

JavaSE on the Desktop: Extreme GUI Makeover

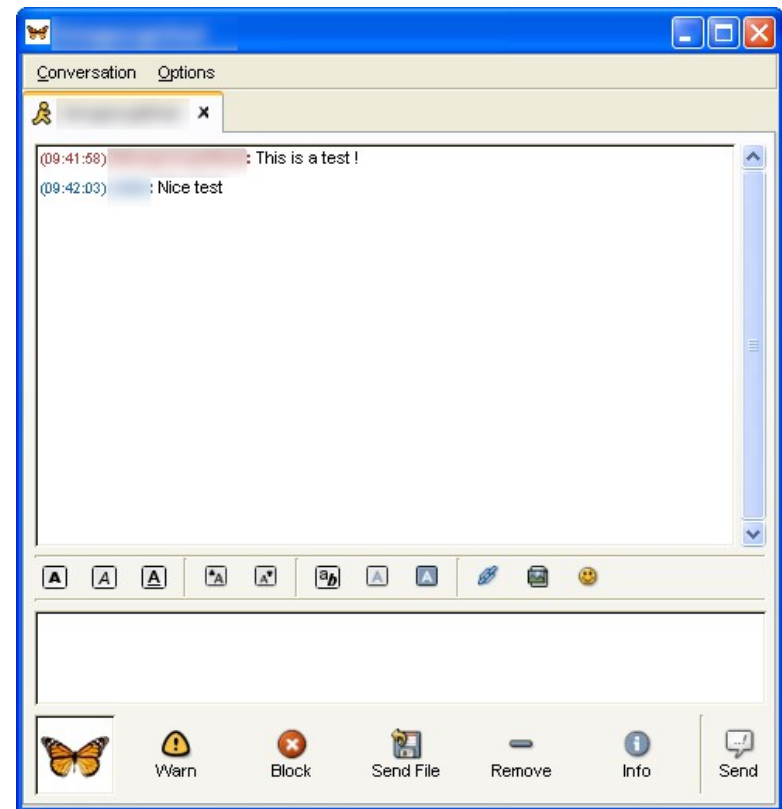
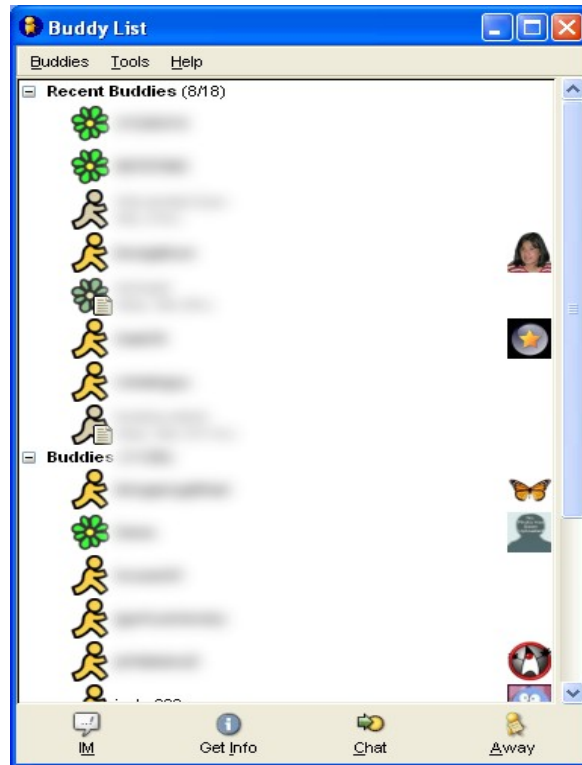
Lee Chuk Munn
Technology Evangelist
Sun Microsystems, Inc.

Swing Applications

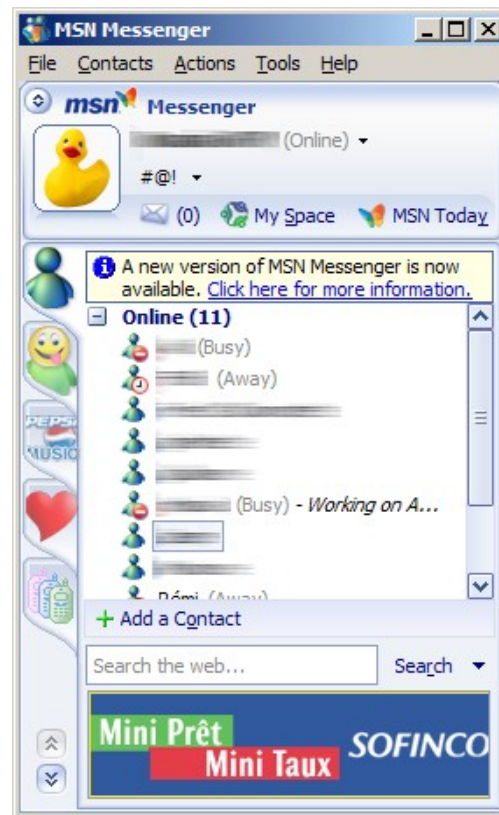
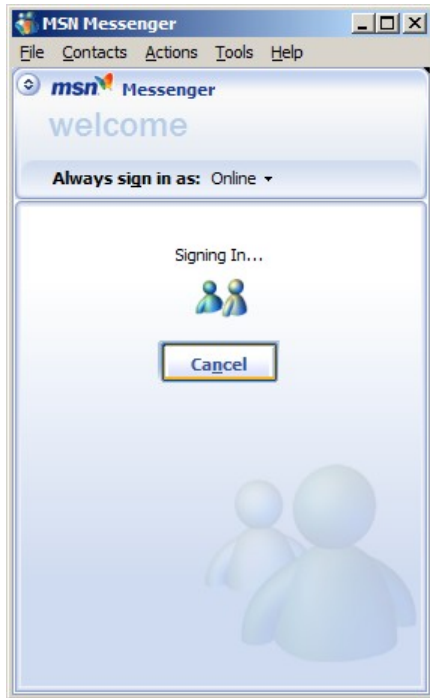
- Bad reputation
 - > Slow and ugly
 - > Complex and difficult
- Reality – Basic API have not change from AWT
 - > Drop in replacement for AWT apps
 - > Append “J” to all widget names
 - > Eg. Button to JButton
- “Advance” widgets and APIs allows you to do interesting things
 - > Need to understand the basic Swing philosophy

Chat Clients

GAIM



MSN Messenger



Demo

How would a
Swing Chat Client
look like?

Details, Details, Details

- Bind buttons to keyboards

```
accountsButton.setMnemonic( 'A' );
```

- Set the login button as the default button

```
frame.getRootPane()  
    .setDefaultButton(loginButton);
```

- Set focus to the password field

```
passwordField.requestFocusInWindow();
```

I18N and L10N

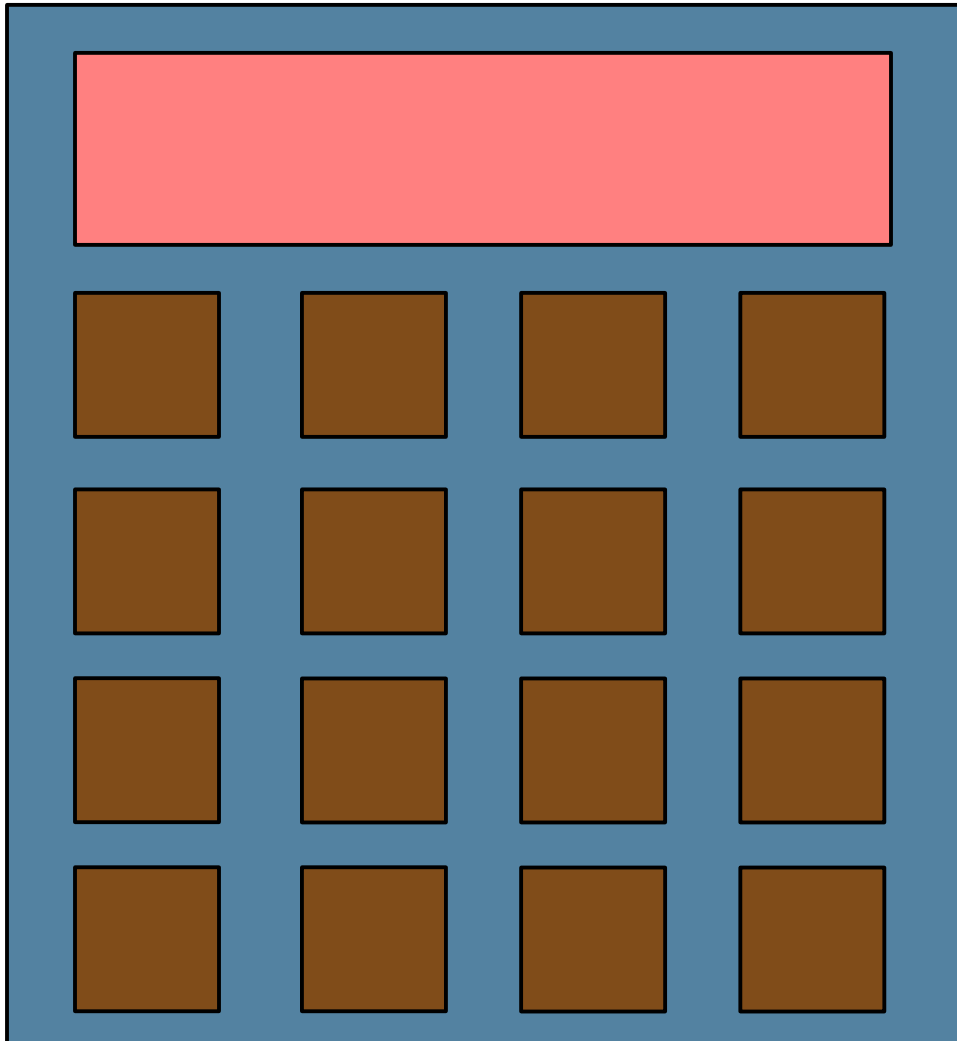
- Never hardcode strings. Use resource bundles and locales

```
01 // load MediumLogin_fr.properties,  
02 // MediumLogin_de.properties...  
03 ResourceBundle.getBundle(  
04     "com/sun/java/chat/ui/login/MediumLogin")  
05 JButton button = new JButton(  
06     bundle.getString("accounts.button"));
```

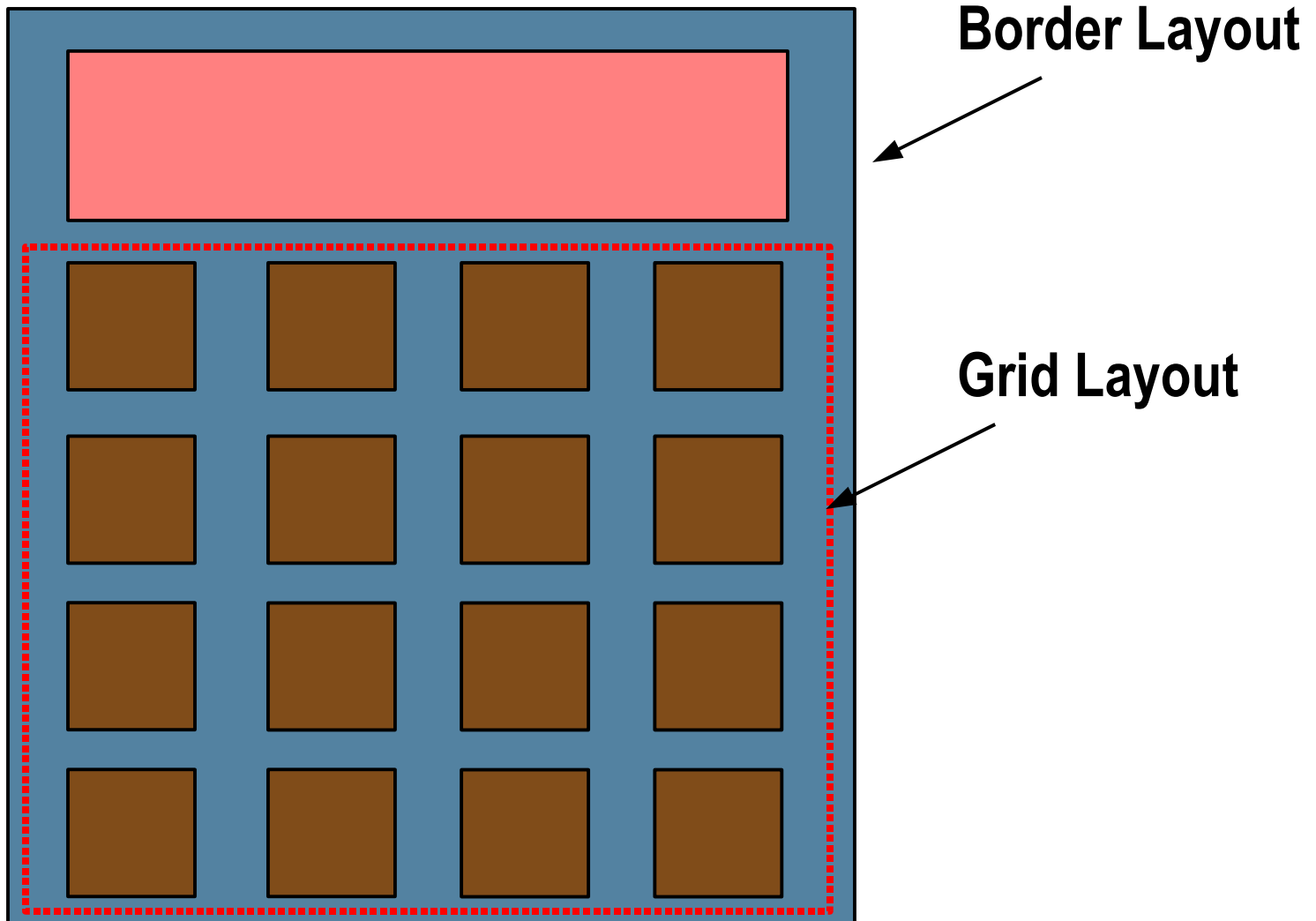
Visual

- Layout alignments
 - > Insets, struts, rigid box for spacing
- Use a combination of layout managers to layout your GUI
 - > FlowLayout, GridLayout, BorderLayout when combined can do some pretty good GUI
- Alternate colors between rows
 - > Combo boxes, tables, trees
- “Less is More”

How Would You Create This?



How Would You Create This?



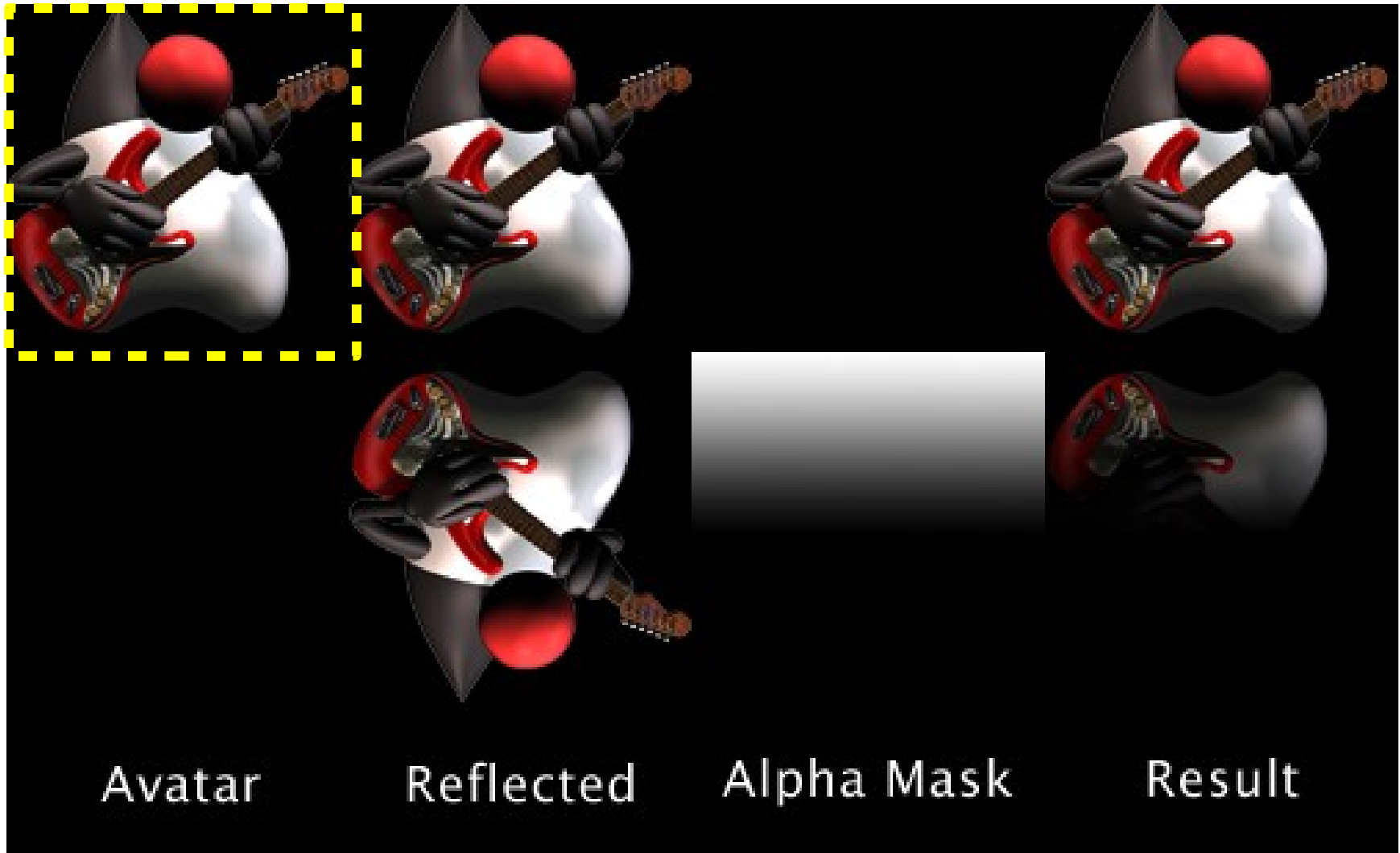
2.5D with Java2D

- Illusion of 3D
 - > Reflect avatar on the ground
- Illusion of depth
 - > Further, smaller
 - > Further, less opaque

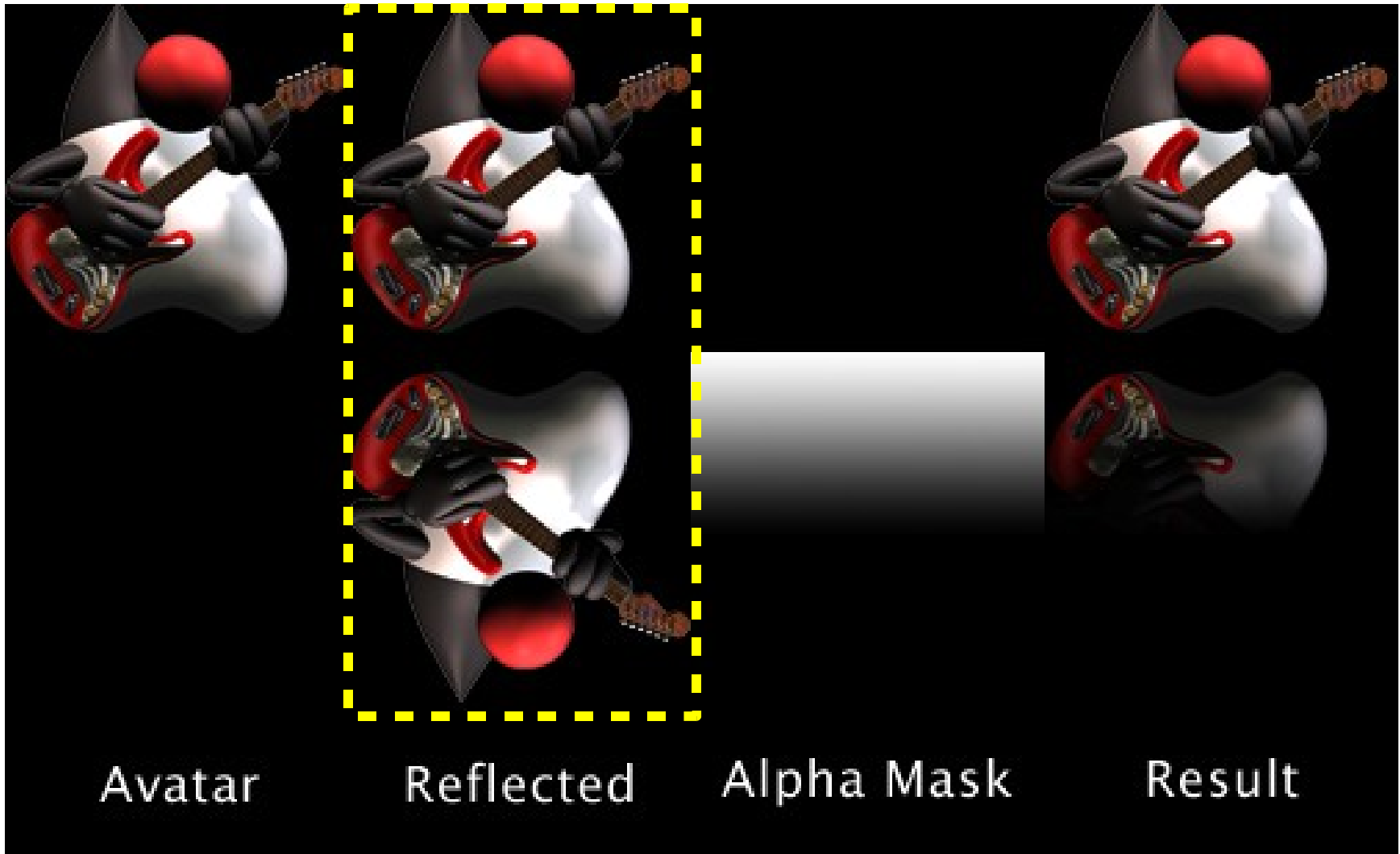


- Java3D will work with Swing and Java2D

Drawing Avatar



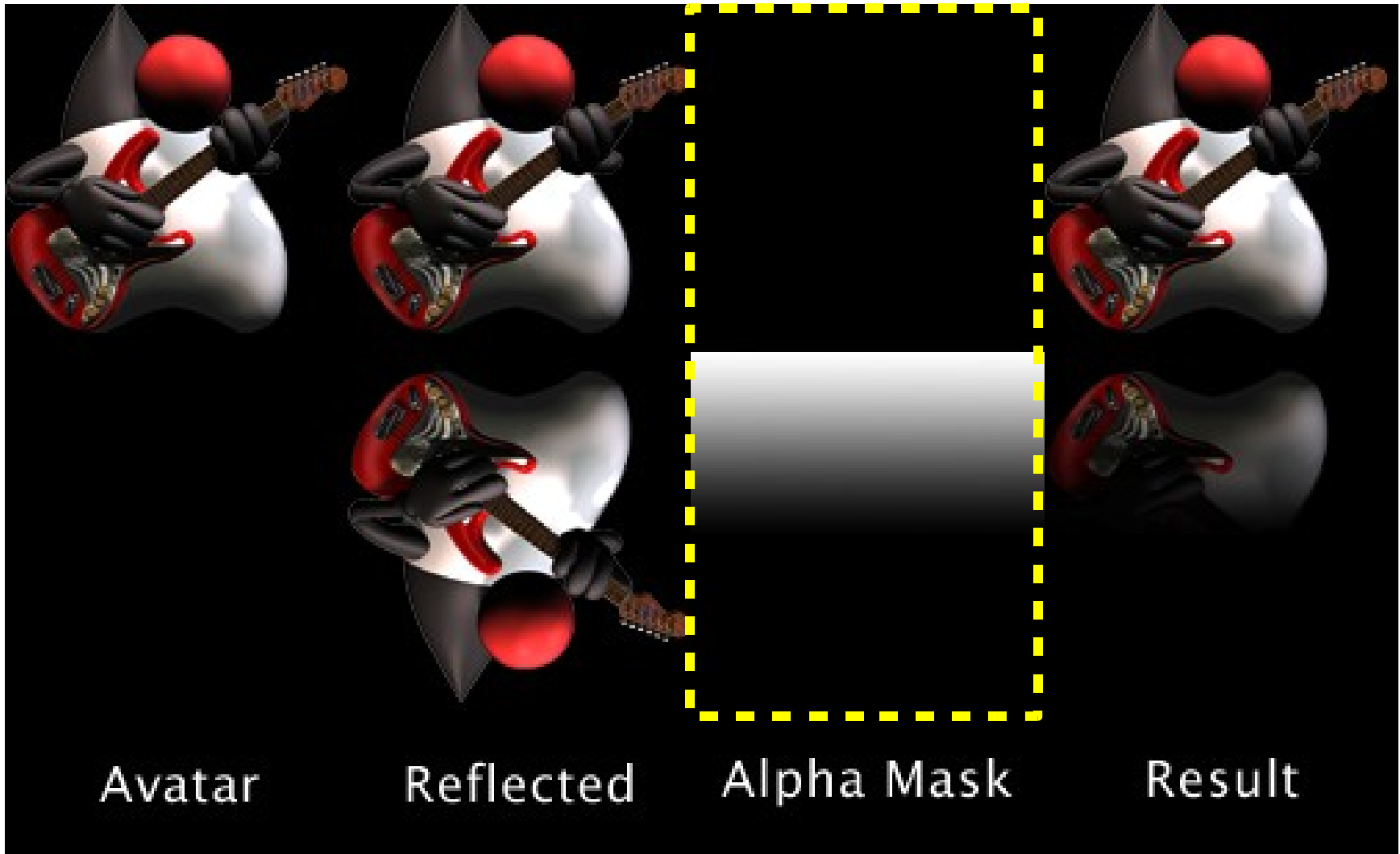
Drawing Avatar



createReflection()

```
01 BufferedImage buffer = new BufferedImage(
02     avatarWidth, avatarHeight << 1,
03     BufferedImage.TYPE_INT_ARGB);
04 Graphics2D g = buffer.createGraphics();
05
06 //Draw original image from top
07 g.drawImage(avatar, null, null);
08 g.translate(0, avatarHeight << 1);
09
10 //Draw reflection from bottom up
11 AffineTransform reflectTransform =
12     AffineTransform.getScaleInstance(1.0, -1.0);
13 g.drawImage(avatar, reflectTransform, null);
```

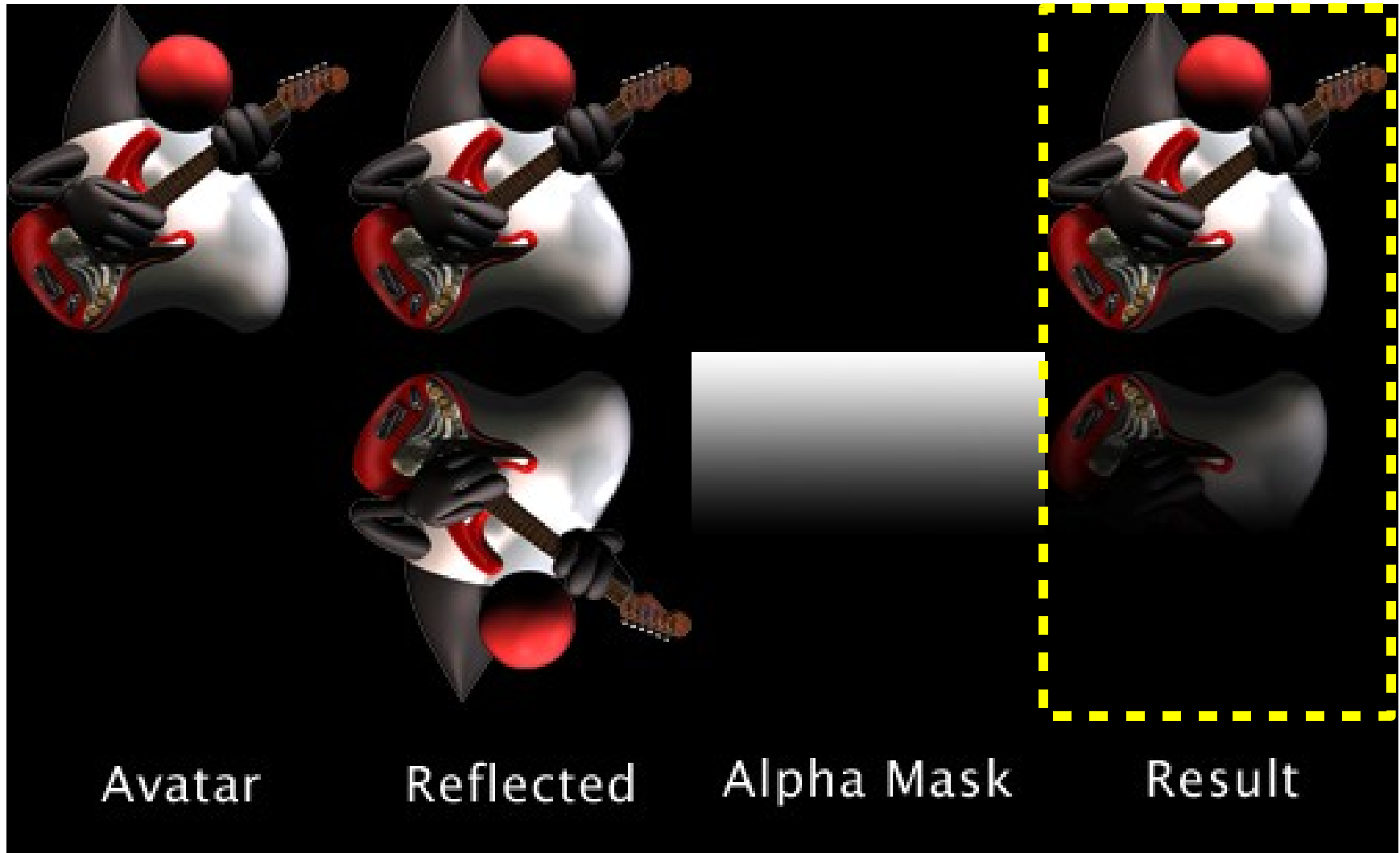
Drawing Avatar



createGradientMask()

```
01 BufferedImage gradient = new BufferedImage(
02     avatarWidth, avatarHeight,
03     BufferedImage.TYPE_INT_ARGB);
04 Graphics2D g = gradient.createGraphics();
05 GradientPaint painter = new GradientPaint(
06     0.0f, 0.0f,
07     new Color(1.0f, 1.0f, 1.0f, 0.5f),
08     0.0f, avatarHeight / 2.0f,
09     new Color(1.0f, 1.0f, 1.0f, 1.0f));
10 g.setPaint(painter);
11 g.fill(new Rectangle2D.Double(
12     0, 0, avatarWidth, avatarHeight));
```

Drawing Avatar



Drawing Avatars

```
01 //Load the image
02 Image avatar = ...;
03
04 //Create a "reflection" of the avatar
05 BufferedImage reflect = createReflection(
06     avatar, avatarWidth, avatarHeight);
07
08 //Create the gradient mask
09 BufferedImage alphaMask = createGradientMask(
10     avatarWidth, avatarHeight);
11
12 //Apply the mask to the reflection
13 Graphics2D g2 = reflect.createGraphics();
14 g2.setComposite(AlphaComposite.DstOut);
15 g2.drawImage(alphaMask, null, 0, avatarHeight);
16 g2.dispose();
```

Summary

- Understand your applications and the target audience
- Take advantage of Swing and the powerful Java2D API to eliminate usability and visual problems
- Understand what you are using could have tremendous impact on the performance of your application
- Use your imagination. Make your application fun

JavaSE on the Desktop: Extreme GUI Makeover

Lee Chuk Munn
Technology Evangelist
Sun Microsystems, Inc.