The Auckland Interface Model (AIM)

HCI Guest Lecture Christof Lutteroth

# **Developing GUIs**

🏶 GUIDesign - Microso	oft Visual Studio
File Edit View Proje	ect Build Debug Data Format Tools Window Community Help
i 🛅 • 🛅 • 💕 📕 🗿	🖡 🔏 🗈 🙈 🍠 🗸 🔍 – 💭 – 🖳 – 🖳 🕨 Debug 🛛 – Any CPU 🚽 🍻 newLink 🚽 🔩 🚰 📸
Toolbox 🗸 🕂 🗙	
Pointer	
(ab) Button	Form1
CheckBox	private void InitializeComponent()
CheckedListBox	
ComboBox	this.button1 = new System.Windows.Forms.Button();
DateTimePicker	button1
A Label	
A LinkLabel	// button1
	this.button1.Location = new System.Drawing.Point(26, 29);
ListView	this.button1.Name = "button1";
#- MaskedTextBox	this.button1.Size = new System.Drawing.Size(121, 81);
MonthCalendar	this.button1.TabIndex = 0;
NotifyIcon	this.button1.Text = "button1";
NumericUpDown	this.button1.UseVisualStyleBackColor = true; //
	// Form1
_	
ProgressBar     RadioButton	this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
•	this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
RichTextBox	this.ClientSize = new System.Drawing.Size(233, 221);
abl TextBox	this.Controls.Add(this.button1);
ToolTip	<pre>this.Name = "Form1"; this.Text = "Form1";</pre>
TreeView	this.ResumeLayout (false);
Containers     Pointer	
Ready	☐ 15, 15 I <sup>II</sup> 186 × 181
nodey	

## Layout Managers

- 1. Get a layout specification as input
- 2. Recalculate the positions and sizes of the controls after each resizing

🔜 Form1	_ 🗆 🗙			
button1	button2			
button3				

🔜 Form1		- 🗆 ×	🔡 Form1	- 🗆 ×
button1	button2		button1	button2
bi	utton3			
			but	ton3

# Specifying GUI Layout

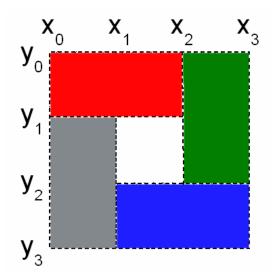
1	My Computer						. (		K
Г	file Edit View Fav	orites	Тр	os	Help			_	1
	🕤 Back 🔹 🌍 👻	٩	J	s	arch 😥 Folders 🔢 🕶				
ĥ	ddress 😼 My Compute	r		Г			~	→ Go	
Г				^	Name	Туре	Total	F	^
	System Tasks		2		Hard Disk Drives				
	View system				Second Disk (C:)	Local Disk	130 GB		
					🐲 Local Disk (U:)	Local Disk	130 GB		
	programs Change a setting				Devices with Removable Storage				E
				_	DVD-RW Drive (D:)	CD Drive			
	Other Places	4	۲	-	Network Drives				
	My Network Place	s			ﷺclut002 on 'afs\ec.auckland. \users\c\' (H:)	Network D	1.99 TB		
	My Documents				Rapps2005s1 on 'afs\ec.auckland.ac.nz\s	Network D	1.99 TB		
	Control Panel				🔀 all on 'afs' (Z:)	Network D	1.99 TB		
	Details		≈	_	Other				
	My Computer			~	Mobile Device	Svstem Fol		>	<b>~</b>
Γ				Γ					
Controls are aligned in a grid									

#### Areas

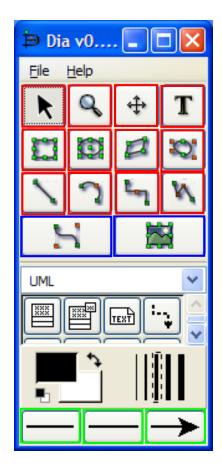
- Grid lines are variables with coordinates (tabs)
- Place controls by choosing left, top, right and bottom tab (area)
- Overlapping areas with layers

$$a =_{def} (x_1, y_1, x_2, y_2, layer, content)$$

 $A = \{(x_0, y_0, x_2, y_1, 0, red), (x_2, y_0, x_3, y_2, 0, green), \\(x_1, y_2, x_3, y_3, 0, blue), (x_0, y_1, x_1, y_3, 0, grey), \\(x_1, y_1, x_2, y_2, 0, empty)\}$ 



# Specifying GUI Layout



Same size

Same size Same height as above Double width as above

Same size 2/3 width as above

### Linear Constraints

 $C \subset \{ a_0 x_0 + \ldots + a_m x_m + b_0 y_0 + \ldots + b_n y_n OP c \\ \mid a_0, \ldots, a_m, b_0, \ldots, b_n, c \in \mathbb{R} \land OP \in \{\leq, =, \geq\}\}$ 

Absolute constraints

 $x_3 = 50. \qquad x_2 - x_1 = 100.$ 

- Relative constraints
  - Relative position  $x_{2} x_{1} = x_{3} x_{2} \Leftrightarrow -x_{1} + 2x_{2} x_{3} = 0.$   $x_{1} x_{2} x_{3}$ Relative size  $x_{2} x_{1} = 2(x_{4} x_{3}) \Leftrightarrow -x_{1} + x_{2} + 2x_{3} 2x_{4} = 0.$ Aspect ratio  $\frac{x_{2} x_{1}}{y_{2} y_{1}} = \frac{16}{9} \Leftrightarrow -x_{1} + x_{2} + \frac{16}{9}y_{1} \frac{16}{9}y_{2} = 0.$

Different units possible per constraint (cm, pixels)

# Problem



Designers working on a GUI:

- "Button A should be twice as wide as button B" (width  $= 2 \text{ width}_B$ )
- "Button B should be 100 pixels wide"
- "The GUI should be no more than 200 pixels wide"

### Infeasible!

# Linear Programming

#### Input:

Set of linear constraints C on variables

 $C \subset \{ a_0x_0 + \ldots + a_mx_m + b_0y_0 + \ldots + b_ny_n OP c \}$ 

 $a_0, \ldots, a_m, b_0, \ldots, b_n, c \in \mathbb{R} \land OP \in \{\leq, =, \geq\}\}$ 

Linear objective function to minimize

 $c_1x_1+c_2x_2+\ldots+c_nx_n$ 

**Output:** variable values so that...

- All constraints are satisfied
- The value of the objective function is minimal

### Soft Constraints

 $width_A = 2 width_B$  is a **hard** constraint

- 1. Allow  $width_A$  to be smaller or bigger  $width_A + d_1 - d_2 = 2 width_B \quad d_1 \ge 0, d_2 \ge 0$
- 2. width<sub>A</sub> should not be much smaller/bigger objective function =  $d_1 + d_2$

# Example 1

💀 Form1	
button 1	button2
butt	on3

💀 Form1		🗷 Test
button1 button2 textBox1 button3 button4 button5 button6 rchTextBox1	textBox2	55         2         7         7         7         7         2           55         63         5         12         75         34         7         19         9           2         47         48         10         17         7         34         7         46         51         52
listView 1		15 <b>143 39 100 41 7 37</b>
L		45 34 43 5 63 2 88 89 53 54 55 90 95 96 57
label1 label2	label3	8 88 89 53 54 50

LayoutSpec ls = new LayoutSpec(); XTab x1 = ls.AddXTab(); YTab y1 = ls.AddYTab();

- Is.AddArea(Is.Left, Is.Top, x1, y1, button1);
- Is.AddArea(x1, Is.Top, Is.Right, y1, button2);
- Is.AddArea(Is.Left, y1, Is.Right, Is.Bottom, button3);
- Is.AddConstraint(new double[] { 2, -1 }, new Variable[] { x1, Is.Right }, OperatorType.EQ, 0);

Is.AddConstraint(new double[] { 2, -1 }, new Variable[] { y1, ls.Bottom }, OperatorType.EQ, 0);

# Example 2

🖶 Form1		
A1		
	A2	
		A3

🐼 GenericEditor 📃 🗖	X
<ul> <li>B PD Metamodel : PD model  </li> <li>B ○</li> <li>C ○</li> <li>C ○</li> <li>C ○</li> <li>PD Metamodel : Object  </li> <li>C ○</li> <li< th=""><th>&lt; 11 12 12 12 12 12 12 12 12 12 12 12 12</th></li<></ul>	< 11 12 12 12 12 12 12 12 12 12 12 12 12
<ul> <li>         Entity type : Entity type</li></ul>	

LayoutSpec Is = new LayoutSpec(); Column c1 = Is.AddColumn(); Row r1 = Is.AddRow(); Row r3 = Is.AddRow(); r1.Next = r3; Row r2 = Is.AddRow(); r2.InsertAfter(r1); // ...

Area a1 = ls.AddArea(r1, c1, b1); a1.HAlignment = HorizontalAlignment.LEFT; a1.VAlignment = VerticalAlignment.TOP; Area a3 = ls.AddArea(r3, c1, b3);

r2.HasSameHeightAs(r1); r3.HasSameHeightAs(r1);

# Conclusion

#### • We can specify GUI layout with:

- □Tabstops
- Areas
- Linear constraints
- Linear programming makes it possible to transform hard constraints into soft constraints
- Link to project website on my homepage: http://www.cs.auckland.ac.nz/~lutteroth/

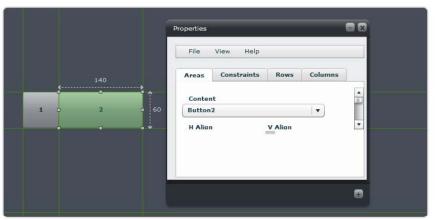
Thank you for your attention  $\bigcirc$ 

# AIM

#### **OTHER AIM PROJECTS**

# AIM Editor

- AIM comes with a built-in GUI layout editor
- Call AIMLayout.Edit() and you can change the current GUI
- Drag&drop of areas and visual specification of constraints
- Can also load/save the layout

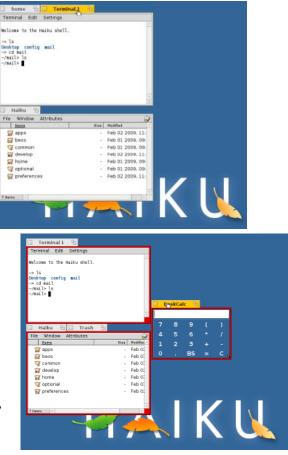


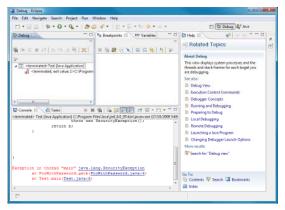
- Work in progress
- Make it a complete GUI editor
- Support for adding completely new controls, more visual constraint editing, …

http://genoupe.se.auckland.ac.nz/aimf

# Stack & Tile

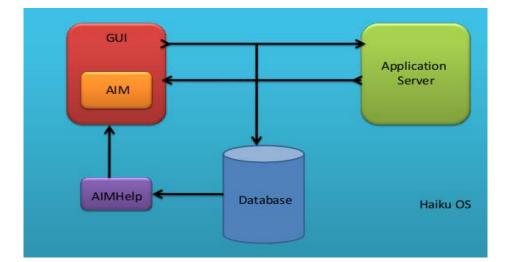
- New feature for a windowing system (in Haiku OS)
- Hold Windows key down to Stack&Tile windows
  - Stack window onto another by overlapping their tabs
  - Tile window to other windows by bringing edges close together
- Stack&Tile configuration is persistent (if supported by app)

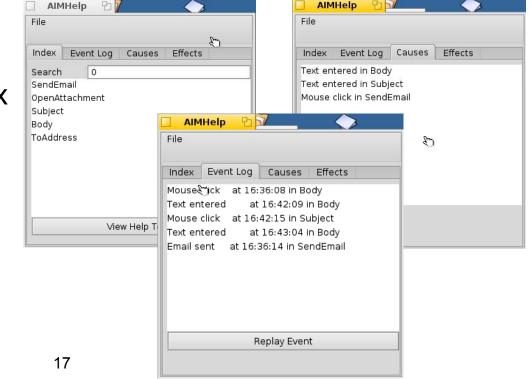




# HaikuHelp

- Generate help information automatically by analyzing input messages and state changes in a GUI
- Features:
  - Automated Help Index
  - Event Log
  - Causes Analysis
  - Effects Analysis







Questions and comments?  $\rightarrow$  lutteroth@cs.auckland.ac.nz