Desktop Java Enhancements

Sander Mak
FELLOW & SOFTWARE ARCHITECT
@Sander_Mak

Deprecation Alert: Applets

Removal of Java browser plugin

Deprecated: java.applet.Applet

Alternative: Java Web Start

Extended HiDPI Support

Virtual vs. physical pixels: high Dots Per Inch (DPI)

MacOS already supported

Java 9 introduces Linux and Windows support

Linux support through GTK+ 3

```
-Djdk.gtk.version=3
```

Graphics Improvements

New OpenType font renderer: Harfbuzz (replaces ICU)

Marlin renderer: improved performance

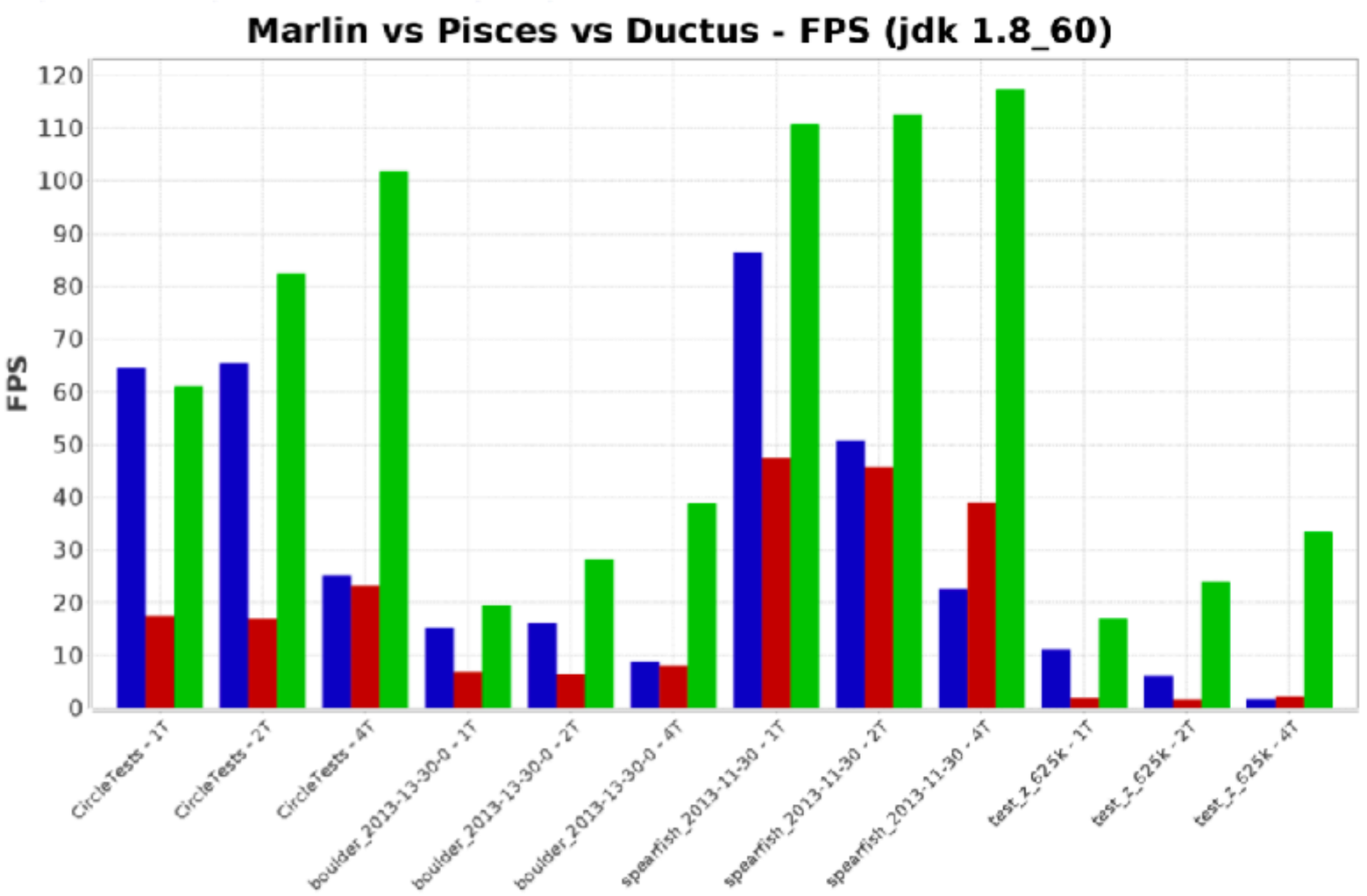
Risks: minor rendering differences

Marlin Renderer Performance

Ductus
(Oracle JDK)

Pisces
(OpenJDK)

Marlin
(OpenJDK 9)



From: Laurent Bourges, used with permission

Platform Specific Desktop Features

java.awt.Desktop new methods:

Showing custom about/preferences windows
Showing files with native file explorer
Open native help viewer application
Listen to system events (sleep, app reopened)

java.awt.desktop new package with interfaces:

Callback types for events and handlers

Not supported on all platforms!

a lot of this is driven by the removal of specific com.apple.eawt functionality

no surprise that almost all features are supported on at least MacOS

Platform Specific Desktop Features

java.awt.Taskbar

Windows

MacOS

Linux (Unity)

Platform Specific Desktop Features

java.awt.Taskbar

Manipulate taskbar icon

Show progress

Manage context menu on taskbar

```
public boolean isSupported(Taskbar.Feature feature)
```

Demo

Taskbar

Show progress in icon

Request user attention when done

Change the Taskbar Icon

```
Taskbar bar = Taskbar.getTaskbar();
if (bar.isSupported(Taskbar.Feature.ICON_IMAGE)) {
    Image icon = ImageIO.read(new File("icon.jpg"));
    bar.setIconImage(icon);
}
```

Show Progress

```
// ..
bar.setIconImage(icon);
// ..
```

```
Thread.sleep(1000);
for (int i = 0; i <= 100; i++) {
    bar.setProgressValue(i);
    Thread.sleep(30);
}
```

Request User Attention

```
// ..
bar.setIconImage(icon);
// ..
```

```
for (int i = 0; i <= 100; i++) {
    bar.setProgressValue(i);
    Thread.sleep(30);
}
```

```
bar.setIconBadge("Done!");
```

Request User Attention

```
// ..
bar.setIconImage(icon);
// ..
```

```
for (int i = 0; i <= 100; i++) {
    bar.setProgressValue(i);
    Thread.sleep(30);
}
```

```
bar.setIconBadge("Done!");
bar.requestUserAttention(true, true);
```

Multi-resolution Images

Represent the same image at different resolutions

```
public interface MultiResolutionImage {

  Image getResolutionVariant(double destImageWidth,
                             double destImageHeight);

  List<Image> getResolutionVariants();

}
```

Implemented by BaseMultiResolutionImage

Multi-resolution Images

sander_lowres.png

sander.png

```
private static MultiResolutionImage loadImages()
    throws IOException {









}
```

Multi-resolution Images

sander_lowres.png

sander.png

```
private static MultiResolutionImage loadImages()
    throws IOException {

  Image[] images = new Image[] {
    ImageIO.read(new File("sander_lowres.png")),
    ImageIO.read(new File("sander.png"))
  };


}
```

Multi-resolution Images

sander_lowres.png

sander.png

```
private static MultiResolutionImage loadImages()
    throws IOException {

  Image[] images = new Image[] {
    ImageIO.read(new File("sander_lowres.png")),
    ImageIO.read(new File("sander.png"))
  };

  return new BaseMultiResolutionImage(images);
}
```

Multi-resolution Images

100x90

157x142

```
MultiResolutionImage mr = loadImages();
```

Multi-resolution Images

100x90

157x142

```
MultiResolutionImage mr = loadImages();

Image lowres = mr.getResolutionVariant(50, 50);
Image highres = mr.getResolutionVariant(200, 200);
```

Multi-resolution Images

100x90

157x142

```
MultiResolutionImage mr = loadImages();

Image lowres = mr.getResolutionVariant(50, 50);
Image highres = mr.getResolutionVariant(200, 200);

System.out.printf("Width for 50: %d%n",
                  lowres.getWidth(null));
System.out.printf("Width for 200: %d%n",
                  highres.getWidth(null));
```

```
Width for 50: 100
Width for 200: 157
```

TIFF Support

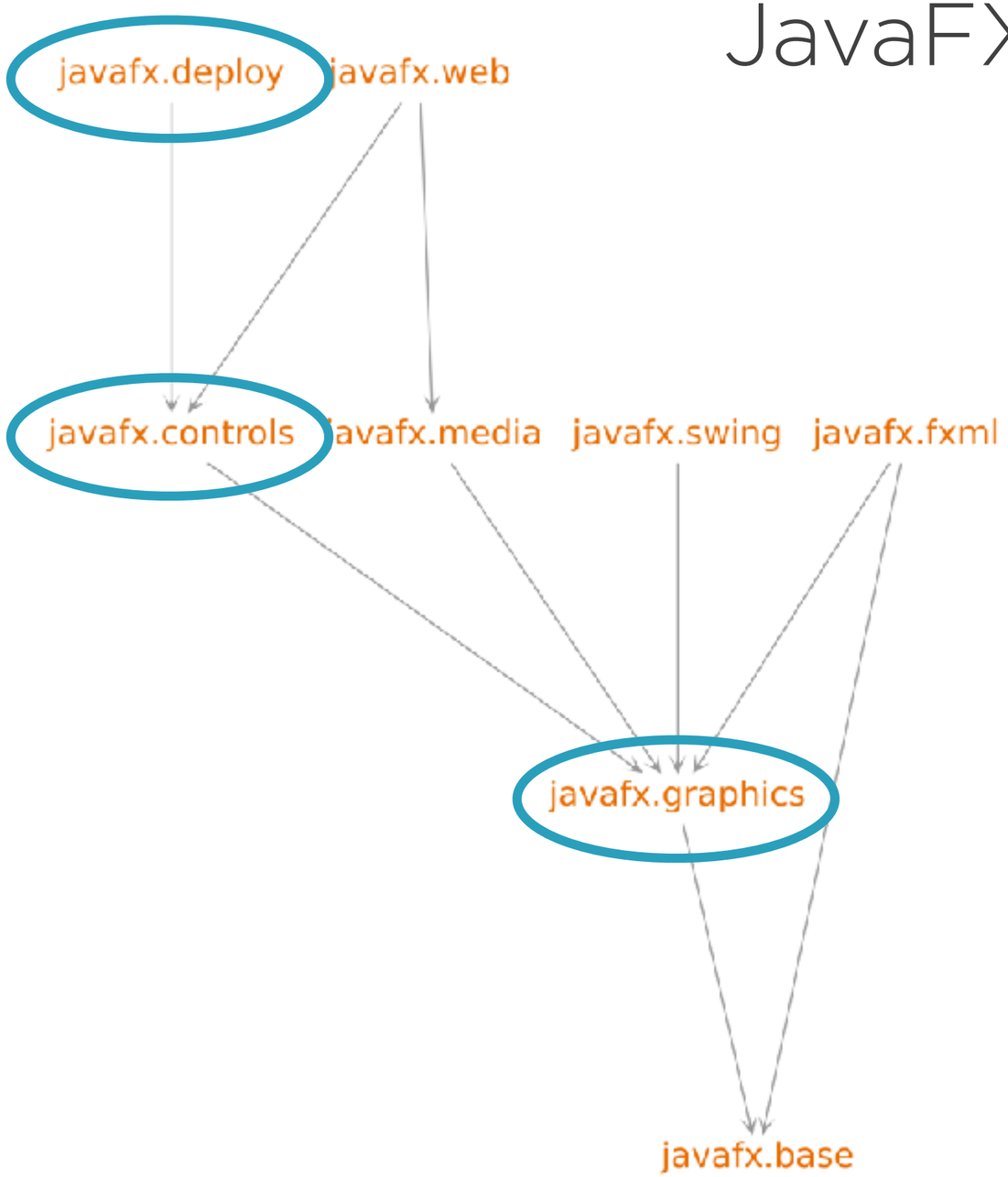
The easy way

```
BufferedImage img =
  ImageIO.read(new File("image.tiff"));
```

The slightly less easy way

```
ImageReader tiffReader =
   ImageIO.getImageReadersByFormatName(
      "tiff").next();
tiffReader.setInput(inputStream);
BufferedImage image = tiffReader.read(0);
```

JavaFX Modules



Single jfxrt.jar split into modules

Exported APIs with encapsulated implementations

Module javafx.deploy has no public API

JavaFX Controls

Skin

Control

Behavior

JavaFX Controls in JDK 8

Skin

Control

Behavior

JavaFX Controls in JDK 9

Skin

Control

Behavior

JavaFX New APIs

Control customization requires use of com.sun.* internal types

Strong encapsulation prevents their use

New APIs are introduced for

UI control skins (`javafx.scene.control.skin`)

CSS styling of controls (`javafx.css`)

JavaFX New APIs

Public but deprecated methods on public APIs

javafx.scene.image.Image.impl_getUrl() -> getUrl()

These are all **breaking changes**,
plan accordingly

Summary

Desktop Java Enhancements

- Applets officially deprecated
- HiDPI and GTK+3 support on Linux
- Improved graphics rendering
- New API for taskbar interaction
- Multi-resolution images
- JavaFX modularized with new APIs