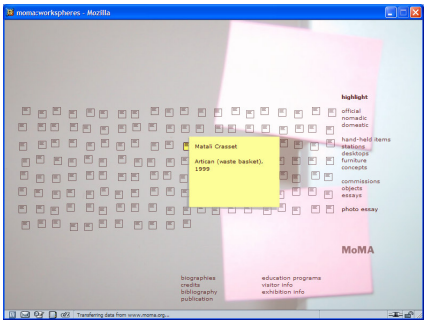
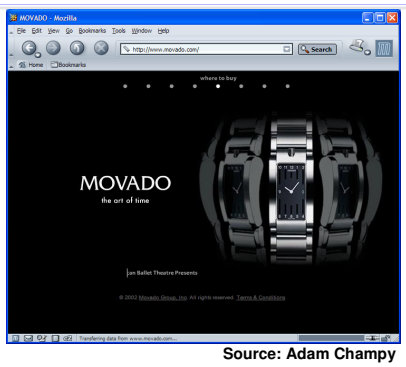
Lecture 10: Constraints and Layout

UI Hall of Fame or Shame?



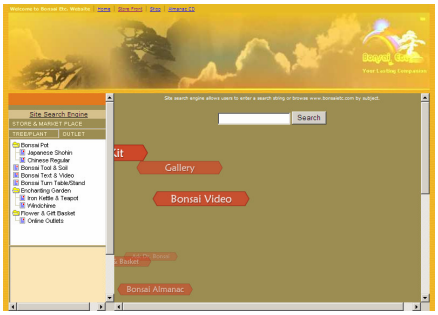
Source: Vishy Venugopalan

More "Mystery Navigation"



Source: Adam Champy

Let's Play a Menu Game



Source: Anson Tsai

Today's Topics

- Automatic layout
- Constraints

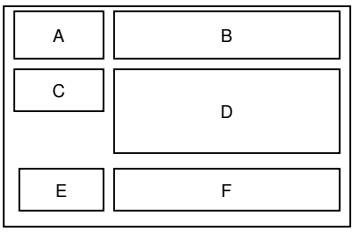
Layout

- Determining the positions and sizes of graphical objects

Layout Ranges in Difficulty

- Fixed constants
  - Many Windows dialog boxes
- Directly computable from model
  - Checkerboard from PS2/PS3
- One pass algorithm
  - Java layout managers, HTML tables
- Dynamic programming
  - paragraph flow with hyphenation
- Nonlinear optimization
- NP-hard
  - Graph with fewest edge crossings

Layout = Constraint Satisfaction



Layout Managers

- Also called geometry managers (Tk, Motif)
- Abstract
  - Represents a bundle of constraint equations
- Local
  - Involve only the children of one container in the view hierarchy

Layout Propagation Algorithm

- layout(Container parent, Rectangle parentSize)
  - for each child in parent,
    - get child's size request
  - apply layout constraints to fit children into parentSize
  - for each child,
    - set child's size and position

Kinds of Layout Managers

- Packing
  - one dimensional
  - Tk: pack
  - Java: BorderLayout, FlowLayout, BoxLayout
- Gridding
  - two dimensional
  - Tk: grid
  - Java: GridLayout, GridBagLayout, TableLayout
- General
  - Java: SpringLayout

Important Concepts

- Anchoring
- Expanding vs. padding
- Invisible components
  - Struts
  - Glue
  - Springs
- Nested containers

Totally Gridbag

Hints for Layout

- Use packing layouts when alignments are 1D
  - borders for top-level
  - nested boxes for internal
- Reserve gridding layouts for 2D alignment
  - unfortunately common when fields have captions!
  - TableLayout is easier than GridBag

Constraints

- Constraint: relationship expressed by the programmer and automatically maintained by the UI toolkit
- Uses
  - Layout
    - field.left = label.right + 10
  - Value propagation
    - deleteAction.enabled = (selection != null)
  - Synchronization of views to models
  - Interaction
    - rect.corner = mouse

One-Way Constraints

- Also called formulas, after spreadsheet
  - y = f(x1, x2, x3, ...)
  - Y depends on (points to) x1, x2, x3, ...
- Algorithms
  - Data-driven
    - Reevaluate formulas when a value is changed
  - Demand-driven
    - Reevaluate formulas whenever a value is requested
  - Lazy
    - When dependent value changes, **invalidate** all values that depend on it
    - When invalid value is requested, **recalculate** it

Variants

- Multi-output formulas
  - (y1, y2, ...) = f (x1, x2, x3, ...)
- Cyclic dependencies
  - Detect cycles and break them
- Constraint hierarchies
  - Some constraints stronger than others
- Side effects
  - If f has side effects, when do they happen?
    - Lazy evaluation makes side effects unpredictable
  - Amulet: eager evaluation

Multiway Constraints

- Each constraint is a multivariate relationship
  - rect.right = rect.left + rect.width − 1
  - Any variable may be used as target (different *method* for each target variable)
  - Planning step decides which variables to target

Variants

- Constraint hierarchy
  - Which value should be changed?
  - Each constraint has a priority
  - "Stay" constraints (highest priority) are used for constants
- Inequalities
  - Label.right <= field.left

Ultra-Lightweight Constraints

- Problem
- Approach
- Evaluation