

GUI Entwicklung

Sascha Just Softwarepraktikum 2014







Register	
First name :	
Last name :	
	R

Register	
First name :	
	R

	Home	Books	Molic	10000	
AH	enta basic Bayannis Oritikas Oritikas Entropedie Entropedie Boltz- Batty		ign af spran einent fer ach ant eisig ach ant Dignumei	Transact. Anatomic Contraction	

2			
Negista	2		
First n	ame :		
last has			-
Last no.			- 1
			E.
			- 6
			A Real Property of
	Registe Firist n Last hav	Register First Name : Last hame :	Register First Name : Last name :













Credits: Nielsen Norman Group







5.































Wireframing

Window Name			

Wireframing

Window Name				
File Edit View	Help			
Open Open Recent • Option One	Open		Desktop	
✓ Toggle Item Disabled Item	Devices	Name simulation.log amazing.log	Date Modified Yesterday, 3:19 PM Yesterday, 7:55 AM	
	Shared Time Places	stunning.log	Friday, September 18, 2014 8:24 AM	
	E Desktop			
	double-click, ENT	ER or F2 to edit		
			Cancel	

 Schnelles Erstellen eines Mockups ohne dazu programmieren zu müssen.

- Schnelles Erstellen eines Mockups ohne dazu programmieren zu müssen.
- Entdeckt frühzeitig Probleme im Design.

- Schnelles Erstellen eines Mockups ohne dazu programmieren zu müssen.
- Entdeckt frühzeitig Probleme im Design.
- Ermöglicht das Einholen von User-Feedback vor dem Implementieren.

- Schnelles Erstellen eines Mockups ohne dazu programmieren zu müssen.
- Entdeckt frühzeitig Probleme im Design.
- Ermöglicht das Einholen von User-Feedback vor dem Implementieren.
- Ermöglicht Zusammenarbeit von Teams aus mehreren Disziplinen.

Storyboards



GUI Design Abnahme

Was wir von Ihnen sehen möchten.

- Paper Prototype/Wireframe (Papier oder Digital) mit klar erkennbaren Elementen.
- Status-Diagramme
- Status-Beschreibungen
- Status-Übergangsbeschreibung
- Interaktionsbeschreibung: GUI zu Datenquelle (Simulator oder Logreader) und umgekehrt.

Status Diagramme

Welchen Status kann Ihre GUI annehmen?

Wie gelangt die GUI in welchen Status?



D

KONFIGURIEREN

BESTÄTIGUNG

Status Beschreibungen

- A. Uninitialisiert: Fenster: Main. Panels: Main:Control. Alle Buttons gesperrt, Menu>File>(Open, Quit) verfügbar.
- B. Pausiert: Play-Button nutzbar, Spielfeld präsent in Panel Main:Map. Spielstände in Panel Main:Info angezeigt.
- C. Laufende Simulation: ...
- D. Warten auf Konfiguration: ...
- E. ...

Status Übergänge

Start der GUI: Erstellung des Hauptfensters (*Main*) mit Panel *Main:Control*.

Konfiguration: Konfigurieren-Button öffnet Dialog:Config

Bestätigung: *Dialog:Config:OK*-Button + gültige Einstellungen schließt Dialog und erstellen Panel Main:Main und Main:Info.

Laufende Simulation: Drücken des Main:Control:Play-Buttons startet Simulation und setzt Pause-Symbol für Main:Control:Play-Button.

• • •

Interaktion mit Datenquelle

In Ihrem Fall zwei Möglichkeiten:

Simulator

Log Reader

Interaktion mit Simulator

GUI zu Simulator

Main:Control:Play#click

startet neuen Simulator-Thread

Main:Control:Pause#click

signalisiert dem Simulator-Thread zu warten

Main:Control:Turn#change

signalisiert dem Simulator-Thread Daten ab der angegebenen Runde zu senden.

Interaktion mit Simulator

Simulator zu GUI

Simulation bereit

Setzt GUI in Pausiert-Status (freigeben der Elemente: Play, ...)

Ende der Simulation

Setzt GUI in End-Status und veranlasst Anzeigen des Punktestands.



GUI Frameworks

Swing JavaFX GWT

. . .

http://docs.oracle.com/javase/8/javase-clienttechnologies.htm

Paper Prototype zu GUI.

00		Untitled		R <u>w</u>
Library	Q 07	No Selection	Inspector	Q 0+
v	Containers		v	Properties
Accordion	0		No Selection	
Accordion (empty)				
± AnchorPane				
🔲 BorderPane				
FlowPane				
I GridPane				
III HBox				
Pane Pane				
ScrollPane				
•	Controls			
•	Menu			
•	Miscellaneous			
►	Shapes	Drag Library items here		
►	Charts			
	3D			

JavaFX Scene Builder Demo

Drag Library items here	
	► Layout
► Controller	► Code

$\Theta \Theta \Theta$

📬 • 🖆 • 🗒 🐚 🏝 🔍 🛱 🚍 1 🗹 • 1 🔐 🍫 • 🕥 • 💁 📽 🞯 • 1 😕 🗁 🖋 • 1 🍄 🌽 🍩 🔳 🕥 1 😓 • 7 • 7 • 7

📙 Package Explor 🖾 🍃 Type Hierarchy 🕂 JUnit 🗖 🗖	♦ pirates.fxml I Main.java S
Package Explor Process PiratesGUI PiratesGUI <td< th=""><th><pre>Primes.txml [] Man,ava 32 24 * Copyright 2014 Sascha Just] 13 package application; 14 14 15 timport javafx.application.Application; 20 21 /** 22 * The Class Main. 23 */ 24 public class Main extends Application { 25 26 /** 27 * The main method. 28 * @param args 30 * the arguments 31 */ 32 public static void main(final String[] args) { 33 launch(args); 34 } 35 36 /** 37 * [@inheritDac] 38 * 8 esee javafx.application.Application#start(javafx.stage.Stage) 40 */ 410 @Override 412 final Parent root = FXMLLoader.load(getClass().getResource("pirates.fxml")); 413 final Scene = new Scene(root); 424 scene = new Scene(root); 435 scene = new Scene(root); 436 scene = new Scene(root); 437 primaryStage.store(scene); 438 primaryStage.store(scene); 439 primaryStage.store(scene); 439 primaryStage.store(scene); 430 primaryStage.store(scene); 431 public acts (final Exception e) { 432 e.printStackTrace(); 433 } 434 435 catch (final Exception e) { 435 e.printStackTrace(); 435 } 435 } 436 } 437 catch (final Exception e) { 438 e.printStackTrace(); 439 } 439 } 430 catch (final Exception e) { 430</pre></th></td<>	<pre>Primes.txml [] Man,ava 32 24 * Copyright 2014 Sascha Just] 13 package application; 14 14 15 timport javafx.application.Application; 20 21 /** 22 * The Class Main. 23 */ 24 public class Main extends Application { 25 26 /** 27 * The main method. 28 * @param args 30 * the arguments 31 */ 32 public static void main(final String[] args) { 33 launch(args); 34 } 35 36 /** 37 * [@inheritDac] 38 * 8 esee javafx.application.Application#start(javafx.stage.Stage) 40 */ 410 @Override 412 final Parent root = FXMLLoader.load(getClass().getResource("pirates.fxml")); 413 final Scene = new Scene(root); 424 scene = new Scene(root); 435 scene = new Scene(root); 436 scene = new Scene(root); 437 primaryStage.store(scene); 438 primaryStage.store(scene); 439 primaryStage.store(scene); 439 primaryStage.store(scene); 430 primaryStage.store(scene); 431 public acts (final Exception e) { 432 e.printStackTrace(); 433 } 434 435 catch (final Exception e) { 435 e.printStackTrace(); 435 } 435 } 436 } 437 catch (final Exception e) { 438 e.printStackTrace(); 439 } 439 } 430 catch (final Exception e) { 430</pre>
	pirates.fxml
File Edit H	Help Untitled Tab 1 Untitled 💌

FXML in Eclipse verknüpfen: Demo

JavaFX Links

http://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm

http://docs.oracle.com/javase/8/javafx/get-started-tutorial/ get_start_apps.htm#JFXST804

https://www.youtube.com/watch?v=ij0HwRAICmo

http://code.makery.ch/java/javafx-8-tutorial-intro/

Swing Links

http://docs.oracle.com/javase/tutorial/uiswing/start/index.html

Java GUI Toolkits

Java GUI Toolkits



Java GUI Toolkits










+ Standard Widget Toolkit (SWT) • + Qt Jambi • ...







JFrame frame = new JFrame("HelloWorldSwing");





JFrame frame = new JFrame("HelloWorldSwing");

Hello World

private static void createAndShowGUI() {
 //Create and set up the window.
 JFrame frame = new JFrame("HelloWorldSwing");
 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

//Add the ubiquitous "Hello World" label.
JLabel label = new JLabel("Hello World");
frame.getContentPane().add(label);

//Display the window.
frame.pack();
frame.setVisible(true);

}

Hello World

```
import javax.swing.*;
```

```
public class HelloWorldSwing {
    private static void createAndShowGUI() { /* see above */ }
    public static void main(String[] args) {
       //Schedule a job for the event-dispatching thread:
        //creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
   }
```

Elemente hinzufügen

Elemente hinzufügen

public Component add(Component c);

Elemente hinzufügen



JFrame

... und hunderte weitere Elemente!

Layout

😤 FlowLayout	
jButton1 jButton2 3 jButton4	Layout
Alignment set to LEFT jButton6 hgap = 15	
vgap = 30 jButton9	





👹 BorderLa	yout	_ 🗆 ×
	NORTH	
WEST	CENTER	EAST
	SOUTH	



<u> BorderL</u> a	yout	- 🗆 ×
	NORTH	
WEST	CENTER	EAST
	SOUTH	

🥞 GridLayout		_ 🗆 🗵
jButton1	jButton7	jButton6
jButton5	jButton4	jButton3
jButton2		

I. Für jede Aktion eigene Klasse implementieren

I. Für jede Aktion eigene Klasse implementieren

public class MyClass implements ActionListener {
 public void actionPerformed(ActionEvent e) {
 ...//code that reacts to the action...
 }

I. Für jede Aktion eigene Klasse implementieren

public class MyClass implements ActionListener {
 public void actionPerformed(ActionEvent e) {
 ...//code that reacts to the action...
 }

2. Objekt der Klasse beim Bedienelement registrieren

I. Für jede Aktion eigene Klasse implementieren

public class MyClass implements ActionListener {
 public void actionPerformed(ActionEvent e) {
 ...//code that reacts to the action...
 }

2. Objekt der Klasse beim Bedienelement registrieren

someComponent.addActionListener(instanceOfMyClass);

auch WindowListener, ChangeListener, MouseListener...

CDU/CSU 41,5%

CDU/CSU	41,5%
SPD	25,7%

CDU/CSU	41,5%
SPD	25,7%
GRÜNE	8,4%

CDU/CSU	41,5%
SPD	25,7%
GRÜNE	8,4%
FDP	4,8%

CDU/CSU	41,5%
SPD	25,7%
GRÜNE	8,4%
FDP	4,8%
LINKE	8,6%

CDU/CSU	41,5%
SPD	25,7%
GRÜNE	8,4%
FDP	4,8%
LINKE	8,6%
AfD	4,7%

CDU/CSU	41,5%
SPD	25,7%
GRÜNE	8,4%
FDP	4,8%
LINKE	8,6%
AfD	4,7%
Sonstige	6,2%

CDU/CSU	41,5%
SPD	25,7%
GRÜNE	8,4%
FDP	4,8%
LINKE	8,6%
AfD	4,7%
Sonstige	6,2%



CDU/CSU	41,5%
SPD	25,7%
GRÜNE	8,4%
FDP	4,8%
LINKE	8,6%
AfD	4,7%
Sonstige	6,2%





Grafik
Grafik



Vektor



Vektor

Pixel







```
/** Returns an ImageIcon, or null if the path was invalid. */
protected ImageIcon createImageIcon(String path, String description) {
    java.net.URL imgURL = getClass().getResource(path);
    if (imgURL != null) {
        return new ImageIcon(imgURL, description);
    } else {
        System.err.println("Couldn't find file: " + path);
        return null;
    }
}
```











import java.awt.Graphics2D;

}

}

public class CellPanel extends JPanel {

public void paintComponent(Graphics g) {
 super.paintComponent(g);
 Graphics2D g2d = (Graphics2D) g;
 drawItall(g2d);
}

private void drawItAll(Graphics2D g2d) {
 g2d.drawImage(...);
 g2d.drawString(...);
 g2d.drawLine(...);