UI Programming

(part of this content is based on previous classes from Anastasia, S. Huot, M. Beaudouin-Lafon, N.Roussel, O.Chapuis)

Assignment 1 is out!

Design and implement an interactive tool for creating the layout of comic strips



https://www.lri.fr/~fanis/teaching/ISI2014/assignments/ass1/

Graphical interfaces

GUIs: input is specified w.r.t. output

Input peripherals specify commands at specific locations on the screen (*pointing*), where specific objects are drown by the system. Familiar behavior from physical world





WIMP interfaces

WIMP: Window, Icons, Menus and Pointing

Presentation

Windows, icons and other graphical objects

Interaction

• Menus, dialog boxes, text input fields, etc

Input

pointing, selection, ink/path

Perception-action loop

feedback



Software layers

Application			Applications/Communication (MacApp)
	Interface Tools & Toolkits		Builders, Java Swing, JavaFX, Qt (C++), GTK+, MFC, Cocoa
		Graphics Library	GDI+, Quartz, GTK+/Xlib, OpenGL
		Windowing System	X Windows (+KDE or GNU)
	Input/	Output	
Opera	iting Sy	/stem	Windows, Mac OS, Unix, Linux, Android, iOS, WindowsCE

Software layers

Application						
[Interface Tools & Toolkits					
	Graphics Library					
	Windowing System					
	Input/Output					
Operating System						

Input/output peripherals



Input: where we give commands

Output: where the system shows information & reveals its state



Interactivity vs. computing

Closed systems (computation):

- read input, compute, produce result
- final state (end of computation)

Open systems (interaction):

- events/changes caused by environment
- infinite loop, non-deterministic

Problem

We learn to program algorithms (computational)

Most languages (C/C++, Java, Lisp, Scheme, Pascal, Fortran, ...) designed for algorithmic computations, not interactive systems

Problem

Treating input/output during computation (interrupting computation) ...

- write instructions (print, put, send,...) to send data to output peripherals
- read instructions (read, get, receive,...) to read the state or state changes of input peripherals

Problem

To program IS in algorithmic/computational form

```
two buttons B1 and B2
finish <- false
while not finish do
   button <- waitClick () //interruption, blocked comp.
   if button
        B1 : print « Hello World »
        B2 : finish <- true
   end
end
```

Managing input



Event based (driven) programming



Event based (driven) programming



Example: Swing (and AWT)

3 threads

Initial thread: main ()

methods (drawing functions)

AWT Event Queue Ţ • EDT manages the events queue: **Event Dispacher Thread** sends events to listeners (functions (EDT) dealing with events) and calls paint Ϋ́ • Worker (or background) threads, where time-consuming tasks are executed paint () Listeners

Software layers

Application						
	Interface Tools & Toolkits					
	Graphics Library					
	Windowing System					
	Input/Output					
Opera	Operating System					

Interface builders

Transation of the			لكلك
le bolt view Navigale source Hefactor Bi	IO Run CIS TOUS WINDOW HEE		
🖎 选 📂 🕼 📌 📦 💷 🦄	P 2 3 3 4 4 3		
rojects 4 × Files Buntime	🔯 Anterna.java 🗶 🗃 Anagrama.java 🗶 😨 Alcost.java 🗶 😨 Find.java 🗶 😭 ContactEditor.java 🗶 💷	Palette	
🖕 Anagrantiane	Server Design Do DA CR HILLE IF IS I CO C	Swing	
in 🔯 Source Packages		w Xabel	nettu@ 🔊
😑 🔛 con.tzy.oragrana.lb		IN Mondebutton	#- Xheddox
Word.brary.java	Position/Direction	to Taket the	17 Differences
B- 🔛 con try anagrams ut	+ Detection [*]: 240.000		C- Calor In Dep
About.jana	Average Fach 110 mm	Ciril X, and allow	182 IS
Anagranic Java	induction and an	I stevtrield	TextArea
Test Packages	Heidurk is rowen state over reade over reader	Dariel Dariel	🛅 XabbedFare
e 🛄 Loraies		SK Noollay	MooiFace
a Garage Coraces	Sydem	IE Manufac	The second data as
a one and the second se	Chanvels: 2 Watte: 12.000 Adjust		EN Propagation
Source replayes		404 2Sider	- Stogresslar
Complete Complete	Antenna Type: Kathain 742151	1 NoltPare	III FornattedTextField
Contact days	Electrical Downilk Prom (*): 0.000 Tec 30.000 Adjust	III Fassword*eid	IX 20pinner
Bool Inca		H Managalar	W TextBane
· Da Lbraries	ADREADOLE (Y +42.	El Mandara	C Maria
	Frequency From (M-q) 943.000 Tec 951.000 Adust	10 Journale	ED Miller
		🖬 Stable	LC Toolfar
uner bee	4X	1991. Hotemallrama	N mara-Dana
Even Antoneo	OK CANAR	Scheddlawl [3Chedd	lox]-Properties
Citter Concepts		Properties Events	Code
T Diamal		- Properties	
E East Davel		action	rul [
- set #abelt [1.abel]		badground	[236,233,216]
war Eddel? [1.del]		buttenGroup	(note)
- Contraid (Textfield)		cosponentPopupPieru	chone>
- Intertheid: [Jintertheid]		fore and	Income Li Pran
- RheckBercl [KheckBoo]		mencoir	[())(1)
Panel Panel		minted	
 platton3 [30utton] 		text	Height is Lower Edge (
Buttoe4[30.tton]		too/TpText	nul
		Other Properties	
		UEClessED	Checistout .
		actionCommand	Height is Lovier Edge (
		alignmer#(0.0
		Juliconact/	- ns

Examples : MS Visual Studio (C++, C#, etc.), NetBeans (Java), Interface Builder (ObjectiveC), Android Layout Editor

Interface builders

Can be used to

- create prototypes (but attention it looks real)
- get the « look » right
- be part of final product
- design is fast
- modest technical training needed
- can write user manuals from it

But: still need to program (and clean code ...)

Interface toolkits

Libraries of interactive objects (« widgets », e.g., buttons) that we use to construct interfaces

Functions to help programming of GUIs

...usually also handle input events (later)

Interface toolkits

Toolkit	Platform	Language
Qt	multiplatform	C++
GTK+	multiplatform	С
MFC later WTL	Windows	C++
WPF (subset of WTL)	Windows	(any .Net language)
FLTK	multiplatform	C++
AWT / Swing	multiplatform	Java
Сосоа	MacOs	Objective C
Gnustep	Linux, Windows	Objective C
Motif	Linux	С
JQuery UI	Web	javascript

Problem with toolkits?

Why Java Swing?

Based on Java (any platform, plenty of libraries)

A lot of online resources and examples

Why Java Swing?

Based on Java (any platform, plenty of libraries)

A lot of online resources and examples

Other alternatives for Java?

→ JavaFX: soon becomes the new standard for Java UI programming, supporting a variety of different devices

« widgets » (window gadgets)



Swing widgets



Swing widgets



Widget complexity

Simple widgets

• buttons, scroll bars, labels, ...

Composite/complex widgets

- contain other widgets (simple or complex)
- dialog boxes, menus, color pickers, ...

Widget tree

Hierarchical representation of the widget structure

a widget can belong to only one « container »



Swing widget classes

- A GUI application has a top-level (container) widget that includes all others
- In Swing there are 3 types: JFrame, JDialog and JApplet
- They all contain other widgets (simple or complex), that are declared in the field **content pane**



Swing widget classes



http://docs.oracle.com/javase/tutorial/ui/features/components.html

Swing JFrame



Useful functions public JFrame(); public JFrame(String name);

public Container getContentPane(); public void setJMenuBar(JMenuBar menu); public void setTitle(String title); public void setIconImage(Image image);

This program does not terminate after "no, still running ..."

Swing JDialog

a message window (dialog) can be "modal" (blocks

interaction)

usually attached to another window (when that closes, so does the dialog)



Widget placement

UI toolkits control widget placement:

- should be independent of widget size (menu at least as big as its largest item, change of scrollbar size with document size, adjusting text flow)
- done in *layout managers* that can be added to container widgets



import javax.swing.*; import java.awt.*;

public class SwingDemo2 extends JFrame {

public void init() this.setTitle("example 2"); getContentPane().add(new JLabel("Swing Demo 2")); Container contentPane = this.getContentPane(); contentPane.setLavout(new FlowLayout()); this.setDefaultCloseOperation(EXIT_ON_CLOSE); contentPane.add(new JButton("clique ici")): contentPane.add(new JButton("clique là")); } public static void main(String[] args) SwingDemo2 frame = new SwingDemo2(); frame.init(); frame.setSize(200.200): frame.setVisible(true); 3 } Bruce Eckel, Thinking in Java, 2nd edition

example 2 Swing Demo 2 clique ici clique là

Widget placement

General guides

- embed geometry of a «child» widget to its parent
- parent controls the placement of its children

Layout algorithm

- natural size for each child (to fit contended)
- size and position imposed by parent
- constraints: grid, form, etc.

ent)	file Cancel OK	}
• • file •	Cancel OK	}

. . .

Layout managers (in Swing)

BorderLayoutDemo			
Button 3 (LINE_START) Button 3 (LINE_START) Button 2 (CENTER)	5 (LINE_END)	Button 1 Button 2 Button	n 3 Long-Named Button 4 5
BorderLayout	F	FlowLayout	
	GridLayoutDem	Button 2	
BoxLayoutDemo	Button 3	Long-Named Button 4	
Button 3 Long-Named Button 4	5 Horizontal gap: Ve 0 🗸 0	ertical gap:	Find Find What: Match Case Wrap Around Whole Words Search Backwards

	example 4	
	Button 0 Button 1 Button 2	Button 3 Button 4
<pre>import javax.swing.*;</pre>	Button 5 Button 6 Button 7	Button 8 Button 9
import java.awt.*;	Button 10 Button 11 Button 12	Button 13 Button 14
public class SwipaDemo4 extends lErame {	Button 15 Button 16 Button 17	(Button 18) (Button 19)
public cluss swingbendt ckeenus strune (🔿 🔿 example 4
<pre>public void init()</pre>		Button 0 Button 1
{		Button 2 Button 3
Container cp = getContentPane();		Button 4 Button 5
		Button 6 Button 7
this.setlitle("example 4");		Button 8 Button 9
this.setberdutteroseoperation(Exi1_ON_CLOSE);		Button 10 Button 11
<pre>cp.setLavout(new FlowLavout()):</pre>		Button 12 Button 13
for(int i = 0; i < 20; i++)		Button 14 Button 15
<pre>cp.add(new JButton("Button " + i));</pre>		Button 16 Button 17
}		Button 18 Button 19
<pre>public static void main(String[] args)</pre>		
{		
SwingDemo4 frame = new SwingDemo4();		
<pre>frame.init();</pre>		
frame.setSize(200,700);		
<pre>frame.setVisible(true);</pre>		
}		
}		

Layout managers (in Swing)

GridLayout gridLayout = new GridLayout(0,2);

JPanel gridPanel = new JPanel(); gridPanel.setLayout(gridLayout);

gridPanel.add(new JButton("Button 1")); gridPanel.add(new JButton("Button 2")); gridPanel.add(new JButton("Button 3")); gridPanel.add(new JButton("Long-Named Button 4")); gridPanel.add(new JButton("5"));

🛃 GridLayoutDemo					
Button 1	Button 2				
Button 3	Long-Named Button 4				
5					
Horizontal gap: Vertical	gap:				
0 🔽 0	Apply gaps				

Layout managers (in Swing)

GridLayout: grid	🕌 GridLayo	utDe	mo				
	But	on 1				Button 2	
	Button 3				Long-Named Button 4		
		5					
	Horizontal gaj	0	Vertical g	gap:			
	0	-	0		-	Apply gaps	



GridBagLayout: sophisticated grid

Placement guides (Mac OS X)



Placement guides (Mac OS X)

Center balance: visual balance of a container's content between the left and right parts



Placement guides (Mac OS X)

Alignement

Column of labels with right alignement Column of controls with left alignment 000 hanger Prefer General Editing: Select existing image Add a margin around image points Reposition windows after change Remember recent items Clipboard Settings: Copy selection from image only Frase selection from image Dither content of clipboard Color Optimization: Calculate best color table Verify color table integrity Notify on loss of color information Notify before CMYK to RGB conversion ?

Placement guides (Mac OS X)

Spacing



Placement guides (Mac OS X)

Alignement and consistency



Consistency between controls of the same type

CRAP contrast, repetition, alignment, proximity

Major sources: Designing Visual Interfaces, Mullet & Sano, Prentice Hall / Robin Williams Non-Designers Design Book, Peachpit Press Slide deck by Saul Greenberg. Permission is granted to use this for non-commercial purposes as long as general credit to E soul Greenberg is clearly maintained. Warning: some material in this deck is used from other sources without permission. Credit to the original source is given if it is known.



World Wide Web Info for Saul Greenberg - Microsoft Internet Explorer		
jle Edit View Favorites Tools Help		4
idress 🚳 C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_desi	gn (index. ł 🍟 🚦	> 0
A first lesson in Graphical Design		
Contrast		
Repetition		
Alignment		
Proximity		
Example: this page,		
home page		
Ditigial Prosanity 2 Algoment 3 Algoment 4 Repetition 5		



CRAP

Contrast Repetition Alignment Proximity

Robin Williams Non-Designers Design Book, Peachpit Press

CRAP



CRAP



CRAP

Robin Williams Non-Designers Design Book, Peachpit Press



CRAP



Robin Williams Non-Designers Design Book, Peachpit Press

Where does your eye go?

title

subtext

three points

main point

sub point

CRAP give you cues about how to read the graphic

Where does your eye go?

Boxes do not create a strong structure • CRAP fixes it



•	
•	
•	
•	
•	
•	

Robin Williams Non-Designers Design Book, Peachpit Press

Where does your eye go?

Some contrast and weak proximity

- ambiguous structure
- interleaved items

Robin Williams Non-Designers Design Book, Peachpit Press

÷÷	Image Preferences
	Image Type:
	Monochrome Gray Scale Color
	EPSF Quality:
	Use Preview Image Use Postscript
	Greeking Text Limit: 7.0 pt.
	Options: Auto Backup on Save
	Auto Save EveryMinutes
	(Apply) (Cancel) (Reset)

x

Good design

Recognize when you're not using them. Put it into words—name the problem.

Learn the principles. They're simpler than you might think.

Apply the principles. You'll be amazed.

Where does your eye go?

---140

Strong proximity (left/right split)

unambiguous

Image Type:	Monochrome Gray Scale Color
EPSF Quality:	Use Preview Image Use Postscript
Greeking	7.0 pt
Options:	Auto Backup on Save
	Auto Save Every Minutes
	(Anniv) (Cancel) (Reset)

✓

Image Preferences

1

Where does your eye go?

The strength of proximity

- alignment
- white (negative) space
- explicit structure a poor replacement

Mmmm:	Mmmm:	Mmmm:
Mmmm:	Mmmm:	Mmmm:
Mmmm:	Mmmm:	
Mmmm:	Mmmm:	Mmmm:
Mmmm:	Mmmm:	Mmmm:



File	Edit Vier	v Favoriter	Tools	Help		
Addres	ss 🐼 c-vo	ocuments and	Settings	isere	hical design\index.t 🗸	
			o c c c c c c c c c c c c c c c c c c c	an a		
A firs	st lesson in	Graphical Des	ign			
Contr	rast					
Repe	tition					
Align	iment					
Proxi	mity					
Exam	ple: <u>this pa</u>	<u>ge</u> ,				
home	e page					
Origin Proxi Align Contr	nal mity 2 ment 3 rast 4 titiza 5					
<u>r.epe</u>	anon 5					

🗿 World Wide Web Info for Saul Greenberg - Microsoft Internet Explorer
Ele Edit View Favorites Iools Help
Address 👹 C: {Documents and Settings Saul Greenberg Wy Documents Wy Webs saul hci_topics powerpoint_presentations 2003 graphical_design version 💌 🛃 Go
CroupLab ubulitviriltyofetigary Saul Greenberg GroupLab Dept Computer Science University of Calgary
Saul Greenberg, Professor Human-Computer Interaction & Computer Supported Cooperative Work Dept. of Computer Science University of Calgary Calgary, Alberta CANADA TXN IN4 Phone: +1 403 224-4070 Fax: +1 403 224-4070 Email: <u>saul@copsc.ucilgary.ca</u>
Research
GroupLab project describes research by my group
Publications by our group: most available in HTML, PDF, and postscript
Project snapshots describes select projects done in Grouplab
Grouplab software repository
Grouplab people
Graduate Students
I have a few openings for MSc and PhD students who are interested in Human Computer Interaction and / or Computer Supported Cooperative Work. <u>Some</u> research and project ideas honors and graduate students
Course offered this year Original
CPSC 481: Foundations and Principles of Human Computer Intera









Example of bad design

Advanced FAX Settings
Aptiva Communication Center
Speaker setting
O Dn 💿 On until connecti C Off
Wait 45 * seconds for connection
<u>Retry</u> after 60 $\frac{A}{V}$ seconds <u>N</u> umber of retries 3 $\frac{A}{V}$
Resolution
Fine O Standard
Maximum transmit rate: 14400 bps
Paper size: Letter (8½ x 11 in)
Use custom editor: xe C:\Phoenix\fax_inst.wii" Bromse
Save Cancel Help

Example of bad design

×b ugton	1 2.8 Beta 2 Server: cimer-bb.Corp
Load v Store (Submit v)	(View) (Print v) (Reset v) (Props) (Gen. Hel
Bug Id: Cc:	Mode : Edit Create Update fists
(Category >) .	Priority: 1 2 3 4 5
(Subcategory)	Severity: 1 2 3 4 5
(Resp Mgr	Bug/Rfe: hus rfe
(Stats p)	Responsible Engineer:
Synops is:	
Keywords:	
(Description) (Work around) (Suggested fix (Commants) (Public summery)
State triggers:	
(Evaluation)	
Commit to fix in rel	
Eivad in reitasser	
United to all and the second	
Cloud be must	
CHARGEN DECRIVER D	2
incompiete because p	<u>ٰ</u>
Root cause	
(Fix affects docs b)	
Duplicate of:	Interast list:
Patch id;	See also (bugids):
History:	
Submitter :	Date:
Generic SVR4 problem2: no yes]
Dispatch operator :	Date:
Evaluator :	Date:
Commit operator :	Date:

(Mullet & Sano, 1995)

IBM's Aptiva Communication Center

Reparing the layout

	Buguo		
Report v Vi	ew ♥ Props ♥ Help <	Mode: Cra	ate Édit
Bug ID: Category Subcategory Release: Status: Synopsis: Keywords: Pub Summary:	XView XView Ibrary 1.0 Submitted	Type: Bu Priority: 1 Soverity: 1	g RFE 2 3 4 5 2 3 4 5
See also: Interest List:			
Description	Work Around Suggested F	lx Commants	Evaluation
Root Cause:	🗑 documentation-confusin	9	
Rott Cause: Same as:	documentation-confusin	9	
Root Cause: Same as: Resp Mgr: Resp Engr:	b documentation-confusin b none b no	g 	
Root Cause: Same as: Resp Mgr Resp Engr Flags:	documentation-confusin one fin Affect Documentation Generic SV88 Problem	g HØvk 1: _ Høok 2: _ N	

Reparing the layout





« widgets » (window gadgets)



Facets of a widget

Facets of a widget

Presentation appearance

Behavior reaction to user actions

Interface with the application notification of state changes

Example: Button border with text inside « pressing » or « releasing » animation when clicked call function when the button is clicked

Variable wrappers (active variables)

two-way link between a state variable of a widget and another application variable (in Tcl/Tk referred to as *tracing*)



problems

- limited to simple types
- return link can be costly if automatic
- errors when links are updated by programmers

Event dispatching

widgets act as input peripherals and send events when their state changes

a while loop reads and treats events

associate an object to a widget, and its methods to changes in the widget state



Event dispatching



divide event sending and treatment better encapsulation (inside widget class)

Callback functions

Registration at widget creation



_____ DoSave (...) { ... }

Call at widget activation





Callback functions

Problem: spaghetti of callbacks

Sharing a state between multiple callbacks by

- global variables that widgets check: too many in real applications
- widget trees: callback functions are called with a reference to the widget that called it (visible in the same tree)
 Fragile if we change the structure of the UI, does not deal with other data not associated to widgets (e.g. filename)
- token passing: data passed with the callback function call

Callback functions

```
/* callback function */
void DoSave (Widget w, void* data) {
    /* retrieve file name */
     filename = (char**) data;
    /* call an application function */
    SaveTo (filename);
    /* close the dialog */
    CloseWindow (getParent(getParent(w)));
/* main program */
main () {
     /* variable with file name */
    char* filename = "";
    /* create a widget and assosiate a callback */
    ok = CreateButton (....);
    RegisterCallback (ok, DoSave, (void*) &filename);
     /* event manager loop */
    MainLoop ();
```

```
}
```

Event listeners (Java)

a variation of callbacks in Java:

- methods of type **AddListener** that do not specify a callback function but an object (the *listener*)
- when a widget changes state, it triggers a predefined method of the *listener* object (e.g. *actionPerformed*)

Event listeners (Java)

```
public class ClickListener implements ActionListener
{
    public void actionPerformed(ActionEvent e){
        JButton button = (JButton)e.getSource();
        ...
    }
}
...
ClickListener listener = new ClickListener();
JButton button = new JButton(''Click me'');
button.addActionListener(listener);
...
```

Event listeners (Java)

Anonymous Inner classes

Event listeners (Java)

Anonymous Inner classes

"new <class-name> () { <body> }"

this construction does 2 things:

creates a new class without name, that is a subclass of <class-name> defined by <body>
creates a (unique) instance of this new class and returns its value

this (inner) class has access to variables and methods of the class inside which it is defined



Events and listeners (Java)

Each has a source (e.g. JButton, JRadioButton, JCheckBox, JToggleButton,JMenu, JRadioButtonMenuItem, JTextField) Can get it with the function getSource() (Listeners) need to implement the interface that corresponds to event e.g. ActionEvent => ActionListener :

```
public interface ActionListener extends EventListener {
    /** Invoked when an action occurs.*/
    public void actionPerformed(ActionEvent e)
}
```

Events and listeners (Java)

all events inherit from the class EventObject

all listeners correspond to an interface that inherits from EventListener

a class receiving notification events of some type needs to implement the corresponding interface:

ActionEvent

ActionListener

MouseEvent

MouseListener KeyListener

- KeyEvent

• ...

Events and listeners (Java)

listeners need to be registered (added) to widgets

- a listener can be added to multiple widgets
 - e.g. one listener handles events from multiple buttons
- a widget can have many listeners
 - e.g. one for "click" events and for "enter" on button events

« drag-and-drop » to think about

What are the affected « widgets »? What are the events?



How to describe this interaction with a « event listener » ?

Interface toolkits

Event-action model

- can lead to errors (e.g. forgotten events)
- difficult to extend (e.g. add hover events)
- complex code

Hard to do things the toolkit was not designed for

e.g., multi-device input, multi-screen applications, advanced interaction techniques (CrossY)