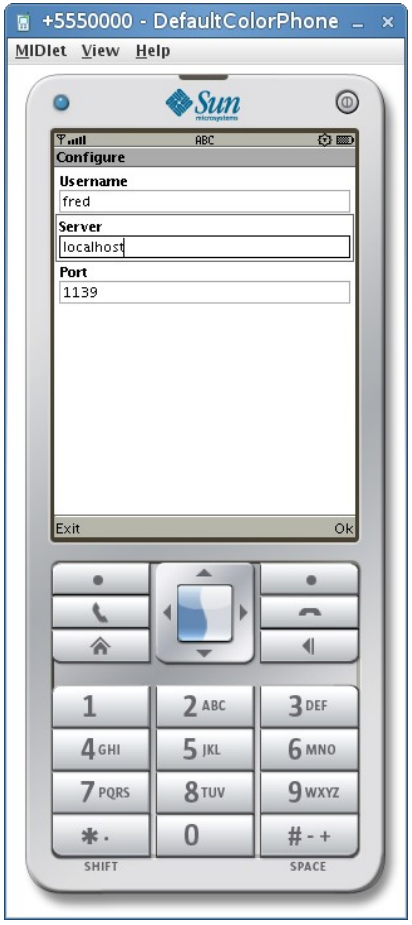
Architecture

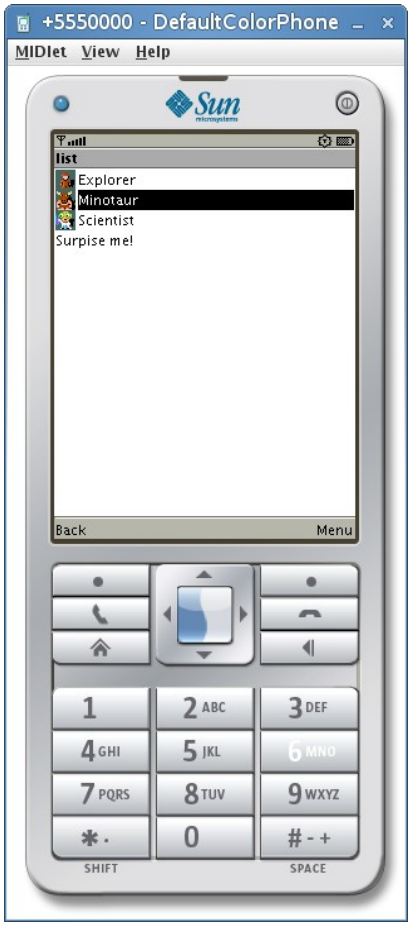


Programming Model and Tools

- Server code written in Java
  - Task, managed objects, communication
  - POJO, Serializable
- Client libraries
  - Java, C (i386, ARM), C#, Python, AS3, etc
- User define byte messages
  - Passed between the Darkstar and game client
- Free to use any framework on game clients
  - Slick2D, GTGE, JMonkey, XNA, etc

Demo Game – MobileHack





Sample screen shots

Darkstar Hack

- Conversion of rogue-like game to MOG
  - Not 'massive', just 3 players
- Each console will login as a character
  - Controls the character
- Position of each players are updated in realtime

Software Used

- Standard CLDC/MIDP 2.1 running on PSP and NDS
- Java Darkstar JME client ported by Max
- Darkstar Pong running on Darkstar Server 0.9.9
- Graphics culled from Nethack
  - http://www.nethack.org

Demo

Summary

- PSPKVM
  - A comprehensive environment for running JavaME commercial applications and homebrews
- DoubleVision
  - Able to squeeze CLDC 1.1/MIDP 2.0 and runtime heap into 4MB of memory
- Project Darkstar
  - Makes writing multi player games extremely easy

Run your applications on different hardware platform with Java Technology

WORA since 1996

Max MU max.mu@sun.com
Liang XU liang.xu@sun.com
LEE Chuk Munn chuk-munn.lee@sun.com