

MOBILE DEVELOPMENT

AUTO LAYOUT USING INTERFACE BUILDER

William Martin
VP of Product, Floored

Angel X. Moreno
EIR, Developer

AUTO LAYOUT

LEARNING OBJECTIVES

- › Distinguish between Springs-and-Struts and Auto Layout.
- › Define what a “constraint” is and how we apply them to views.
- › Devise layouts using Auto Layout and Interface Builder.
- › Distinguish very clearly the differences between frame and constraint bounds.

AUTO LAYOUT

REVIEWING VIEWS

AUTO LAYOUT

UIVIEW

- A *class* that represents a drawable, interactive widget on a screen.
- Typically a rectangular region with a position and size.
- Subclass of UIResponder, that manages events and interactivity.
- Useful by itself, but subclassed into many, more-useful types:
 - UILabel
 - UITextField
 - UITextView
 - etc.

AUTO LAYOUT

UIVIEW

- superview: The UIView that contains a view in question.
- subviews: The UIViews a view contains. A view can contain multiple views. Sometimes they're called "child" views.
- frame: The position and size of the view within its superview (the *external* coordinate system).
 - Has: origin (x and y coordinates), size (width and height)
 - Property used most often.
- bounds: The view's *internal* coordinates system.
 - Usually, just the frame but with (0, 0) as the origin.

AUTO LAYOUT

POINTS, NOT PIXELS

- All UIView work we do in iOS uses **points**, *not* **pixels**.
 - A “point” is a virtual unit that may actually be rendered by multiple physical pixels.
 - <http://www.paintcodeapp.com/news/ultimate-guide-to-iphone-resolutions>

AUTO LAYOUT

MANAGING VIEWS IN IOS

- Two ways to manage view hierarchies:
 - Interface Builder (Storyboards and NIBs)
 - Code
- Three ways to lay out views:
 - Springs & struts (the older way)
 - Auto Layout (the newer way, what we'll cover)
 - Manually, with code.

AUTO LAYOUT

BEGINNING AUTO LAYOUT

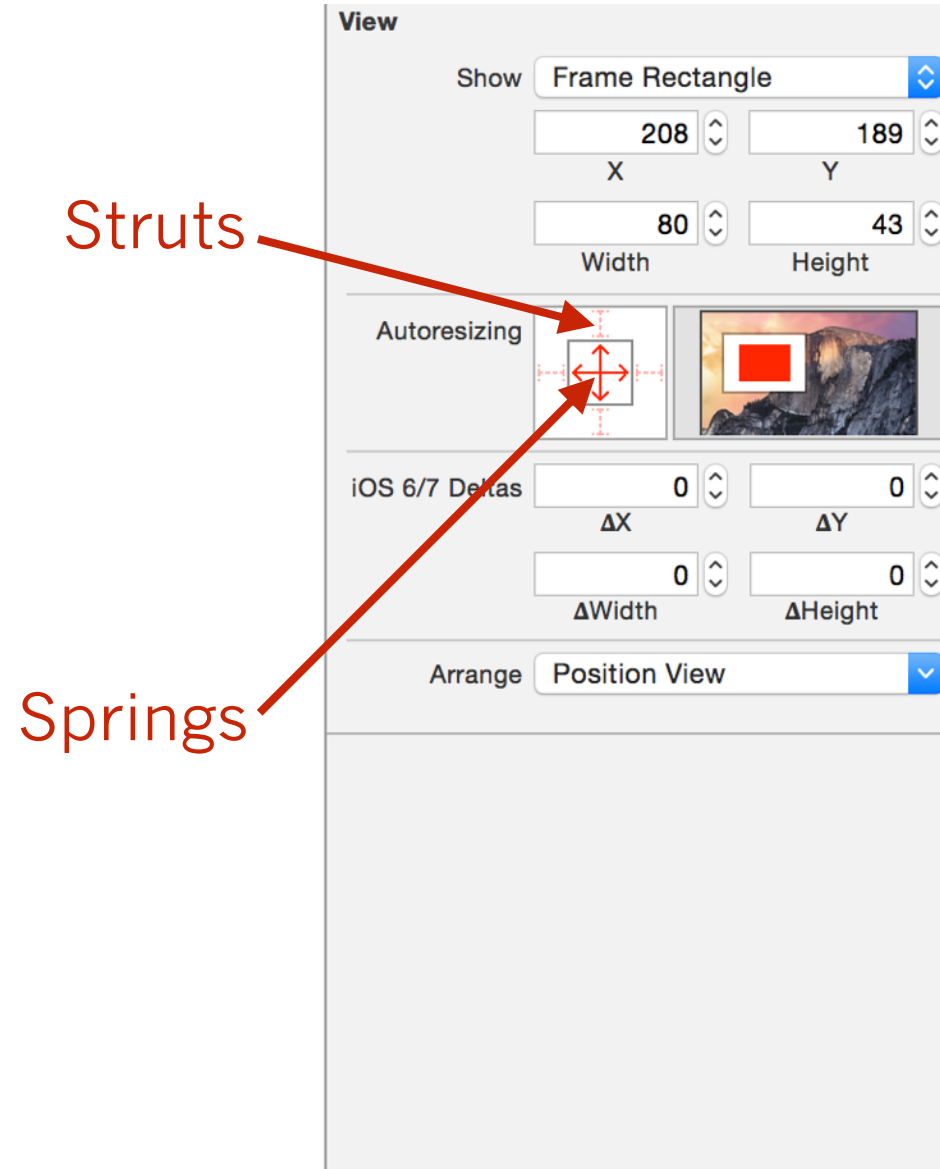
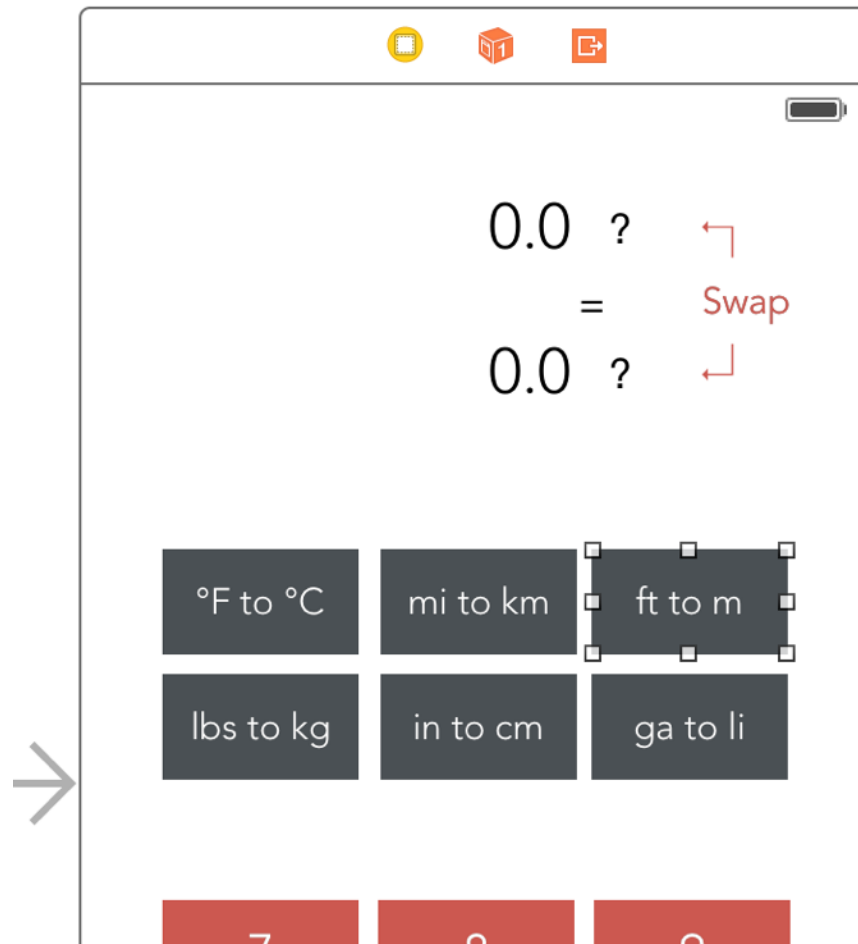
AUTO LAYOUT

SPRINGS + STRUTS

- Lays out views with respect to their superviews.
 - Springs are flexible and can scale to the superview (width + height).
 - Struts are inflexible, used for margins.
- Pros:
 - Much simpler to understand and implement than Auto Layout.
 - Good for a large set of cases.
- Cons:
 - Can't describe relationships between sibling views explicitly.
 - Can become problematic when views start changing size (e.g. text views that change size).

AUTO LAYOUT

SPRINGS + STRUTS



AUTO LAYOUT

WHAT IS AUTO LAYOUT?

- A newer, complex, more general system to lay out views.
- It leverages the concept of “constraints” and a solver that updates view positions and sizes *automatically*.
- When view sizes change, the device reoriented, or the app run on different devices, Auto Layout figures out how to adjust the views’ positions and sizes.

WHAT IS AUTO LAYOUT?

UT?

Architectural floor plan of the Silver Jeans Showroom, 1,314 S.F. The plan shows a large central showroom area with a balcony, two offices, a kitchen, restrooms, and a freight elevator. Dimensions are provided for all major areas and walls.

Key Features and Dimensions:

- Overall Dimensions:** 72'-9 1/4" (Total Width) x 47'-8 1/2" (Total Depth).
- Central Showroom:** 61'-3 1/2" (Width) x 15'-7 1/2" (Depth). Includes a balcony (10'-0 1/2" x 4'-8 1/2").
- Offices:** Two offices, each 19'-3 1/2" x 10'-11 1/2".
- Kitchen:** 8'-1" x 8'-11 1/2". Includes a counter, ref., and storage room.
- Restrooms:** Women's Lav., Men's Lav., and a general Lav.
- Elevators:** Freight Elevator and two standard Elevators.
- Stairs:** Stair A (UP/DN) and Stair B (UP).
- Other Rooms:** Mech. Room, Storage Room, and a Corridor.

Notes:

- EXPOSED STRUCTURE ABOVE @ 10'-9" ± AFF PAINTED WHITE
- CONC. FLOOR (FAIR CONDITION)

AUTO LAYOUT

WHAT IS AUTO LAYOUT?

A few words about Auto Layout:

- › First, “Auto Layout” is a misnomer. It’s quite manual, tedious, and dumb to set up and use.
- › Auto Layout is counter-intuitive; the terminology is not obvious; and the tools are non-deterministic.
- › Auto Layout is hard. Every time you hear or read “easily” or “as simple as” with regards to Auto Layout itself or the tools IB gives you to manage it, *it’s a lie.*

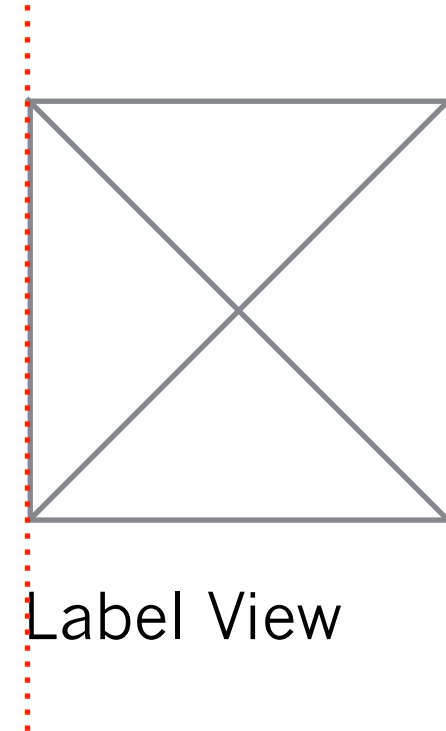
CONSTRAINTS

- A single constraint describes a one-dimensional relationship between views.
- E.g. “The left edge of the body should equal the right edge of the sidebar.”
- Constraints consist of:
 - A “first” view and attribute.
 - A “relation.”
 - A “second” view and attribute.
 - Multiplier
 - Constant
 - Priority

AUTO LAYOUT

CONSTRAINT EXAMPLES

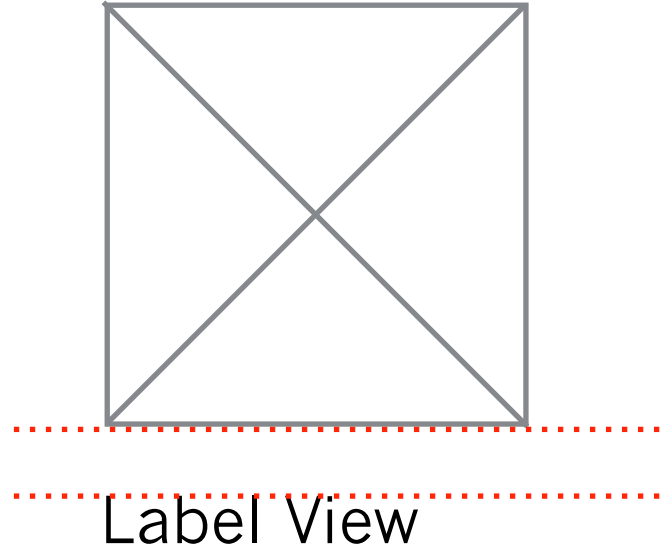
- First view + attribute: `MyLabelView.Left`
- Relation: Equal
- Second view + attribute: `MyImageView.Left`
- Multiplier: 1
- Constant: 0



- What does it mean?
 - The left coordinate of a Label should equal the left coordinate of an Image.

AUTO LAYOUT

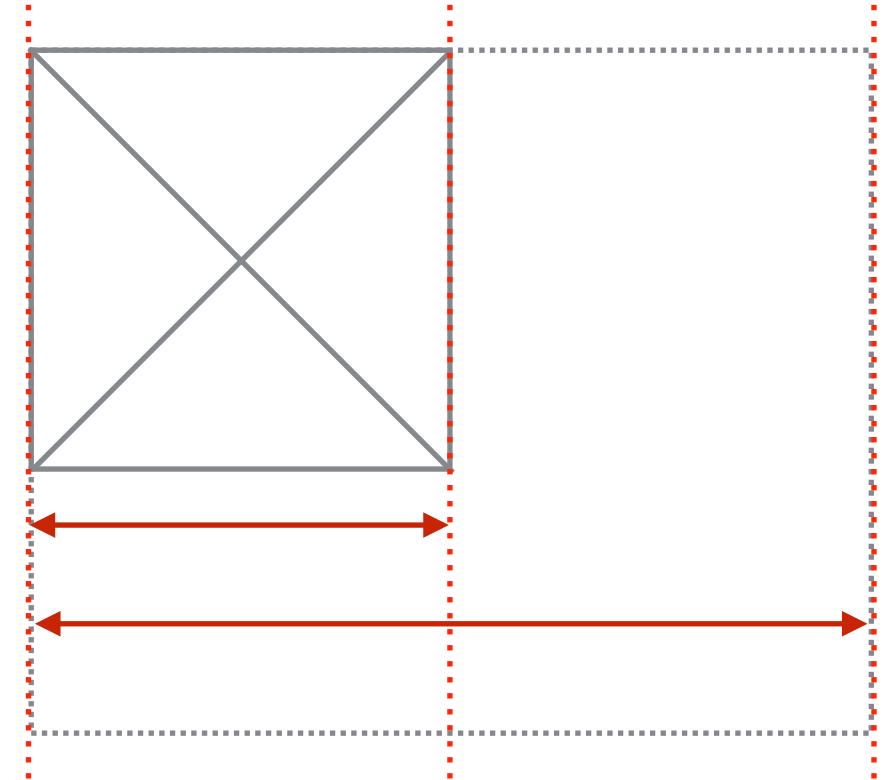
CONSTRAINT EXAMPLES

- First view + attribute: `MyLabelView.Top`
 - Relation: Equal
 - Second view + attribute: `MyImageView.Bottom`
 - Multiplier: 1
 - Constant: 10
- 
- What does it mean?
 - The top of the Label should be 10 points below the bottom of the Image.

AUTO LAYOUT

CONSTRAINT EXAMPLES

- First view + attribute: `MyImageView.Width`
 - Relation: Equal
 - Second view + attribute: `MySuperView.Width`
 - Multiplier: 1:2
 - Constant: 0
-
- What does it mean?
 - The width of the image should be 50% of the width of its superview.



AUTO LAYOUT

AUTO LAYOUT WHITEBOARD EXERCISE

AUTO LAYOUT

DESCRIBE WITH CONSTRAINTS

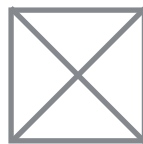
Custom Table Cell View



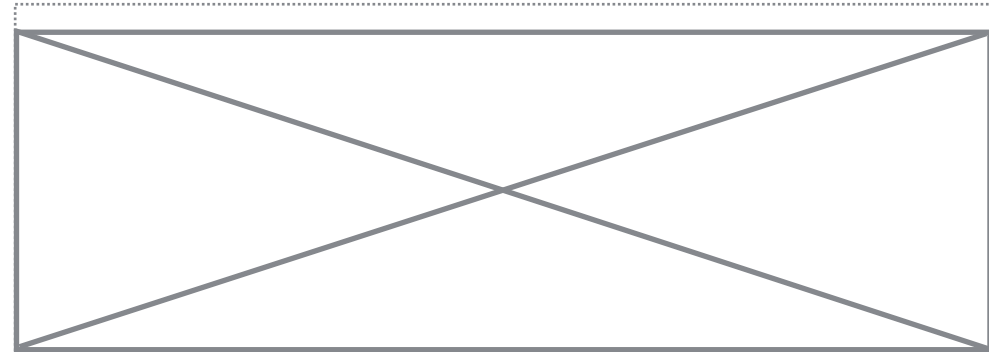
Title of a post

Body of a post. A few sentences that give a preview of what this article is about. Perhaps called an “excerpt?”

Tag Picker View



Detail View



Title of a post



Body of a post. This is the entire body as written by the author.

Multiple paragraphs, scrollable, and all that good stuff.

AUTO LAYOUT

AUTO LAYOUT IN IB

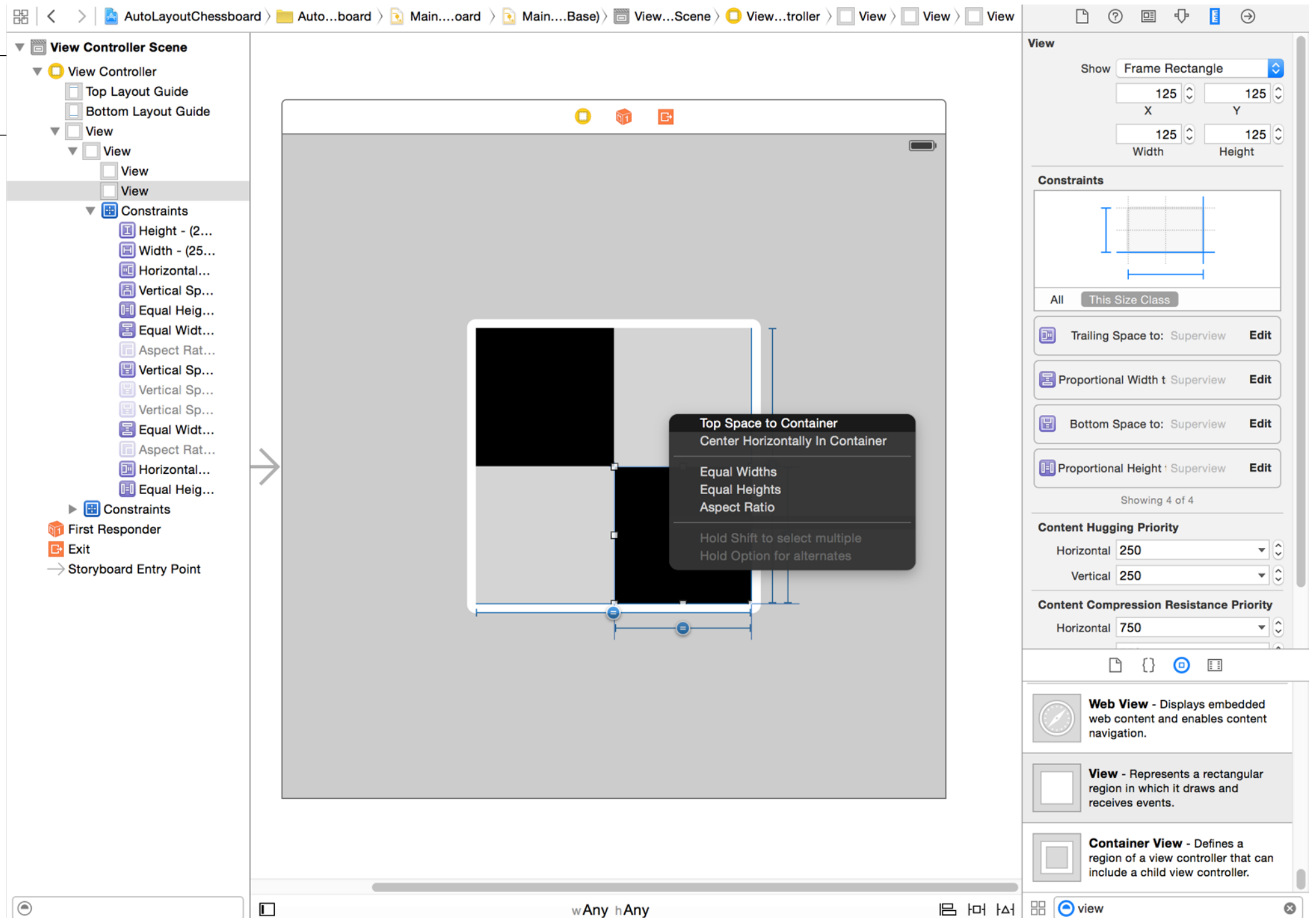
AUTO LAYOUT

AUTOLAYOUT IN IB

- › We can create and adjust constraints in Interface Builder to create dynamic layouts without using code.
- › IB is dumb. It nearly always guesses incorrectly when you create constraints.
 - › e.g. When trying to match a leading edge to a trailing edge, it will create a constraint from the leading edge to a *leading edge plus a constant* instead.
 - › e.g. Sometimes it will create extra constraints for you that you simply don't need.
 - › *Every time you create a constraint, check it!*
- › IB will warn about conflicting/insufficient constraints while you author constraints, but this isn't always 100% comprehensive.

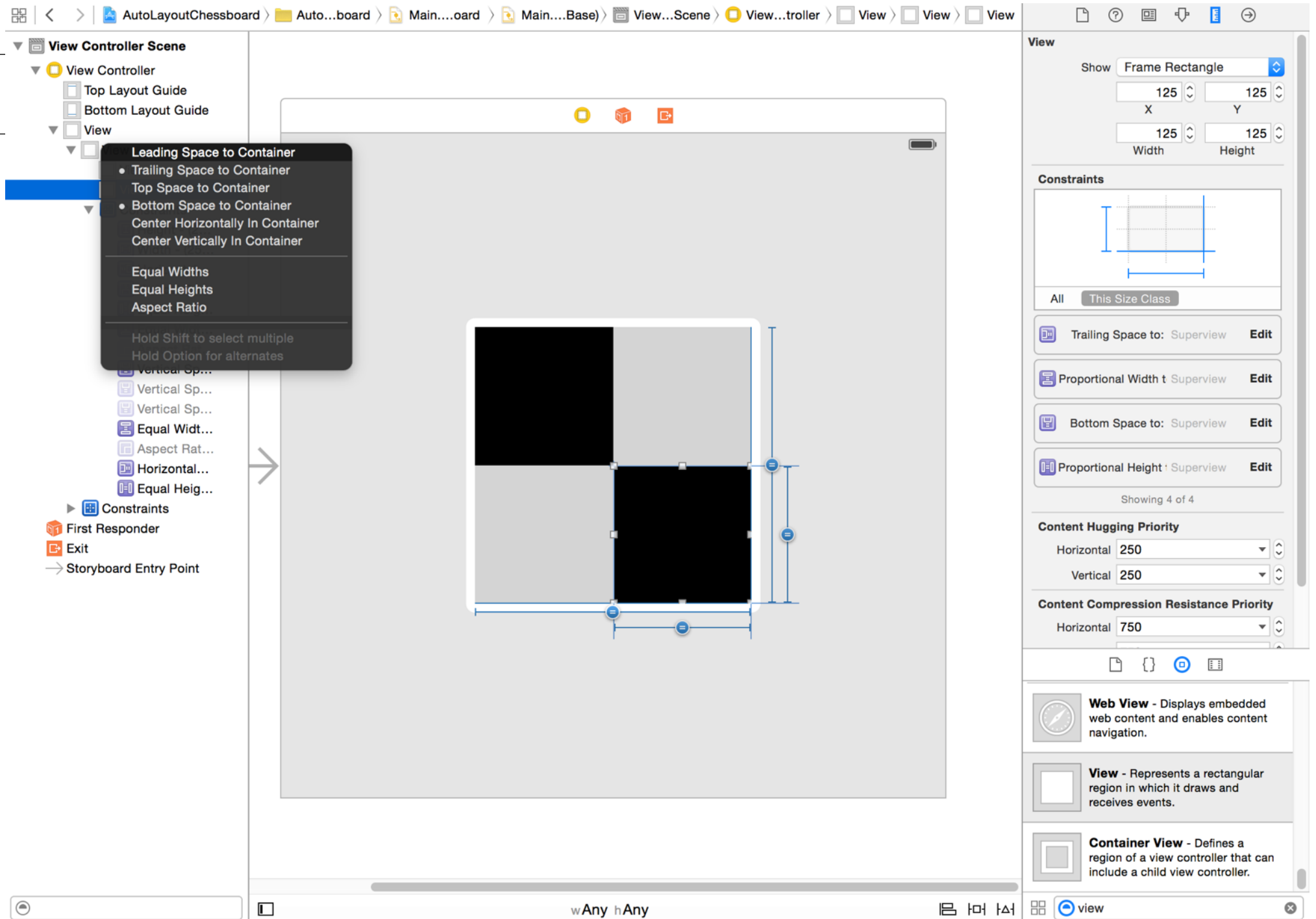
AUTO LAYOUT

Ctrl-drag from
view to view in the
IB canvas.



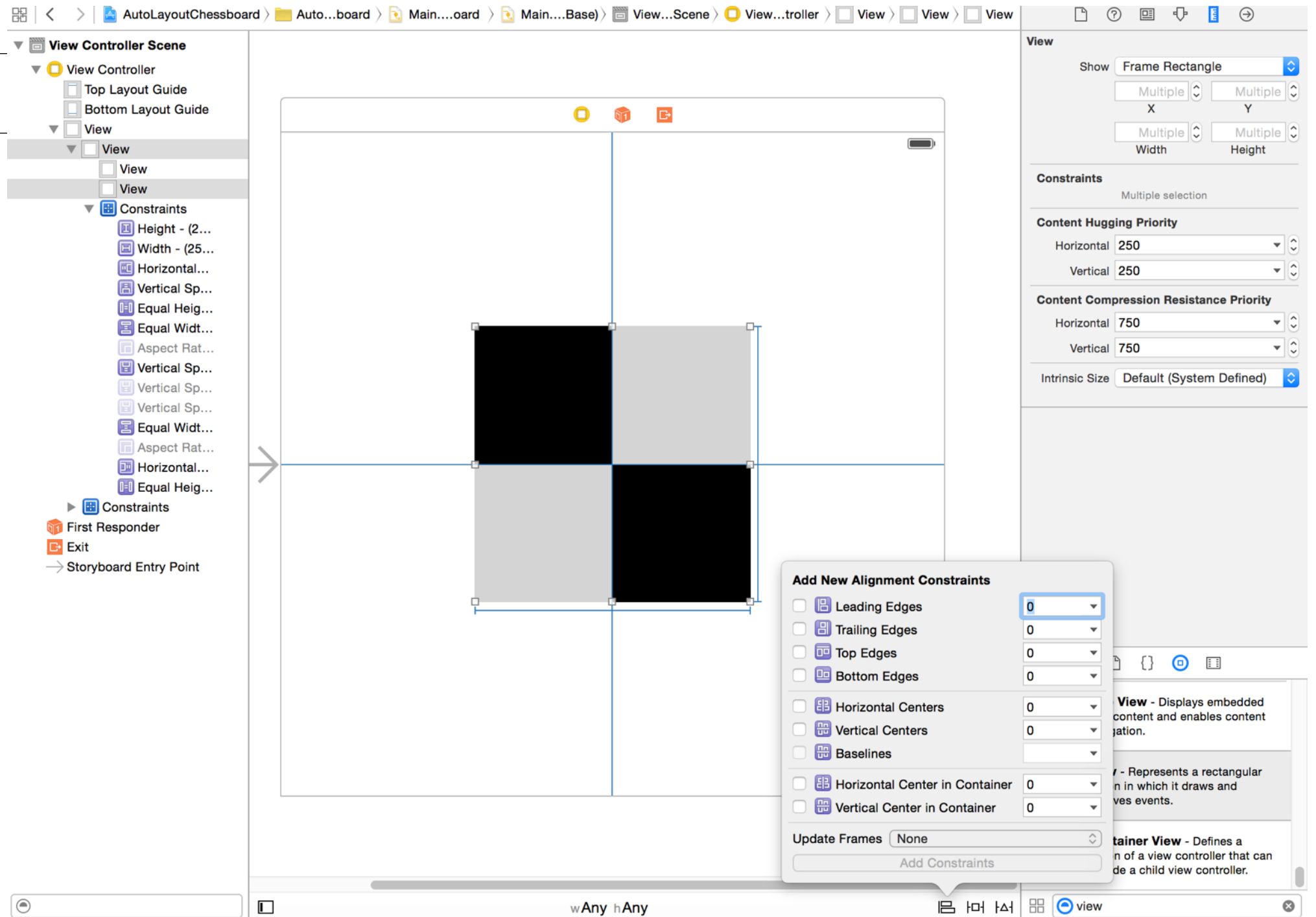
AUTO LAYOUT

Ctrl-drag from
view to view in the
document outline.



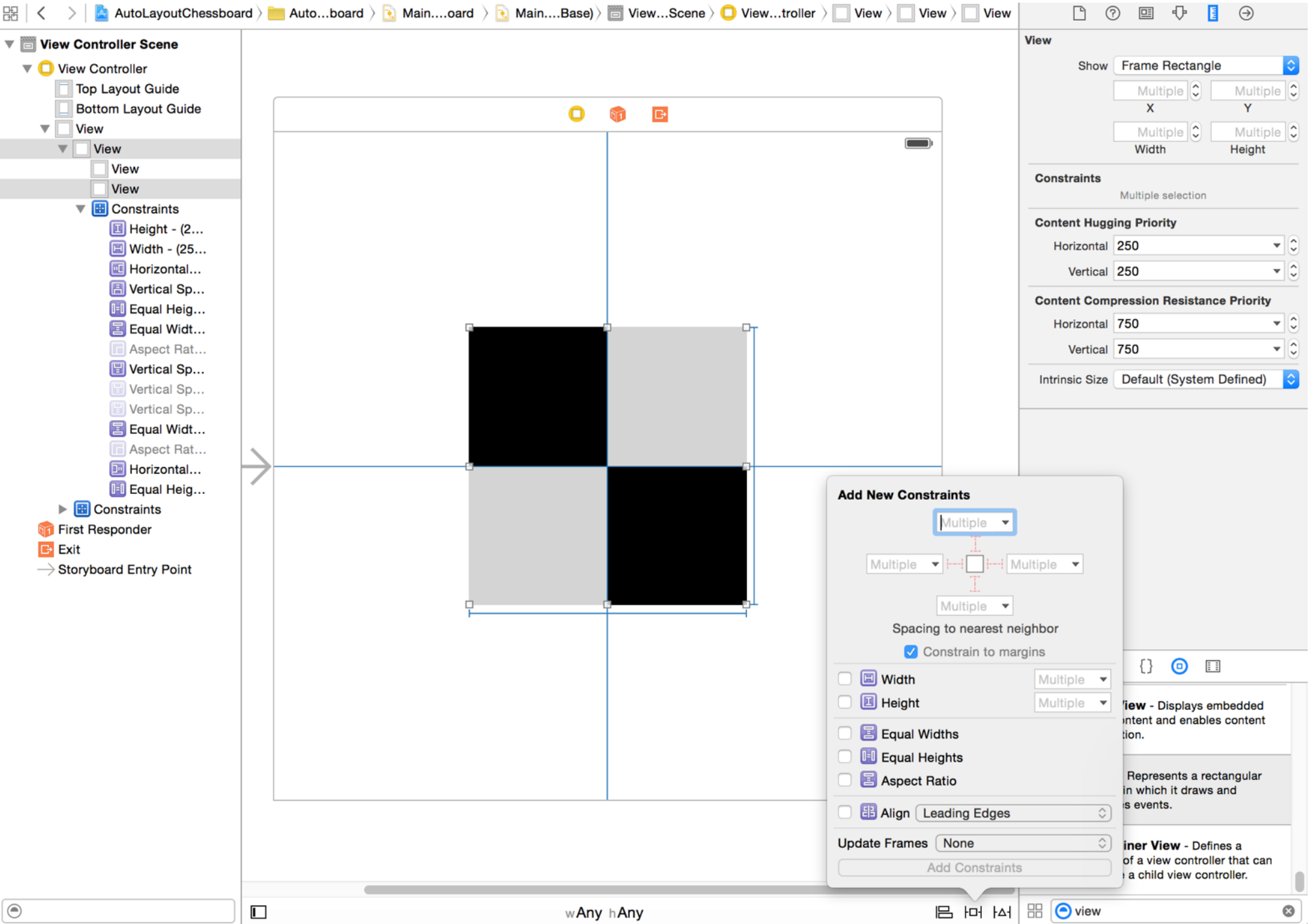
AUTO LAYOUT

Select two views
and create using
the alignment
constraint menu.



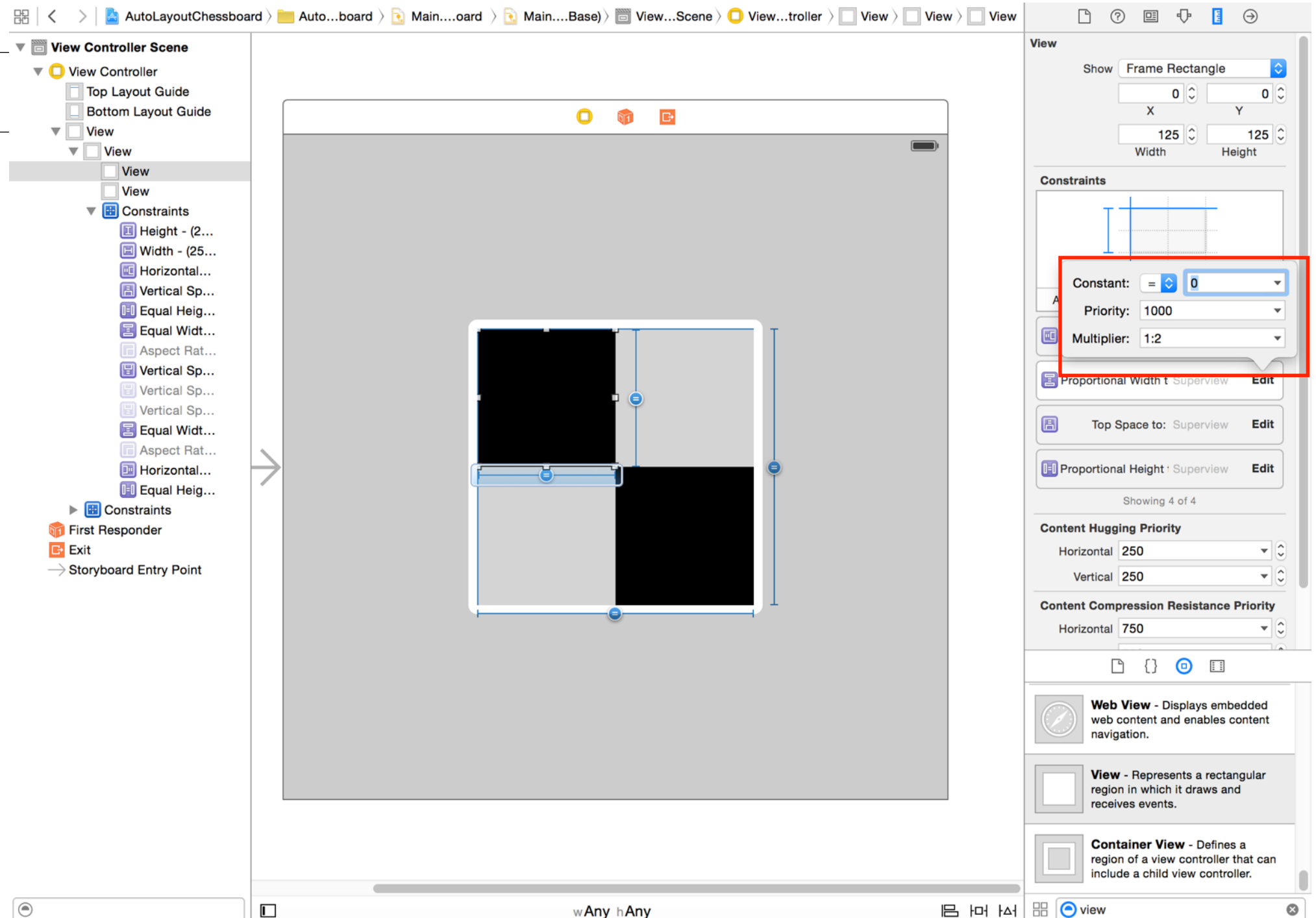
AUTO LAYOUT

Select two views and create using the size constraint menu.



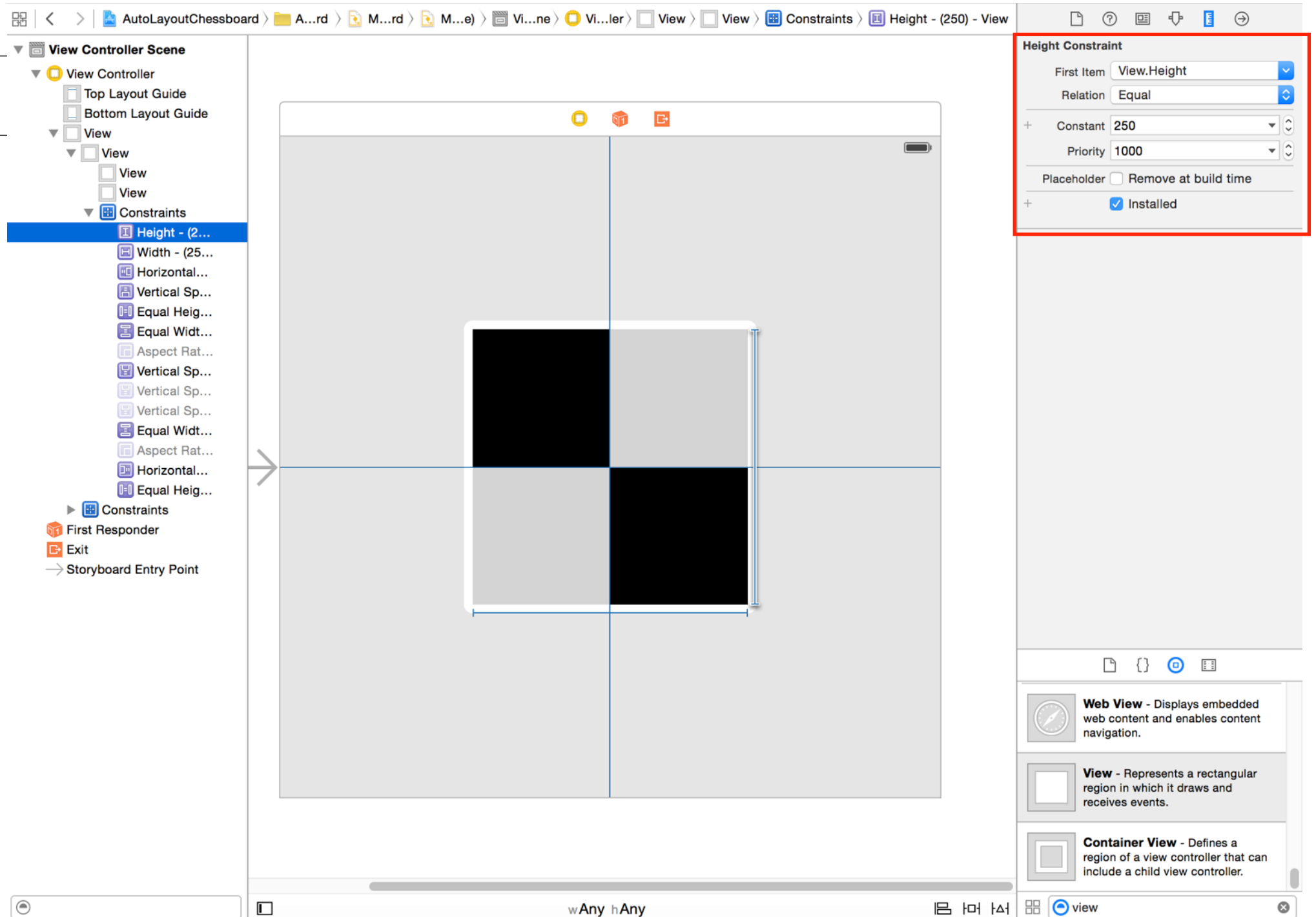
AUTO LAYOUT

Once a constraint is created, adjust some of its parameters by clicking on Edit in the inspector.



AUTO LAYOUT

Once a constraint is created, adjust *all* of its parameters by double-clicking in the inspector.



CONSTRAINT ERRORS

- Two common types of errors:
 - *Insufficient* constraints: We have too few constraints on at least one view, so the solver cannot figure out X, Y, width, or height.
 - *Conflicting* constraints: We have constraints on at least one view that result in *different* solutions for X, Y, width, or height.
- You should fix both of these when you see them, even if visual bugs do not result from them.

AUTO LAYOUT

AUTO LAYOUT IN IB WALKTHROUGH

AUTO LAYOUT

AUTO LAYOUT IN IB ACTIVITY

AUTO LAYOUT

YOUR ASSIGNMENT

- › Create a “face” with Auto Layout in IB.
- › The face must contain:
 - › Two eyes
 - › A nose
 - › A mouth
- › They must stay in place when the device rotates.