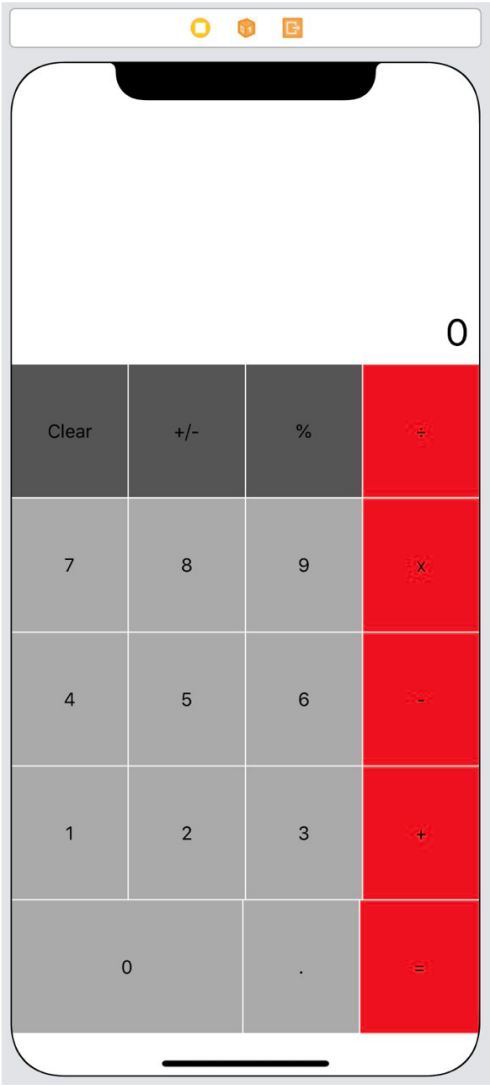
Lab—Calculator

The objective of this lab is to use Auto Layout to create a view that scales with the size and layout of any screen. You'll use view objects, constraints, and stack views to create a simple calculator that maintains its layout on all device sizes.

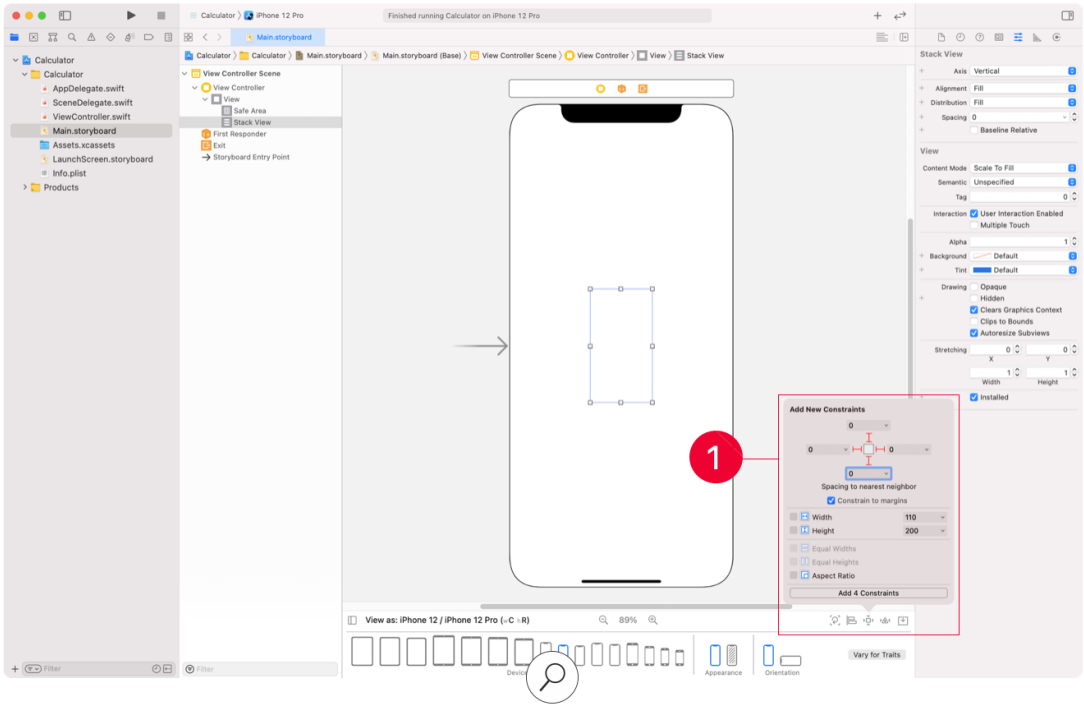
Create a new project called "Calculator" using the iOS "App" template.

Step 1

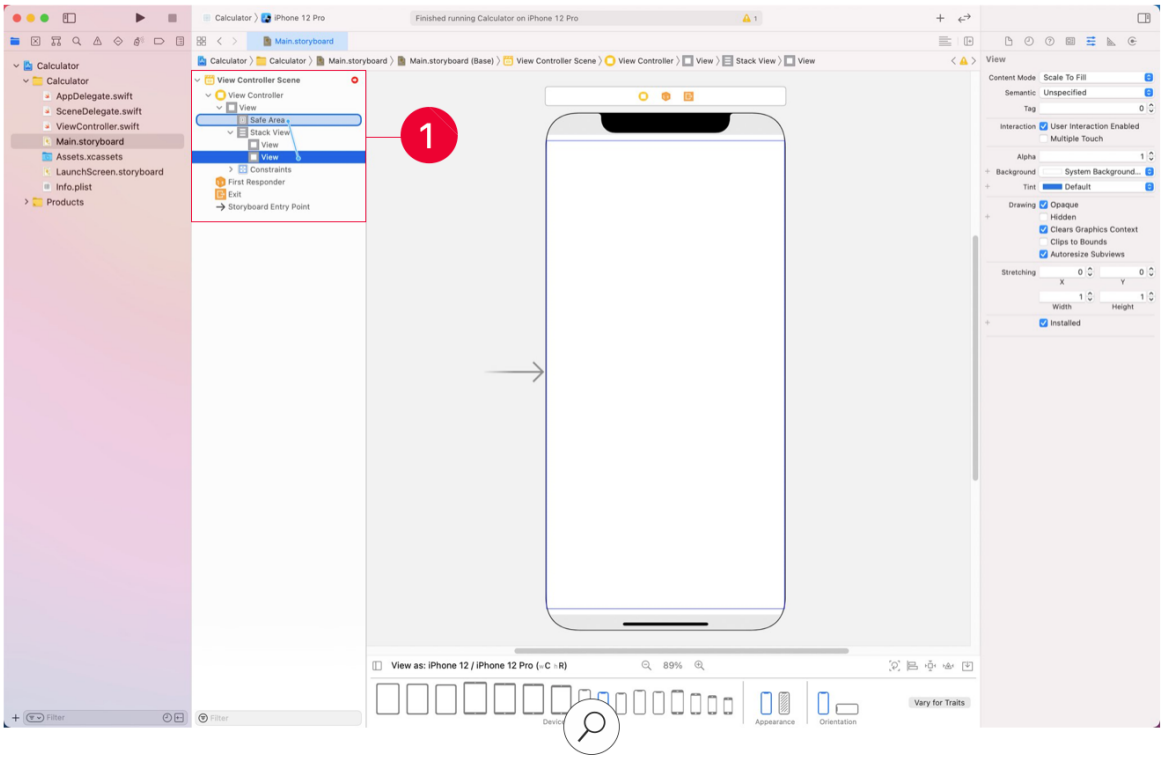
Create The Outer Containers



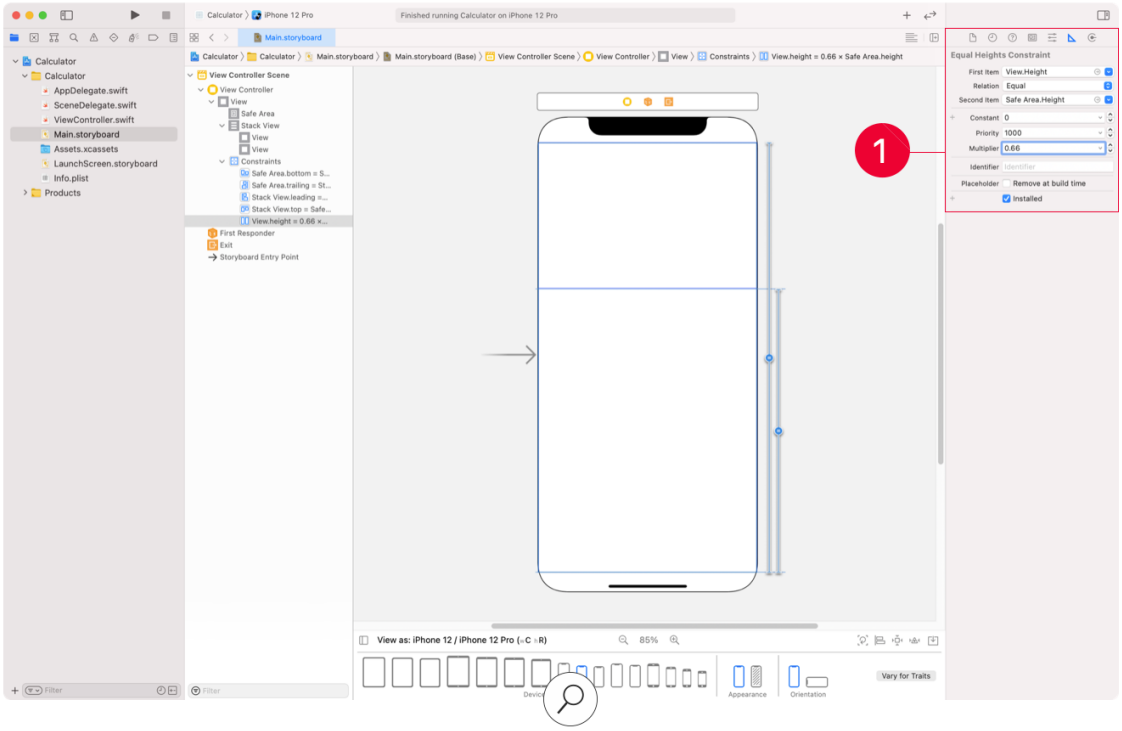
- In Interface Builder, set the "View as" device to 12 / iPhone 12 Pro.
- Ignoring the complexity of the buttons for now, it's easiest to break the calculator down into two distinct sections: the top third displays the digits that are entered, and the bottom two-thirds are used for input.
- Drag a vertical stack view from the Object library onto the scene. Use the Add New Constraints tool to add four constraints that align the top, leading, bottom, and trailing edges of the stack view to the the outer view's respective edges with 0 spacing. [1] Click the Update Frames button, and the stack view should cover the entire screen.



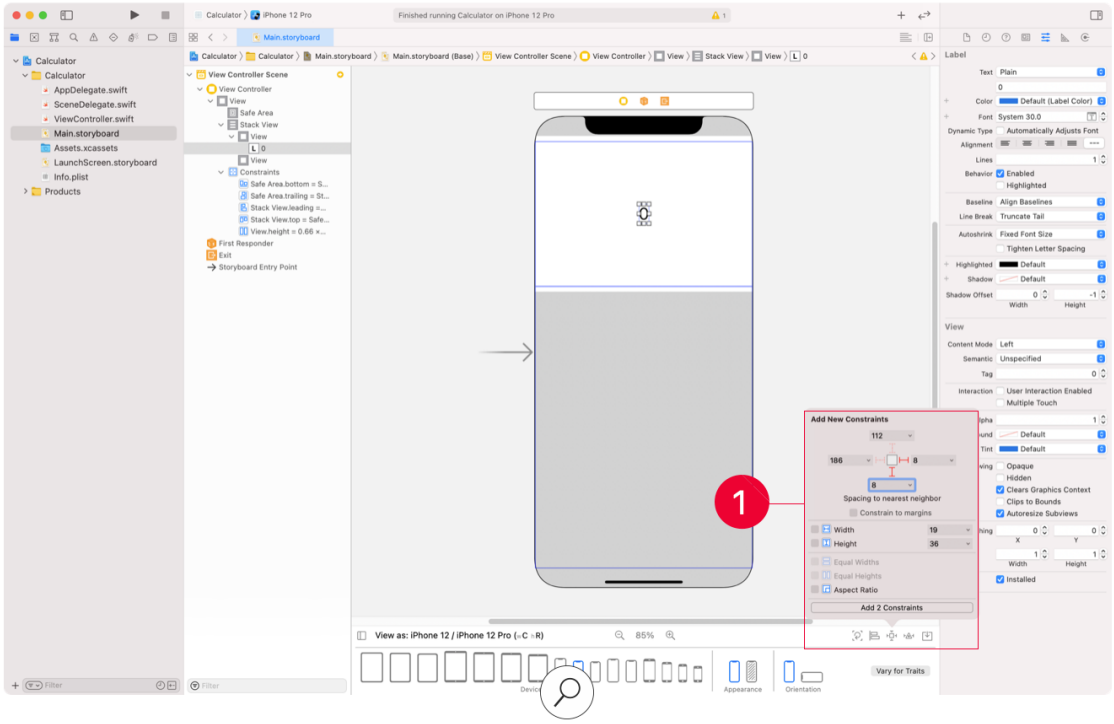
- Drag two `UIView` objects from the Object library into the stack view. **Control-drag** from the bottom view to the outer view, and select Equal Heights in the popup menu. This will set the inner view's height equal to the outer, which you will fix in the next step. [1]



- Select the bottom view and open the Size inspector. Locate the height constraint and double-click it. Change the Multiplier value from 1 to 0.66, which will set the inner view's height equal to the outer view's height, multiplied by 0.66.[1] The height of the bottom view will always be two-thirds the size of the view controller's view.



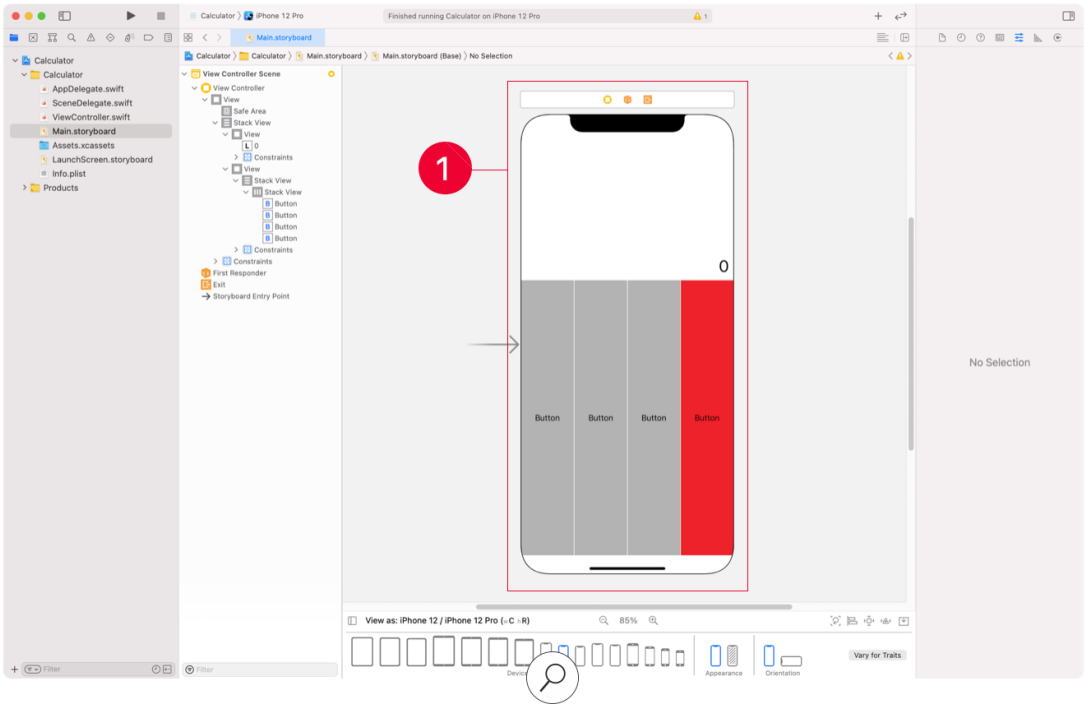
Add a `UILabel` from the Object library into the top container view. Set the text to "0" and the font size to 30.0, then open the Add New Constraints tool. Add bottom and trailing constraints, with 8 pixels of spacing between them. [1]



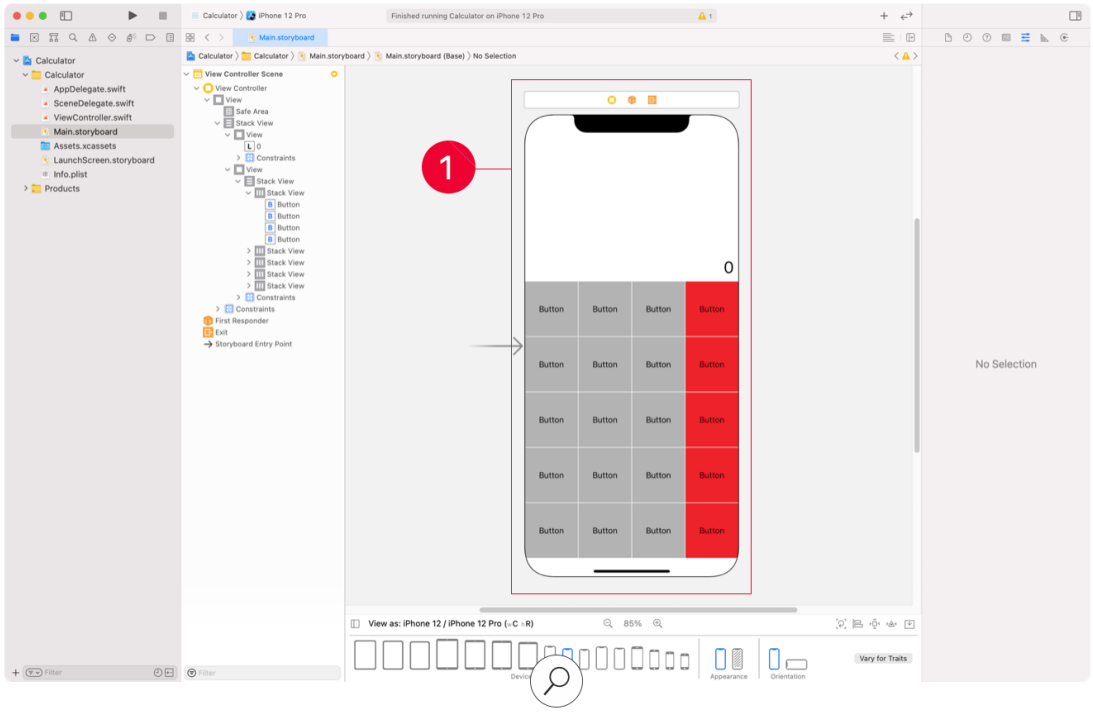
Step 2
Add And Size Buttons

- The finished calculator has five rows of buttons, where each row has an equal height. You can use another vertical stack view to accomplish this. Drag a vertical stack view into the bottom container view, then use the Add New Constraints tool to constrain th top, leading, bottom, and trailing edges of the stack view to the the bottom container view's respective edges with 0 spacing. Click the Update Frames button, and the stack view should cover the entire bottom container.
- Select the newly-added vertical stack view, and open the Attributes inspector. Set the "Distribution" property to "Fill Equally" so that all subviews within the stack view will have the same height. Set the "Spacing" to 1, which will create the horizontal separation lines between each row.
- You can use horizontal stack views for each row of buttons. Add a horizontal stack view into the vertical stack view, and set its "Distribution" property to "Fill Equally" so that all subviews within the stack view have the same width. Set the "Spacing" to 1, which will create the horizontal separation lines between each subview.
- Add a `UIButton` into the horizontal stack view. Set the tint color to Dark Text and the background color to Light Gray.

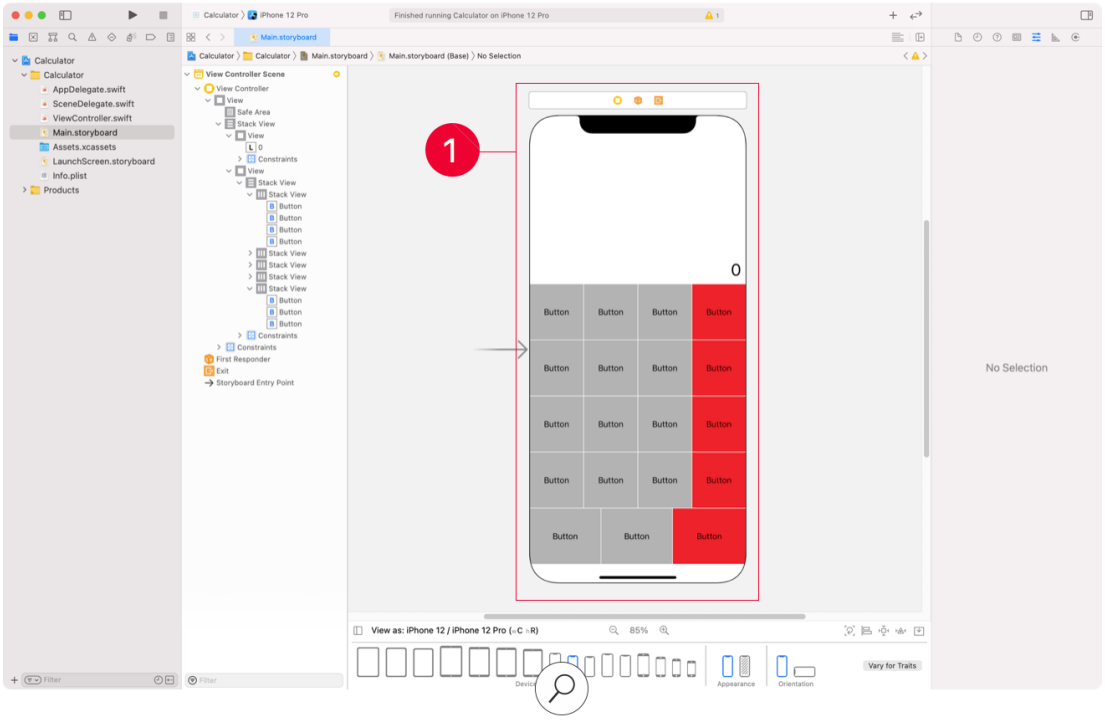
- Copy the button to your clipboard (**Command-C**), then paste (**Command-V**) three additional buttons into the horizontal stack view. Change the background color of the last button to System Red. [1]



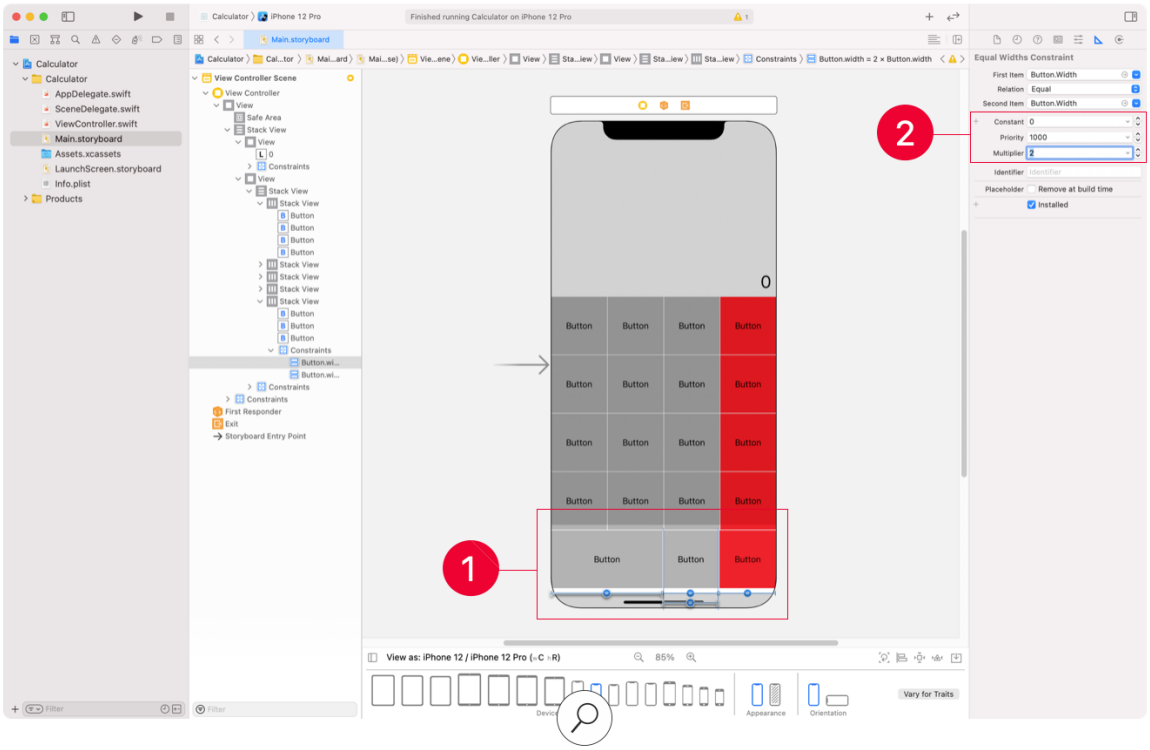
- Select the horizontal stack view in the Document Outline, and copy it to your clipboard (**Command-C**). Then paste (**Command-V**) four additional horizontal stack views into the vertical stack view. [1]



- The bottom horizontal stack view uses only three buttons instead of four. Since some buttons are sized differently than others, you need to update the "Distribution" of this stack view to "Fill Proportionally". Then, delete one of the light gray buttons from the bottom so the stack view contains the proper number of controls. [1]



- **Control-drag** from the bottom-leftmost light gray button to the button next to it, and then select "Equal Widths." This ensures that these two buttons are always the same size. Repeat the process for the middle light gray button and the red button. [1] Then select the leftmost gray button, and open the Size inspector. Double-click its width constraint and change the multiplier to 2. [2] This ensure that the left gray button is twice the width of the other gray button.



- Finally, update the top three buttons to use a dark gray color, and update the button titles to match those on a traditional calculator.

Congratulations! You've used both constraints and stack views to create a simple calculator. Be sure to build and run your application on multiple iOS devices to verify that the constraints you've defined make sense on all screen sizes. Save your work to your project folder.