

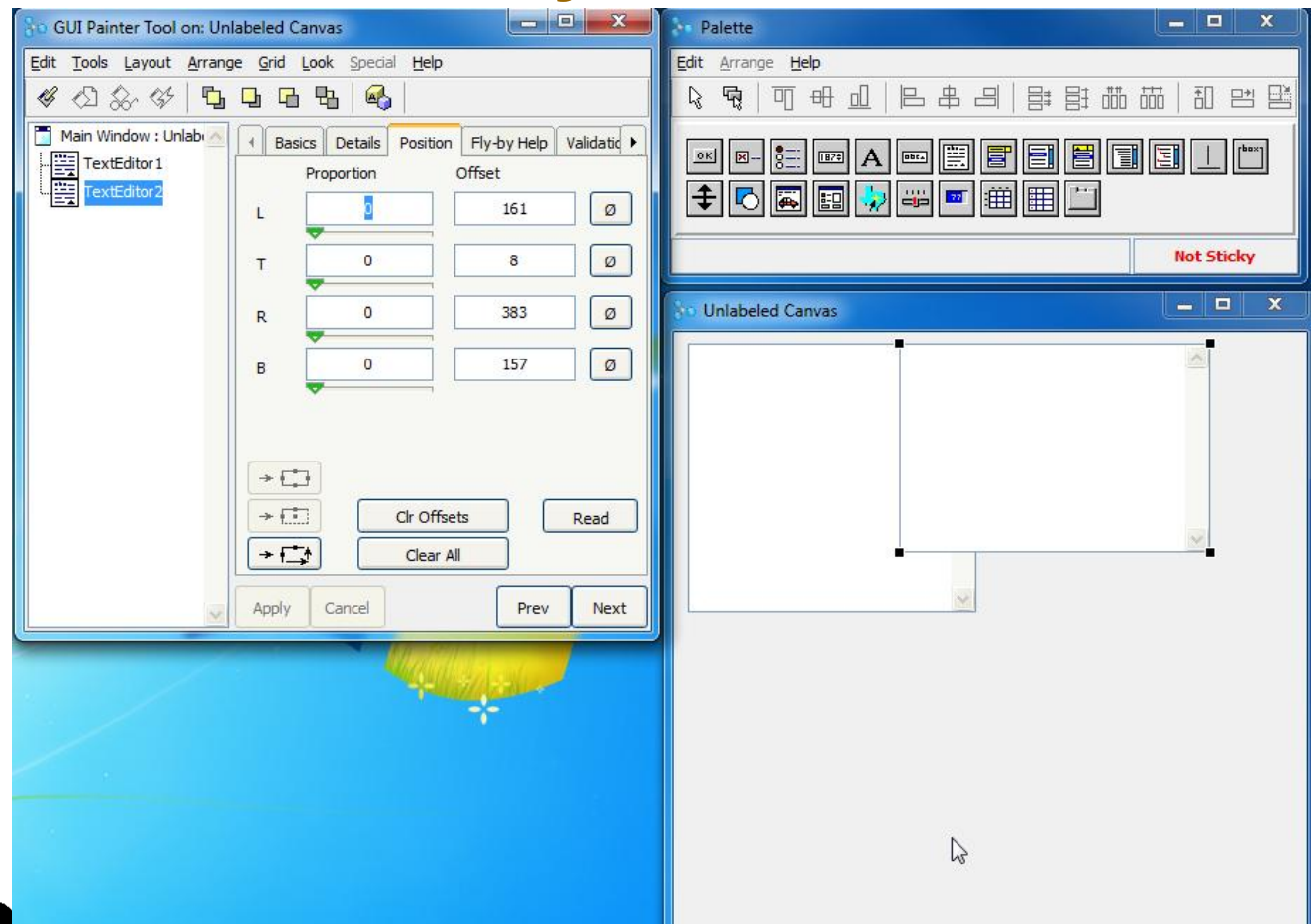


# **Fluid Positioning**

**Simberon Incorporated**

# Visual Works Layout

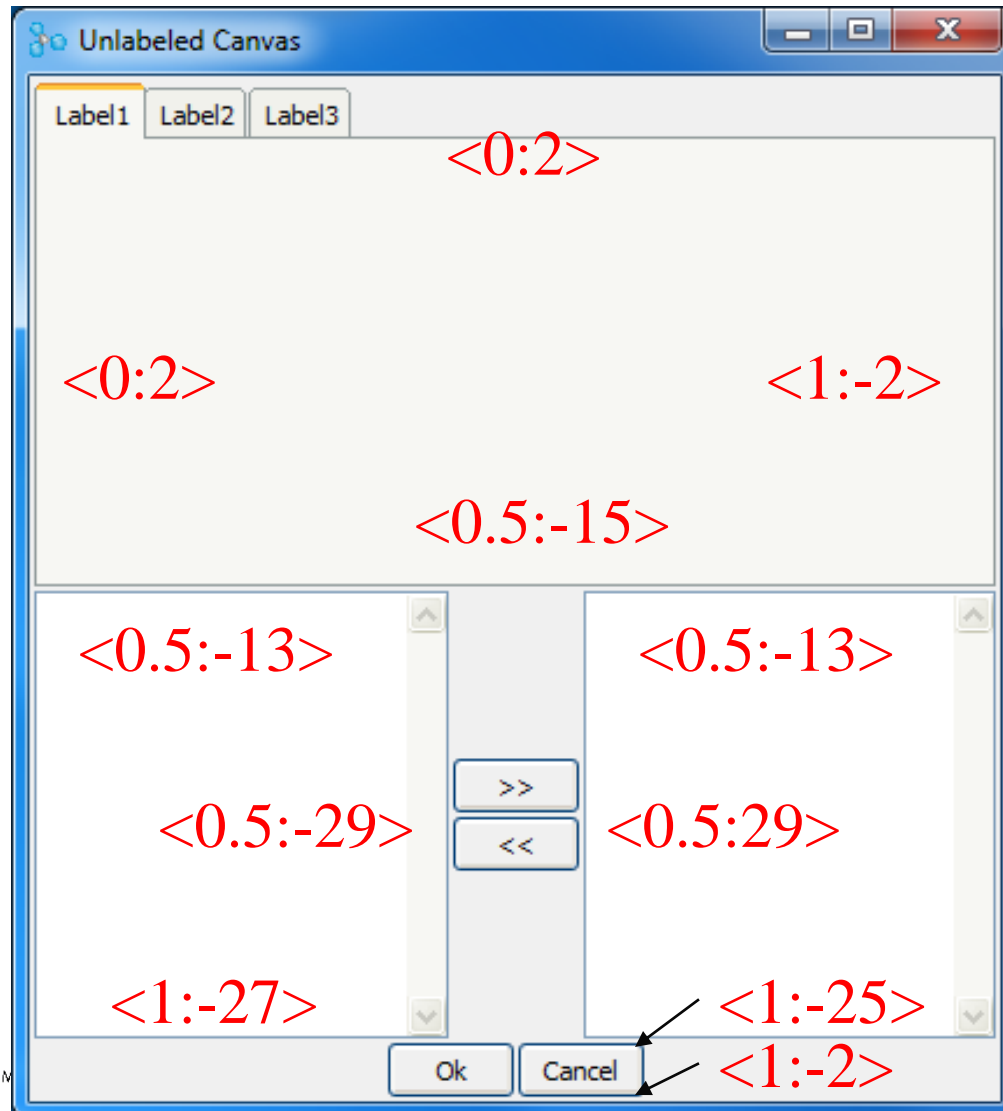
- Use UI Painter to layout UI's



# Position Parameters

- Each edge has a fraction and offset
- For left and right
  - $x = \text{width} * \text{fraction} + \text{offset}$
- For top and bottom
  - $y = \text{height} * \text{fraction} + \text{offset}$

# Hard to Calculate



# Goals

- **Easy automatic layout**
- **No fiddling with position parameters**
- **Variable sized widgets share available space**
- **Fixed size widgets snap into place**
  - **Along borders**
  - **At natural fractions**

# Research

- **Aukland layout algorithm**
  - **Uses constraints**
  - **Each constraint is an inequality**  
 $\text{box2left} > \text{box1right}$
  - **Constraints specified by code**
  - **Solved using Simplex Method**

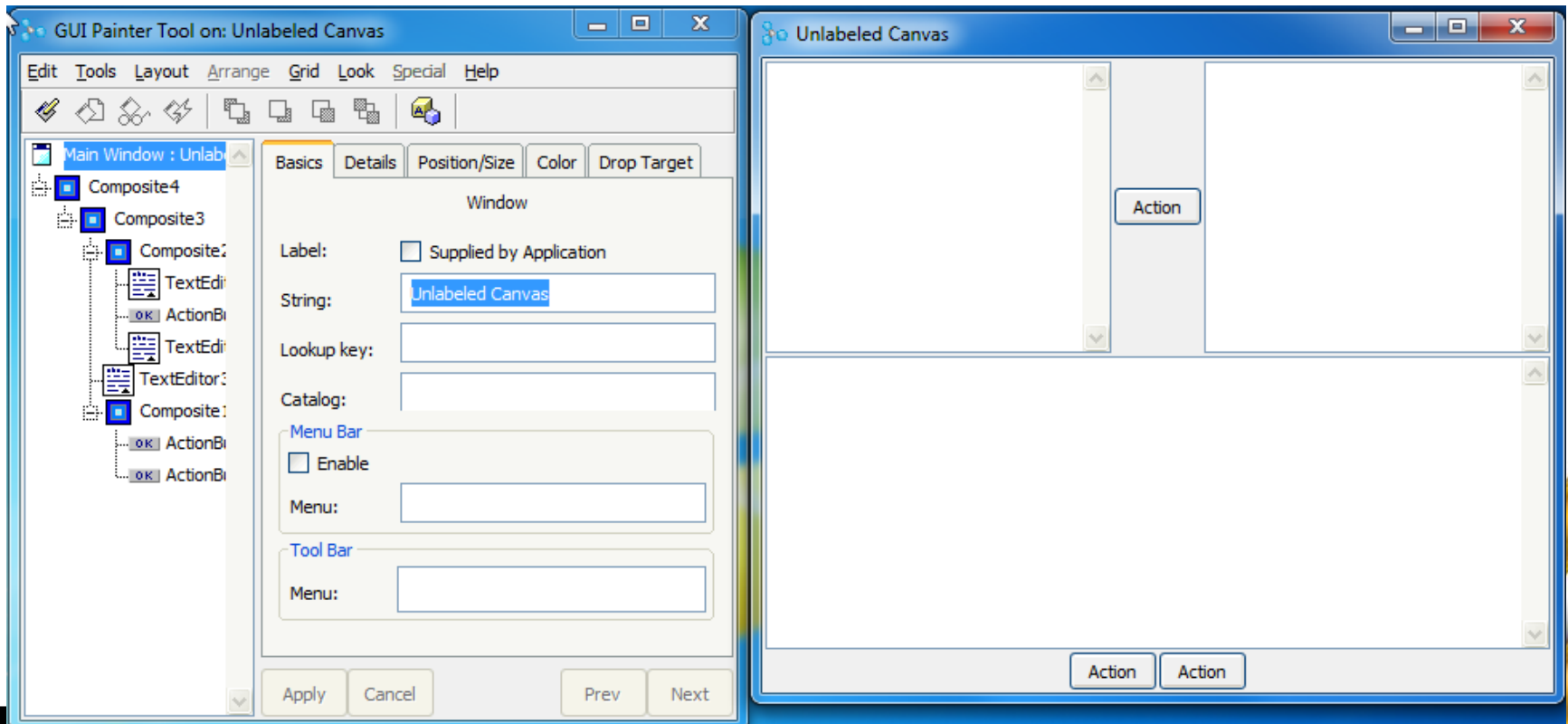
# Auckland Algorithm

- **Not suitable for VisualWorks**
  - No obvious way to use a UI Painter tool
  - Layout is calculated at run time
    - Algorithm doesn't translate to fraction/offset



# Layout Algorithm 1

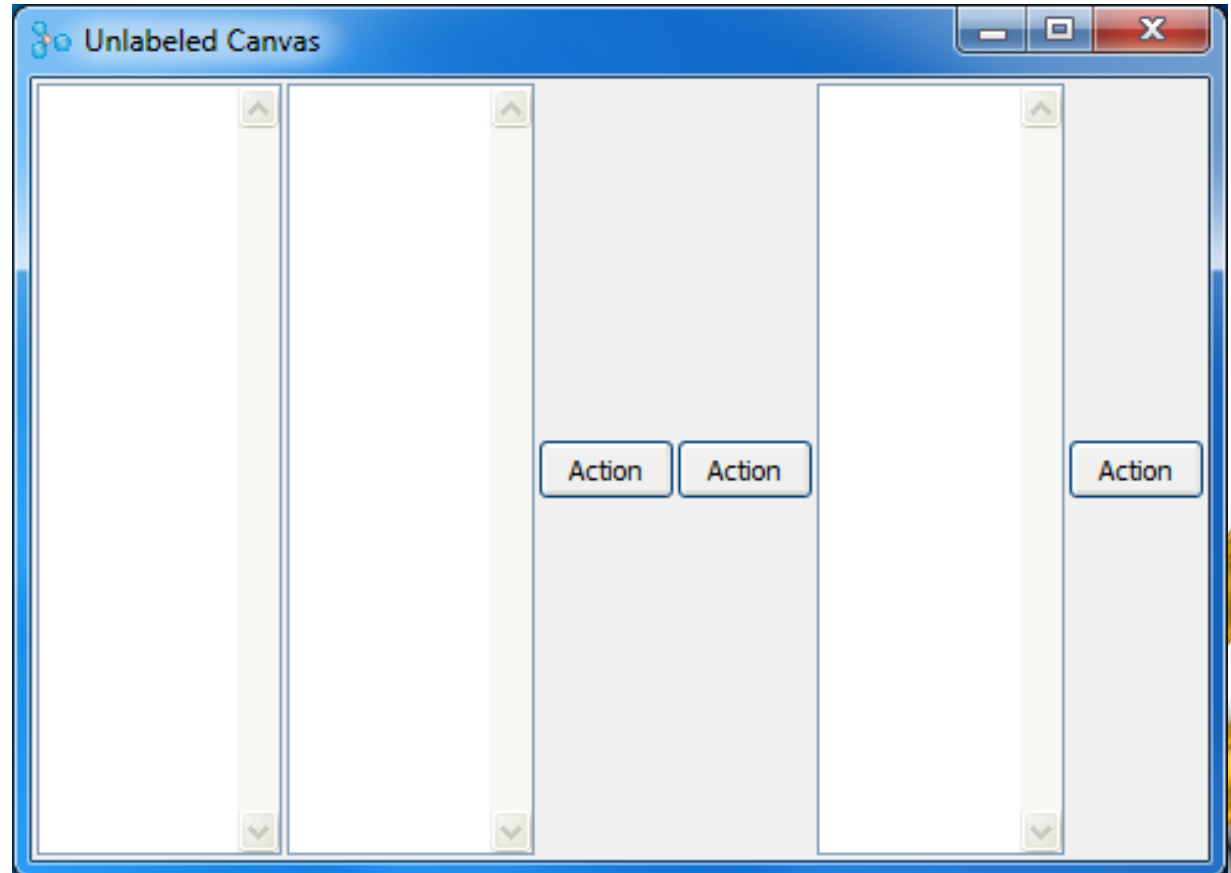
- Widgets are laid out in groups
- Each group is horizontal or vertical





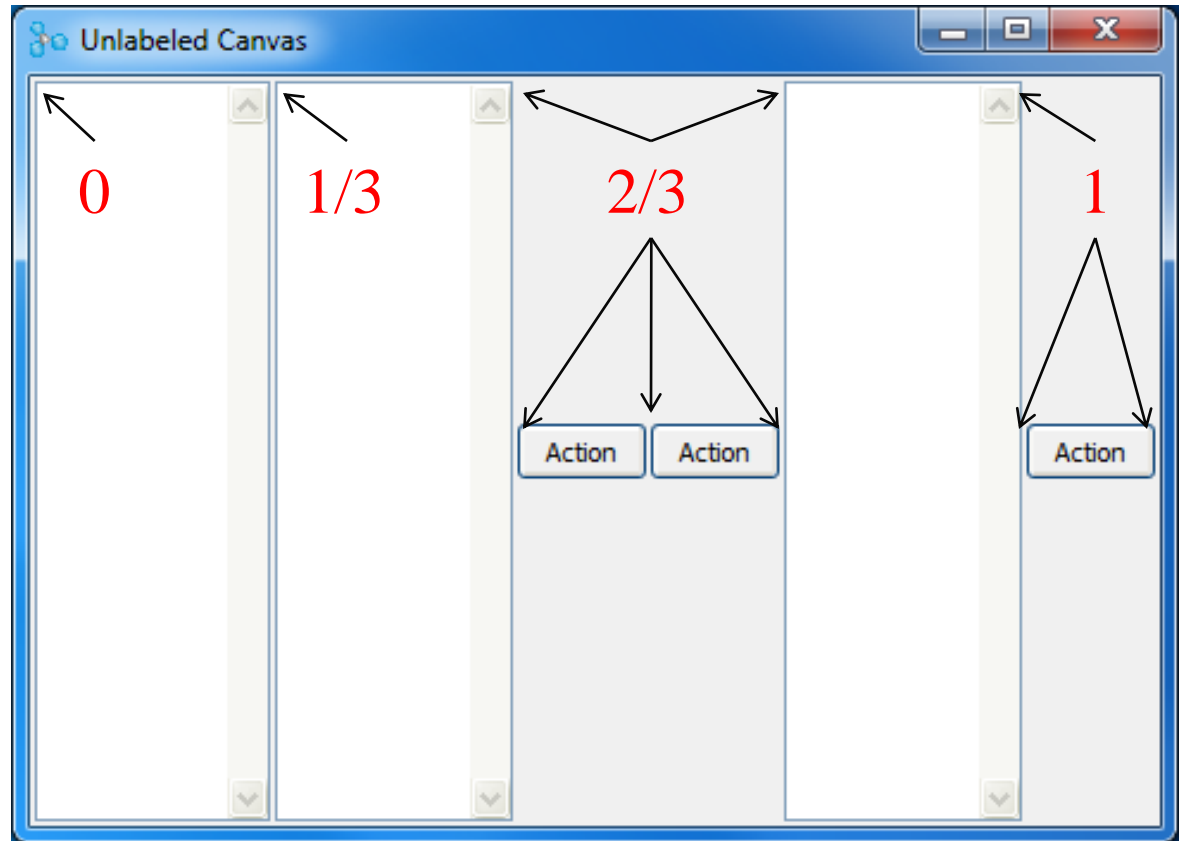
# How to Layout

- How do you compute fractions and offsets?



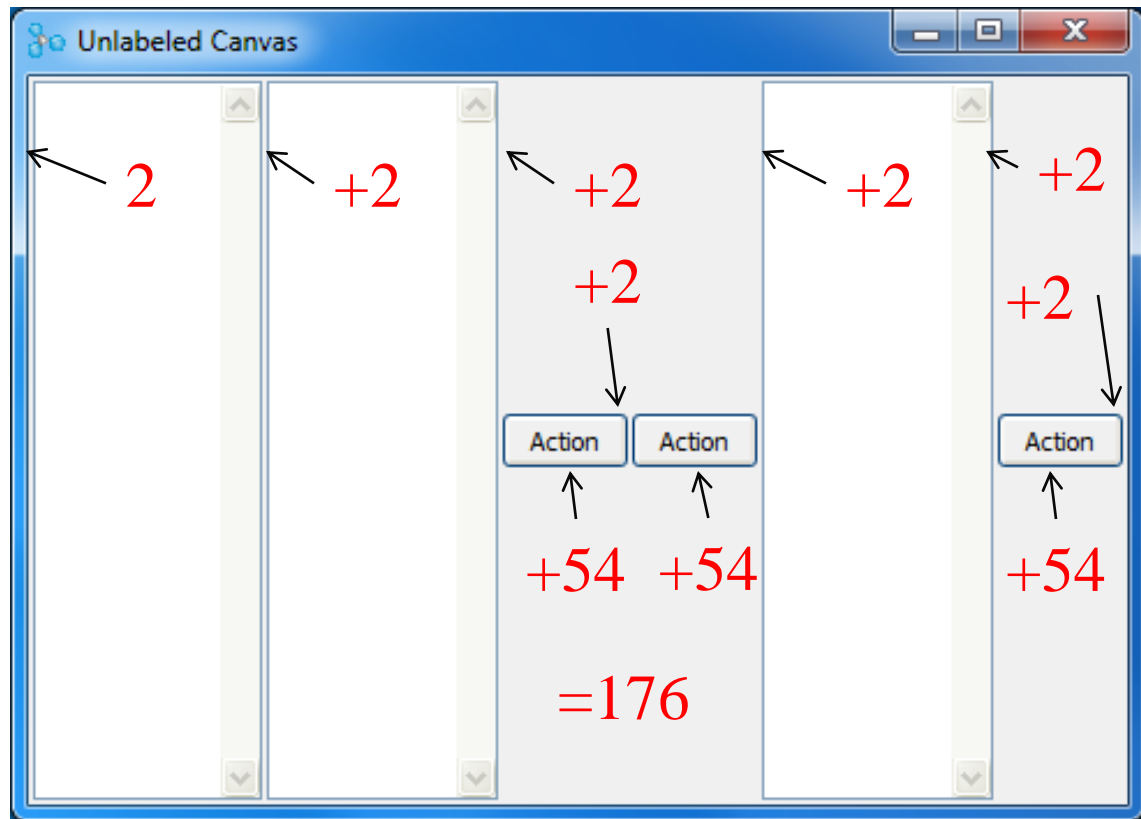
# Fractions

- Based on number of variable widgets



# Offsets

- Calculate total fixed size
  - Fixed widths + margins



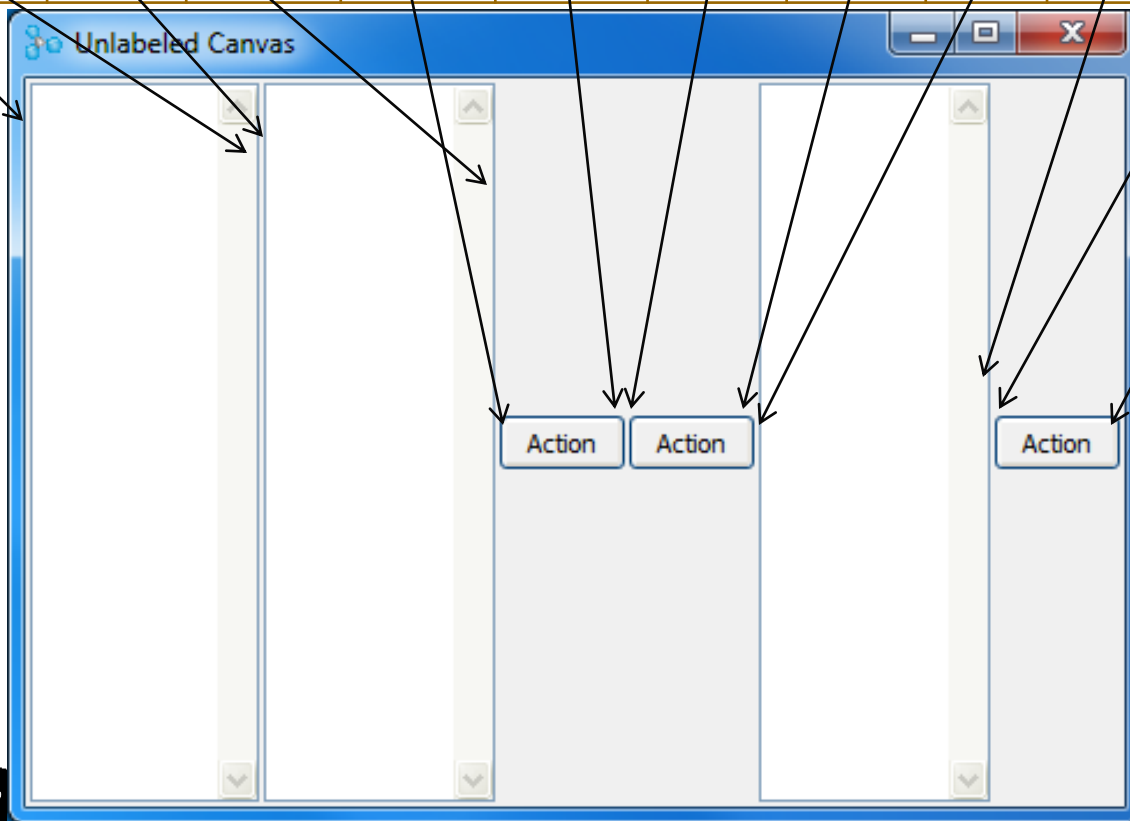
# Offsets

- **Keep a running offset**
  - **Add fixed offsets as you pass them**
    - Margins and fixed size widgets
    - $\text{offset} = \text{offset} + \text{fixedWidth}$
  - **For variable widgets, subtract a fraction of the total fixed width**
    - $\text{offset} = \text{offset} - (\text{fixedWidth} / \text{numVarWidgets})$
    - Round to nearest integer

# Offsets

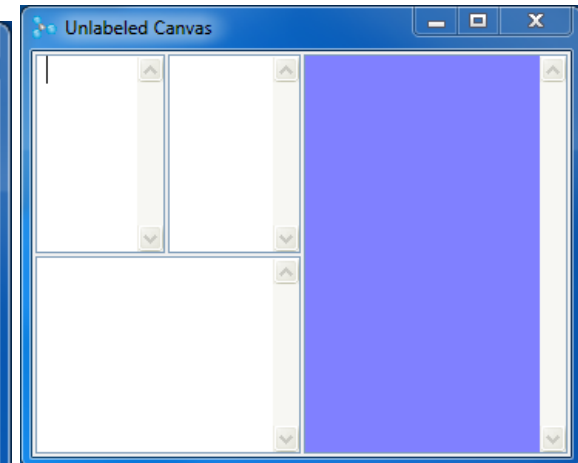
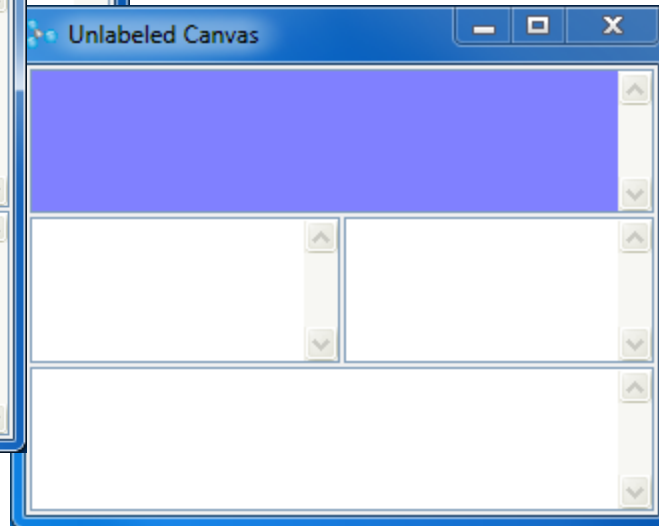
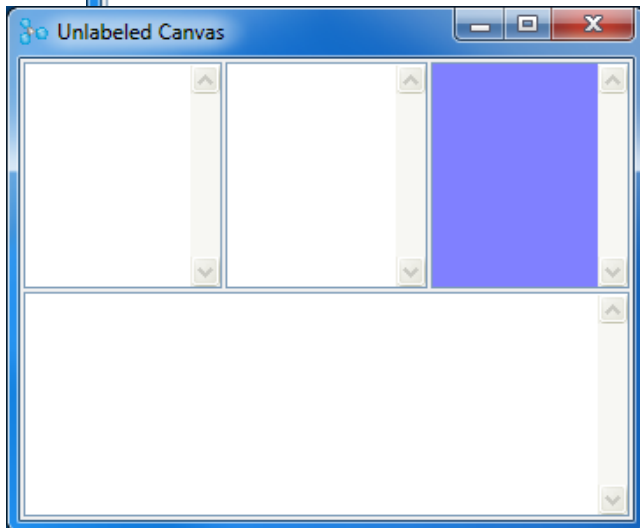
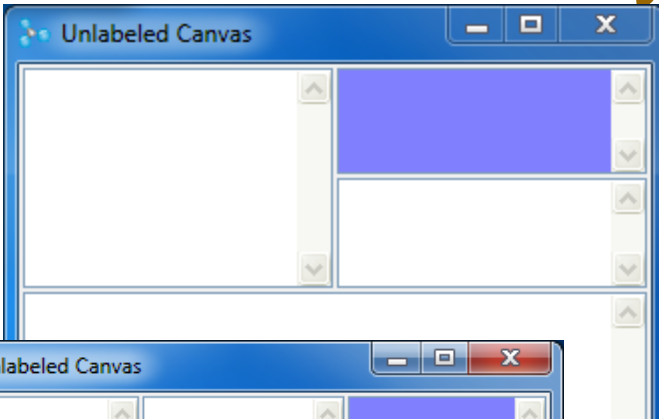
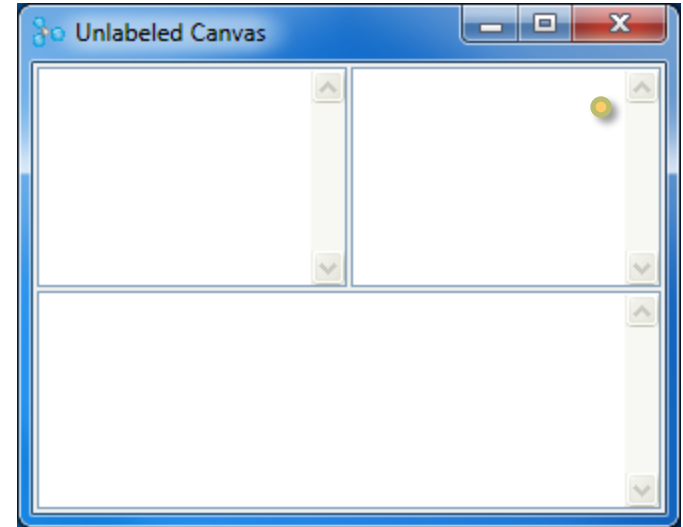
$$176 / 3 = 58.67$$

Initial	0	2	-56.67	-54.67	-113.33	-111.33	-57.33	-55.33	-1.33	0.66	-58	-56
Delta	2	-58.67	2	-58.67	2	54	2	54	2	-58.67	2	54
Current	2	-56.67	-54.67	-113.33	-111.33	-57.33	-55.33	-1.33	0.66	-58	-56	-2



# Ambiguity

- What do you mean?



# Difficulties

- **How do you move widgets?**
  - Coordinates are relative to parent
  - Changing parents requires recalculating layout
- **Bugs in VisualWorks**
  - Ungroup doesn't actually work right
  - Can't resize widgets in groups

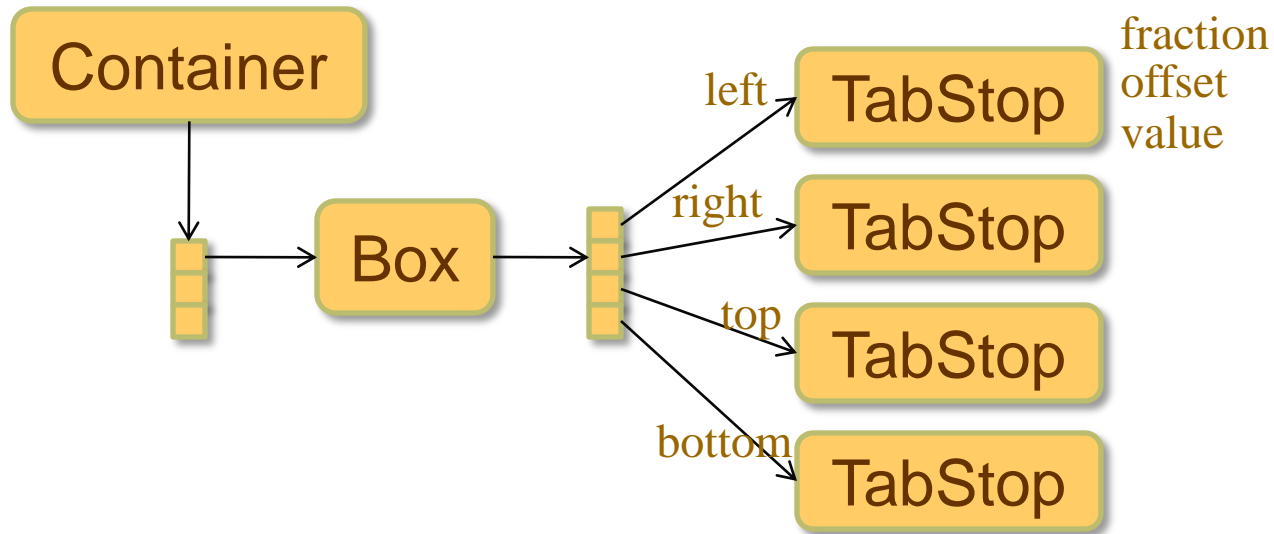


# Instabilities

- **Layout is done in each mouse move**
- **Major structure changes**
  - **Widgets grouped and ungrouped**
  - **Groups added and removed**
- **Result:**
  - **Instability**
  - **Small movements cause big changes**
  - **Hitting a moving target**

# Algorithm 2

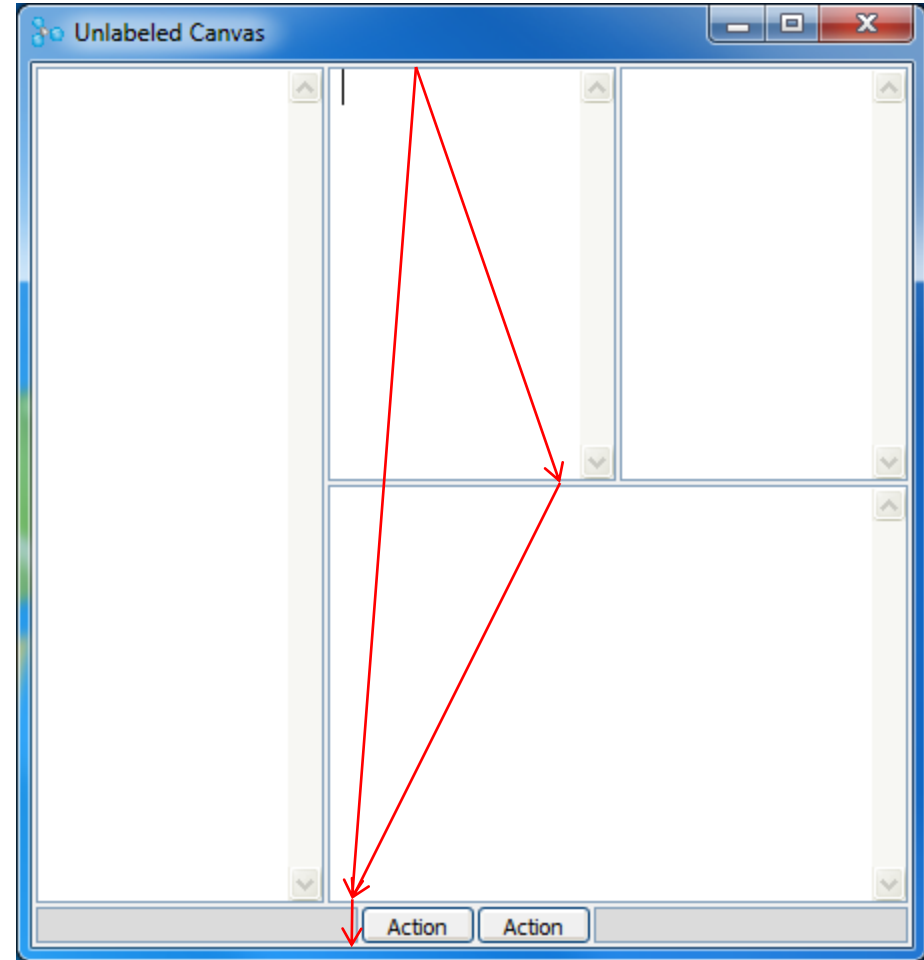
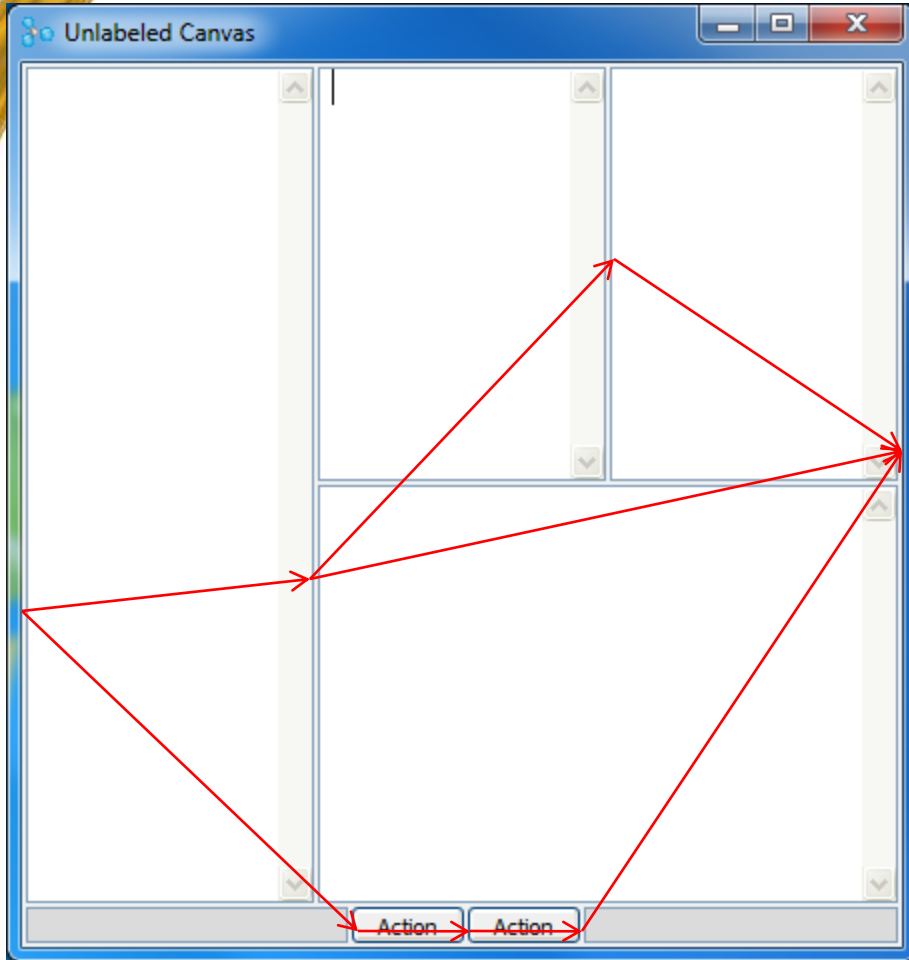
- No more oriented composites
- All layout is flat – no nesting
- Use an explicit model of the layout



# Tab Stops

- **Box edges are tab stops**
- **Tab stops can be shared**
- **Empty areas have hidden boxes**
- **Layout based on paths**
  - **Left to right**
  - **Top to bottom**

# Paths



# Layout

- **Choose best path to layout**
  - Longest? Shortest? Still experimenting
  - Prefer variable paths over fixed paths
- **Apply layout rule**
  - Calculate fractions and offsets
  - Calculate absolute values
- **Map results to real UI**
- **Refresh**

# Left to do

- **Finish adding fixed size widgets**
- **Ability to install and re-launch**
  - After you save and quit, how do you reload it?
  - What about windows not designed with fluid positioning?
- **Cleanup code removing old code**

# Lessons Learned

- **Developing an algorithm is hard**
- **Incremental development is limited**
- **Even though it's hard, write tests**
- **Can't depend solely on tests**



# Current State

- **80/80/80 rule**
  - Algorithm is 80% finished
  - Only the remaining 80% to go
  - Remember the final 80% that's hard
- **Code will be open sourced**
  - Will release once reasonably stable