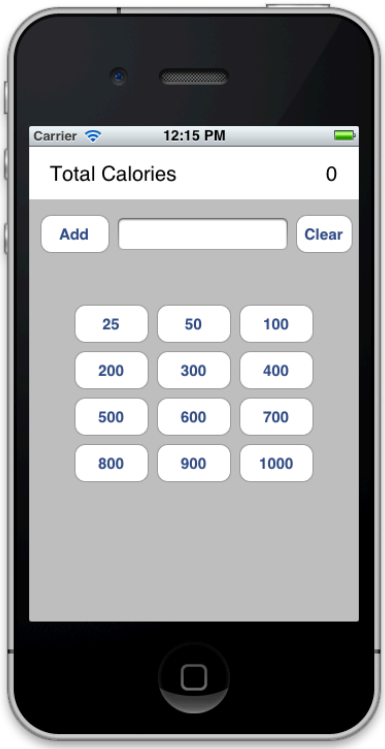
Assignment 1



Create a simple calorie calculator app that allows quick calorie tracking. It's important for iOS interfaces to design around the lowest amount of taps to perform an action. This calorie tracker will include buttons to add large increments, which can help enter quick amounts. The ideal use-case is 2 taps per entry. i.e the user taps the *500 calories button* and the *Add button*.

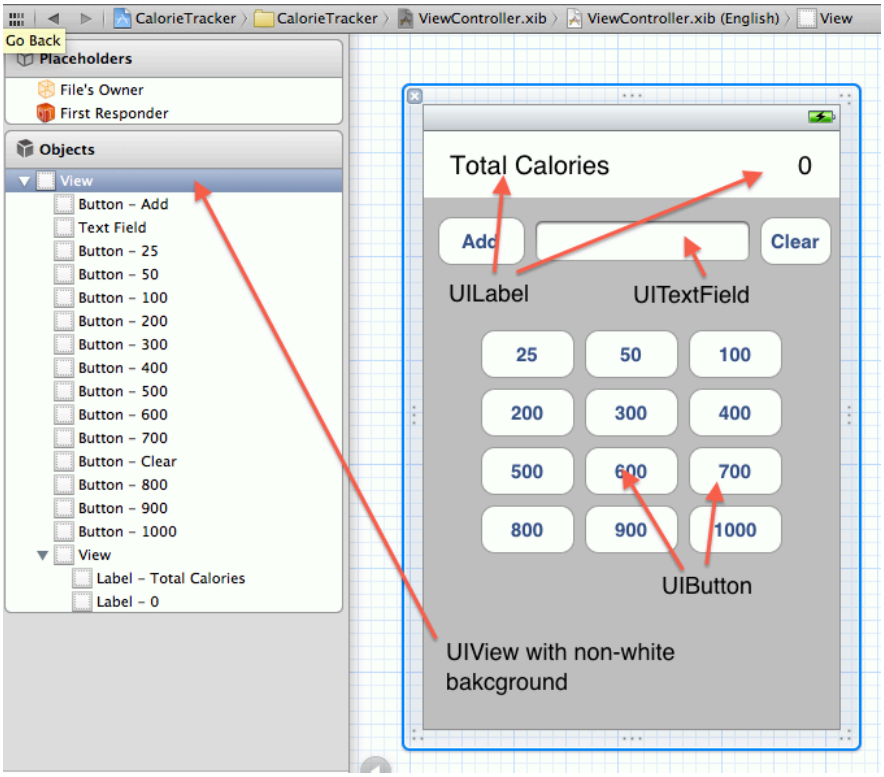
Total Calories UILabel

A text label that will update its display with the sum of all the calories added via the *Add button.* The add button will take the value from the scratchpad textfield and add it on to the current total sum tracked by the app.

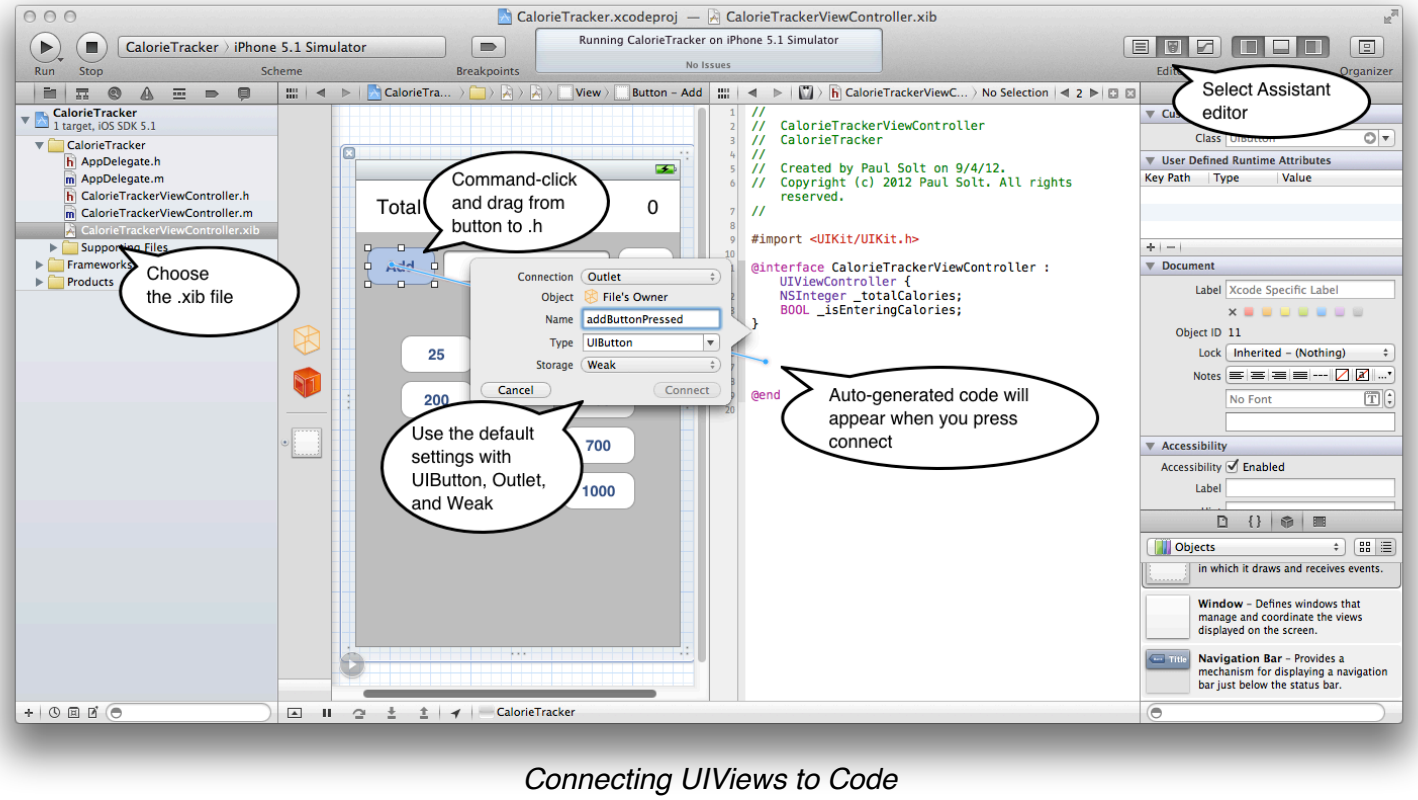
Scratchpad UITextField

A text field that displays the current sum from all of the *calorie buttons* that have been pressed. The user can edit this value from the keyboard before pressing the *Add button.*

1. Create the user interface using Xcode's Interface Builder. (iOS Ch 1) The markup and view hierarchy layout screens should provide one way in which the user interface can be constructed.
   a. Use the UILabel, UITextField, UIView, and UIButton objects to design the interface.



2. Link the calorie buttons to the following methods using the *Assistant Editor.* You can link multiple buttons to the same action.
   a. - *(IBAction)addButtonPressed:(id)sender;*
   b. - *(IBAction)calorieButtonPressed:(id)sender;*
   c. - *(IBAction)clearButtonPressed:(id)sender;*
3. Convert the value of the calorie button and use it to calculate the current amount of calories entered. Track the add amount with each button press and display the result in the *scratchpad UITextField.*
4. Implement the logic to add values from the *scratchpad UITextField* to the *total calories UILabel.*



Connecting UIViews to Code

Hints

1. Start with the Single-View iOS App Template (New -> Project)
2. Command-click or right-click a UI element to see it's connections.
3. Use the assistant editor to make connections between the user interface and the code.

Evaluation (100 points)

1. Assignments will lose points for the following:
   a. Build errors.
   b. Build warnings.
   c. NSLog() print statements.
   d. Lack of javadoc-style comments.

   i. Assume the reader is unfamiliar with your code. Explain what methods do in your header files and any complex logic in your implementation files.
e. Missing required specifications.
f. Code style does not match “Coding Guidelines for Cocoa.”

Grading

- Style/Documentation 30%
- Functionality 50%
- Organization 20%

Extra Credit

1. Make the keyboard hide after pressing *Add* or *Clear. (5 points)*
2. Switch from the default keyboard to numeric or numbers and punctuation. (2 points)