Developing Faster and Flashier Dynamic Graphics With the Java Platform, Micro Edition (Java ME) Personal Basis Profile by Optimizing Java ME CDC

Hinkmond Wong

Senior Staff Engineer
Sun Microsystems, Inc.
http://phoneme.dev.java.net

TS-5724

Goal of This Talk

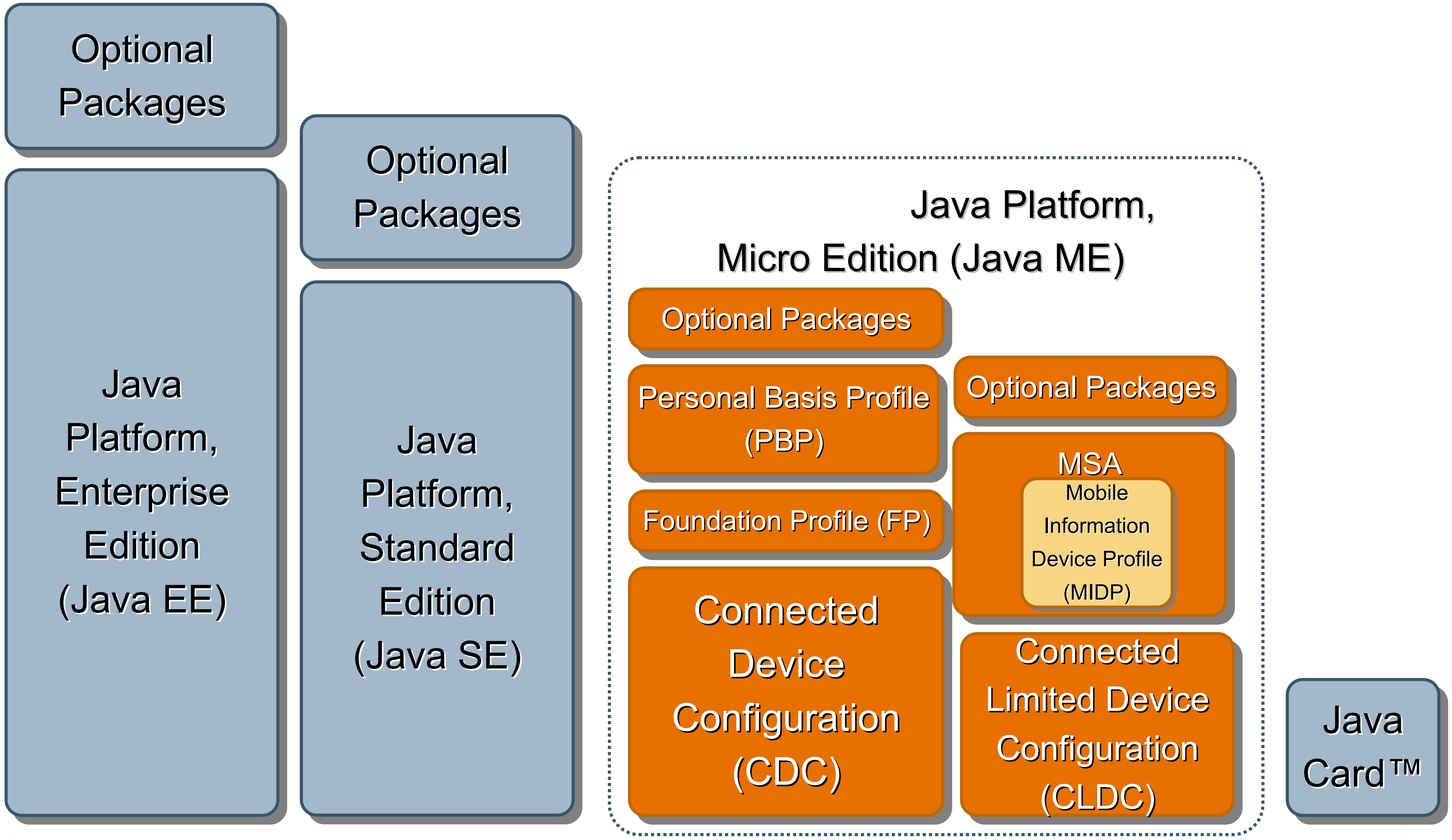
Learn how to use Java™ Platform, Micro Edition (Java ME platform) CDC platform (a.k.a. phoneME™ Advanced Software) to develop flashy and fast graphics with Personal Basis Profile as the example

Agenda

Overview of Java ME Platform CDC
Overview of Personal Basis Profile
Building and Running Xlets
Optimizing Images
Optimizing Animation
Optimizing Games
Optimizing Garbage Collection
Conclusion

Agenda

Java Platform, Micro Edition (Java ME Platform)


Optional Packages
Java Platform, Enterprise Edition (Java EE)
Optional Packages
Java Platform, Standard Edition (Java SE)
Java Platform, Micro Edition (Java ME)
Optional Packages
Personal Basis Profile (PBP)
Foundation Profile (FP)
Connected Device Configuration (CDC)
Optional Packages
MSA
Mobile Information Device Profile (MIDP)
Connected Limited Device Configuration (CLDC)
Java Card™

Agenda

Overview of Java ME Platform CDC

Overview of Personal Basis Profile

Building and Running Xlets

Optimizing Images

Optimizing Animation

Optimizing Games

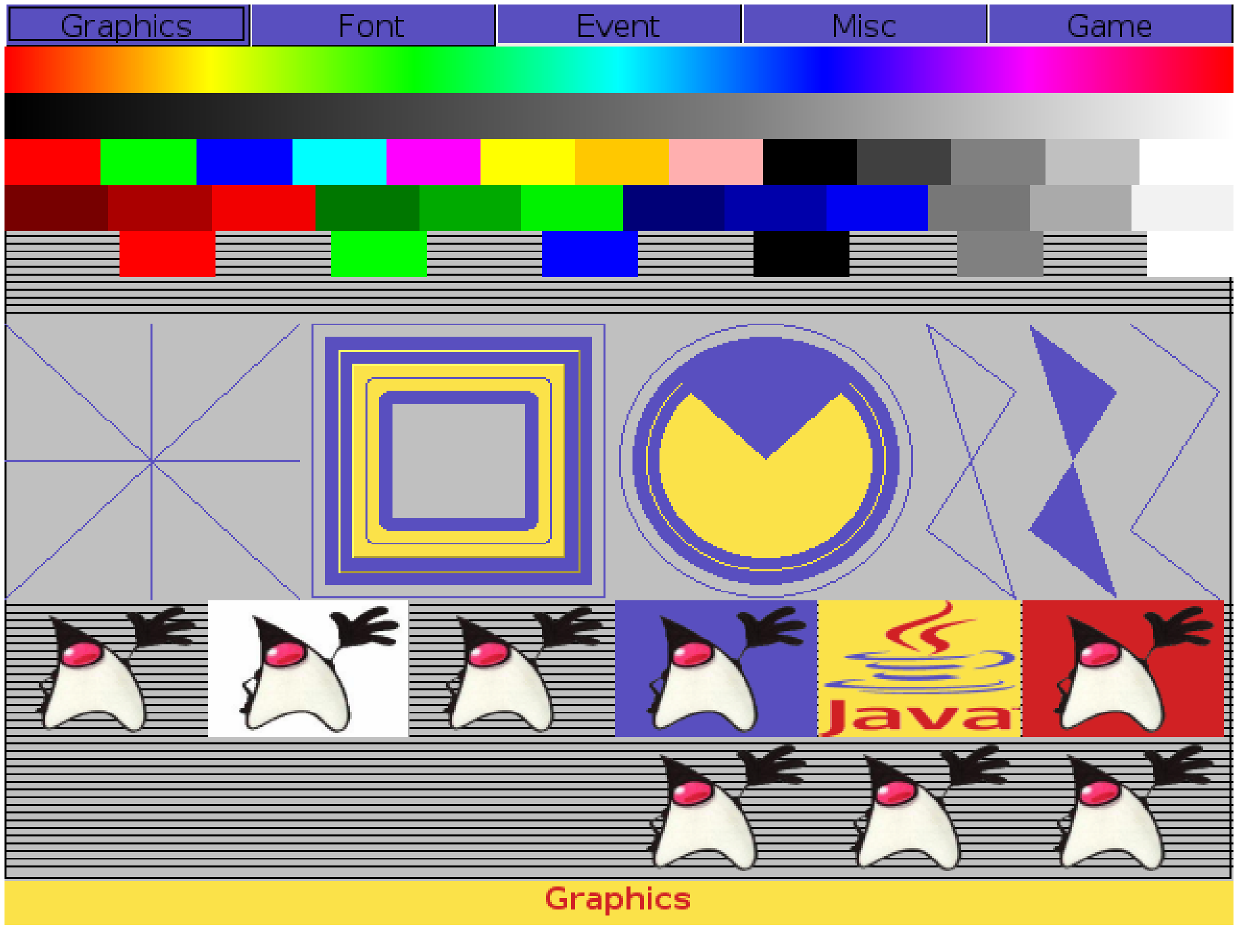
Optimizing Garbage Collection

Conclusion

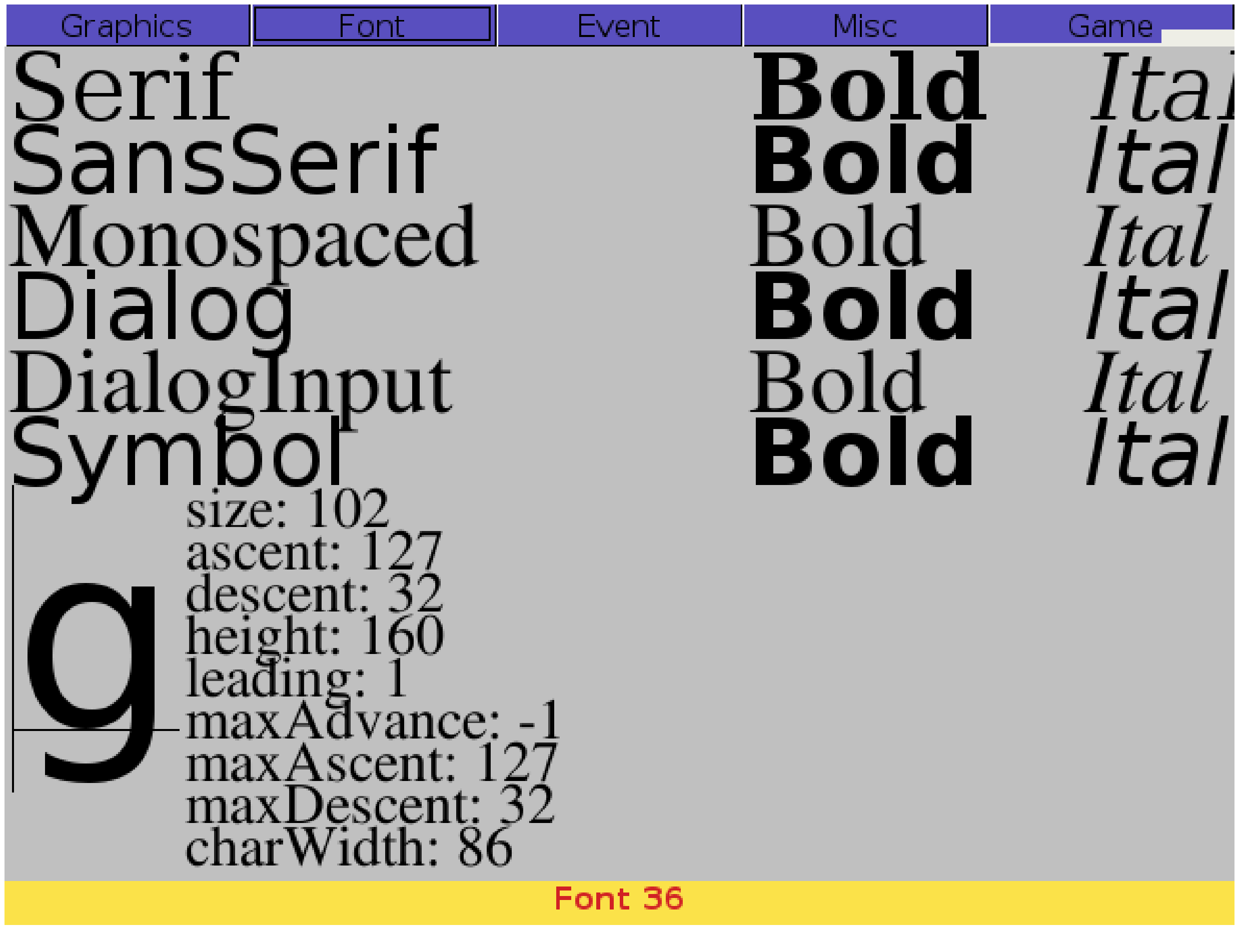
Overview of Personal Basis Profile

- Superset of Java ME Platform CDC/ Foundation Profile
- Standards based GUI
  - Lightweight component toolkits
  - Xlet programming model
  - Core library in Interactive TV and Blu-ray Disc Players (Ex. Sony PlayStation 3)
- API overview
  - Java SE Platform subset
  - java.awt package subset: lightweight components
  - Xlet support

Overview of Personal Basis Profile



Overview of Personal Basis Profile



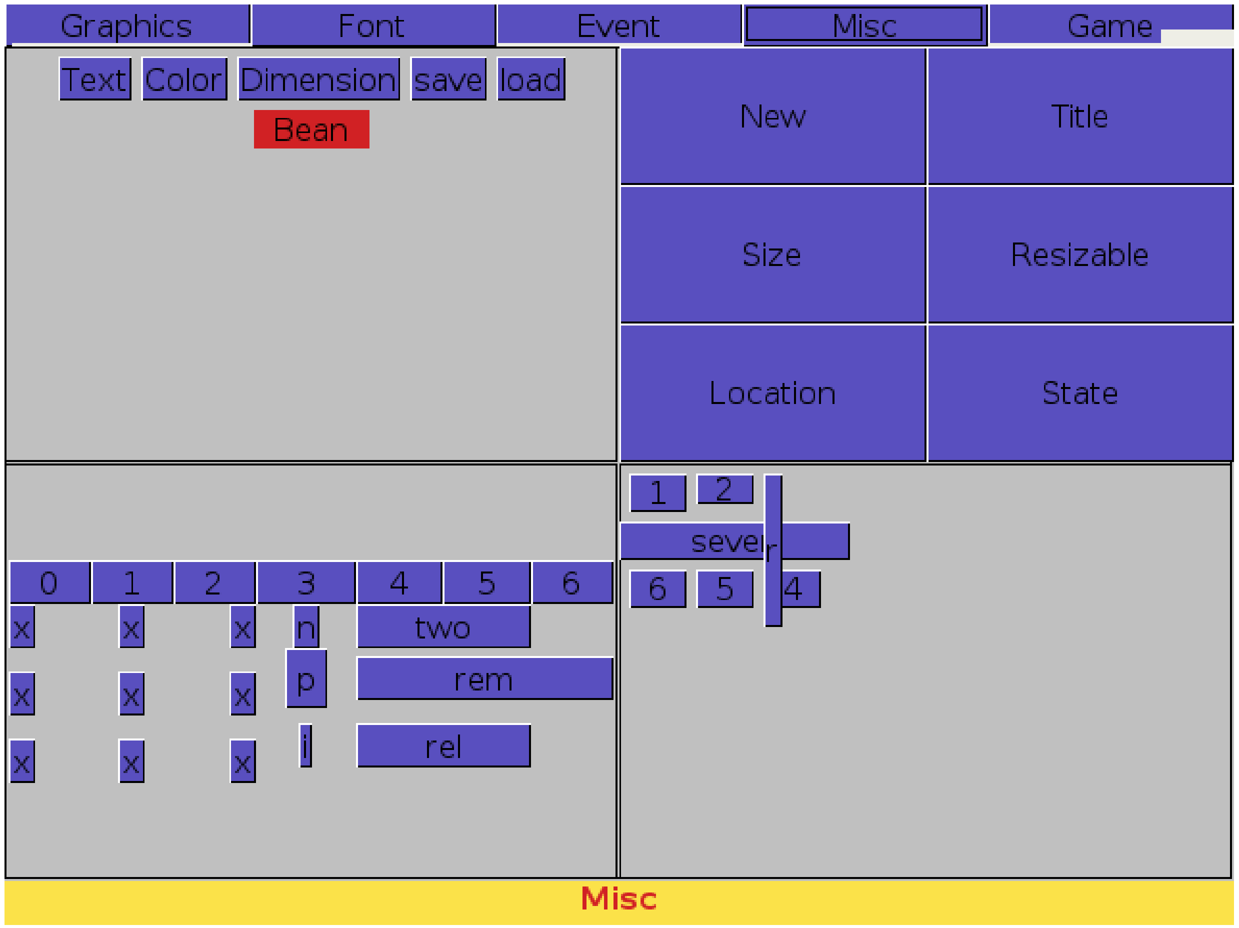
Serif Bold Ital
SansSerif Bold Ital
Monospaced Bold Ital
Dialog Bold Ital
DialogInput Bold Ital
Symbol Bold Ital

g

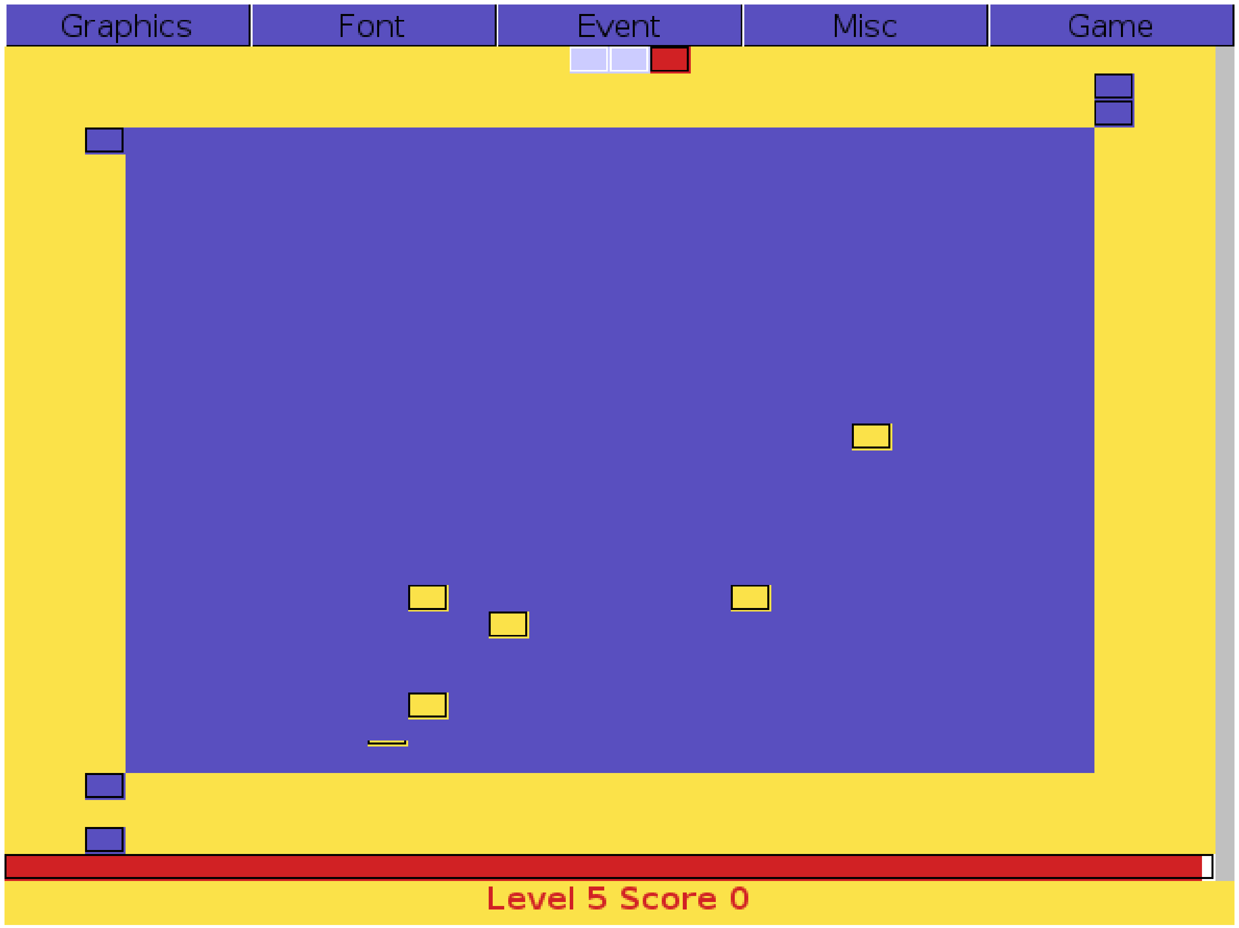
size: 102
ascent: 127
descent: 32
height: 160
leading: 1
maxAdvance: -1
maxAscent: 127
maxDescent: 32
charWidth: 86

Font 36

Overview of Personal Basis Profile



Overview of Personal Basis Profile



Agenda

Overview of Java ME Platform CDC

Overview of Personal Basis Profile

Building and Running Xlets

Optimizing Images

Optimizing Animation

Optimizing Games

Optimizing Garbage Collection

Conclusion

Building and Running

```
# Checkout files
svn co https://phoneme.dev.java.net/svn/phoneme/components/cdc/trunk
cdc
svn co https://phoneme.dev.java.net/svn/phoneme/components/tools/trunk
tools

# Build the platform
cd build/linux-x86-generic
make J2ME_CLASSLIB=basis
```

Building and Running

```
# Build an Xlet
javac -source 1.4 -target 1.4 *.java -classpath
lib/btclasses.zip:lib/basis.jar

# Run an Xlet
bin/cvm com.sun.xlet.XletRunner -name basis.DemoXlet -path
<location_of_JAR_or_class_files>
```

Agenda

Overview of Java ME Platform CDC

Overview of Personal Basis Profile

Building and Running Xlets

Optimizing Images

Optimizing Animation

Optimizing Games

Optimizing Garbage Collection

Conclusion

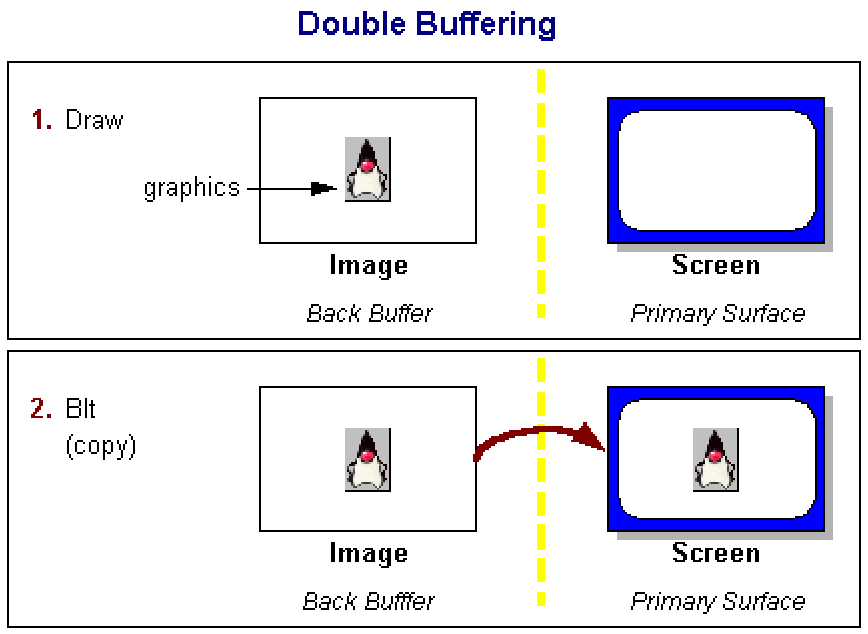
Optimizing Images

- Using MediaTracker (deprecated in Java SE Platform)
  - Track status of a number of media objects
  - Controls priority order in which images are fetched
- Double-buffering
  - BufferedImage and VolatileImage compatibility with Java SE Platform
  - Using off-screen images
  - Primary and back buffers
  - Blitting

Optimizing Images

- Imaging model used in Personal Basis Profile
  - Typical use in apps
  - Blitting pre-defined images
  - Other models
    - Calculated graphics
    - Vector graphics
    - 3D graphics

Overview of Personal Basis Profile



Double Buffering

```
Image buffer = null;
Graphics bufferGC = null;

public void paint(Graphics gc) {
  if (bufferGC == null) {
    buffer = createImage(image.getWidth(c),
                 image.getHeight(c));
    bufferGC = buffer.getGraphics();
  }
  bufferGC.drawImage(image, 0, 0, c);
  gc.drawImage(buffer, 80, 80, c);
}
```

Agenda

Overview of Java ME Platform CDC

Overview of Personal Basis Profile

Building and Running Xlets

Optimizing Images

Optimizing Animation

Optimizing Games

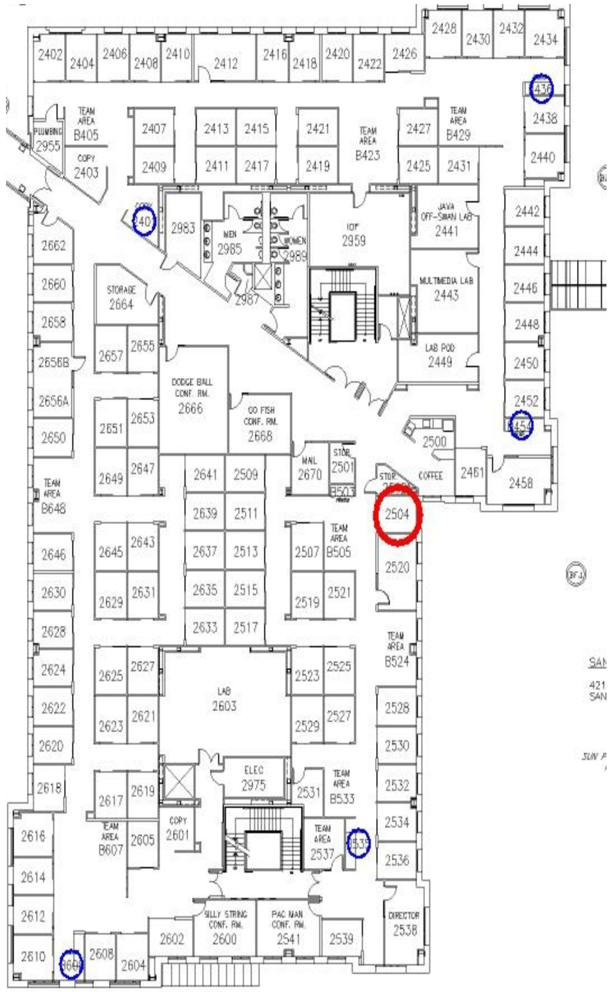
Optimizing Garbage Collection

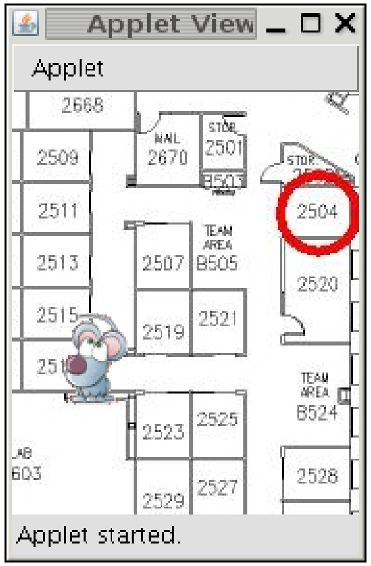
Conclusion

Optimizing Animation

- Sprites and backgrounds
  - Images with transparent colors
  - Background images
  - Only painting dirty areas
- Swoopy graphics
  - Linear movement vs. accelerate/decelerate
  - Scaling and zooming

Overview of Personal Basis Profile





Optimizing Animation

```
Image buffer = null;
Graphics bufferGC = null;

public void paint(Graphics gc) {
  if (bufferGC == null) {
    buffer = createImage(SCREEN_WIDTH,
                   SCREEN_HEIGHT);
    bufferGC = buffer.getGraphics();
  }
  bufferGC.drawImage(mapImage, mapX, mapY, c);
  bufferGC.drawImage(mouseImage, mouseX, mouseY, c);
  gc.drawImage(buffer, 0, 0, c);
}
```

Agenda

Overview of Java ME Platform CDC

Overview of Personal Basis Profile

Building and Running Xlets

Optimizing Images

Optimizing Animation

Optimizing Games

Optimizing Garbage Collection

Conclusion

Optimizing Games

- Reduce number of files
  - Sprite animation images in one file
  - Clipping series of images from one file
- Reduce number of Java class files
  - Object oriented vs. fast
  - Inner classes vs. interfaces
- Event loops and counters
  - Tight loops for drawing
  - Tight loops for processing

Optimizing Games

- Profiling
  - Debug log lines
  - Tools
- Analyzing the paint() method
  - Get rid of waste
  - Coalesce calls
- Threading issues
  - Synchronization
  - Keeping track of sprites

Agenda

Overview of Java ME Platform CDC

Overview of Personal Basis Profile

Building and Running Xlets

Optimizing Images

Optimizing Animation

Optimizing Games

Optimizing Garbage Collection

Conclusion

Optimizing Garbage Collection

- Letting the VM work
  - Calling System.gc()
  - Generation garbage collection
- Profiling using debug version
  - Watching GCs happen
  - Analyzing data
- Managing heap size
  - Array sizes
  - Image buffers

Agenda

Sidenote:

Java ME Platform CDC in the Open Source

Java ME Platform CDC in the Open Source

- The open source version of Java ME platform
- Hosted in subversion on java.net
  - https://phoneme.dev.java.net/source/browse/phoneme/
- Part of the “Mobile and Embedded” community
  - www.mobileandembedded.org
- Other projects part of the M&E community
  - CqME™ project
  - ME Application Developer
- Java.net includes email lists, forums, wikis, etc

Java ME Platform CDC in the Open Source

- phoneME Advanced Software
  - For advanced phones and other consumer devices
  - The “CDC stack”
  - Includes Java ME Platform Personal Basis Profile
- phoneME Feature Software
  - For mass-market “feature” phones
  - The “CLDC stack”

phoneME Software Source Code License

- GPL version 2
- No classpath exception
  - Differs from OpenJDK™: GPLv2 with classpath exception
  - phoneME software is an embedded system
  - You don’t ship it with your application
- GPL's “viral” nature
  - *Does apply to modifications to phoneME software*
  - *Does not apply to applications running atop phoneME software*

Source Code at Java.Net

- Snapshots available as source code bundles
  - https://phoneme.dev.java.net/downloads_page.html
- Source code can be retrieved from svn repository
  - Use an svn client to access https://phoneme.dev.java.net/svn/phoneme
- Typical repository structure
  - Don't just grab the whole thing
  - We have lots of branches, tags, and supertags
  - You'll get about 100 copies of the source code…

Agenda

Sidenote:

Java ME Platform Personal Basis Profile Deployments

Java ME Platform Personal Profile Deployments

- Java TV™ API
- Sony PlayStation 3
- Blu-ray Disc Players

Agenda

Overview of Java ME Platform CDC

Overview of Personal Basis Profile

Building and Running Xlets

Optimizing Images

Optimizing Animation

Optimizing Games

Optimizing Garbage Collection

Conclusion

Conclusion

- Flashy and fast graphics developed in Personal Basis Profile
- Optimizations on the Java ME platform CDC platform
- Optimizations of images, animation, games, and garbage collection

For More Information

- **TS-5114:** Welcome to the UI Theme Park: Customizing the Java ME User Experience With JSR 258
- **TS-5525:** Mobile Ajax for Java Technology
- **TS-5628:** Developing Flashy Mobile Applications, Using SVG and JSR 226
- **TS-5109:** Java Platform, Micro Edition (Java ME): Optimizing Midlets for Size and Performance
- **TS-5913:** Tools for Developing Advanced Mobile Multimedia Applications

DEMO

Optimizing Java ME Platform Personal Basis Profile

Q&A

Developing Faster and Flashier Dynamic Graphics With the Java Platform, Micro Edition (Java ME) Personal Basis Profile by Optimizing Java ME CDC

Hinkmond Wong

Senior Staff Engineer
Sun Microsystems, Inc.
http://phoneme.dev.java.net

TS-5724