

# Mobile Application Development

Lecture 24  
Auto Layout

# Lecture Summary

- Auto Layout
- DEMO



# Hard-coded Layout

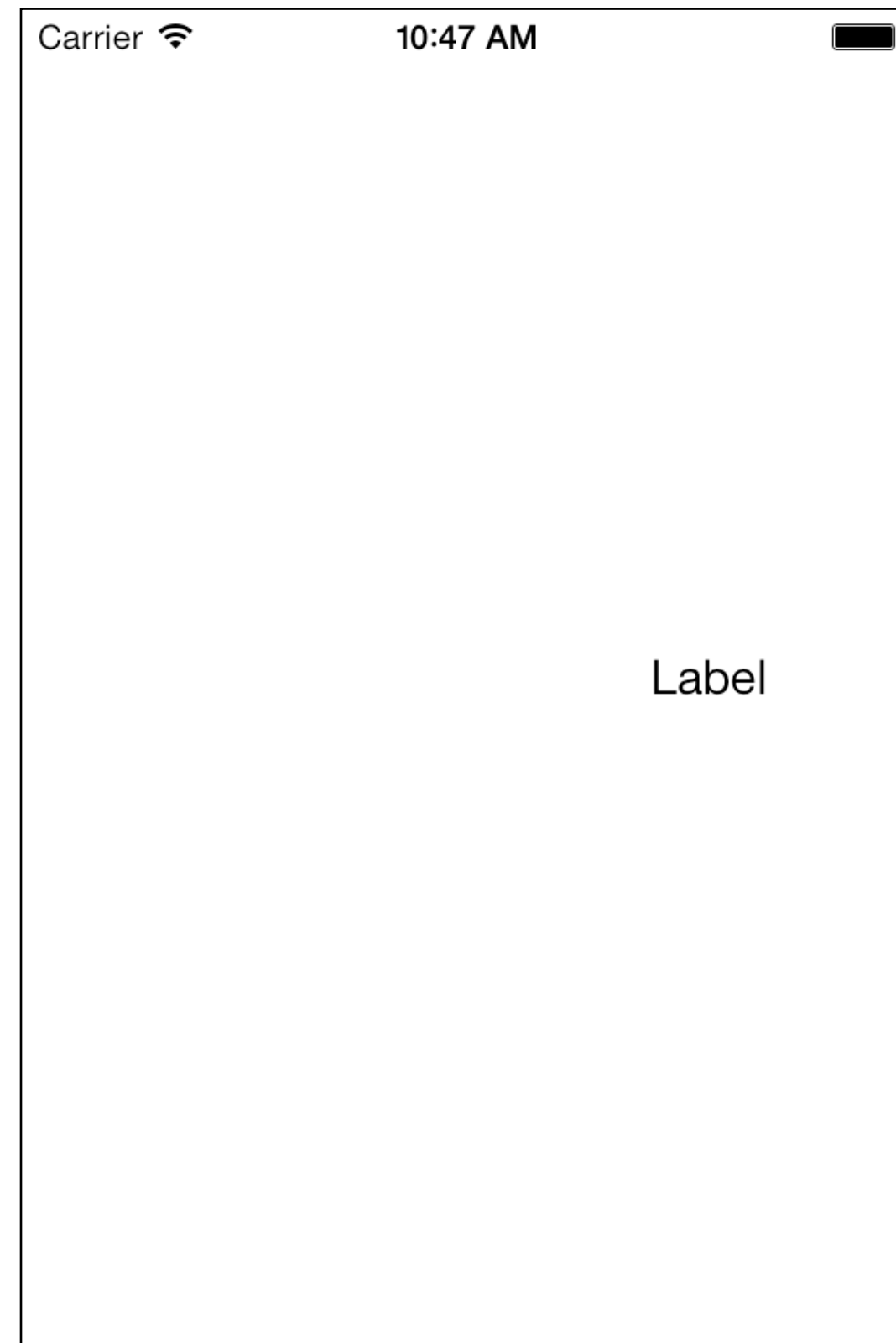
- In storyboard, we can add subviews to a view controller's view by dragging it a view object from the object palette
- Let's place it in a "random" place - also using the blue guidelines
- The label has a fixed (static) frame starting at point (226,229)



<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/AutolayoutPG>

# Hard-coded Layout

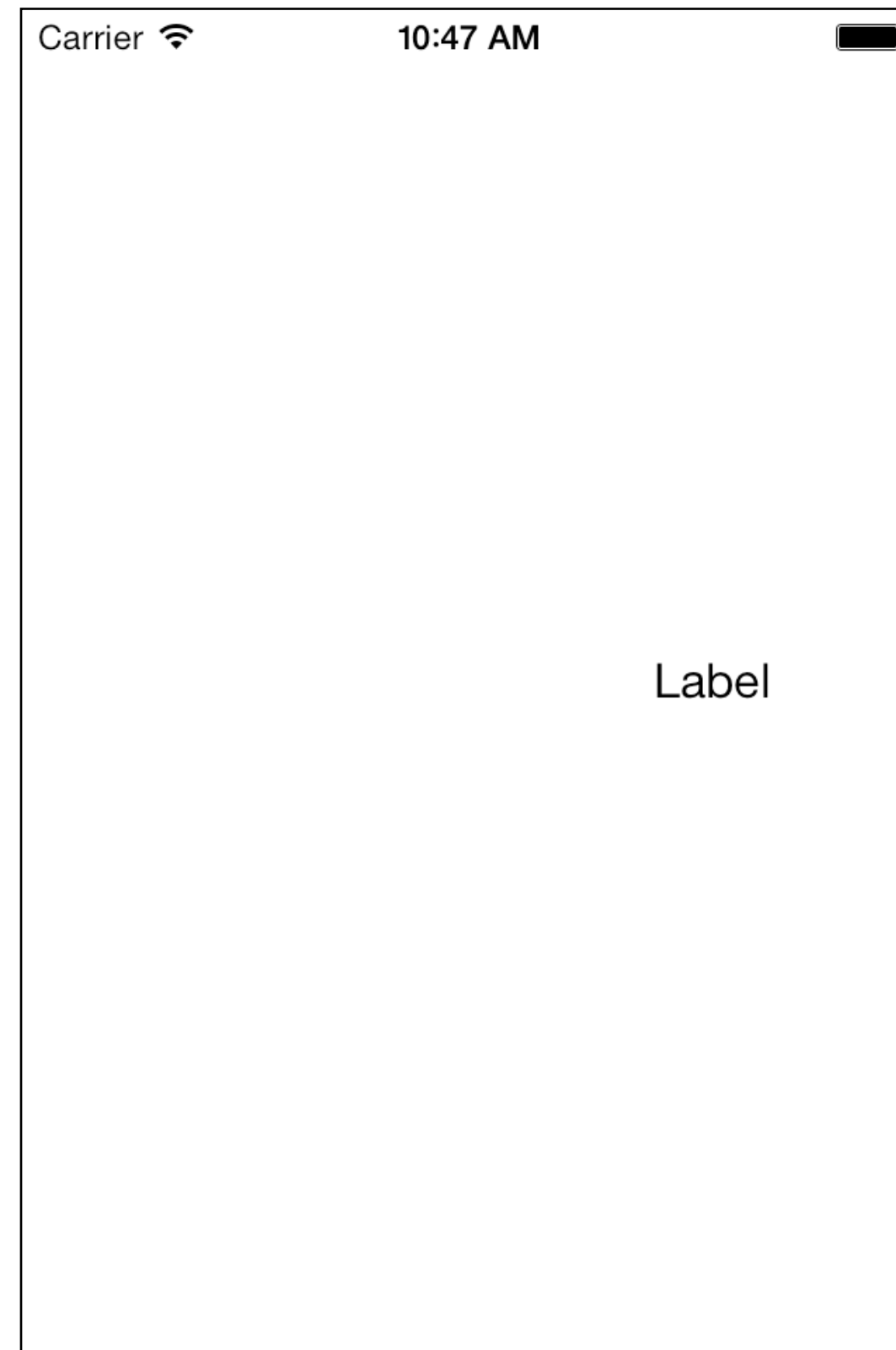
- Everything ok, the label is just where it was placed



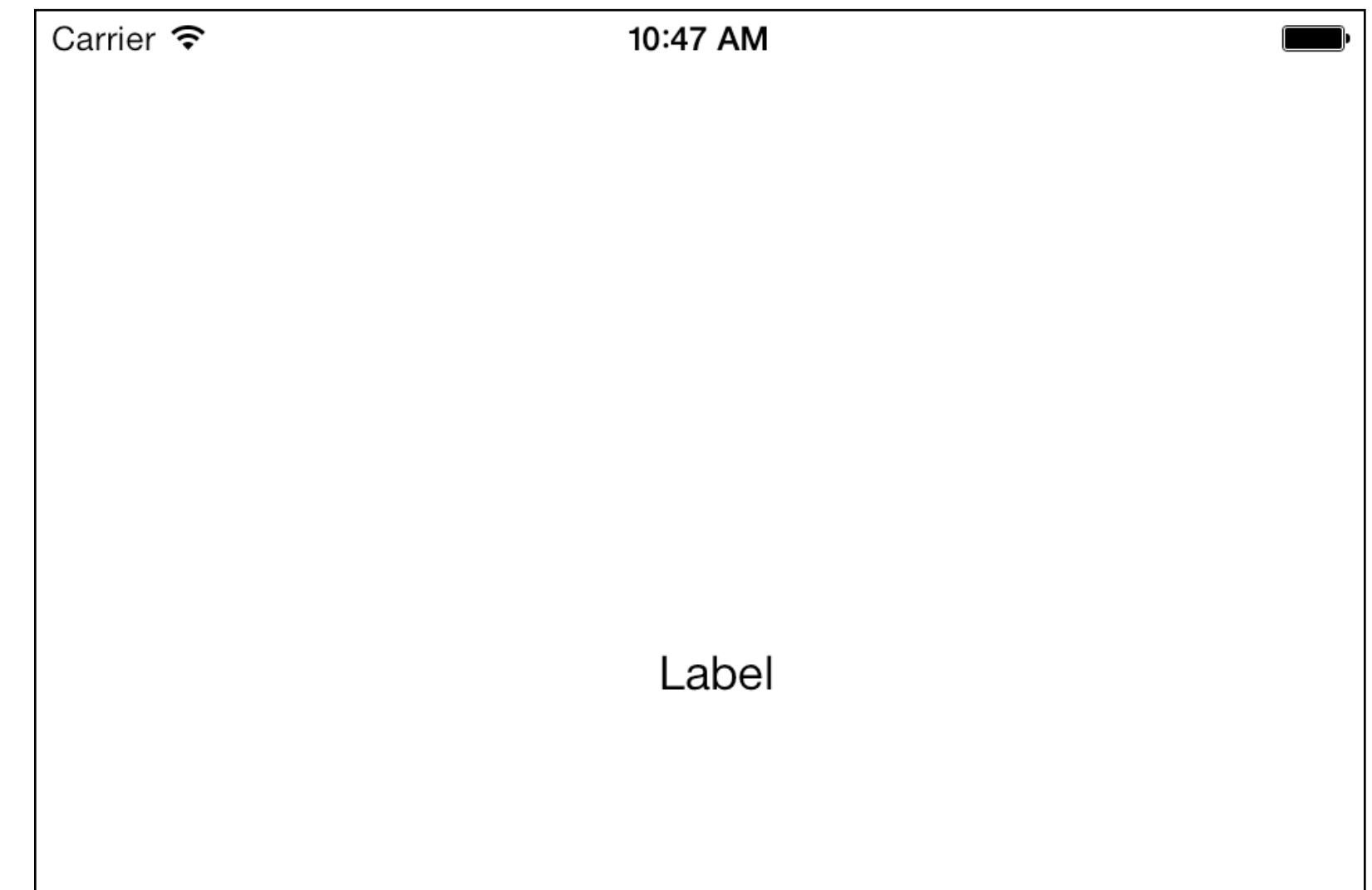
Portrait

# Hard-coded Layout

- What happen when the device is rotated?



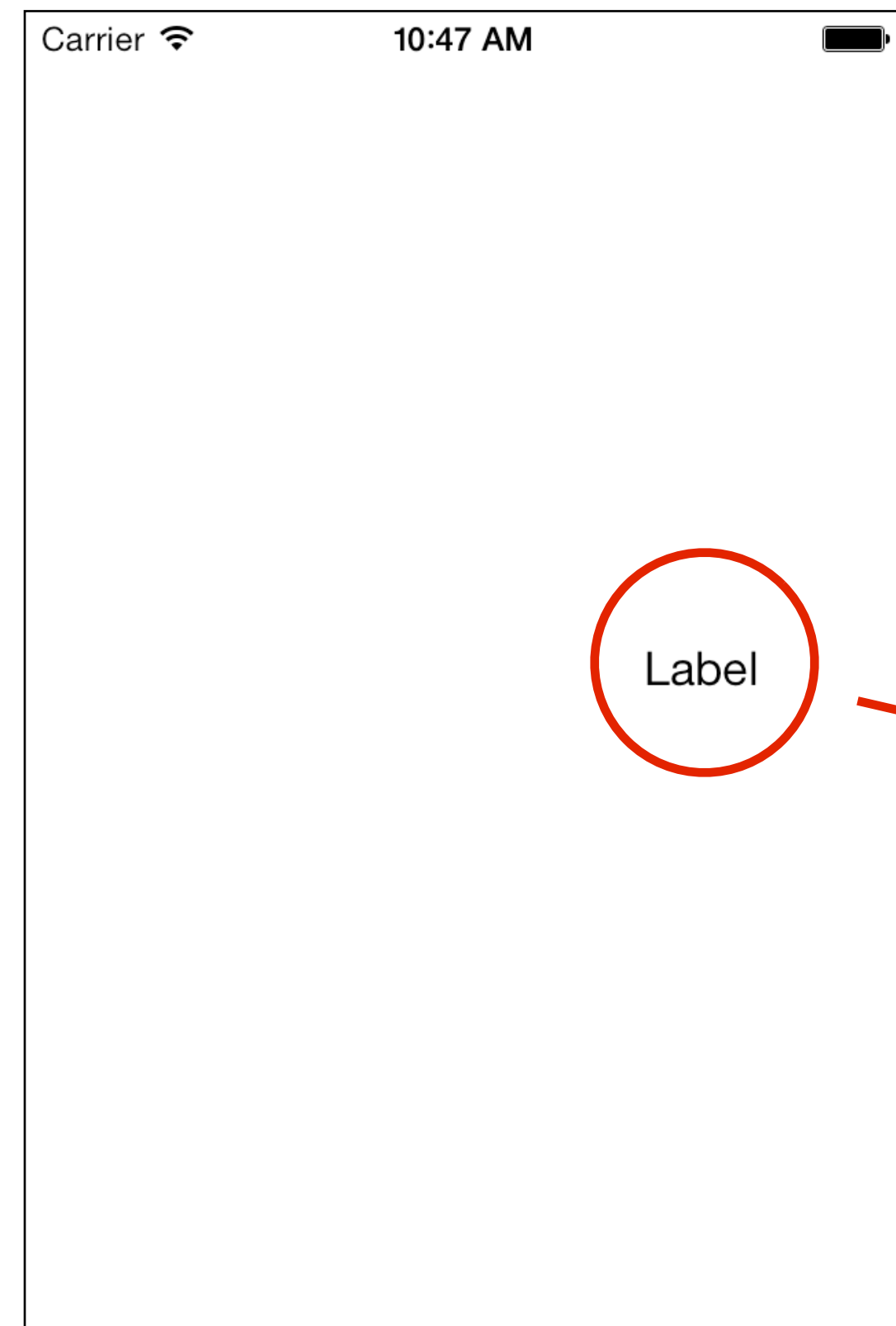
Portrait



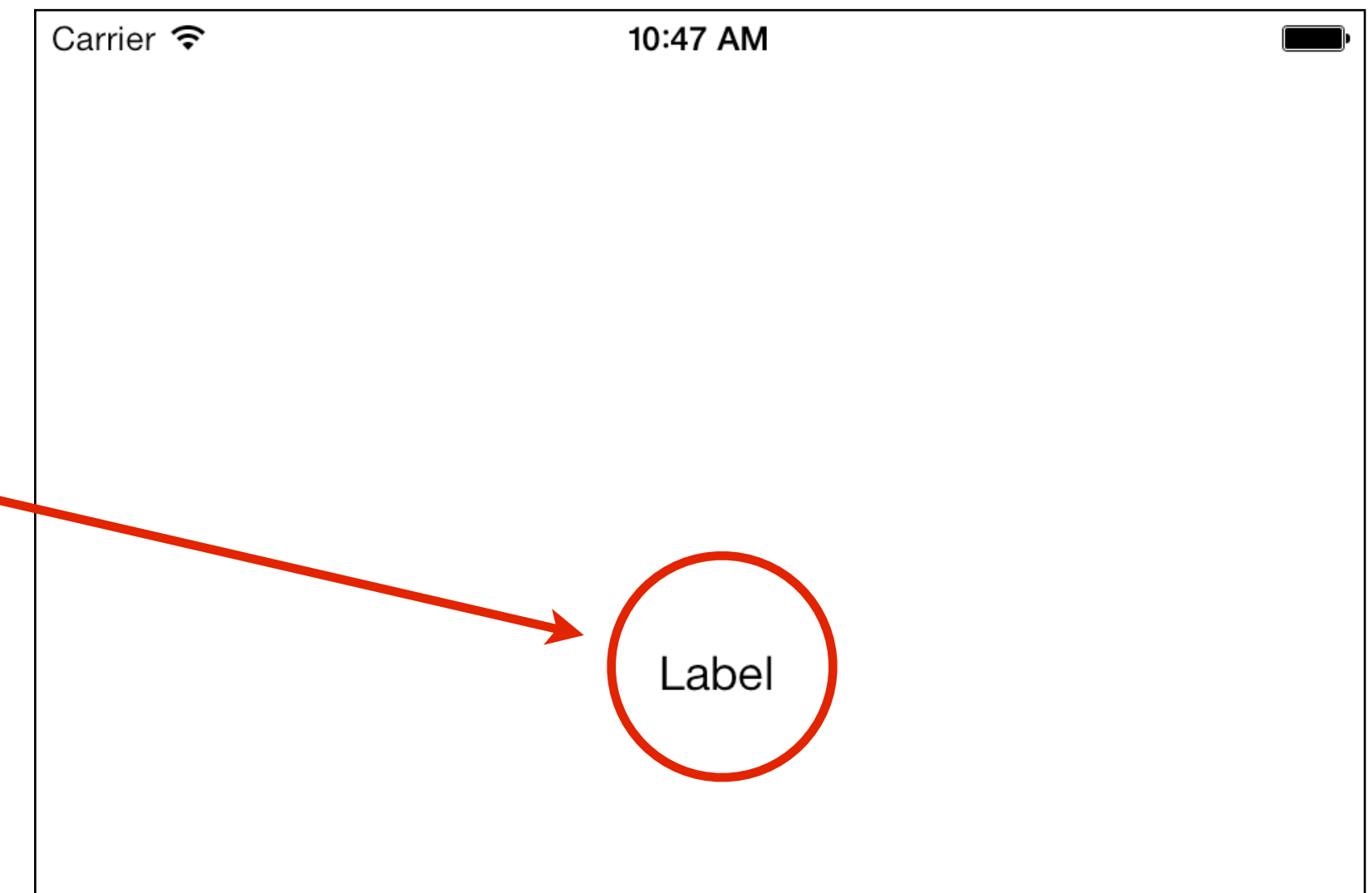
Landscape

# Hard-coded Layout

- The label has a fixed (static) frame starting at point (226,229)
- When switching to landscape mode, the label maintains its original frame
- It would be best to have the UI adapt to the orientation automatically (e.g. stay at the center vertically)

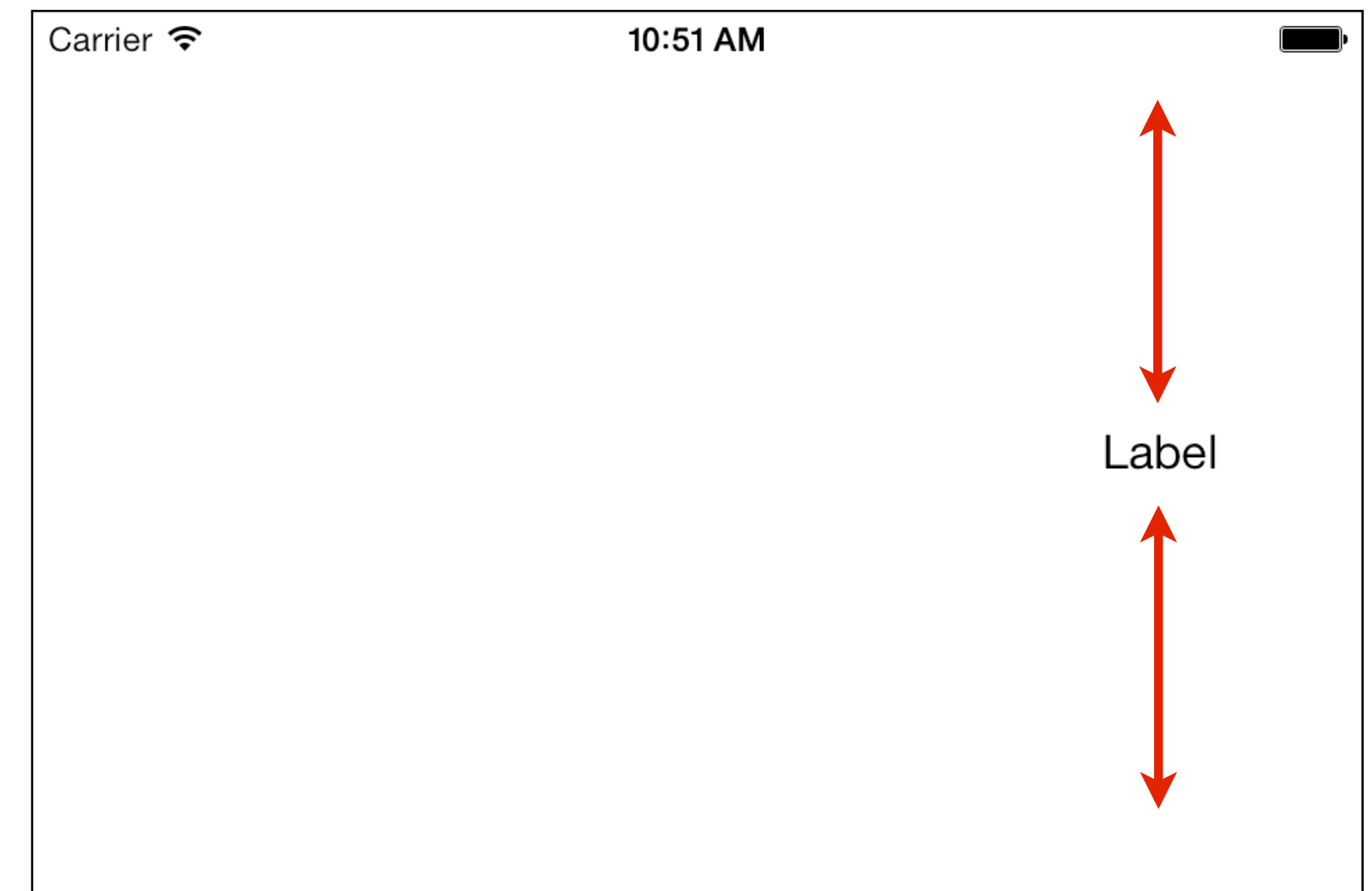
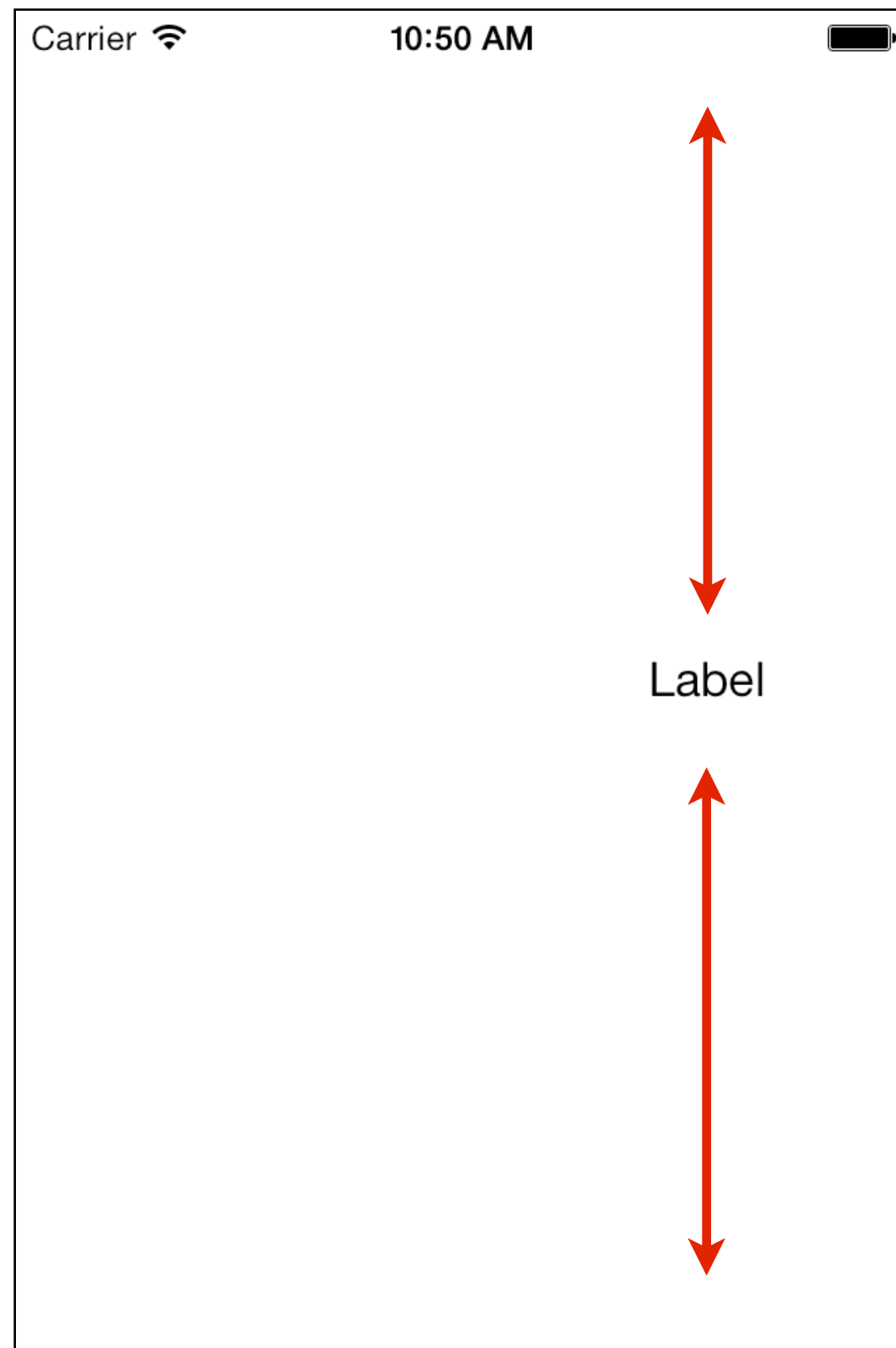


Portrait



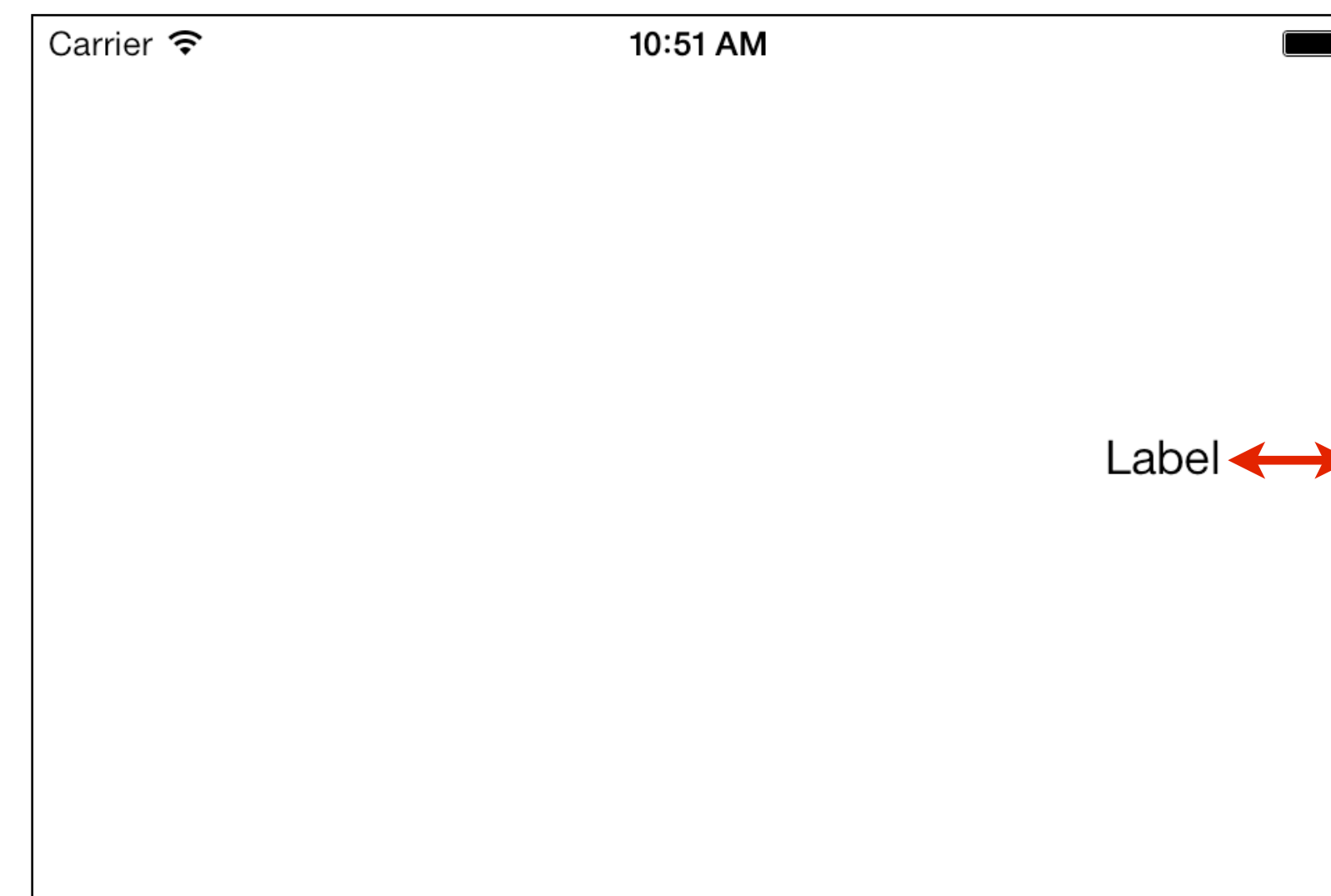
Landscape

# Auto Layout



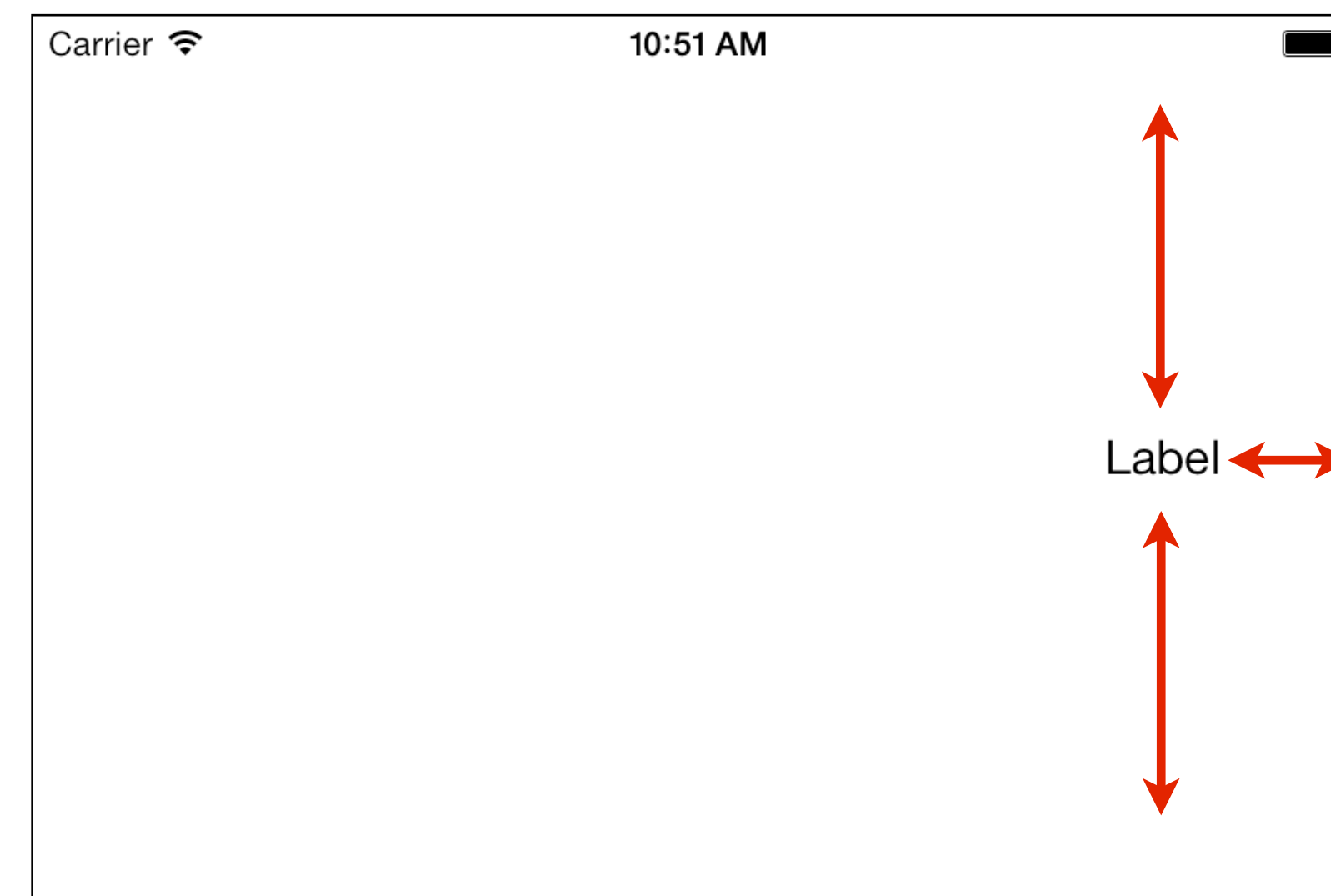
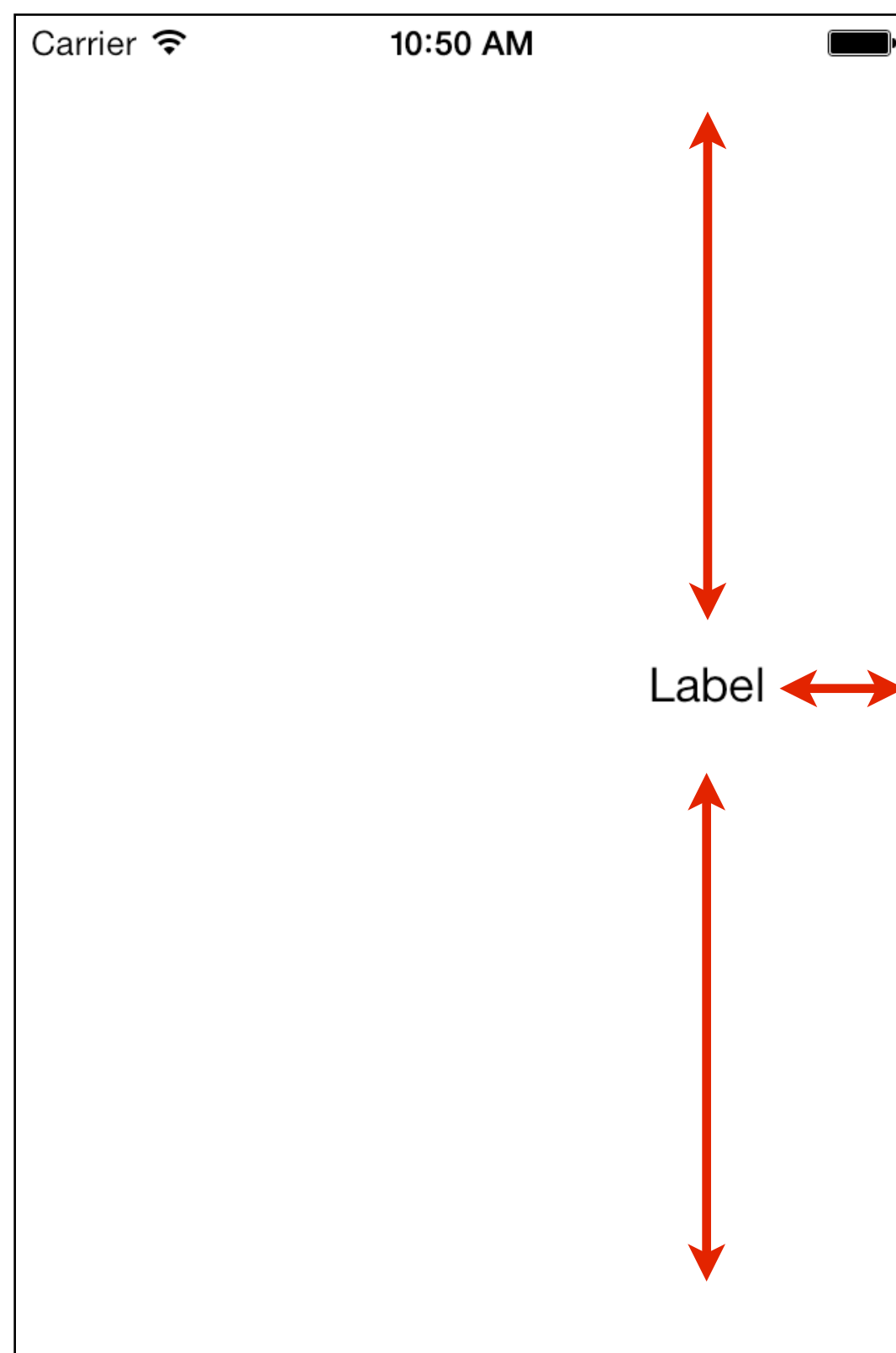
- The label is vertically centered in the container view

# Auto Layout



- The label has a fixed trailing space (to the right edge of the container)

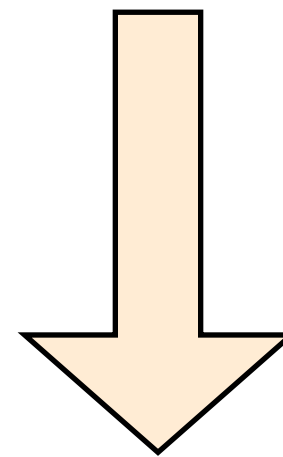
# Auto Layout



- The label is vertically centered in the container view
- The label has a fixed trailing space (to the right edge of the container)

# Auto Layout

- The label is vertically centered in the container view
- The label has a fixed trailing space (to the right edge of the container)



```
Control.centerY = Superview.centerY  
Control.right = Superview.right - <padding>
```

- Auto Layout takes these human-readable rules defined and transforms them into **constraints**

# Auto Layout

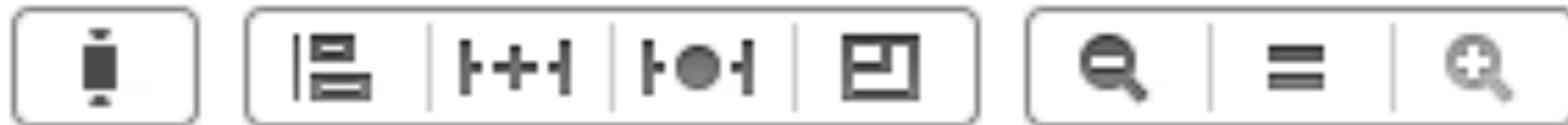
- Auto Layout is a constraint-based, descriptive layout system
- Auto Layout makes it easy to solve many complex layout problems automatically, without the need for manual view manipulation
- With Auto Layout, the layout with constraints is described, and frames are calculated automatically
- Constraints are defined as relationships between the view objects (allows to describe very complex relationships)
- Auto Layout keeps software less bug-prone: it avoids bugs that might be introduced if manual view manipulation were adopted (no code required!)
- Auto Layout makes it easy to deal with:
  - orientation (portrait vs. landscape orientation)
  - different screen sizes (iPhone 4 (3.5" screen) vs. iPhone 5 (4" screen))

# Constraints

- A constraint is a sort of mathematical representation of a human-expressible statement
- *The left edge should be 20 points from the left edge of its containing view* translates to `button.left = (container.left + 20)`
- The constraint can be therefore represented by a formula:  $y = m * x + b$ , where
  - $y$  and  $x$  are attributes of views
  - $m$  and  $b$  are floating point values
- An attribute is one of *left*, *right*, *top*, *bottom*, *leading*, *trailing*, *width*, *height*, *centerX*, *centerY*, and *baseline*

# Using Auto Layout

- Auto Layout is managed visually in Xcode
- In storyboard, in the lower-right corner, are the controls for using Auto Layout



# Using Auto Layout

- Auto Layout is managed visually in Xcode
- In storyboard, in the lower-right corner, are the controls for using Auto Layout



Switch between iPhone 3.5" and iPhone 4" screen sizes

# Using Auto Layout

- Auto Layout is managed visually in Xcode
- In storyboard, in the lower-right corner, are the controls for using Auto Layout



**Set alignment constraints:** create alignment constraints, such as centering a view in its container, or aligning the left edges of two views

# Using Auto Layout

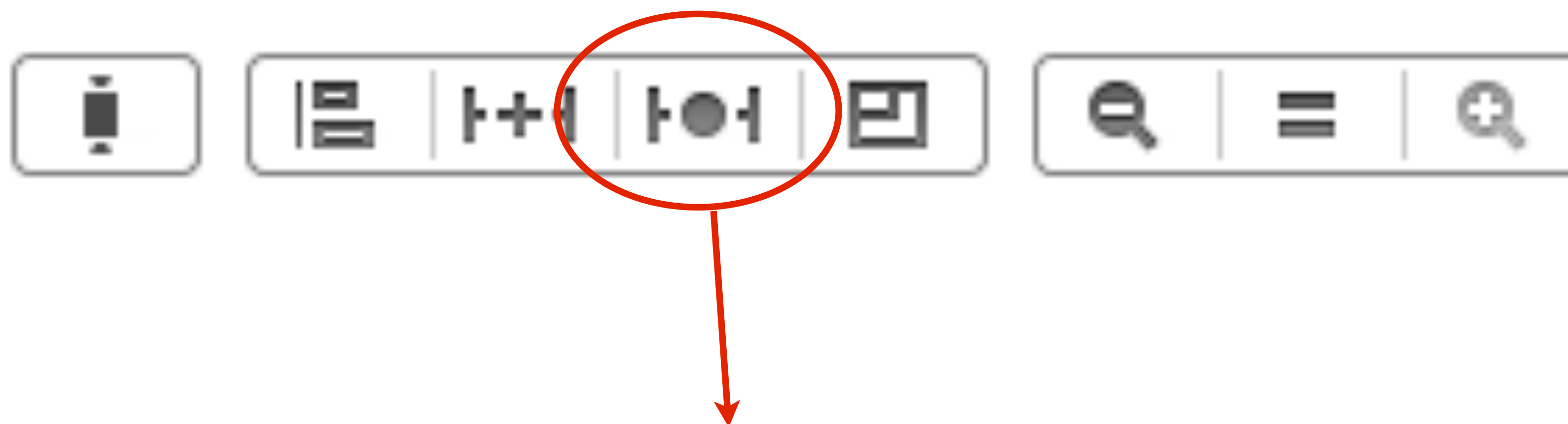
- Auto Layout is managed visually in Xcode
- In storyboard, in the lower-right corner, are the controls for using Auto Layout



**Set pin constraints:** create spacing constraints, such as defining the height of a view, or specifying its horizontal distance from another view

# Using Auto Layout

- Auto Layout is managed visually in Xcode
- In storyboard, in the lower-right corner, are the controls for using Auto Layout



**Resolve Auto Layout issues:** resolve layout issues by adding or resetting constraints based on suggestions

# Using Auto Layout

- Auto Layout is managed visually in Xcode
- In storyboard, in the lower-right corner, are the controls for using Auto Layout



**Resizing:** specify how resizing affects constraints

# Adding constraints

- Constraints can be added in 3 ways:
  - Control-Drag
  - Align and Pin Menus
  - Adding Missing or Suggested Constraints

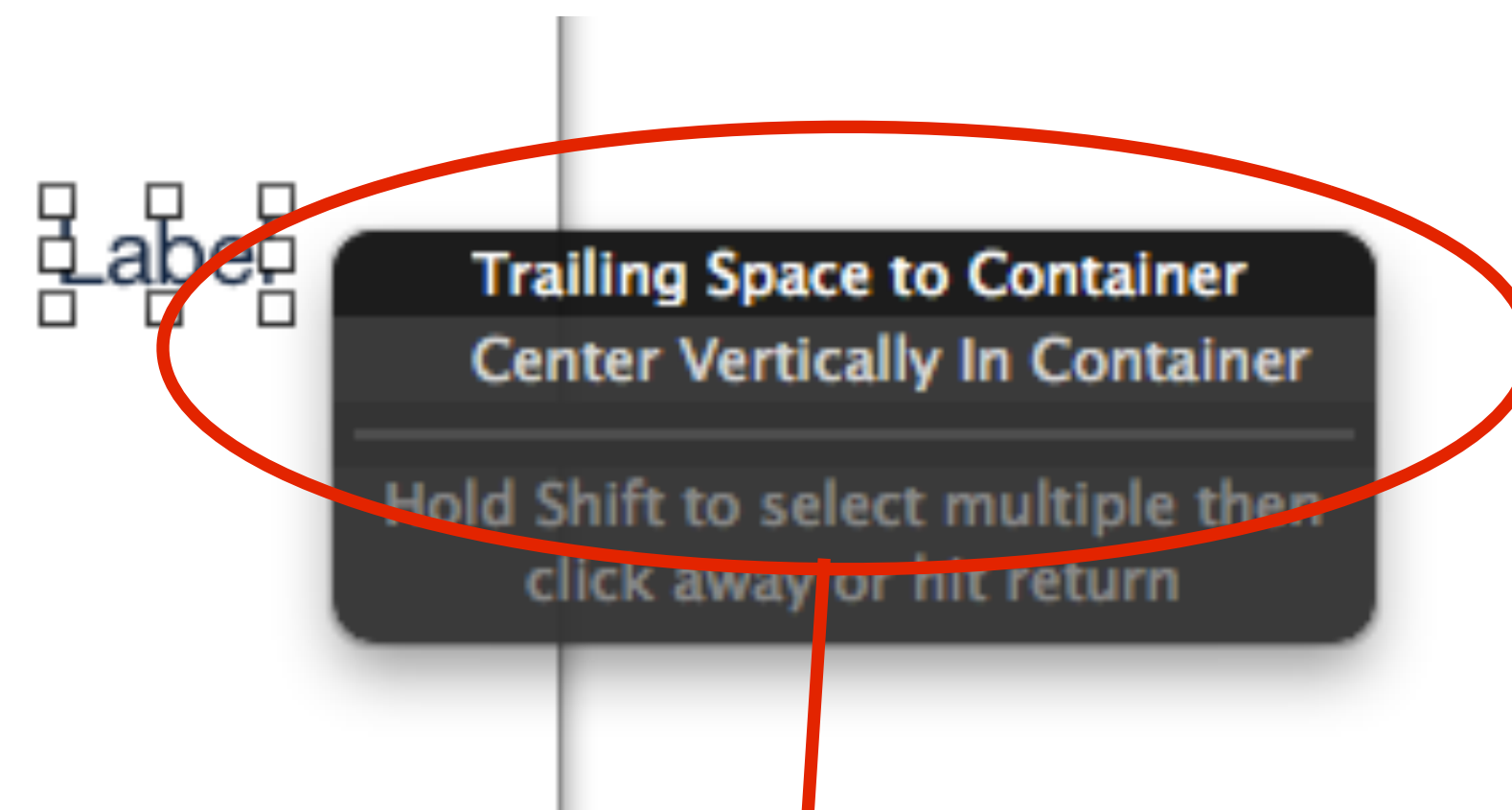
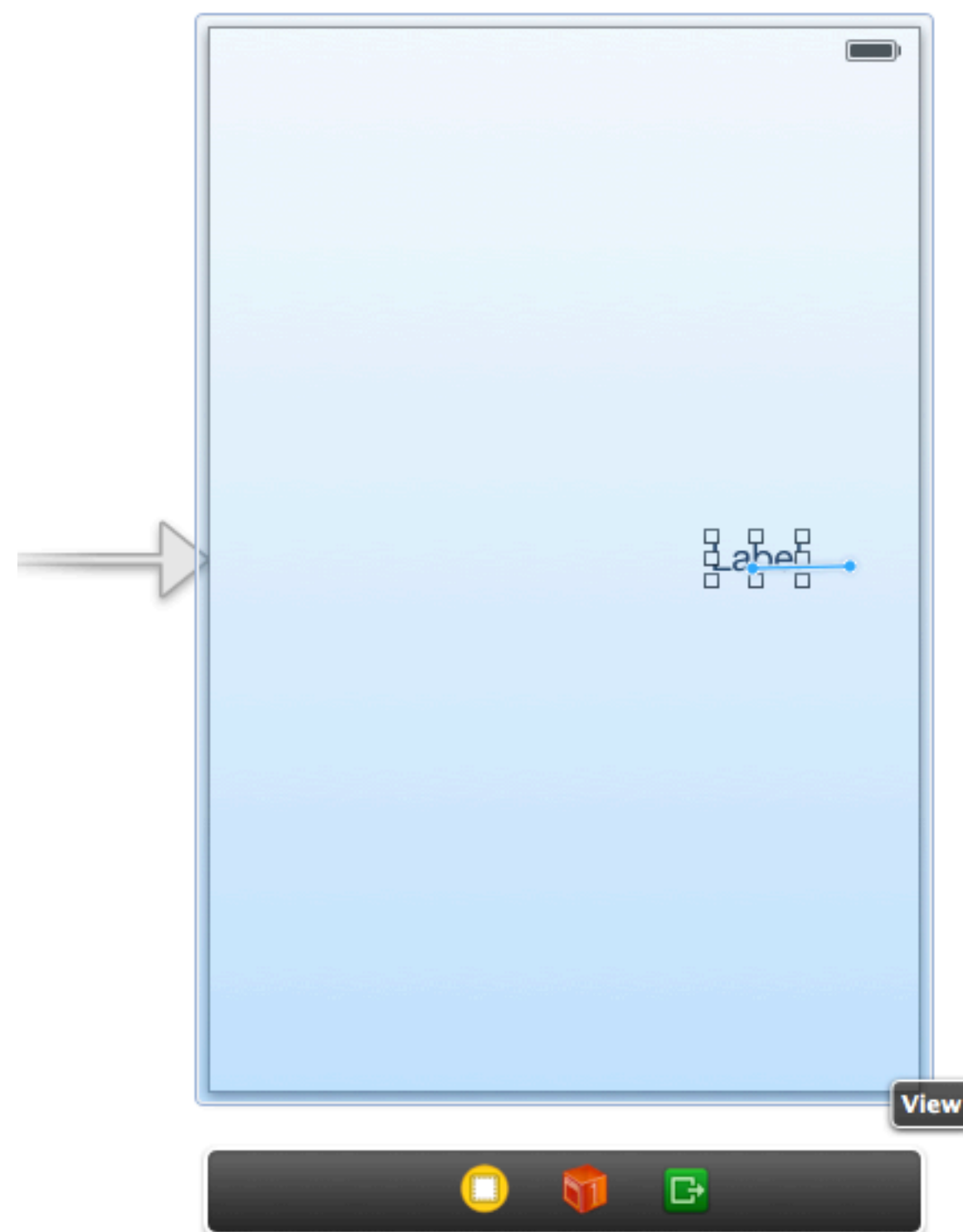
# Adding Constraints with Control-Drag

- Fastest way to add a single, specific constraint
- Hold down the Control key and dragging from a view on the canvas (or another view, for relative positioning)



# Adding Constraints with Control-Drag

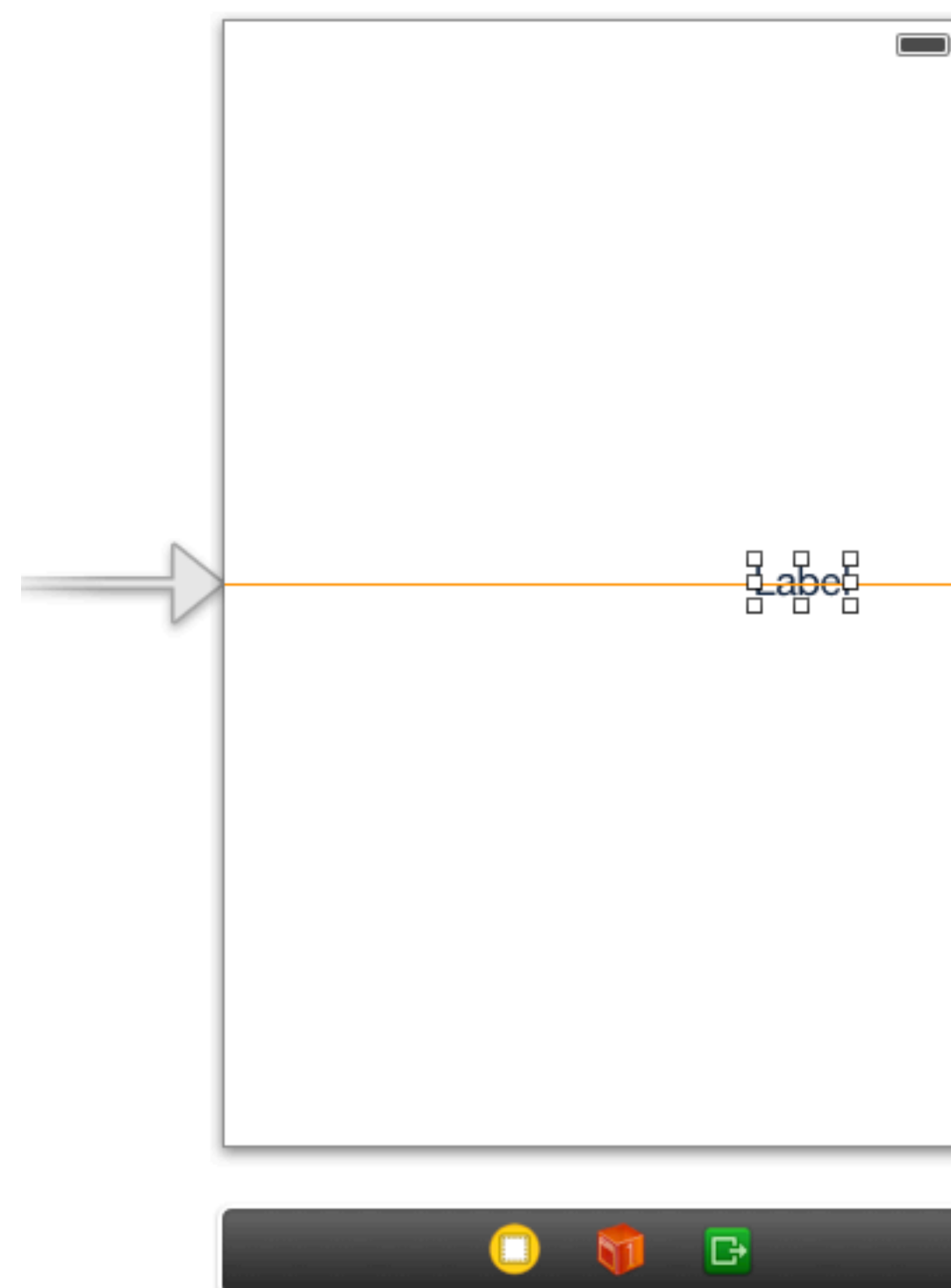
- Fastest way to add a single, specific constraint
- Hold down the Control key and dragging from a view on the canvas (or another view, for relative positioning)



Auto Layout limits the possibilities of constraints appropriately, depending on the type of dragging performed

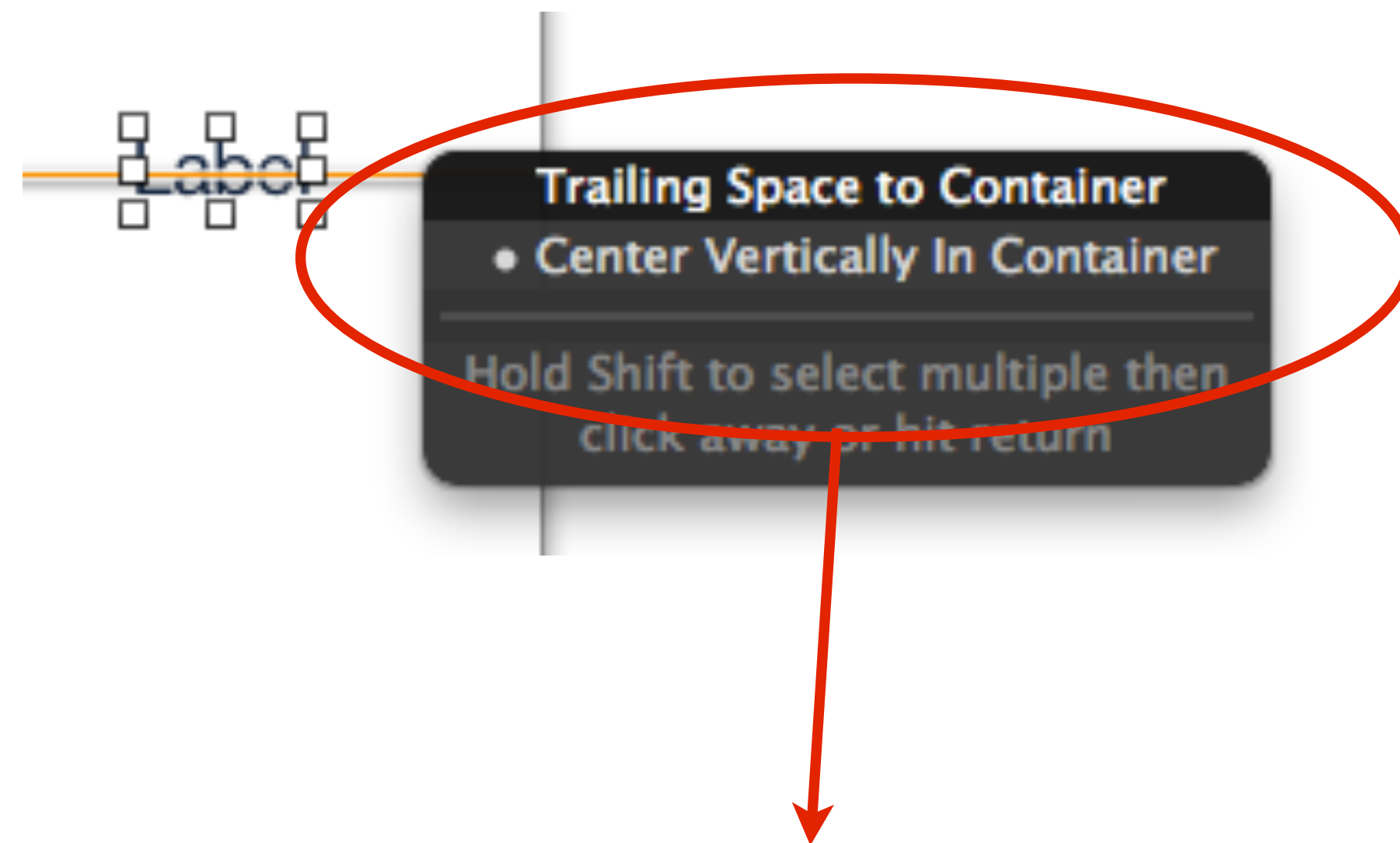
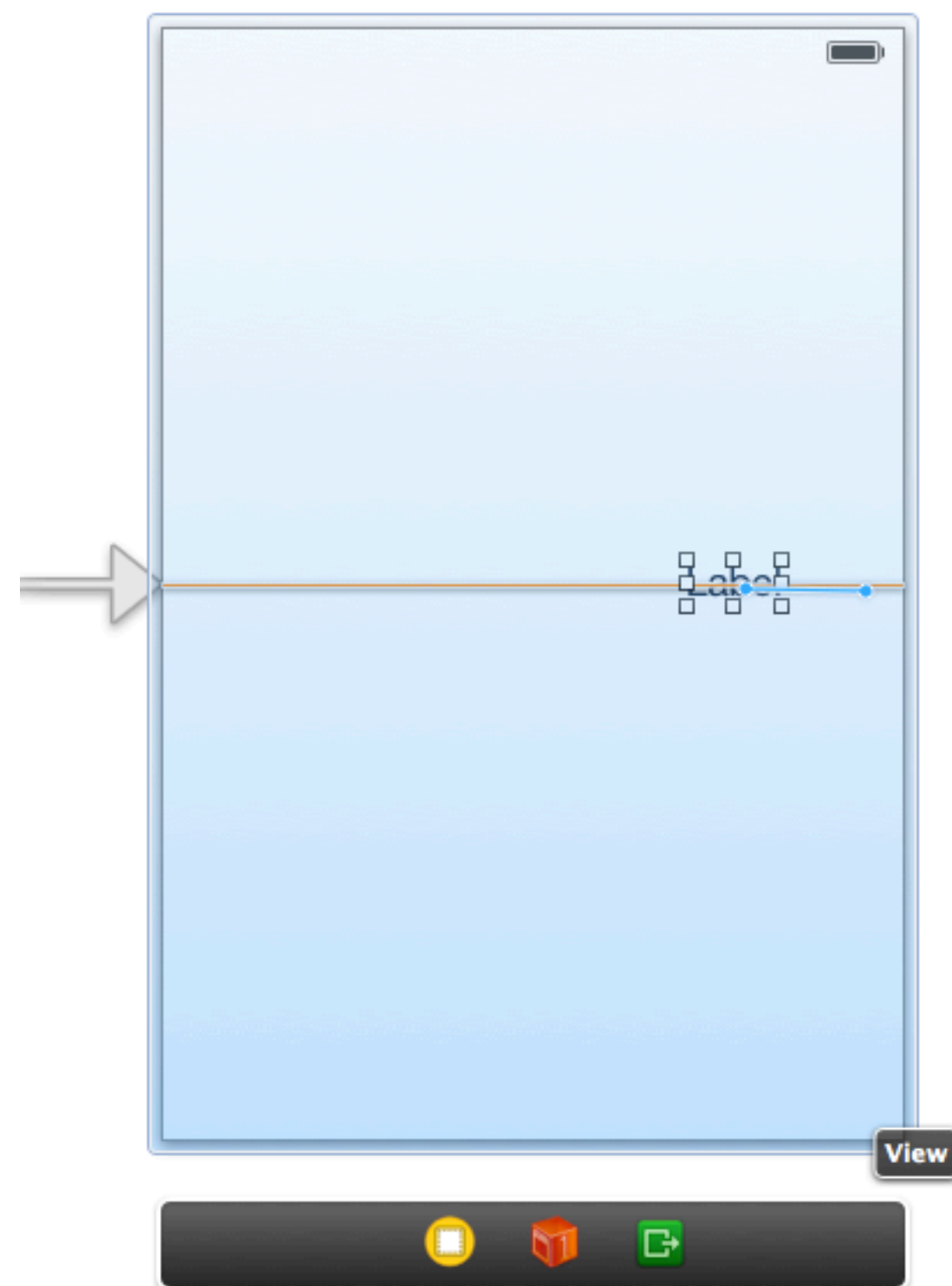
# Adding Constraints with Control-Drag

- Select **Center Vertically in Container**
- The constrained is added
- The yellow line is an warning: some constraint is missing (what about the horizontal position?)



# Adding Constraints with Control-Drag

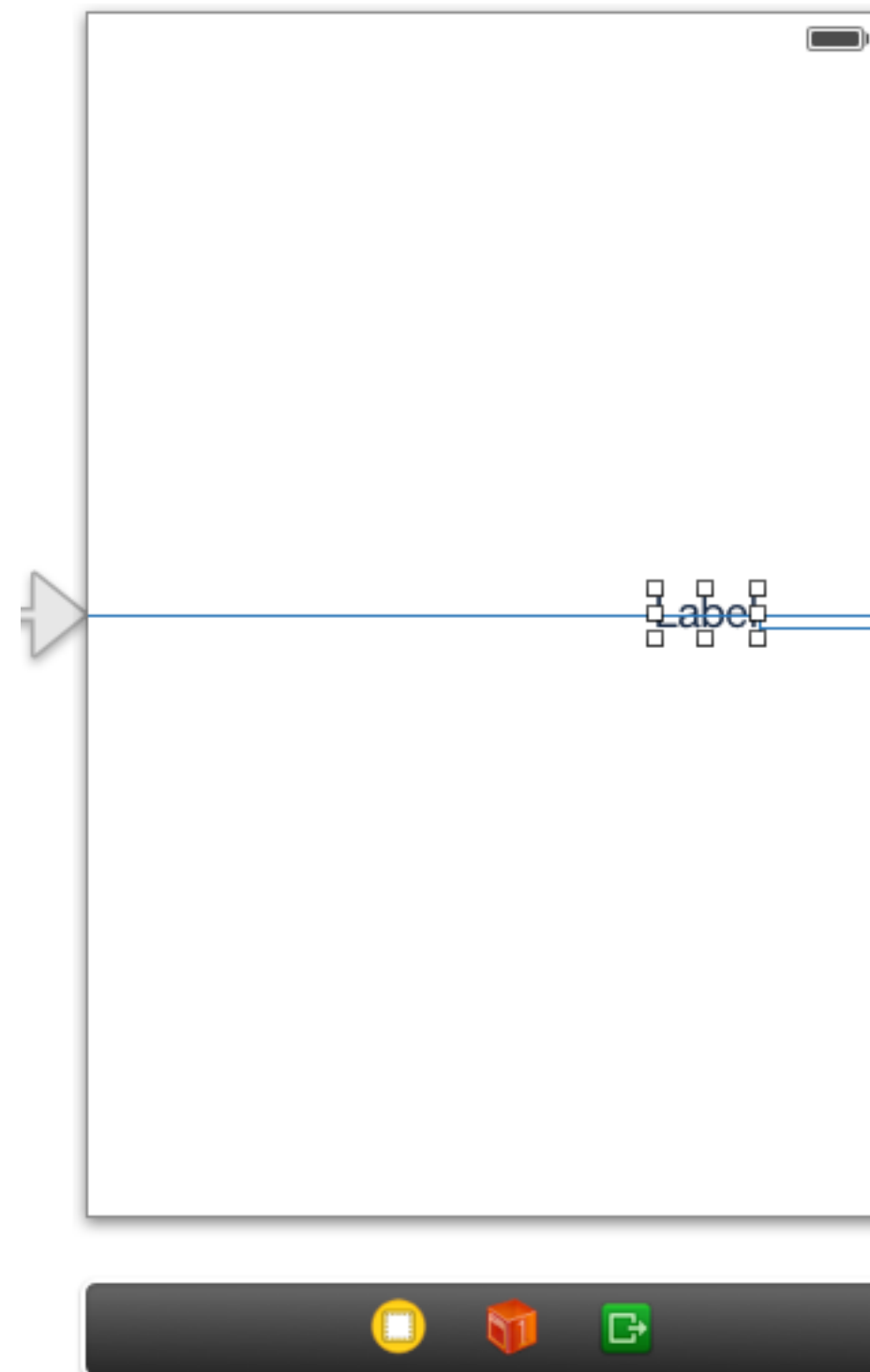
- Repeat the Ctrl-drag procedure
- Select **Trailing Space to Container**



Note that one option is already selected

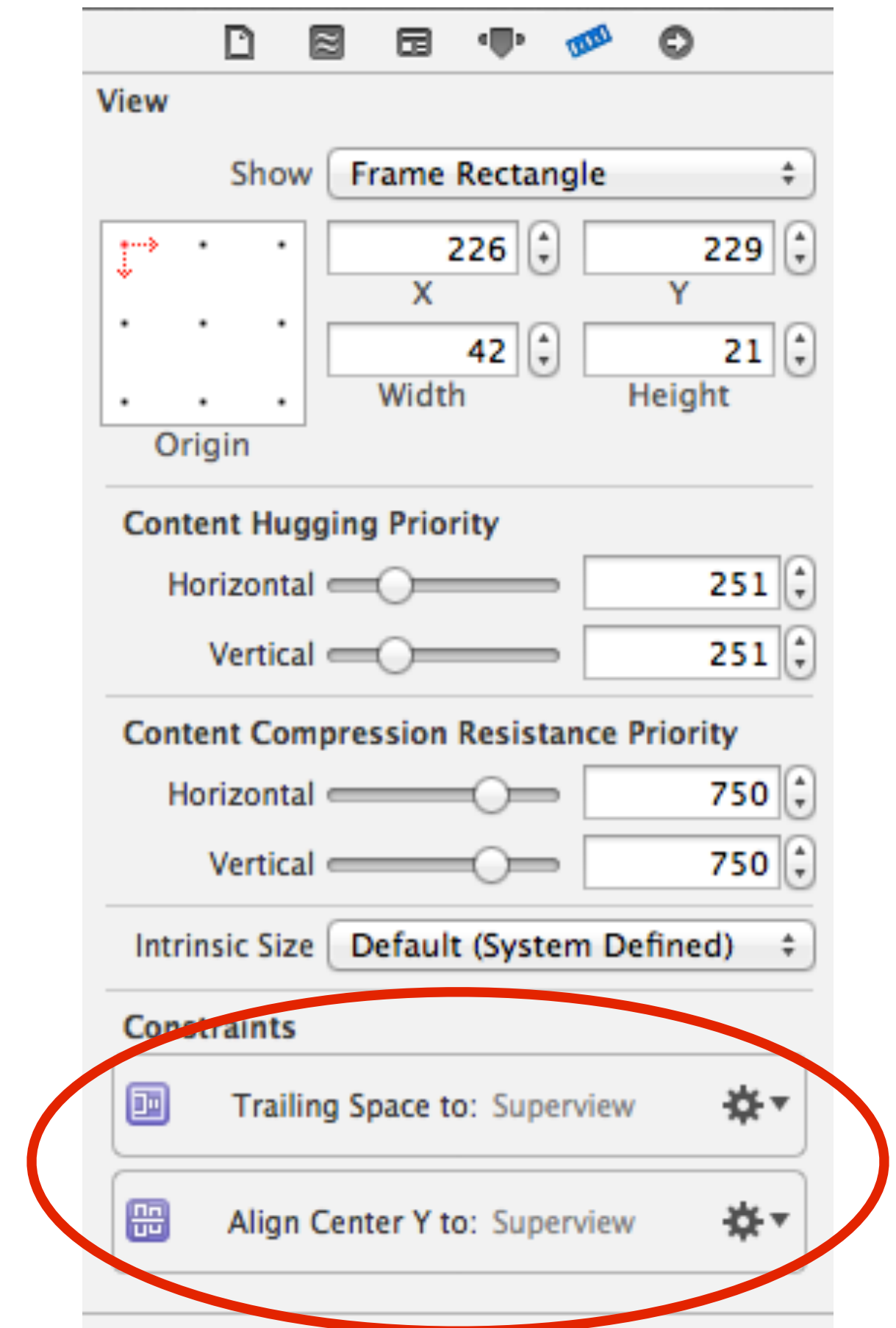
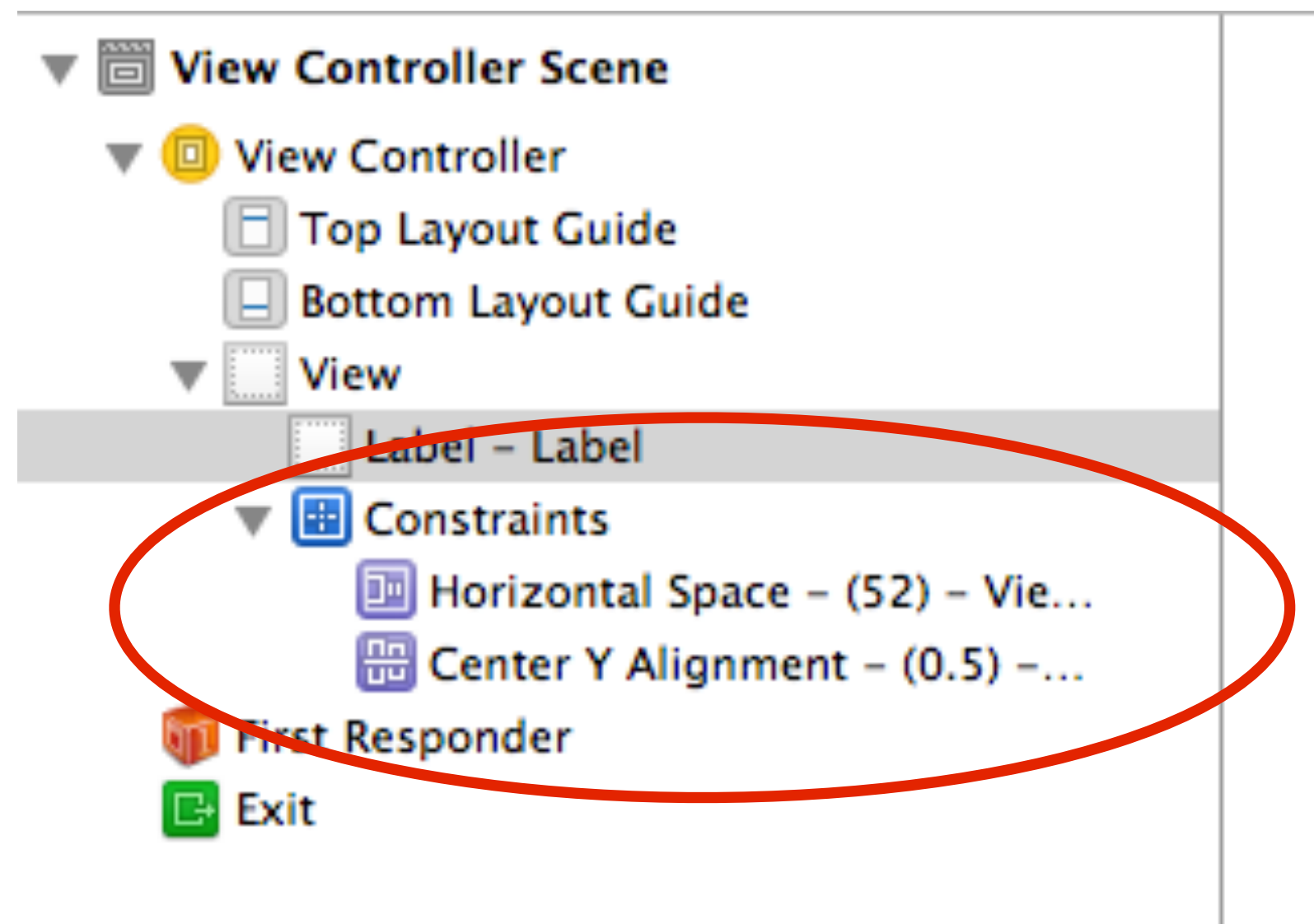
# Adding Constraints with Control-Drag

- The second constrained is added
- The yellow line has disappeared and has become blue: all good!
- A new line has appeared for the second constraint



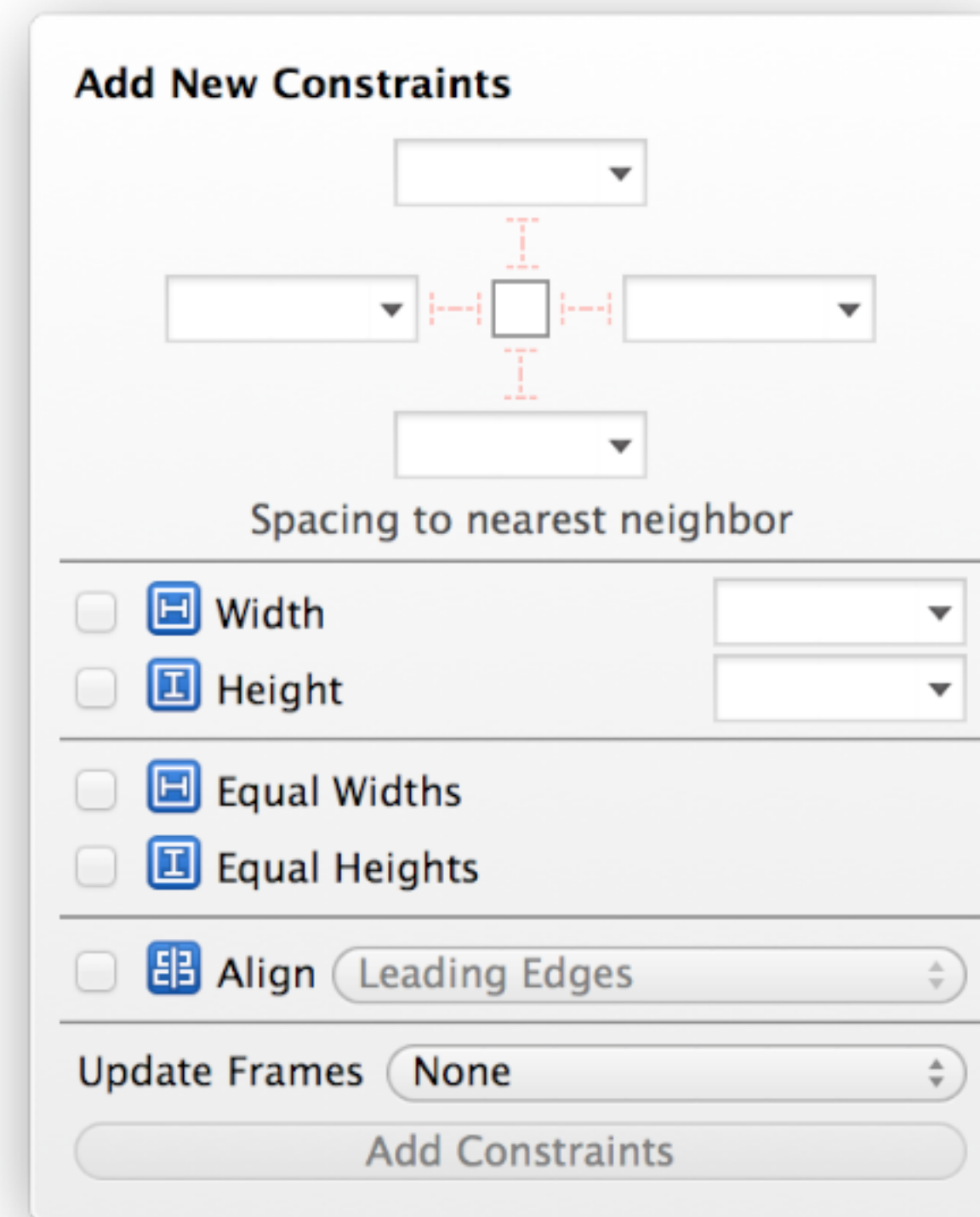
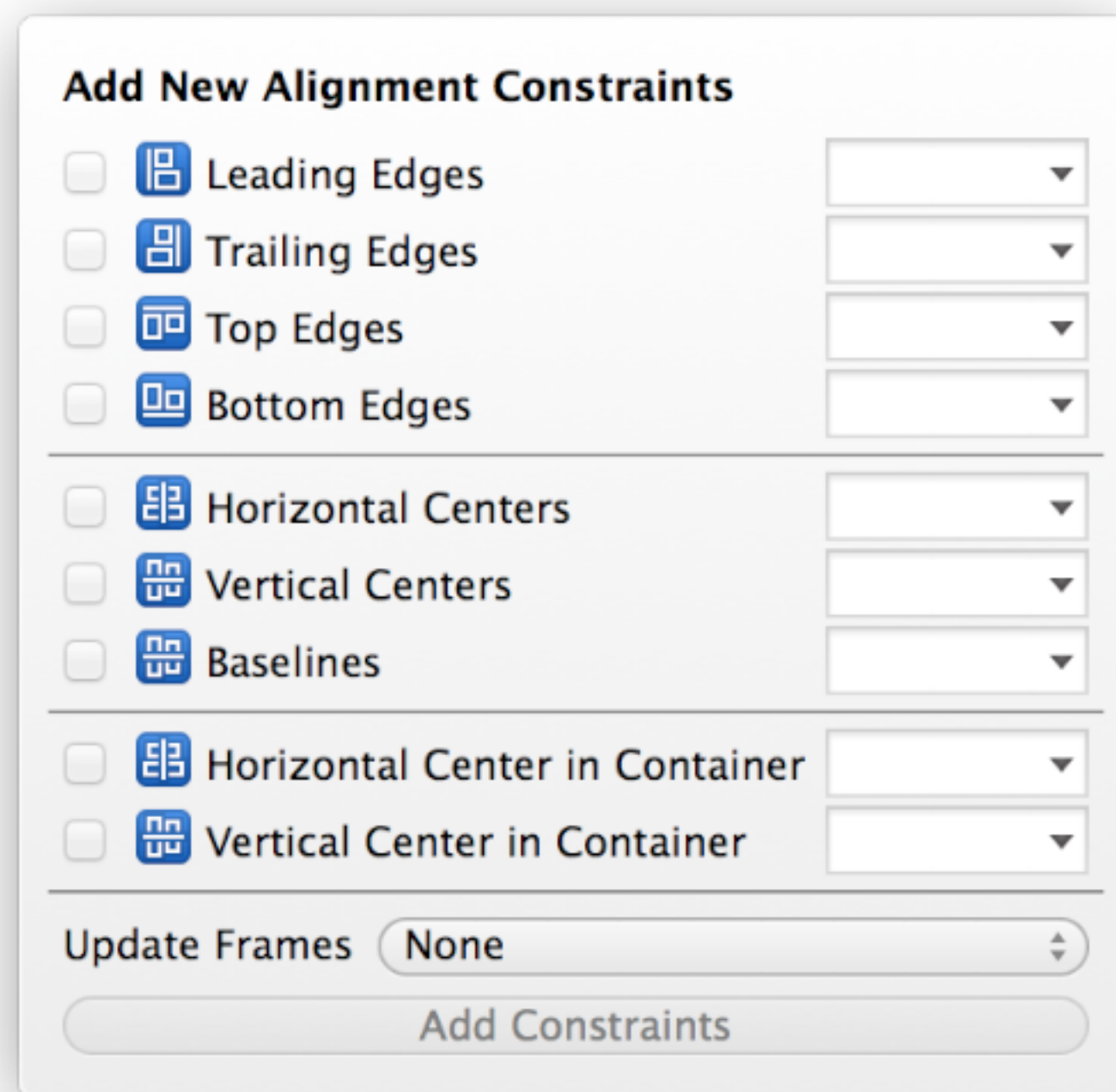
# Adding Constraints with Control-Drag

- Constraints are also visible:
  - in the Document Layout view in storyboard
  - in the Attributes inspector



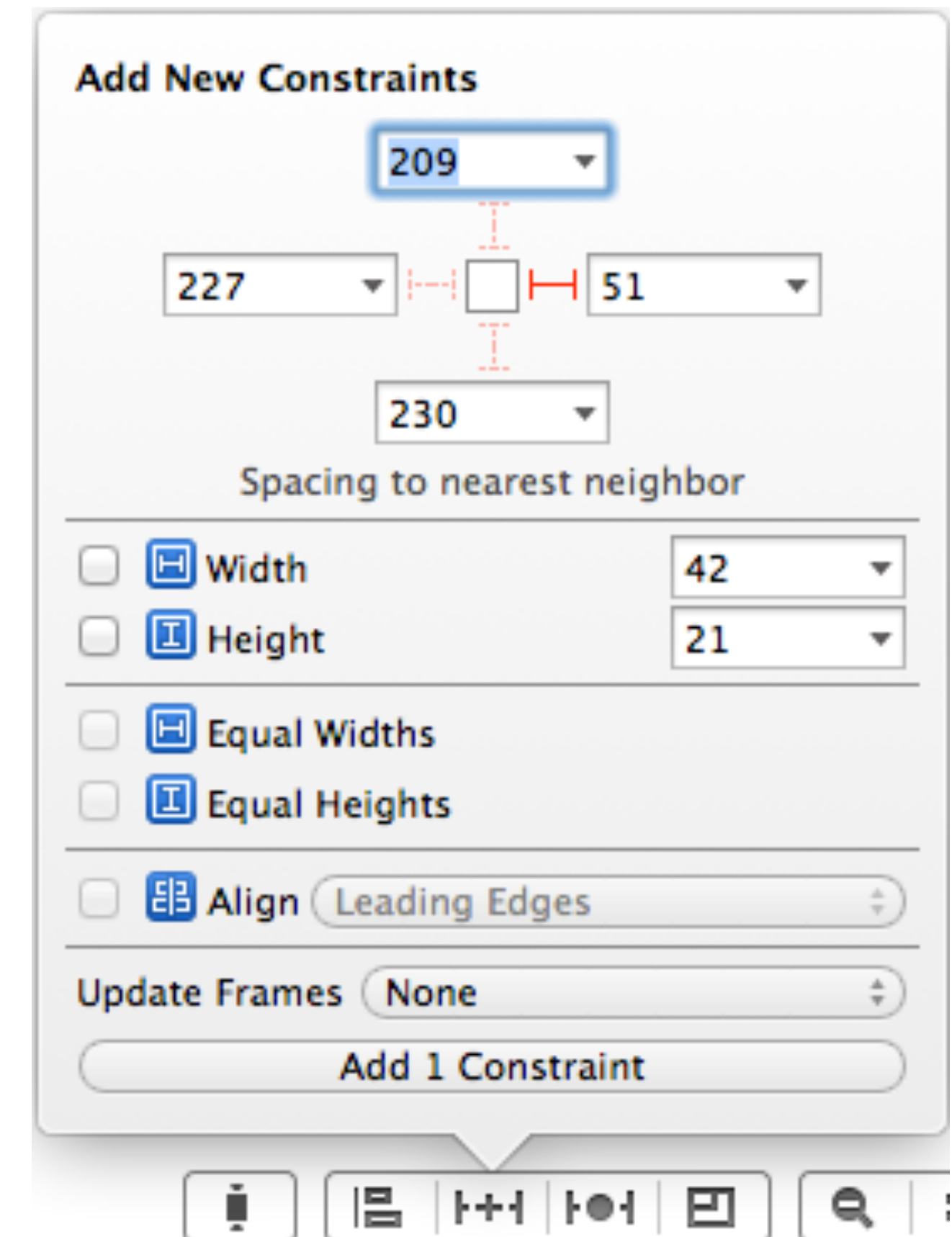
# Adding constraints with align and pin

- Let's go back when only the first constraint was set
- In the Document Layout a red arrow appears (it means there are problems)



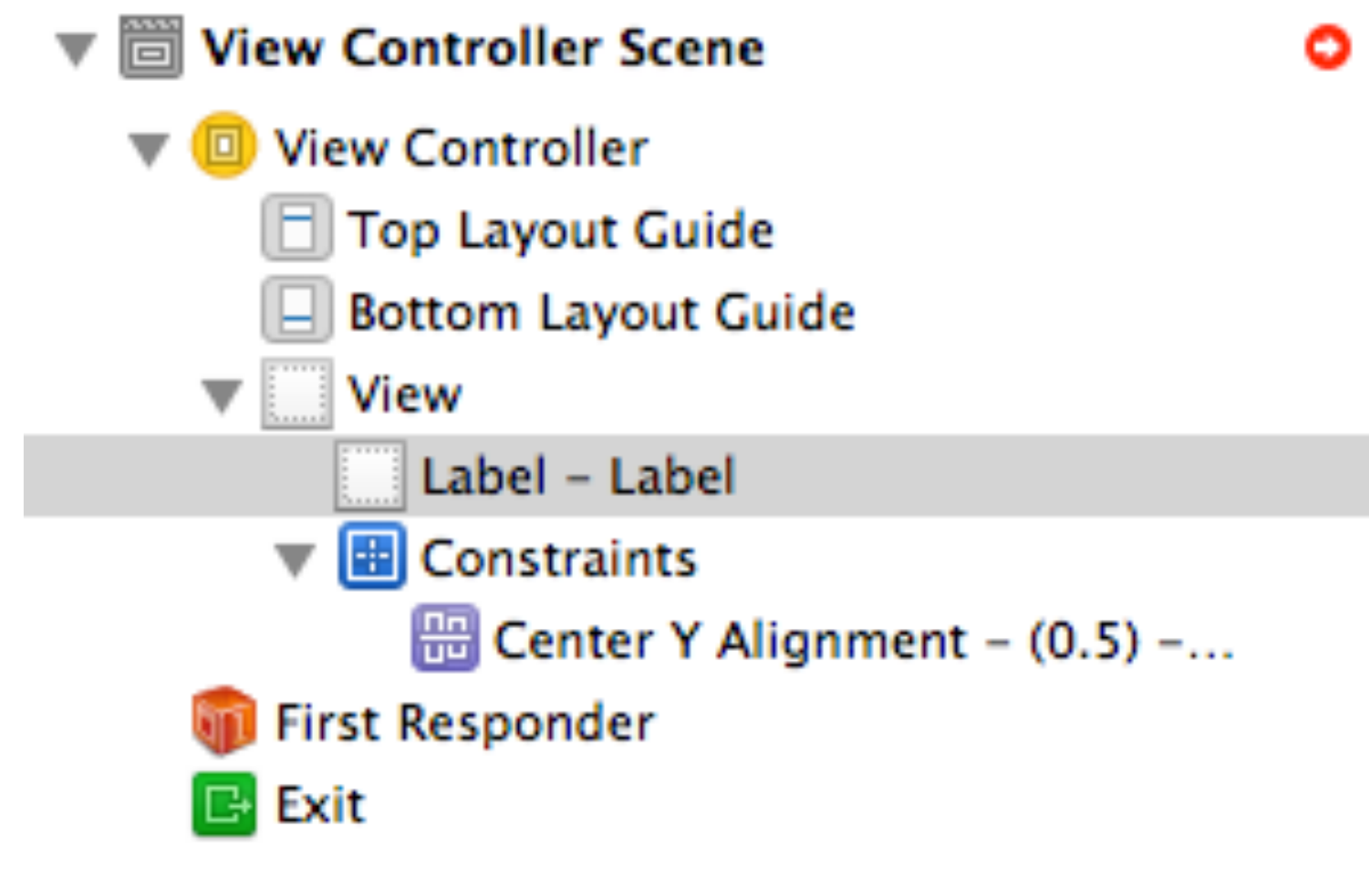
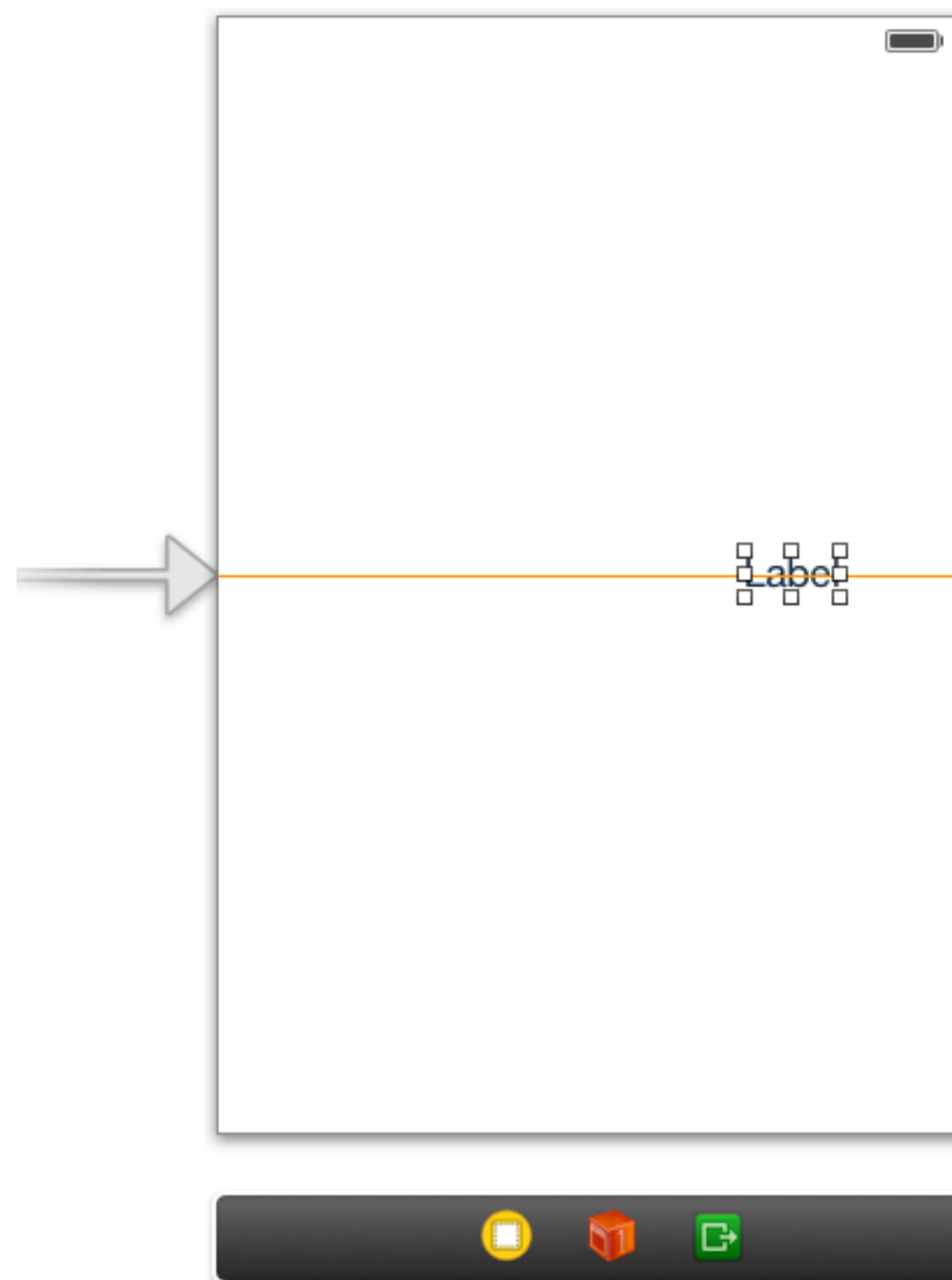
# Adding constraints with align and pin

1. Select the checkbox next to the appropriate constraint:
  - to select a “Spacing to nearest neighbor” constraint, select the red constraint corresponding to the appropriate side of the element
2. Enter the constant value
3. Create the constraints by clicking the Add Constraints button adds the new constraints to the selected elements



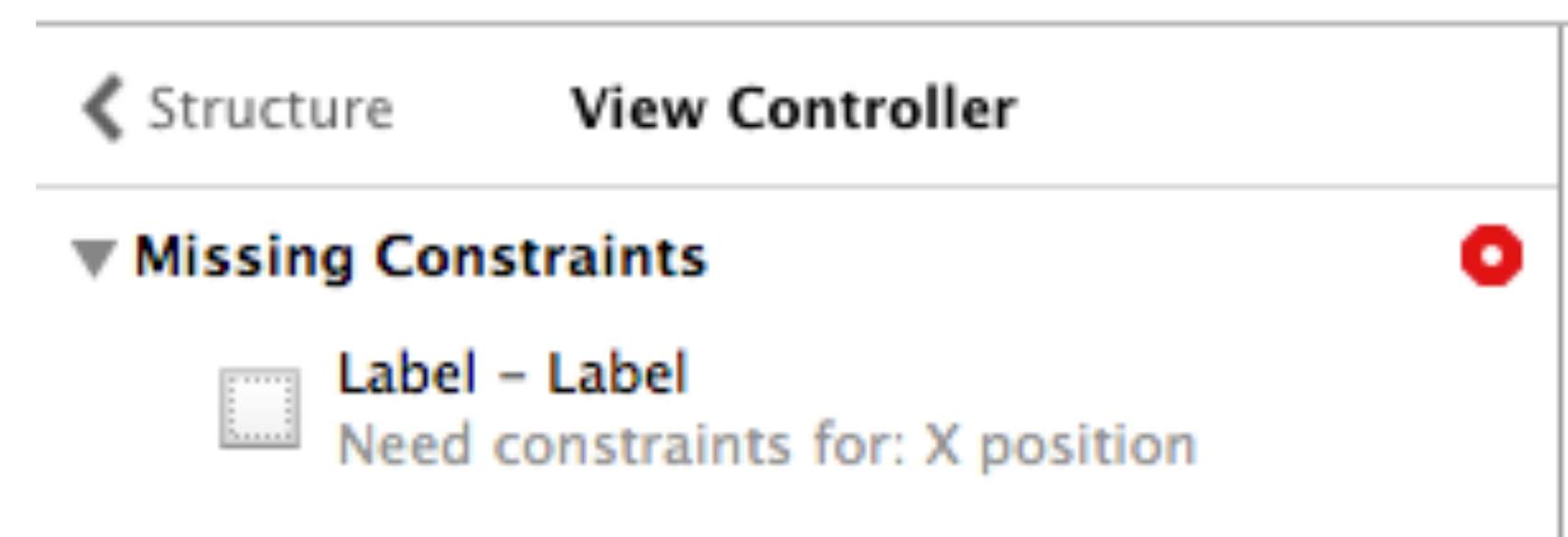
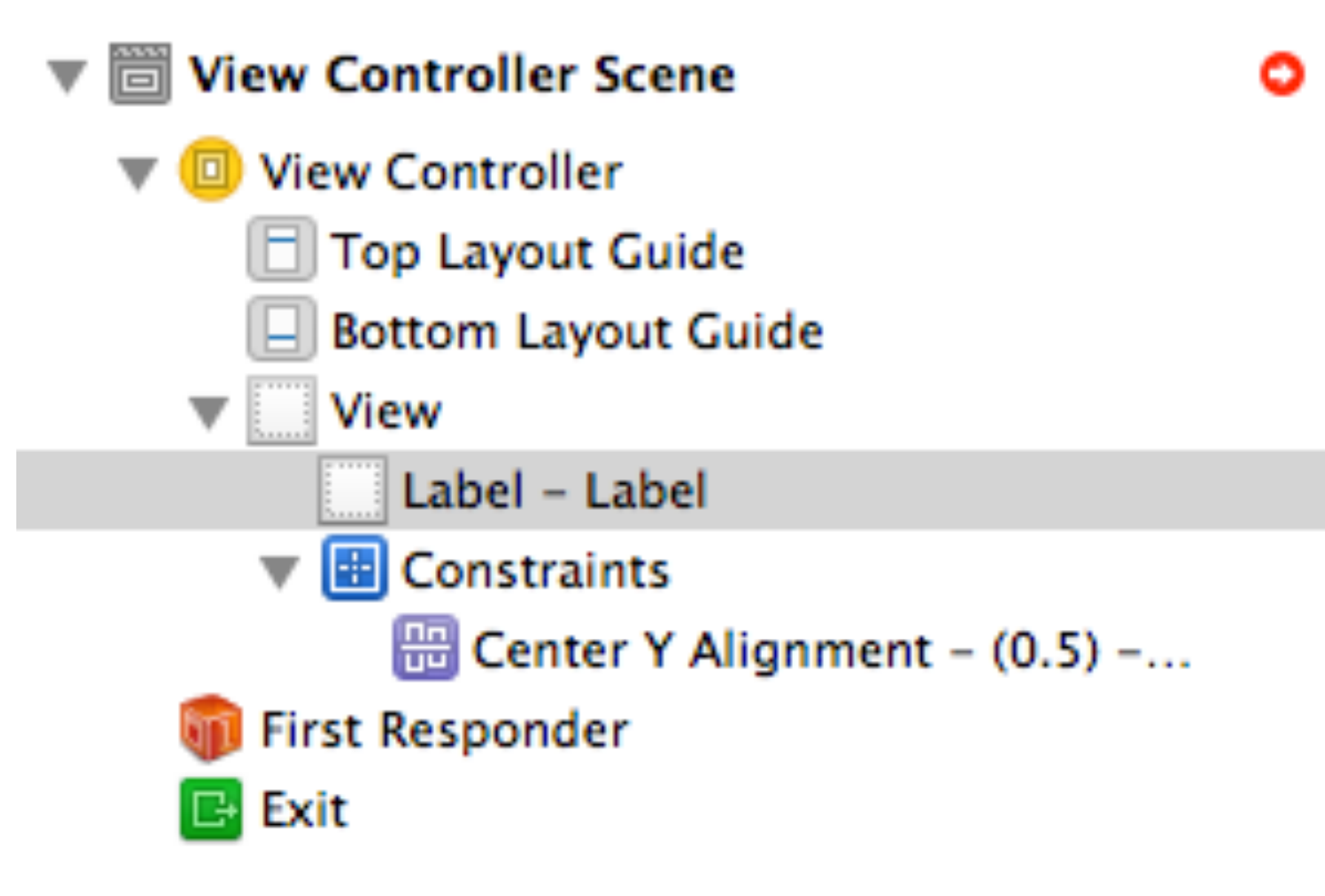
# Adding missing constraints

- Let's go back when only the first constraint was set
- In the Document Layout a red arrow appears (it means there are problems)



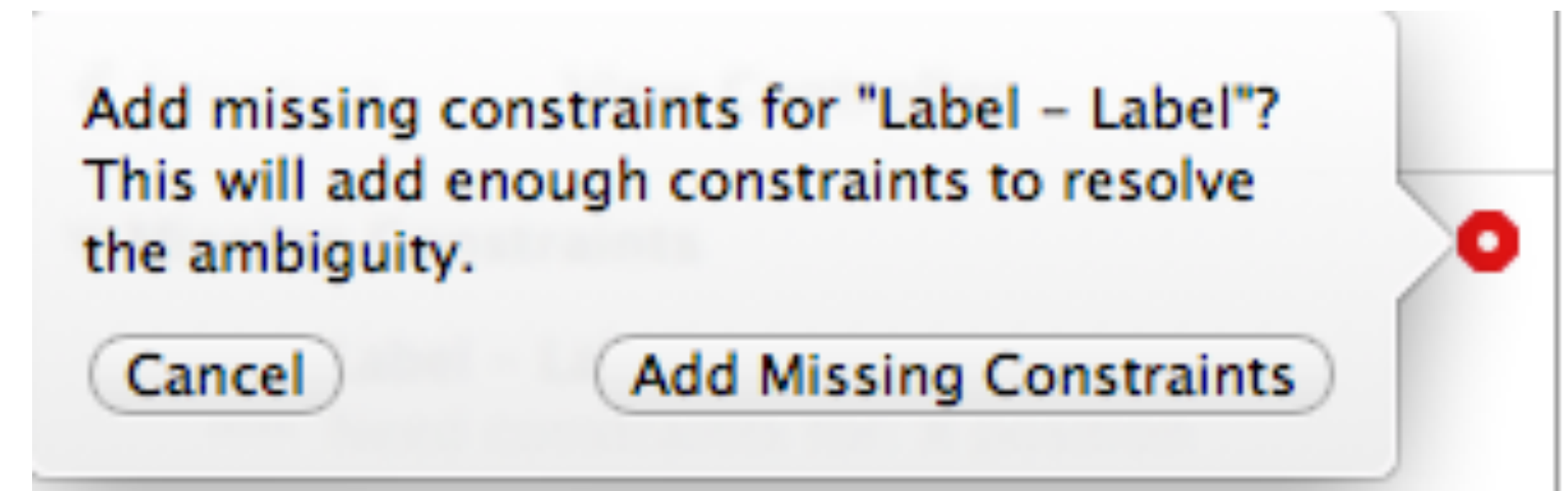
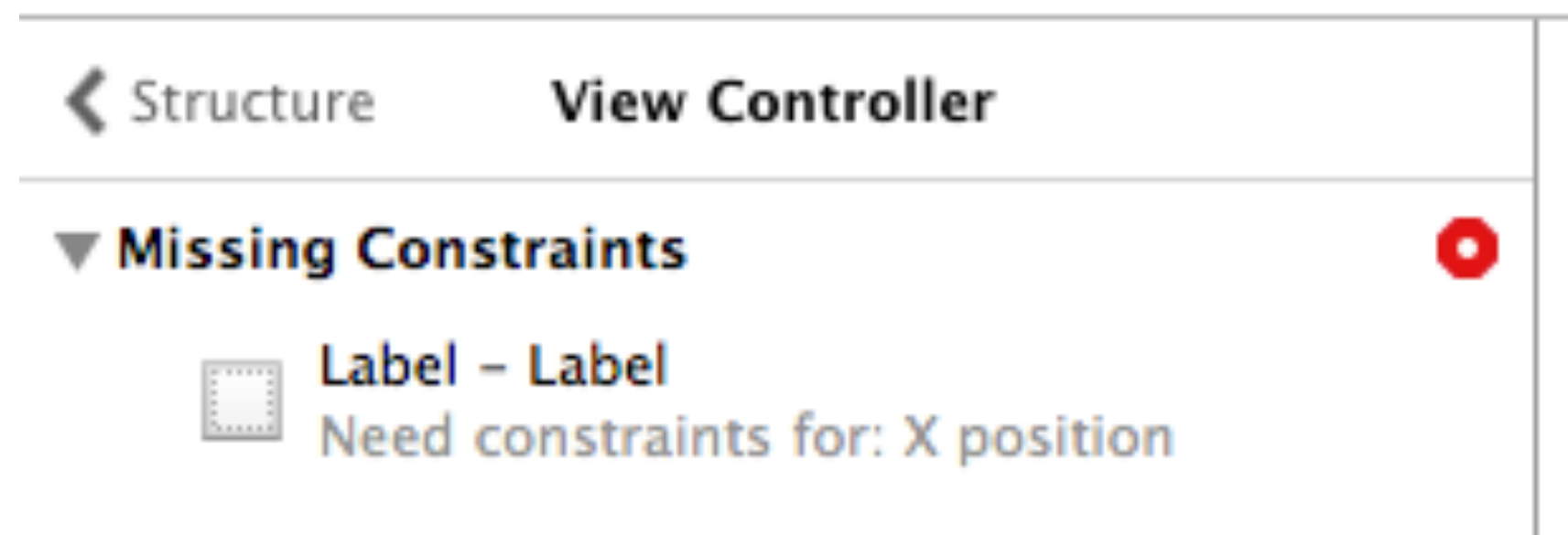
# Adding missing constraints

- By clicking on the arrow a new view appear, showing which elements have layout issues (only one in this case)

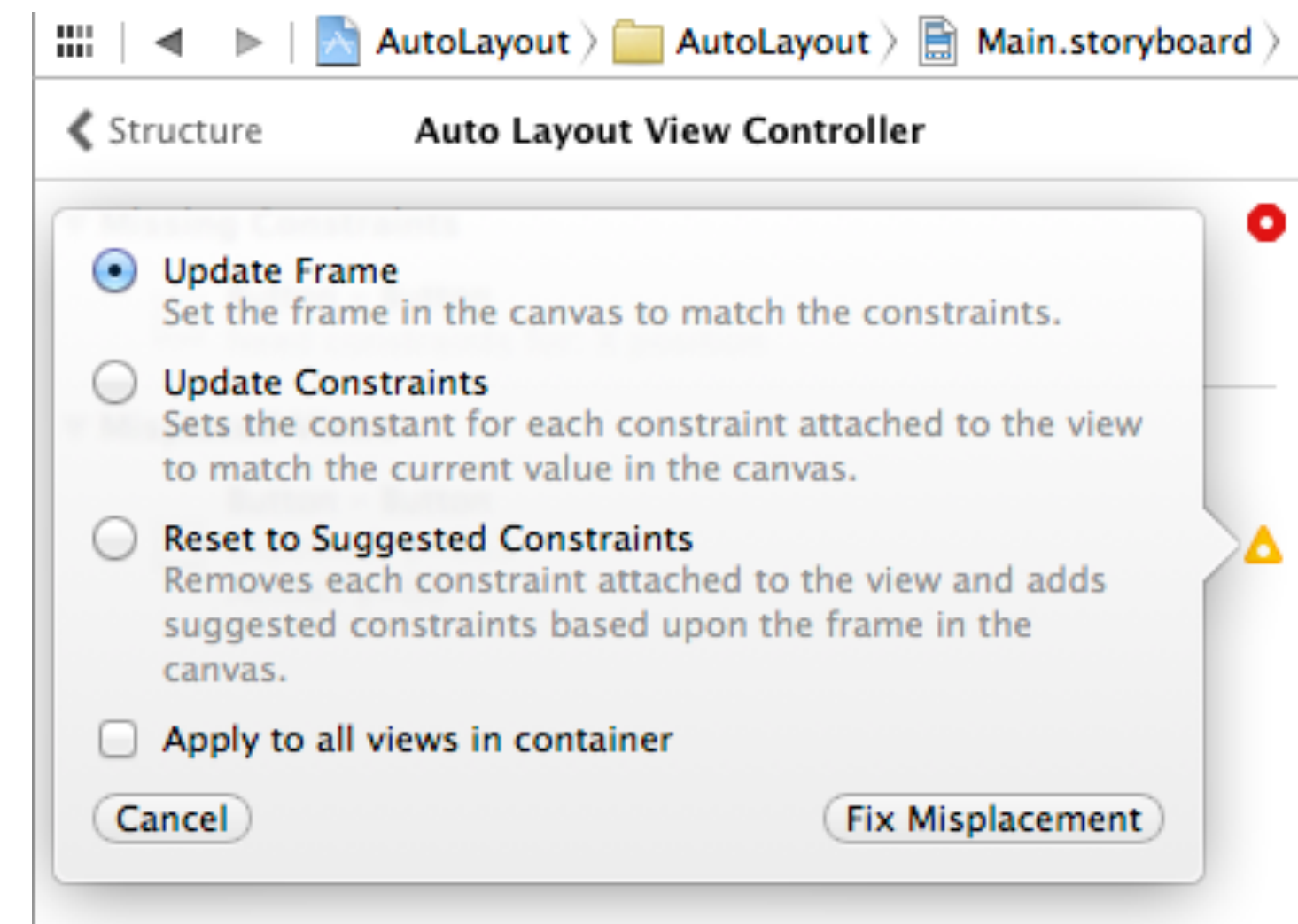


# Adding missing constraints

- By clicking on the red circle, a popup appears
- Auto Layout is able to add the missing constraints automatically

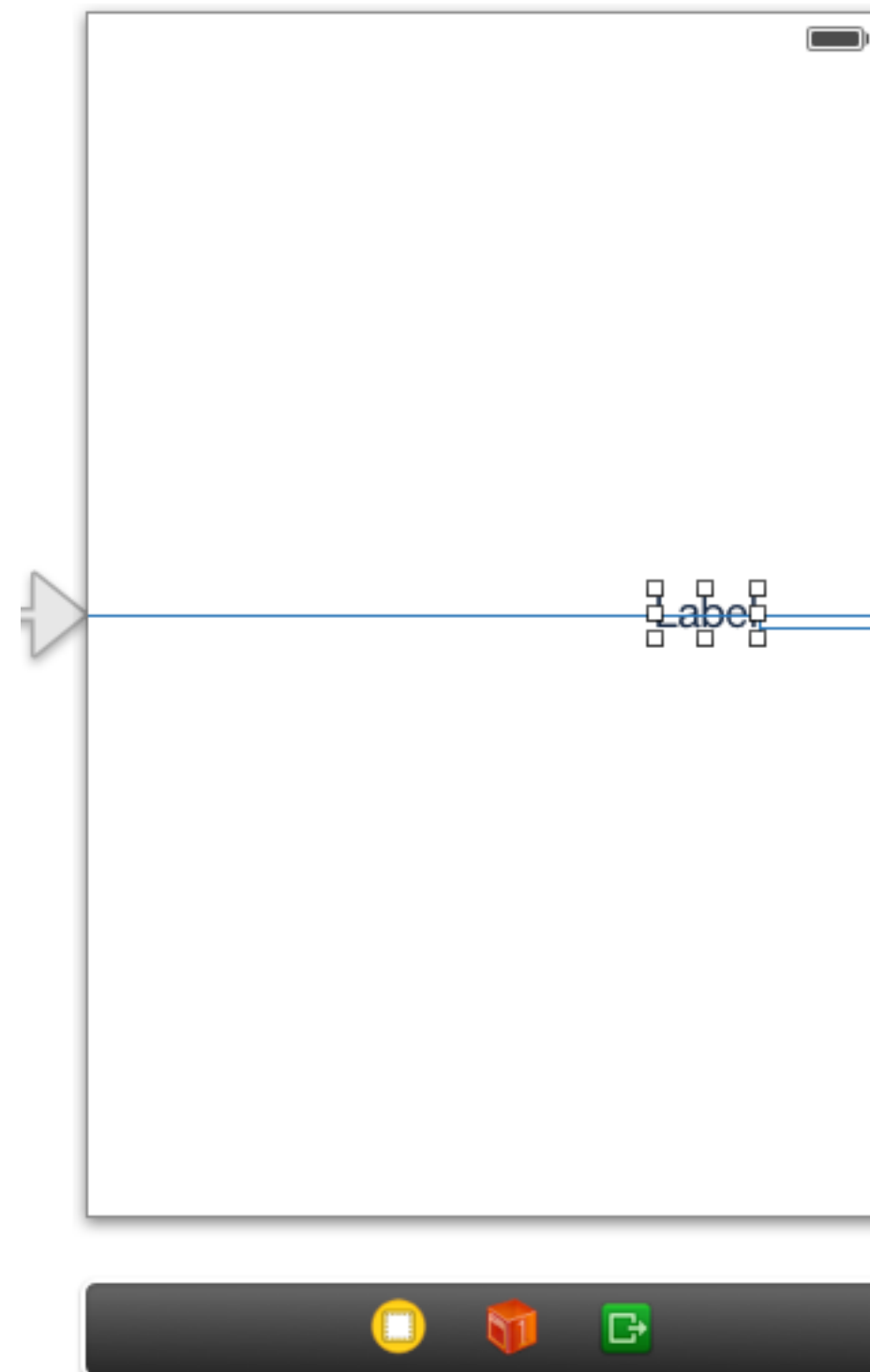


- If many solutions are available (not in this case), another popup shows all the possible actions that may be taken



# Adding missing constraints

- By selecting Add Missing Constraints, Auto Layout creates the new constraints
- Since this was the only issue, all lines in the view become blue



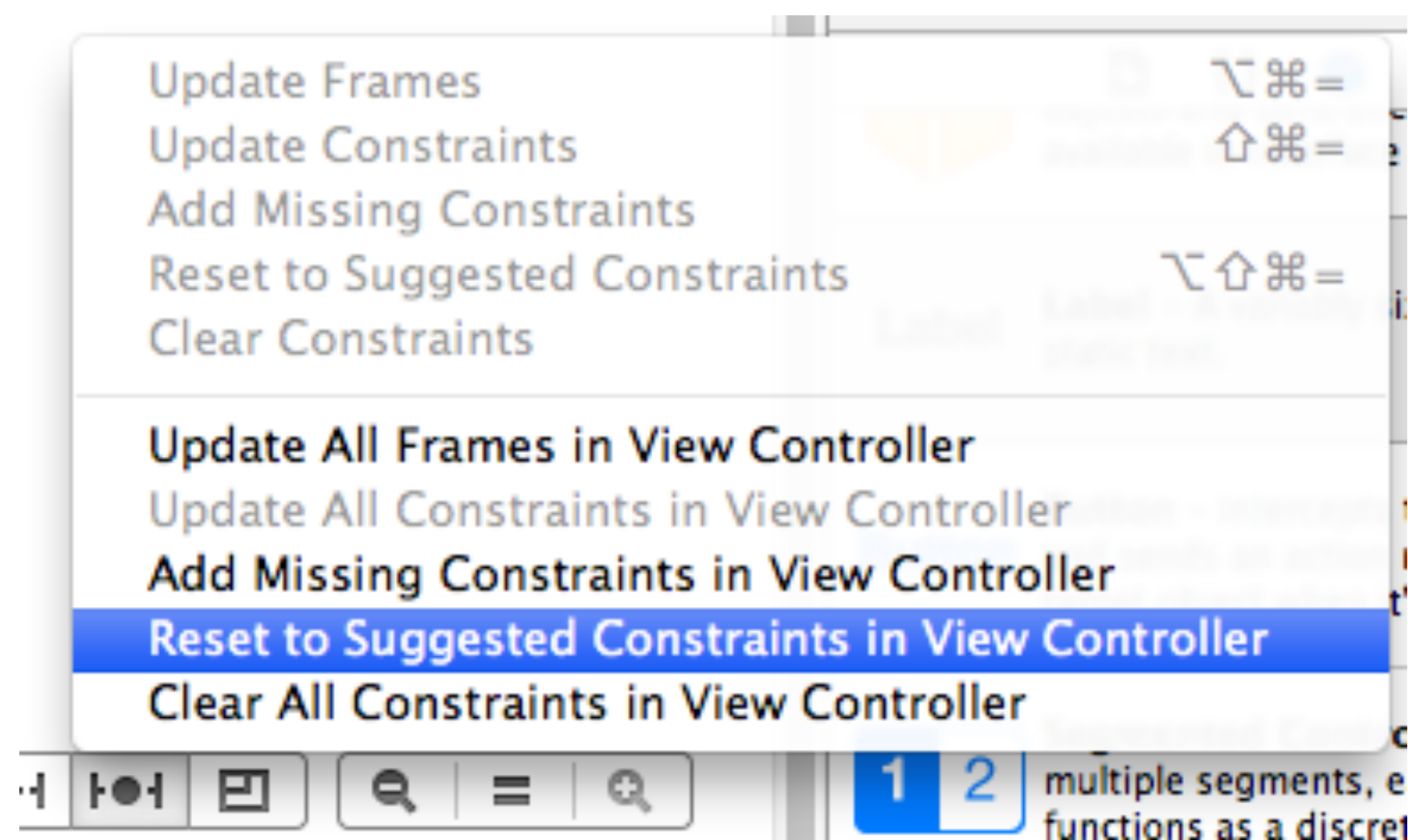
# Adding missing constraints

- If no constraints have been, it is possible to ask Auto Layout to set all the constraints for the view:
  1. Select the View Controller, by clicking on the View Controller icon



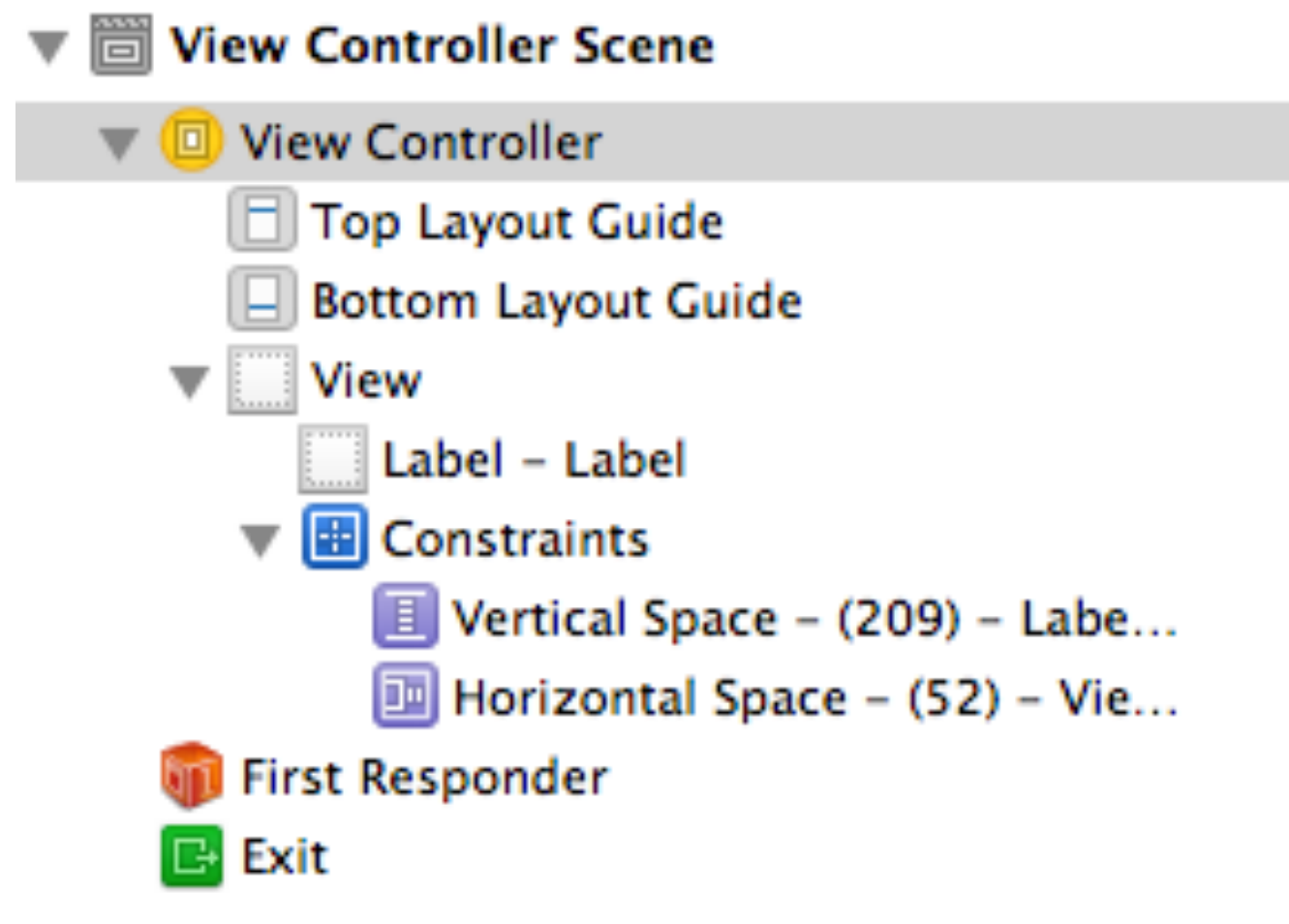
# Adding missing constraints

- If no constraints have been, it is possible to ask Auto Layout to set all the constraints for the view:
  1. Select the View Controller, by clicking on the View Controller icon
  2. In the Resolve Issues menu, select **Reset to Suggested Constraints in View Controller**

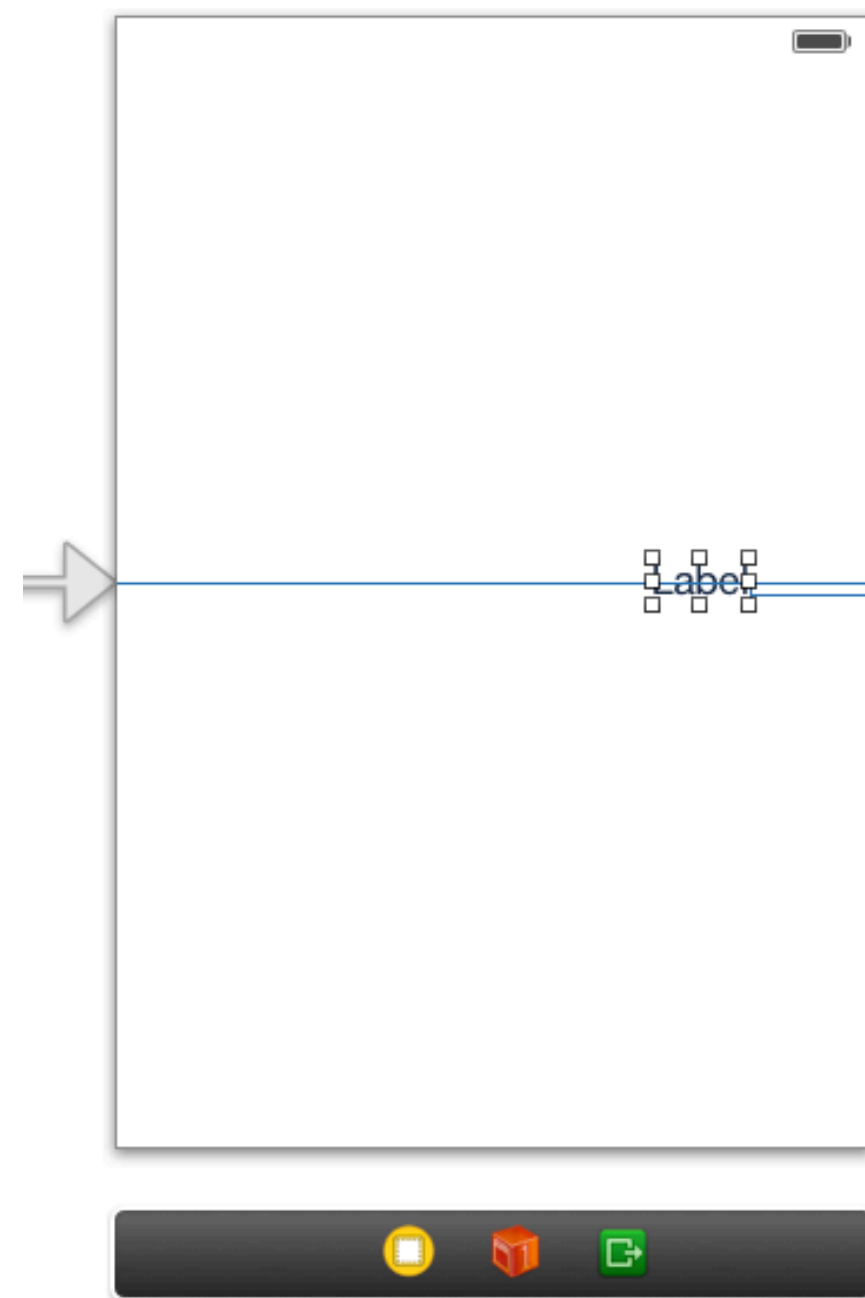


# Adding missing constraints

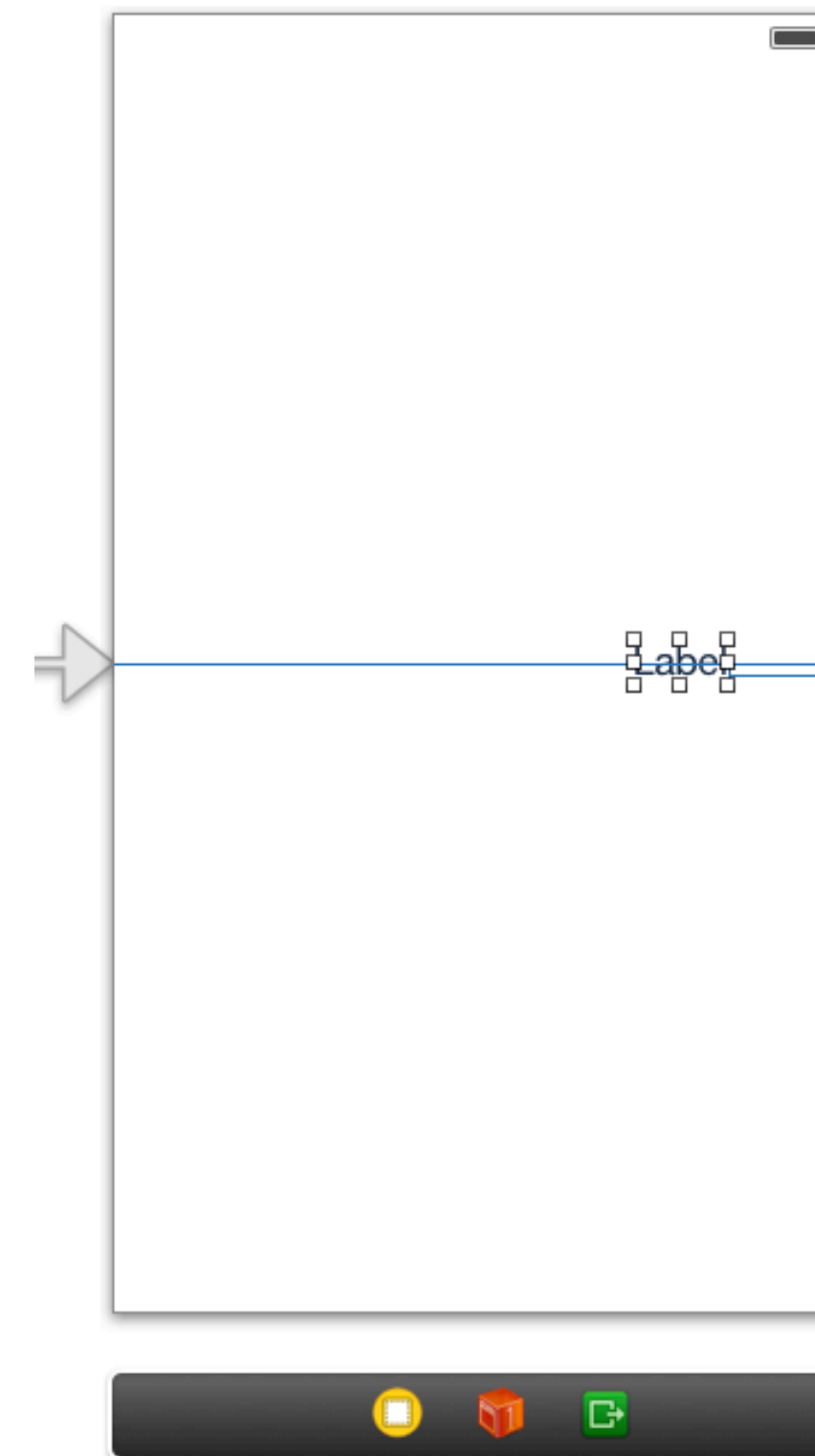
- If no constraints have been, it is possible to ask Auto Layout to set all the constraints for the view:
  1. Select the View Controller, by clicking on the View Controller icon
  2. In the Resolve Issues menu, select **Reset to Suggested Constraints in View Controller**
  3. The constraints are added (note that center vertically is not there!)



# Checking constraints for different screen sizes

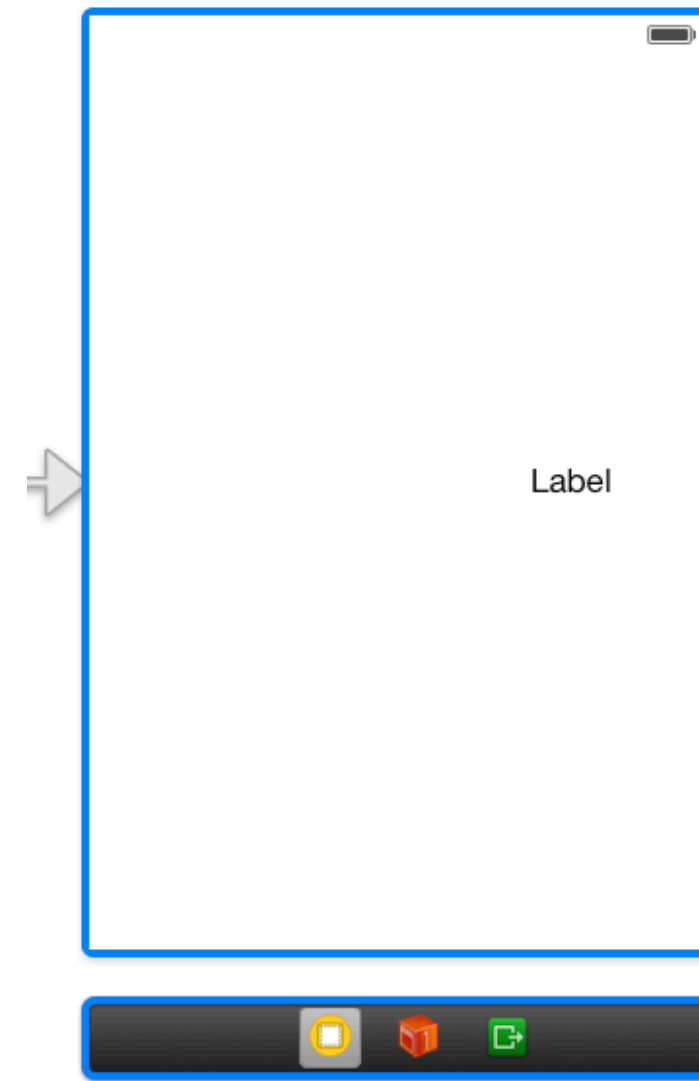
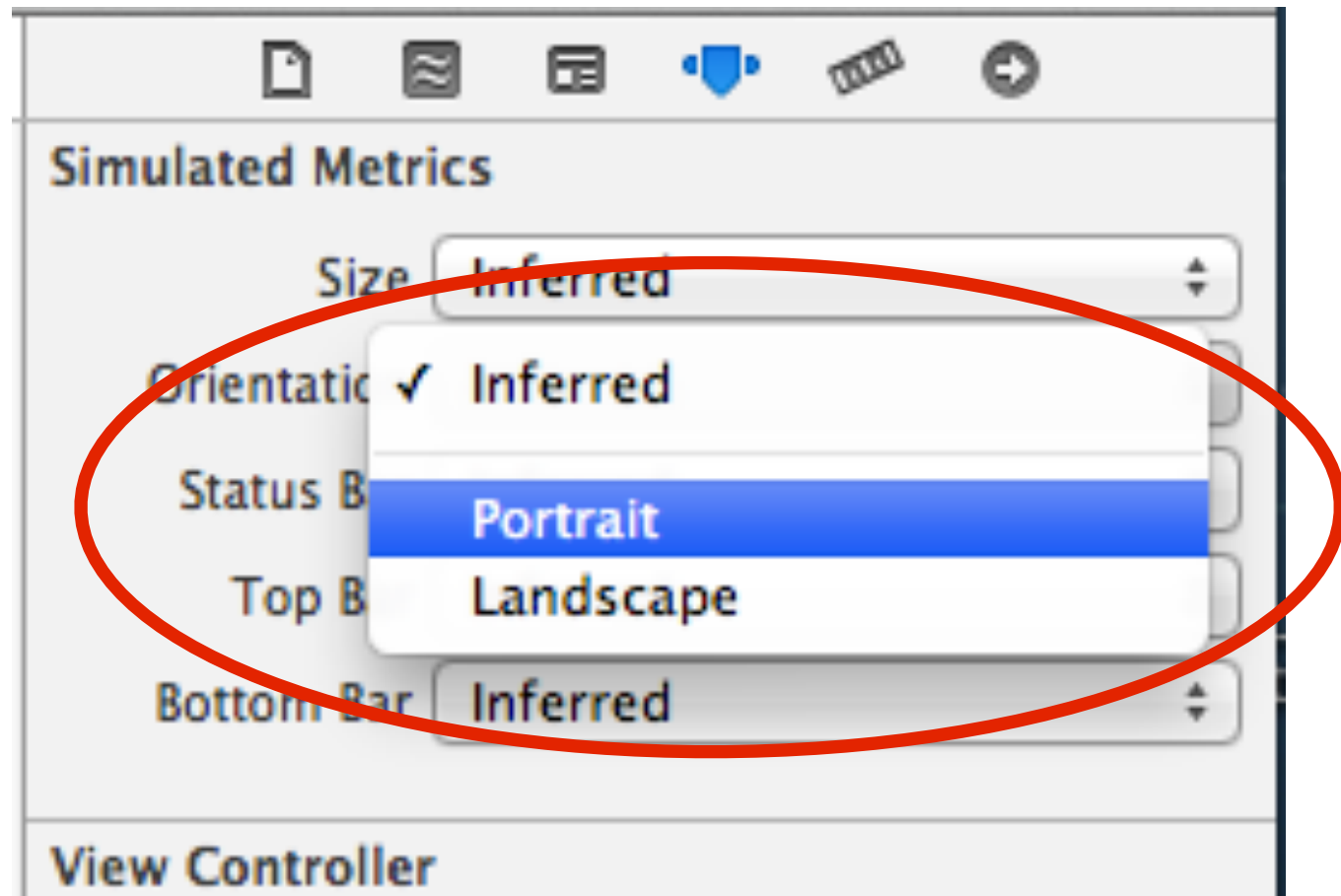


3.5" screen

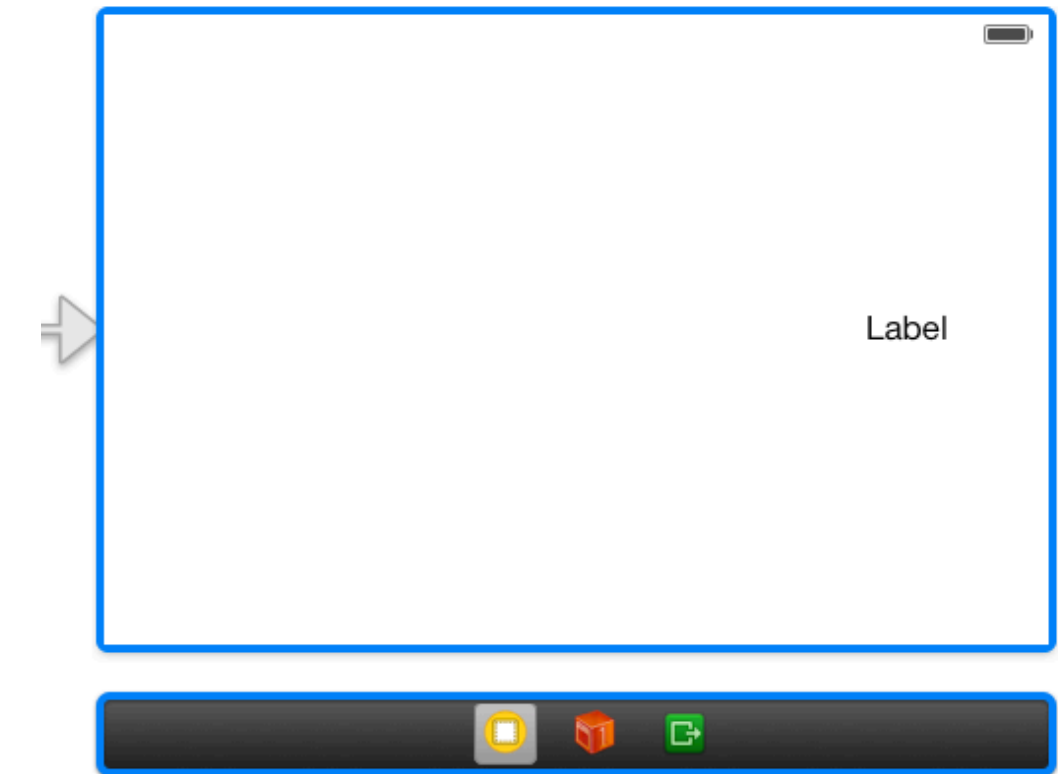


4" screen

# Checking constraints for different orientations



Portrait



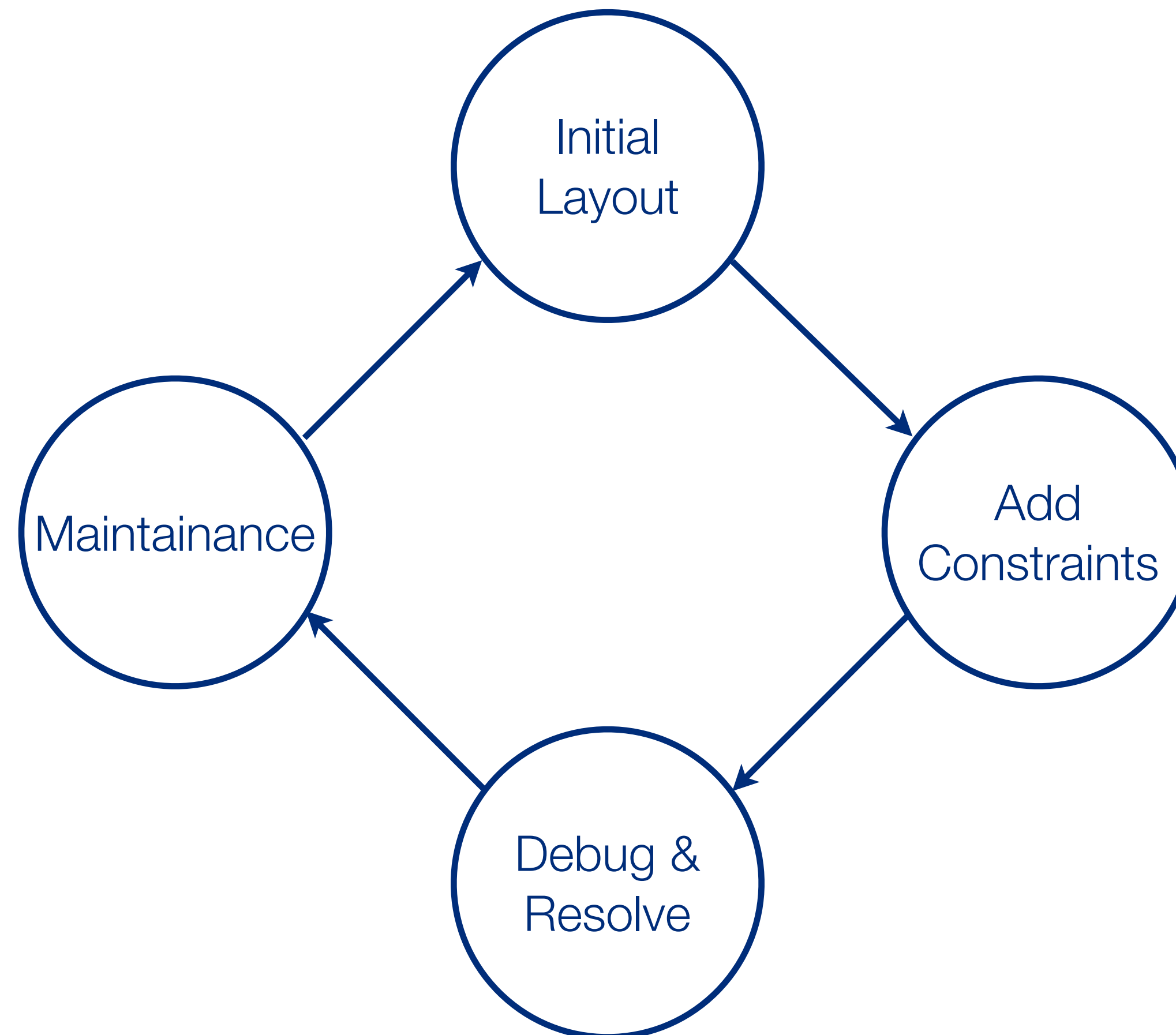
Landscape

# Deleting and editing constraints

- Constraints can be selected and deleted or edited
- Delete by using “delete” on the keyboard
- Editing can be made by using the menus shown previously

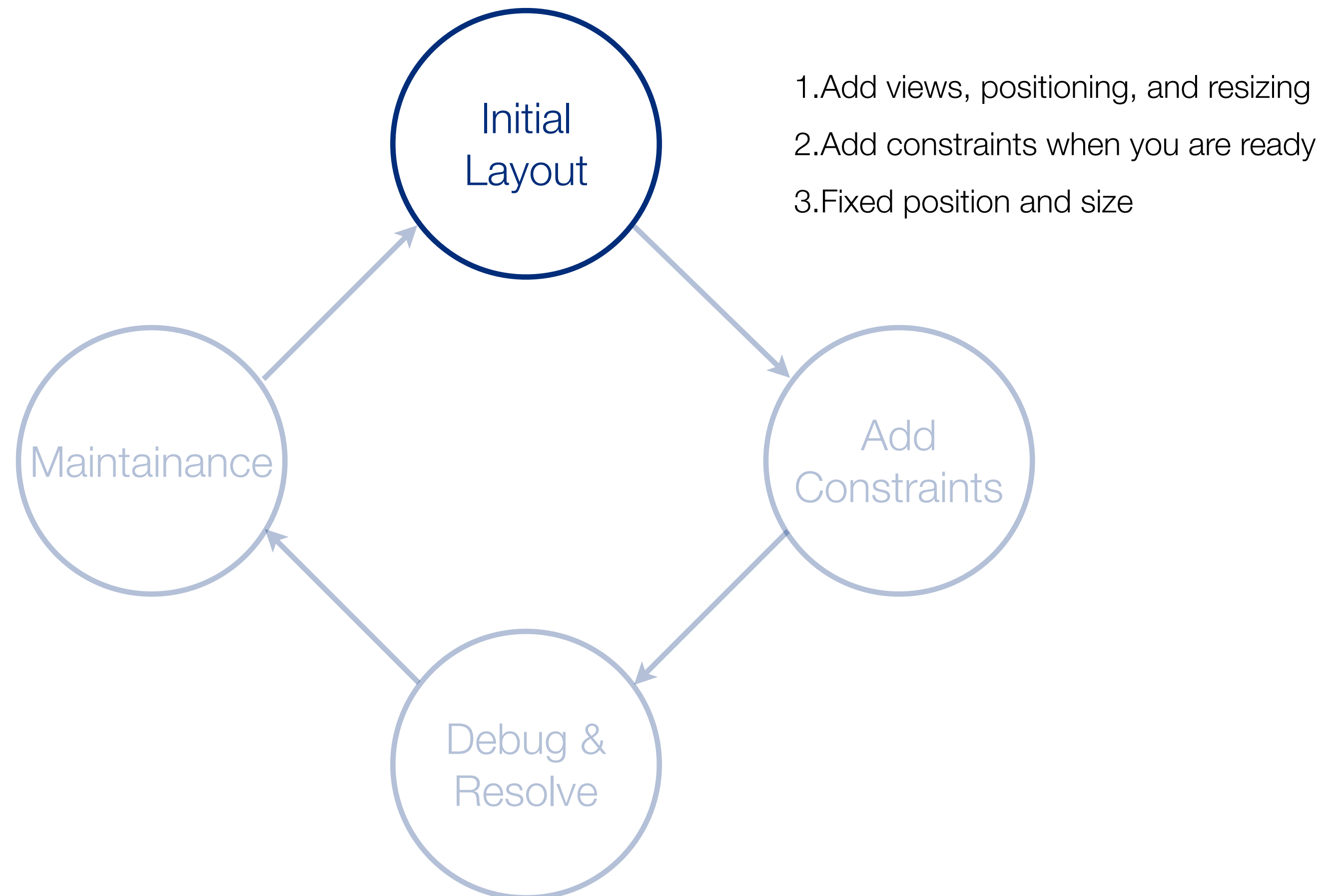
# Auto Layout process

- When working with Auto Layout, typically an iterative process is performed



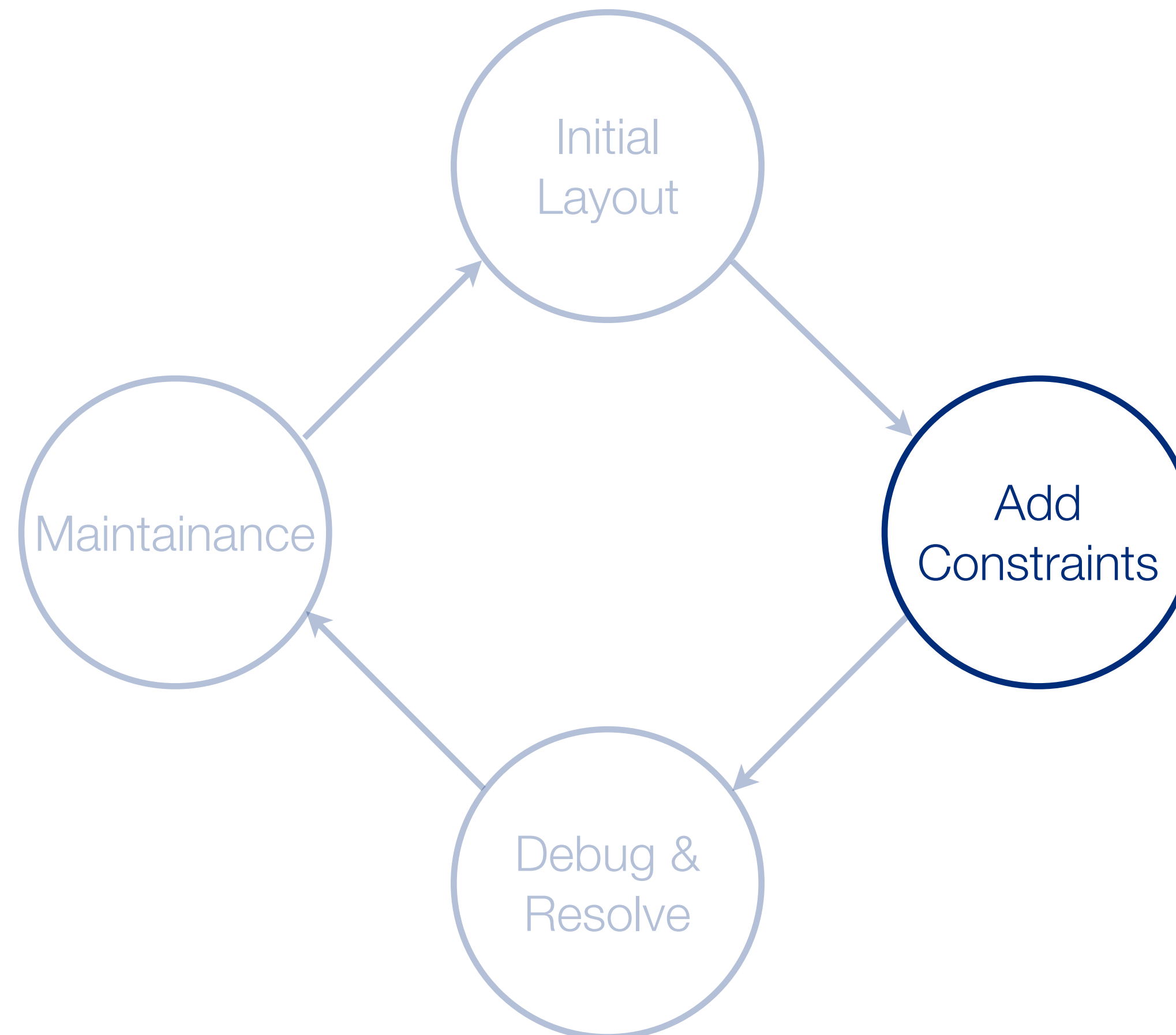
# Auto Layout process

- When working with Auto Layout, typically an iterative process is performed



# Auto Layout process

- When working with Auto Layout, typically an iterative process is performed

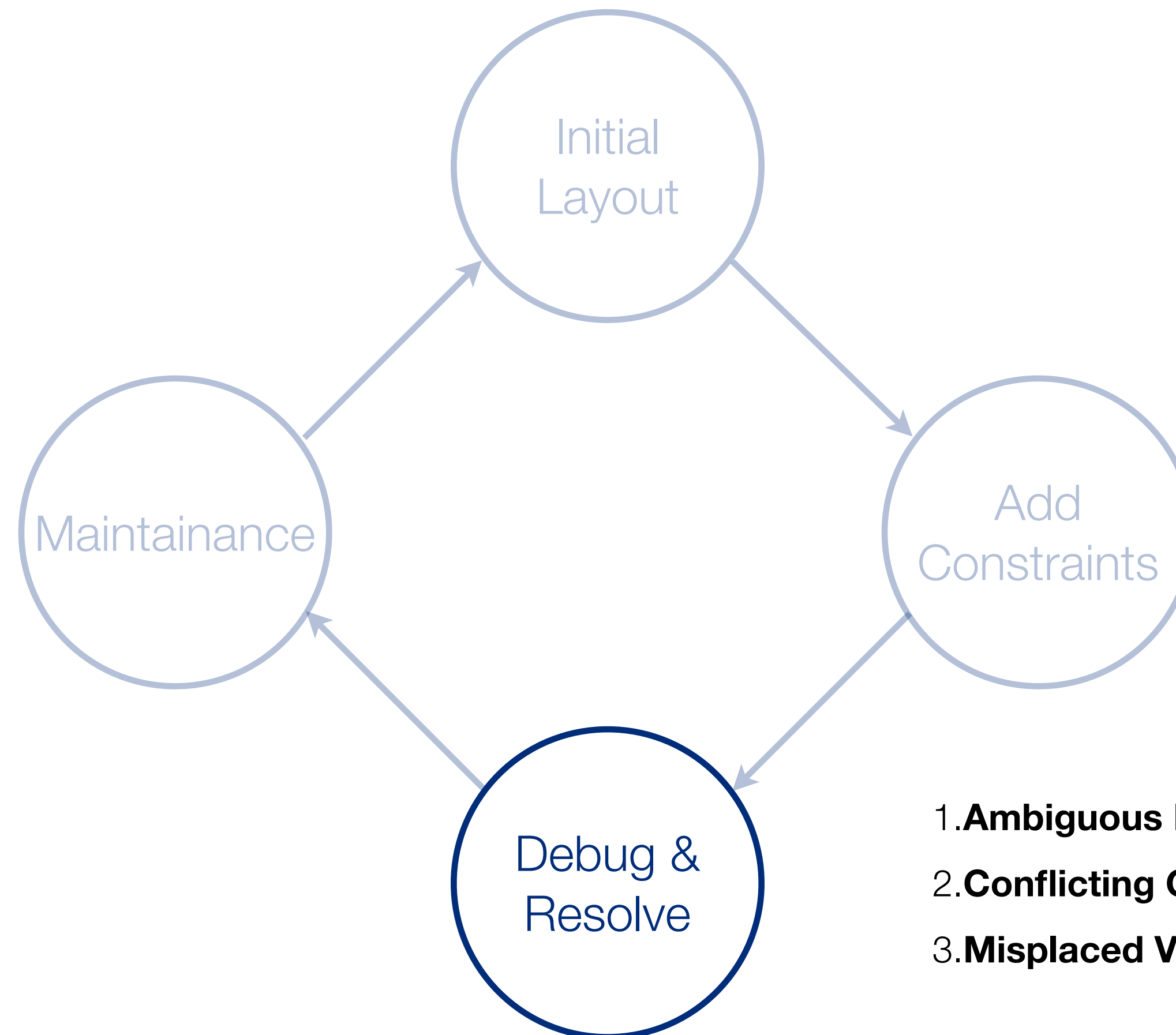


1. Check all changes keep things as they are supposed to be

1. Direct manipulation: Control drag between views
2. Auto Layout resolving menu
3. Constraint addition popovers

# Auto Layout process

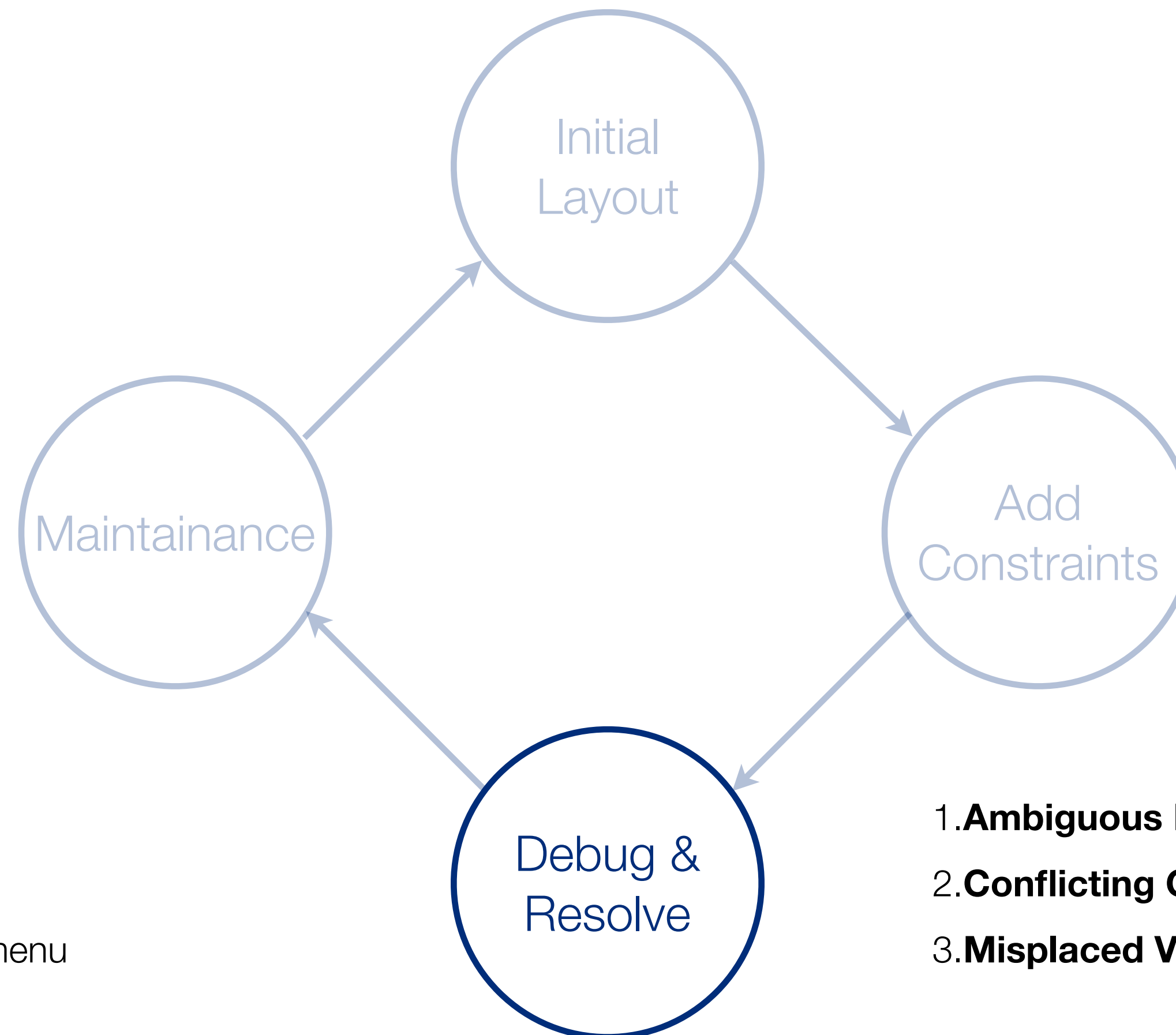
- When working with Auto Layout, typically an iterative process is performed



1. **Ambiguous Frames:** Not enough information
2. **Conflicting Constraints:** Too much information
3. **Misplaced Views:** Mismatched position or size

# Auto Layout process

- When working with Auto Layout, typically an iterative process is performed

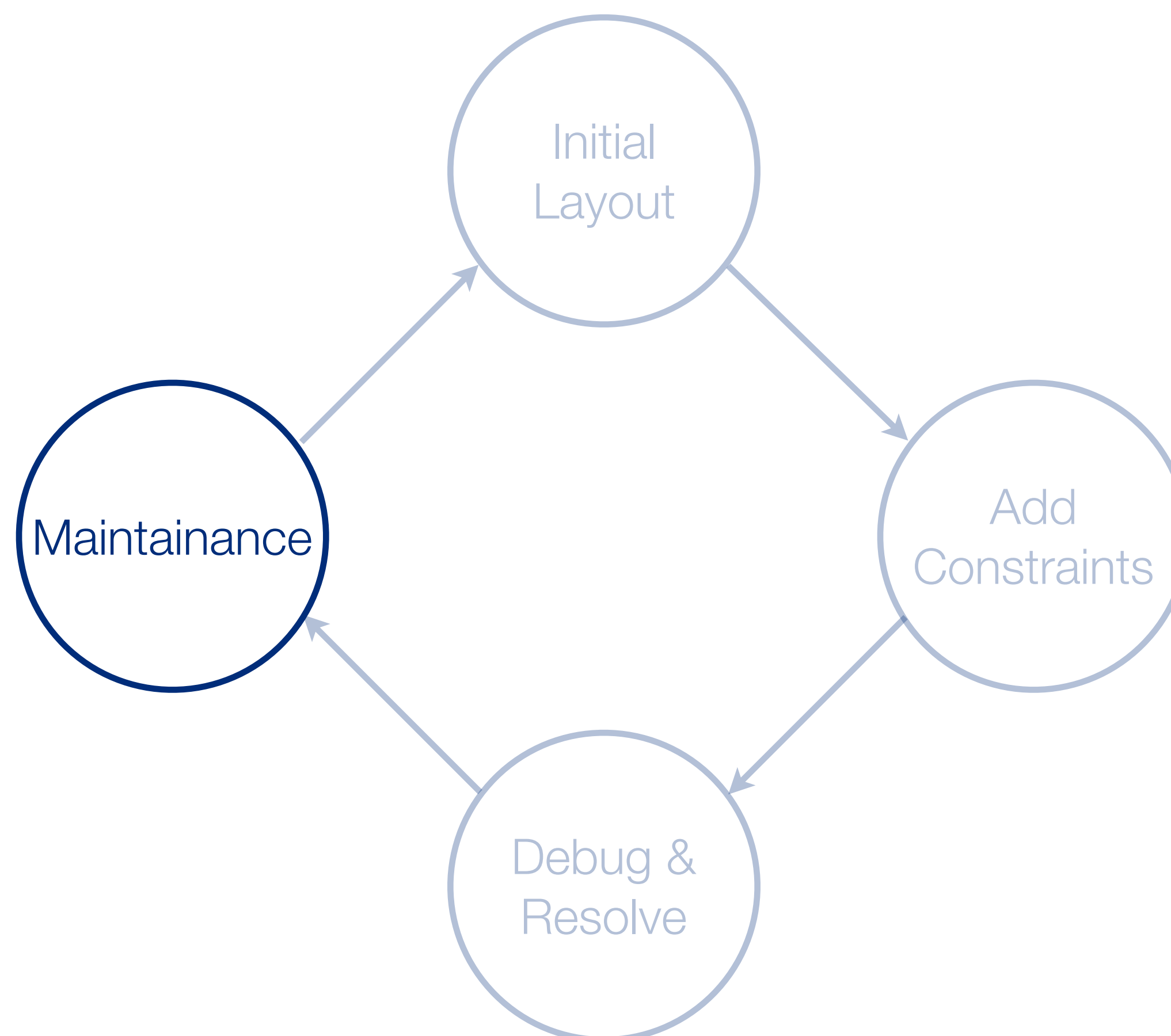


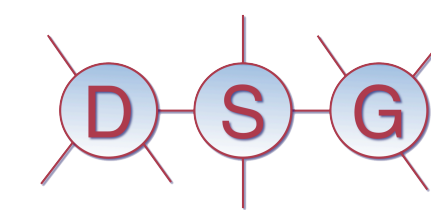
1. Canvas decorations
2. Xcode Issues Navigator
3. Quick fixes via the canvas resolving menu

1. **Ambiguous Frames:** Not enough information
2. **Conflicting Constraints:** Too much information
3. **Misplaced Views:** Mismatched position or size

# Auto Layout process

- When working with Auto Layout, typically an iterative process is performed





# Mobile Application Development

Lecture 24  
Auto Layout