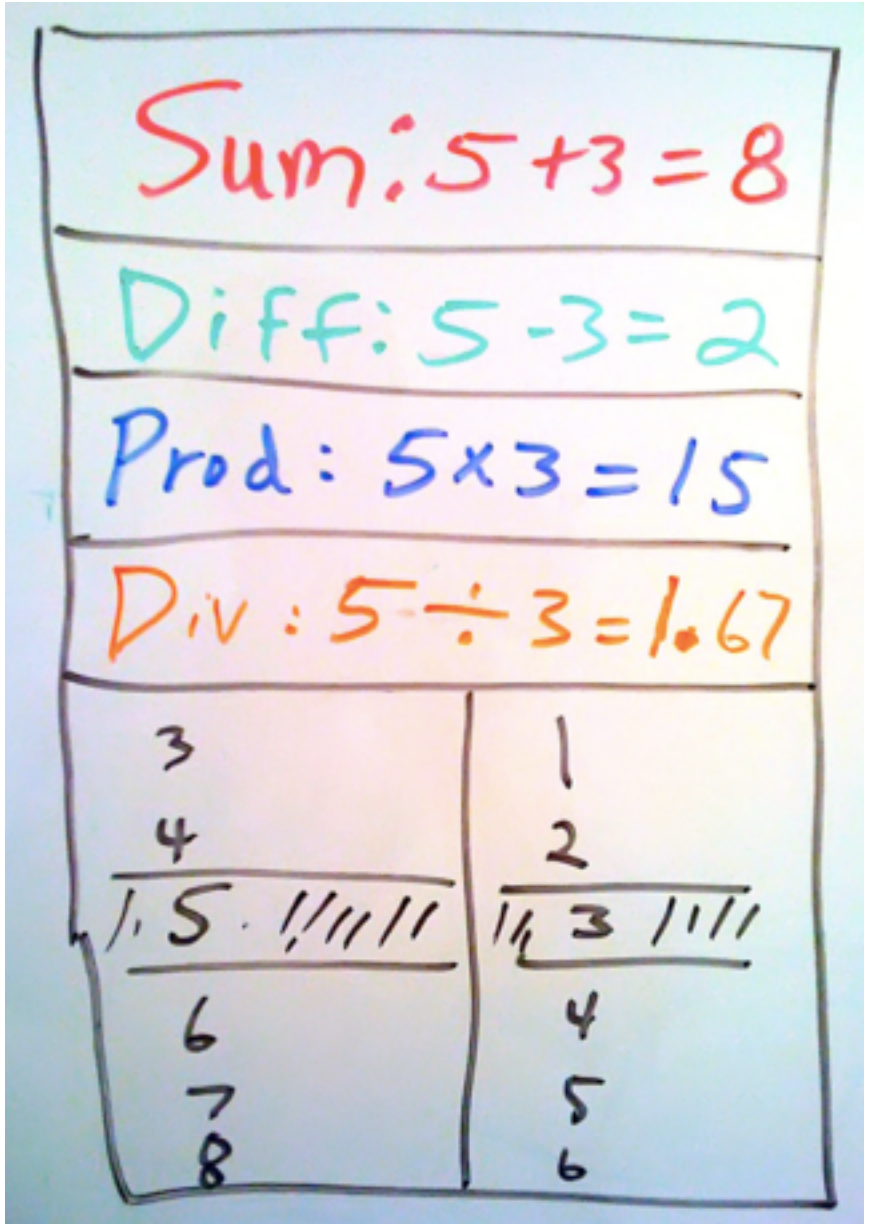
SpinCalc™ Case Study

Control Flow
Components
Code Fragments

The Movie (in the app)

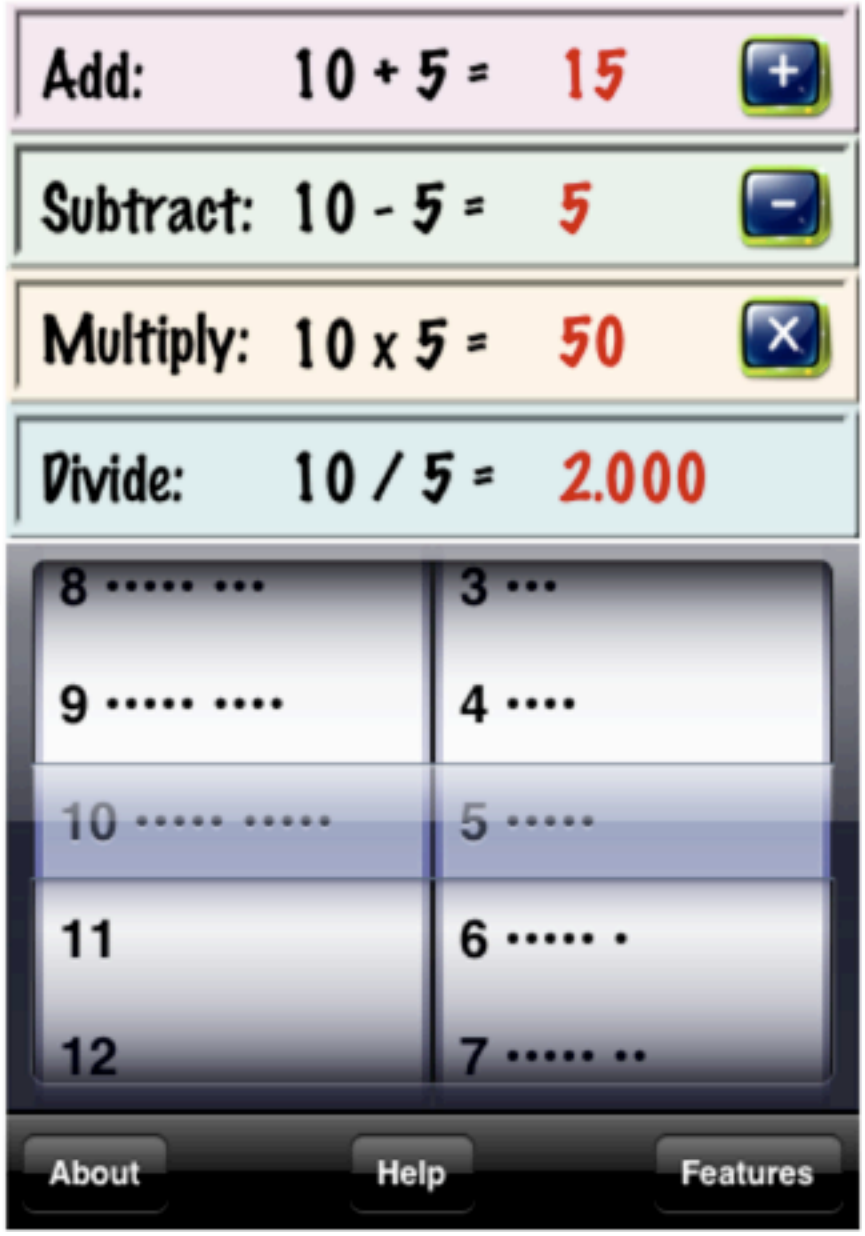
Whiteboard Design



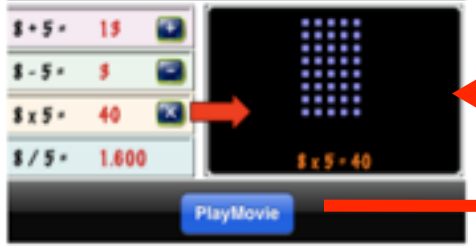
Startup

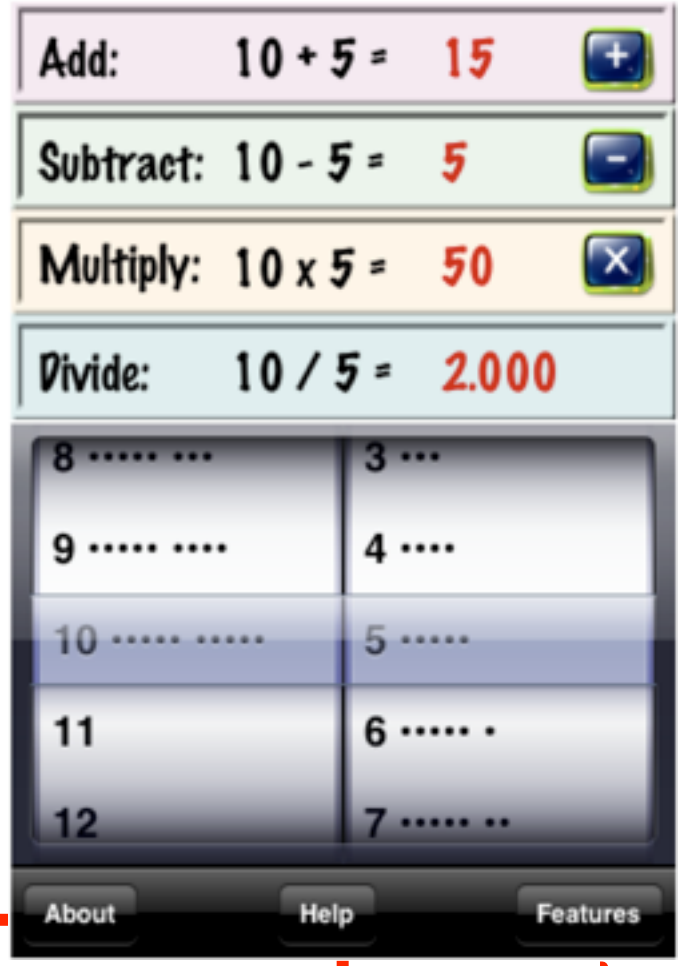
Icon

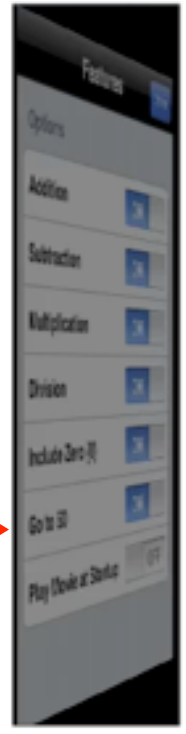
Splash Screen



Toolbar Buttons

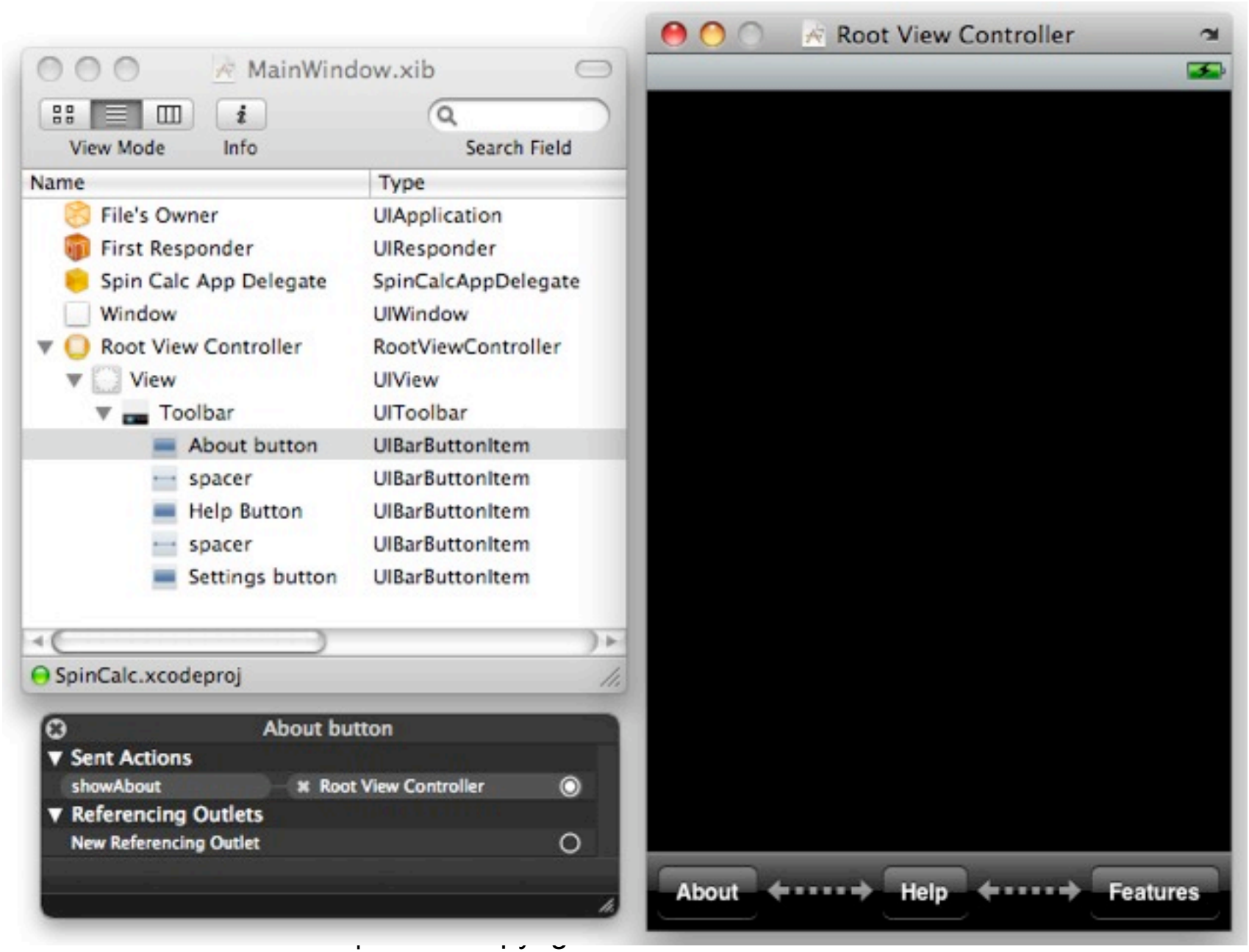




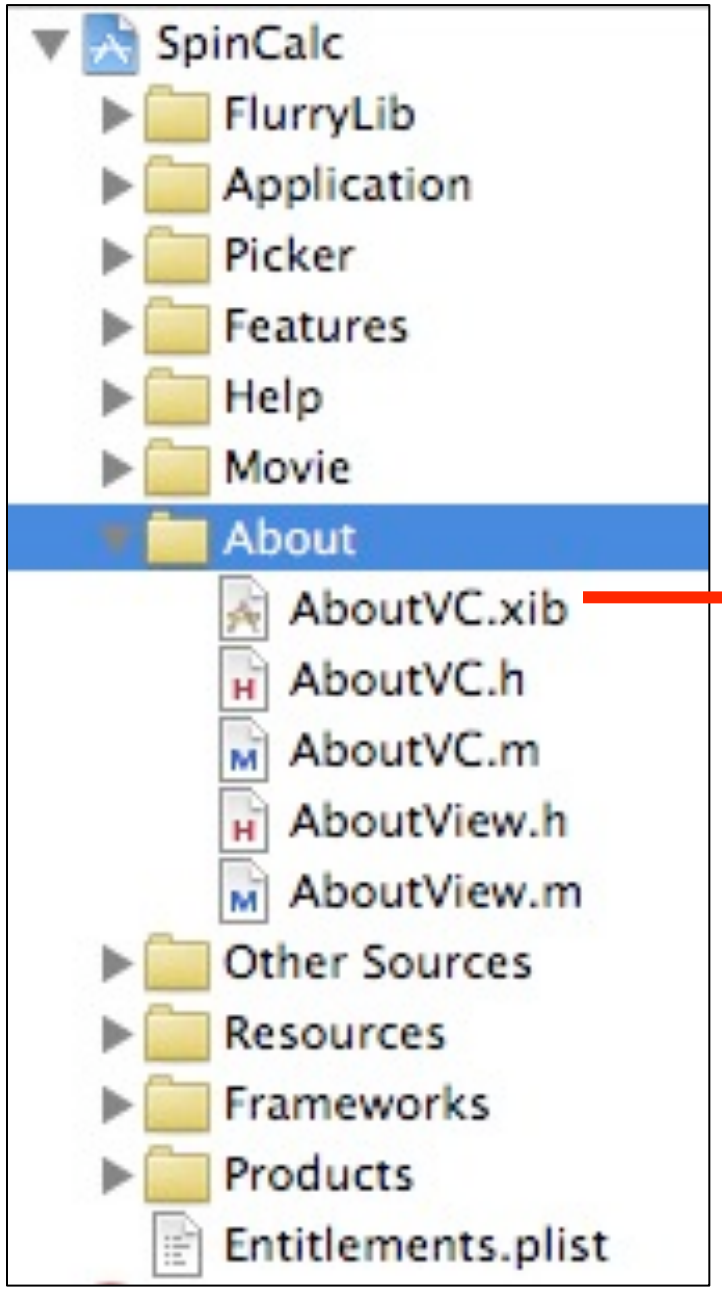


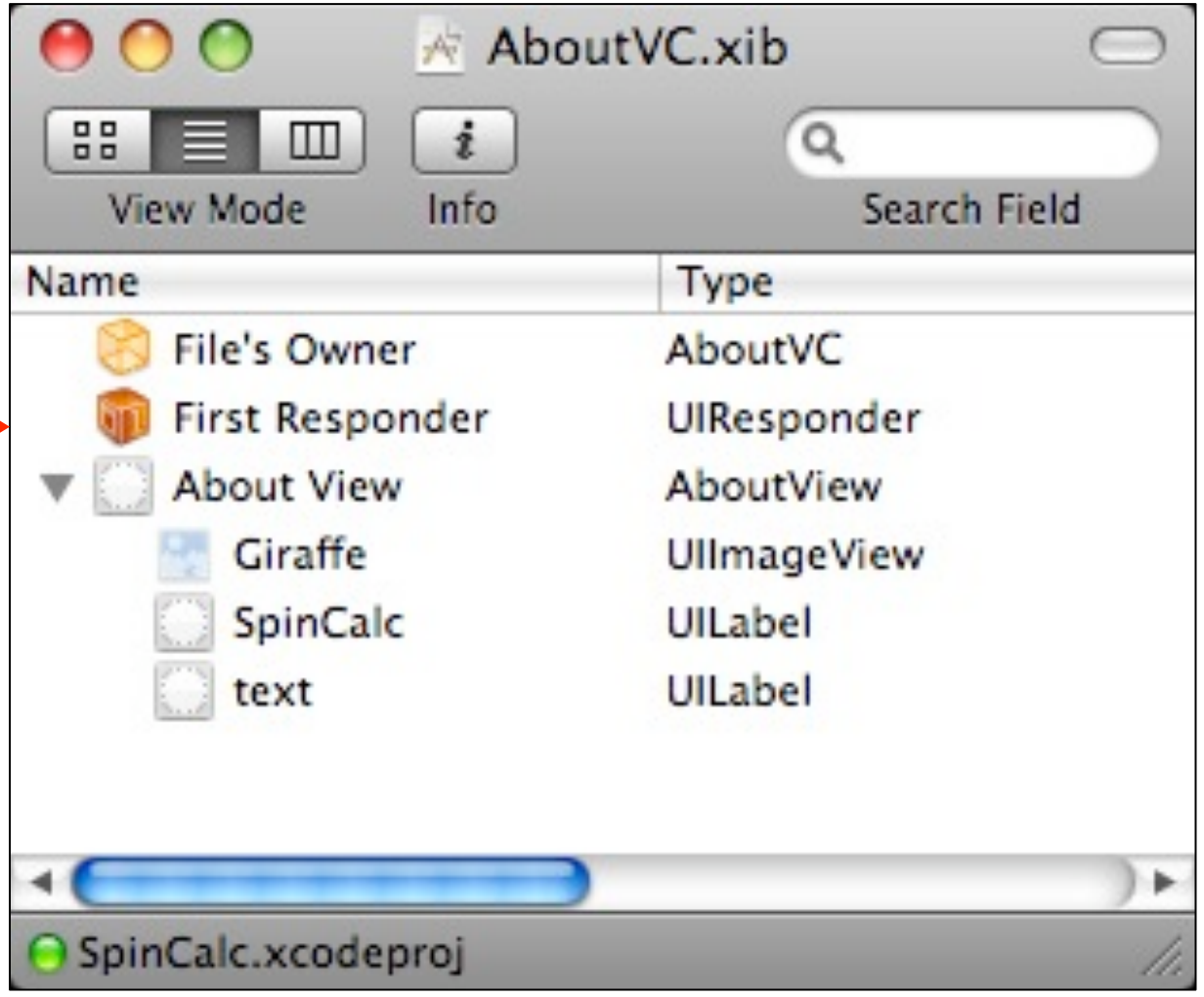
Intro Movie

IB: Toolbar Design

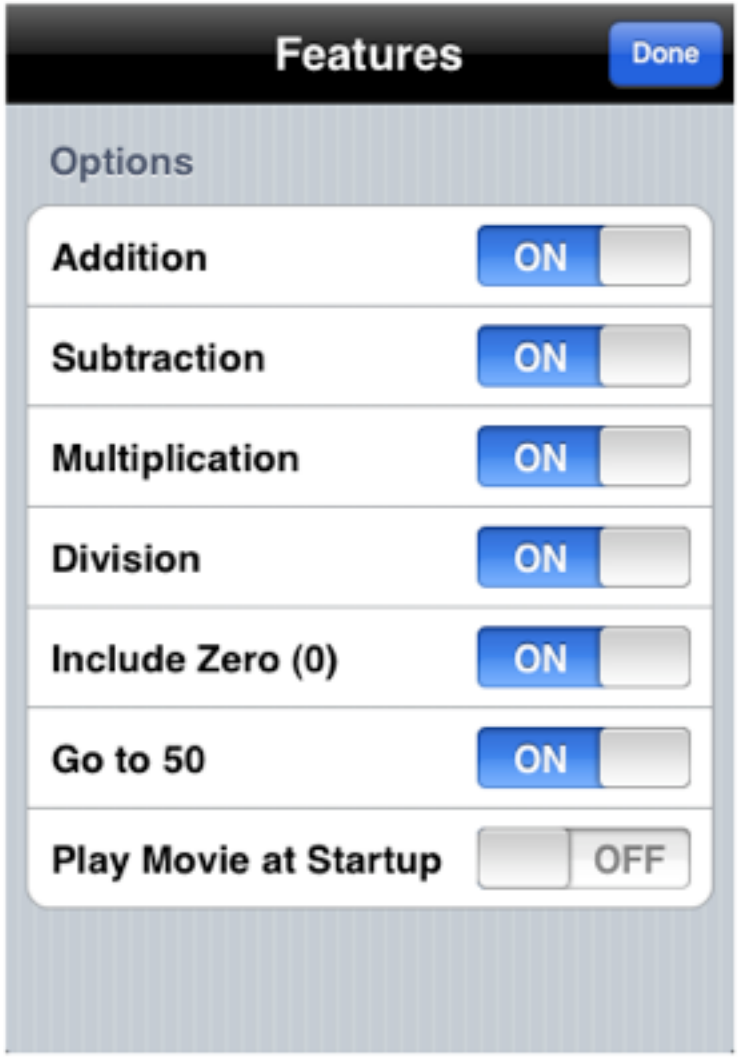


AboutVC.xib



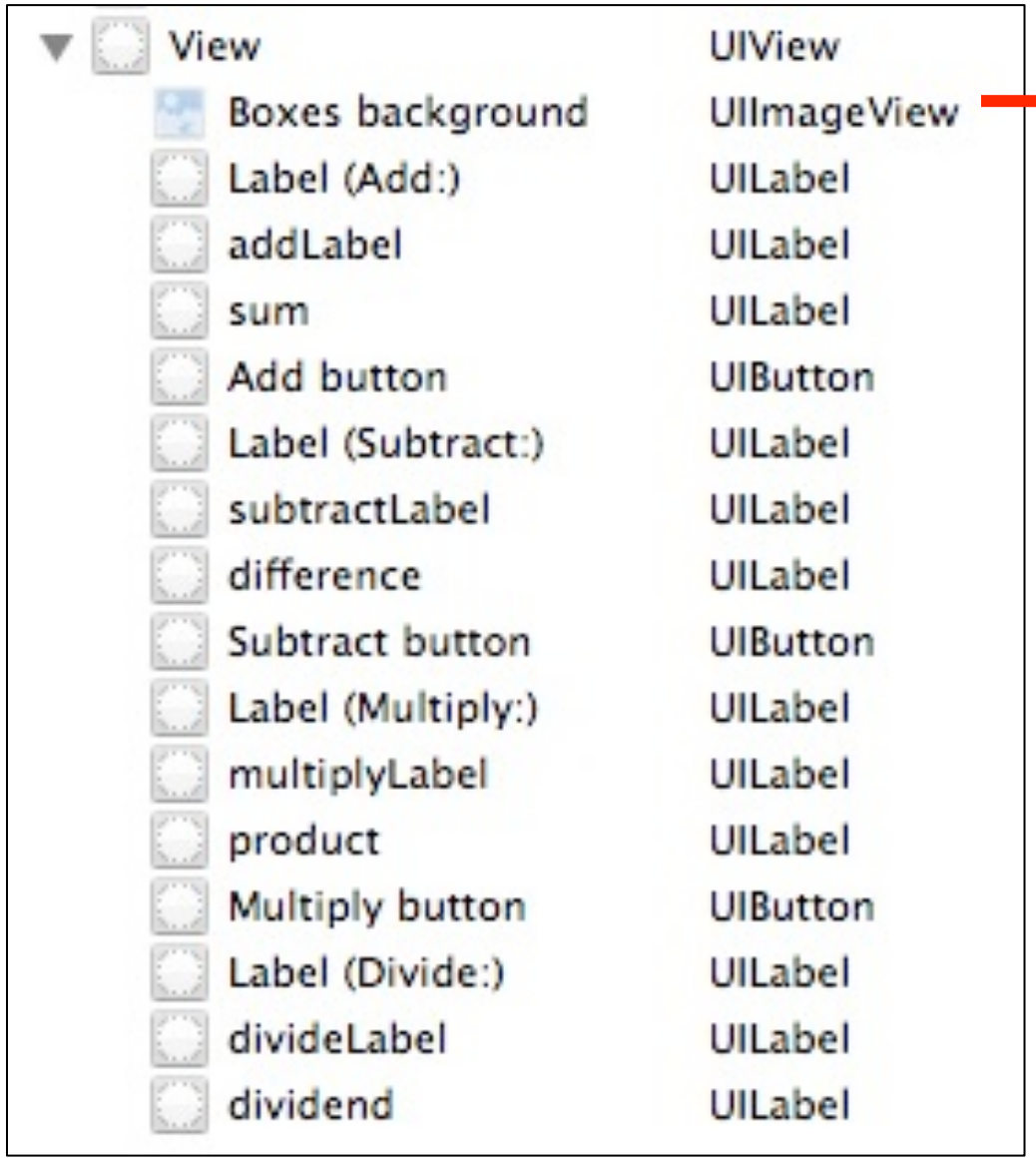


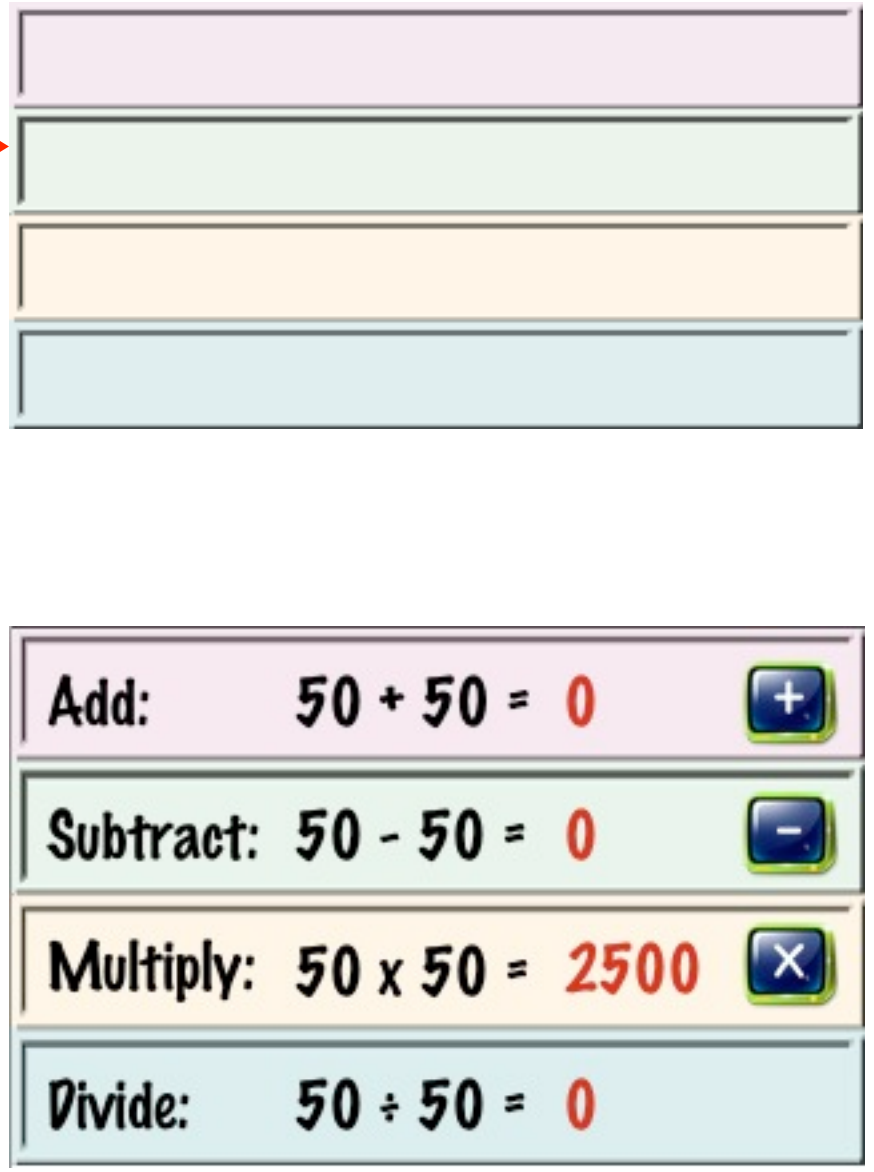
Features



- Get "flipping" behavior using the Utility project template.
- Settings are stored in NSUserDefaults
  - iPhone database
  - Each app has a section
- Screen is a UITableView

IB: Main Results View

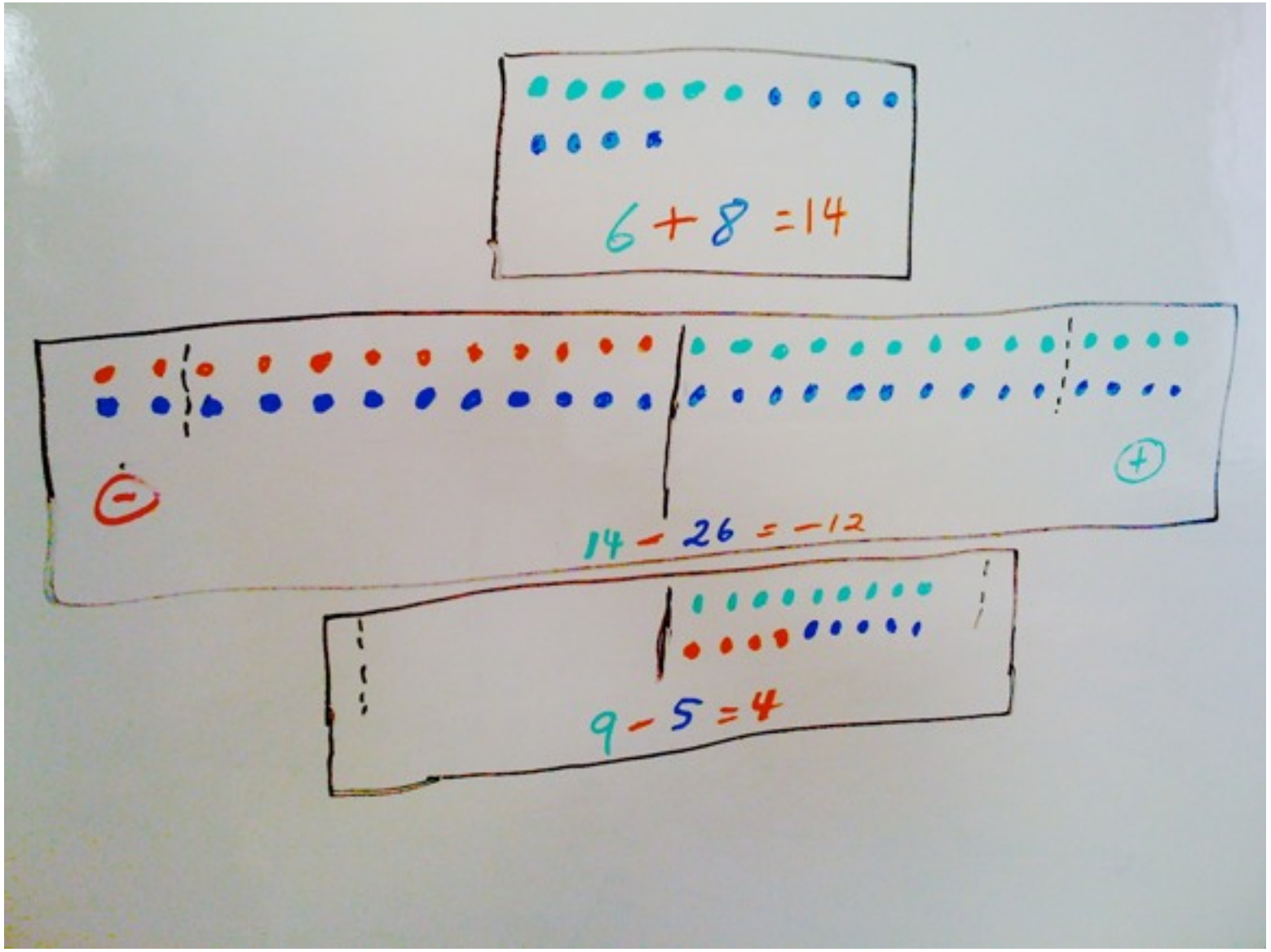




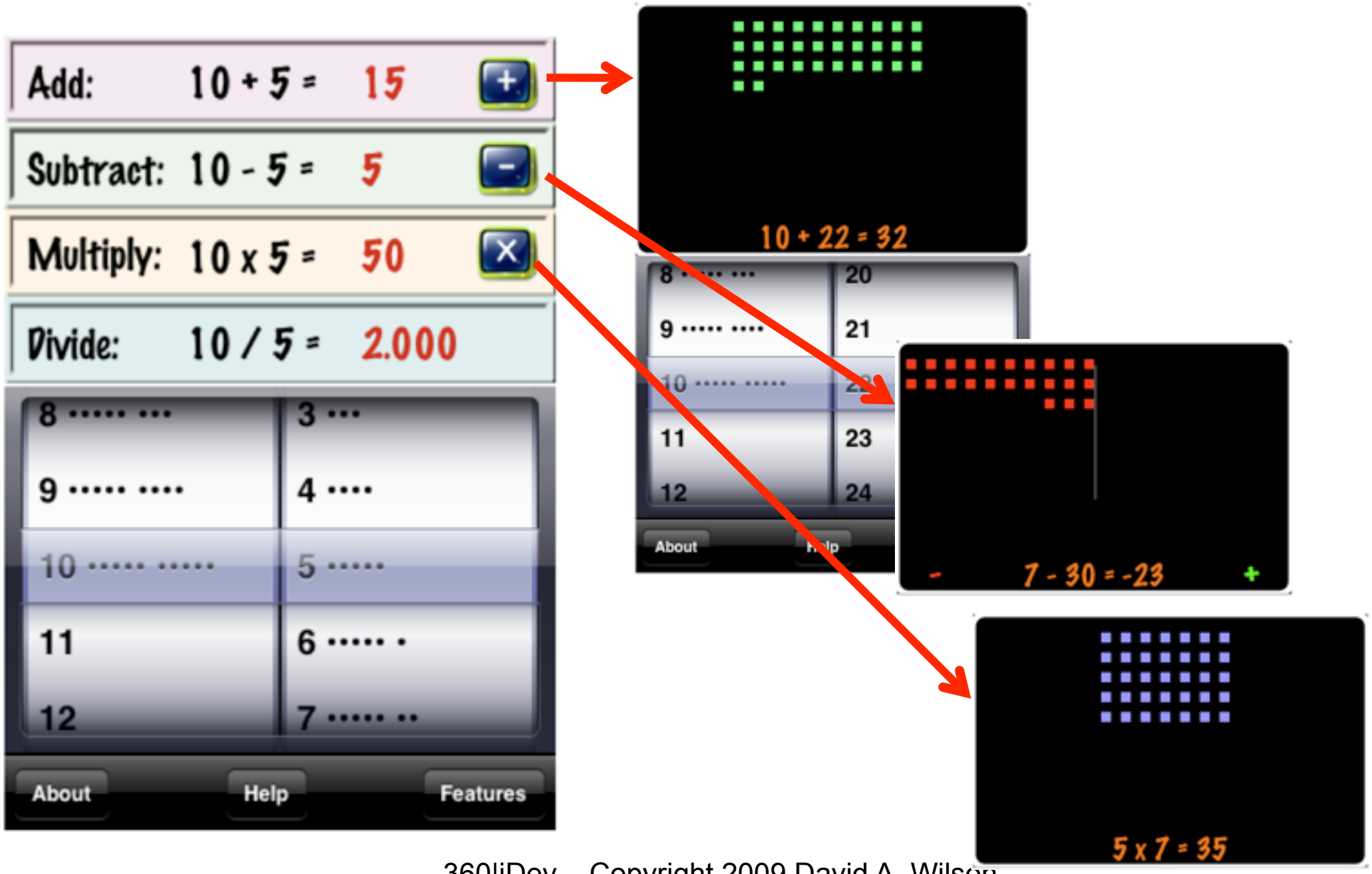
New Requirements

- The requirements always evolve!
- Need a TV Mode
- Show "objects" to represent quantities
- Better explain what the math operations really mean

Back to the Whiteboard



"TV" Modes

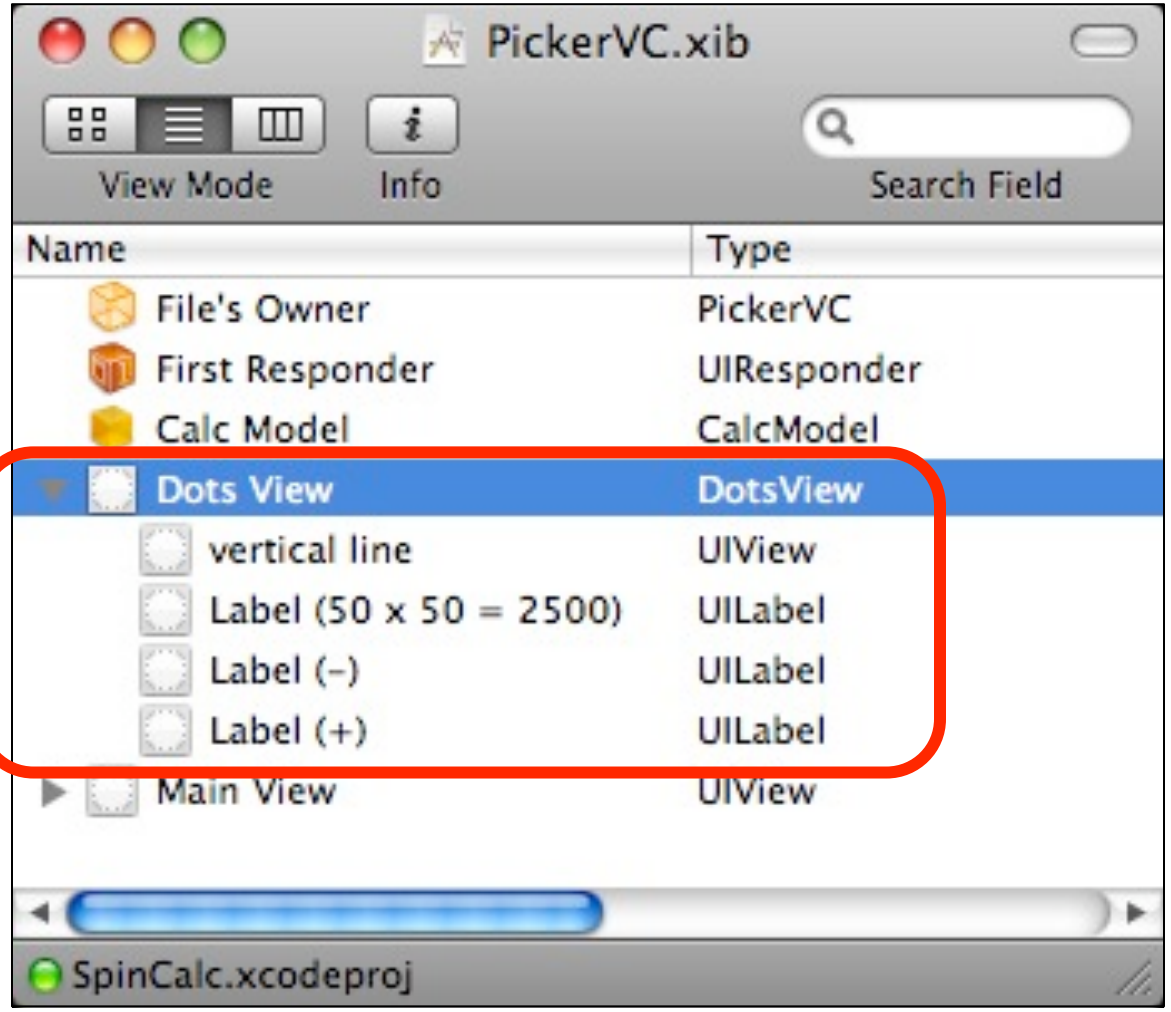


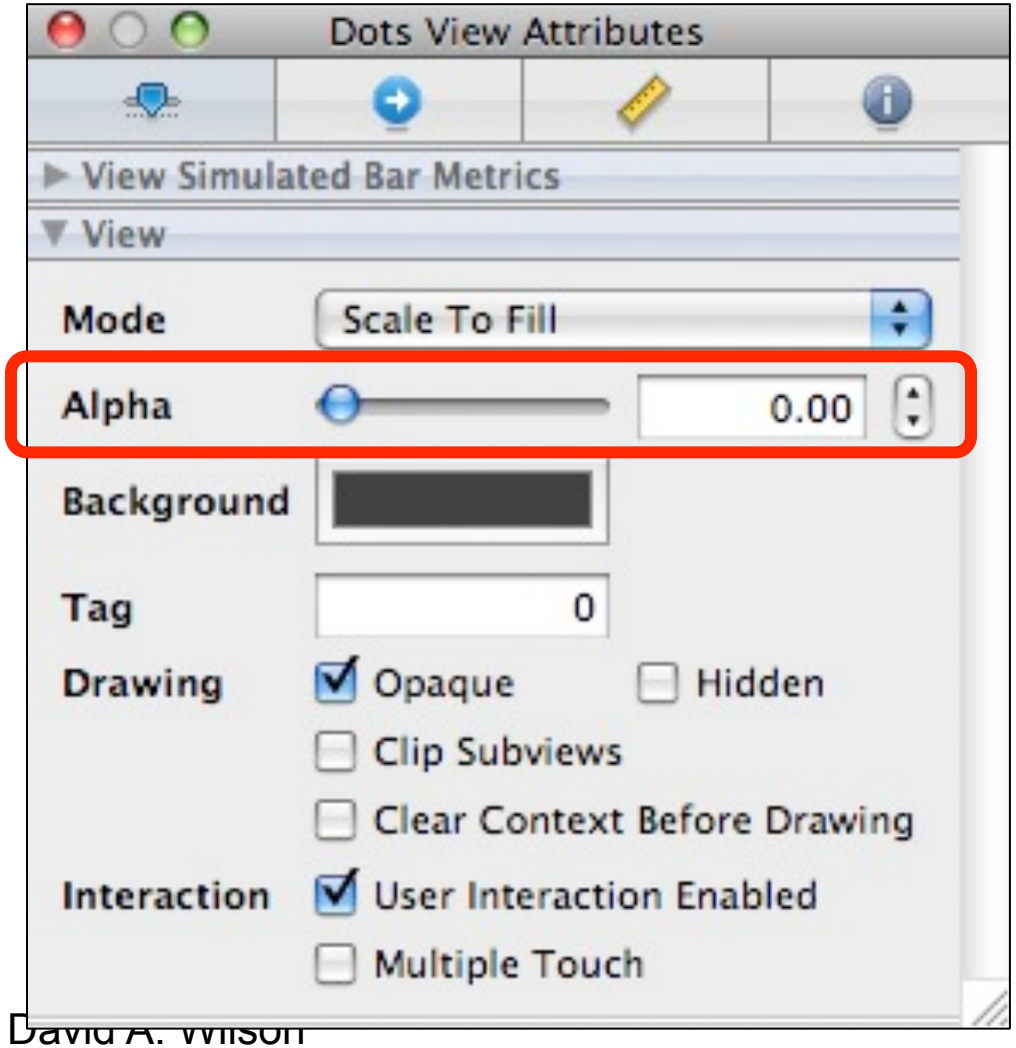
What To Do With Dots View?

- Button press displays DotsView.
  - Plus button sets DrawMode = plus
  - Minus button sets DrawMode = minus
  - Multiply button sets DrawMode = multiply
- Touch makes it disappear.

IB: Dots View Always There

- Initially invisible: alpha = 0.0





Xcode: DotsView (TV)

- TV screen appear when a button is pressed
  - Core Animation used when view appears
  - DotsView alpha set to equal 1.0 (opaque)
- Dots drawn with Quartz drawing library
- DotsView set to transparent when touched.

Xcode: Show Dots View

- Make DotsView appear in ½ second
- Simple example of Core Animation

```
-(IBAction)showDots {
  [UIView beginAnimations:nil
                  context:NULL];
  [UIView setAnimationDuration:0.50];
  dotsView.alpha = 1.0;
  [UIView commitAnimations];
```


Xcode: Hide Dots View

- Make DotsView disappear in ½ second
- Simple example of Core Animation

```
- (void)touchesBegan:(NSSet *)touches
                withEvent:(UIEvent *)event {
   [UIView beginAnimations:nil context:NULL];
   [UIView setAnimationDuration:0.50];
   self.alpha = 0.0;
   [UIView commitAnimations];
}
```

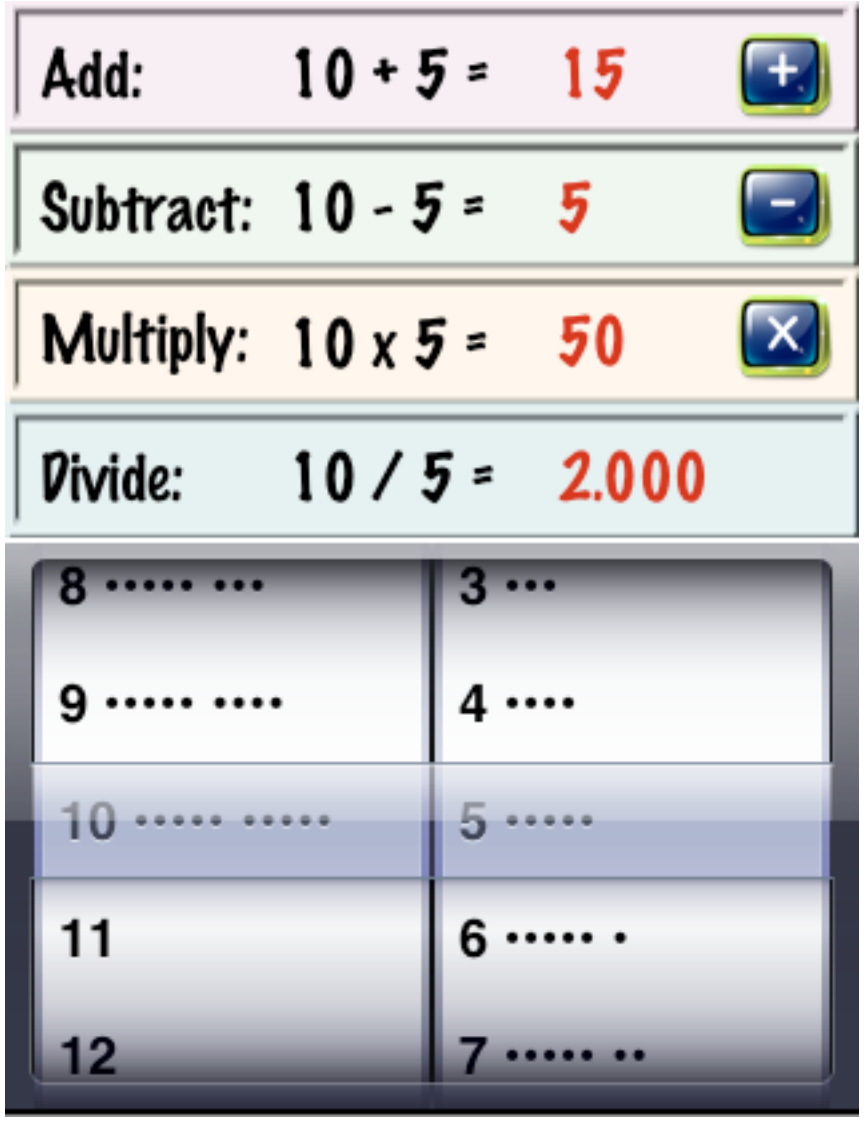

Xcode: Draw DotsView w/Quartz

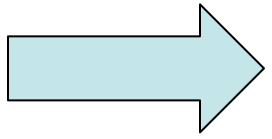
```
- (void)drawRect:(CGRect)rect {
    CGContextRef context = UIGraphicsGetCurrentContext();
    [self drawRoundedCornerRect: context];    // TV "Frame"
    switch (mode) {
        case add:CGContextSetRGBStrokeColor(context, 0.5, 1.0, 0.5, 1.0);
                [self drawAddDots:context];
                break;
        case subtract:
                [self drawSubtractDots:context];
                break;
        case multiply:CGContextSetRGBStrokeColor(context, 0.6, 0.6, 1.0, 1.0);
                [self drawMultiplyDots:context];
                break;
        }
}
```


New Requirements

- Explain addition more clearly
  - One wheel green
  - Other wheel blue
  - Show addition as blue dots + green dots
  - Color code the text
- Use consistent color scheme throughout
- More visual impact

Extreme Makeover





Dots View

Movie on Help Screen

- Record simulator in action
  - iShowU
  - ScreenFlow
- $20 USB microphone for voice over
  - Type out a script
- Use QuickTime Pro
  - Rotate movie 90 degrees to left
  - Export for iPhone (m4v)

SpinCalc Analysis

- Components Used
  - UIPickerView + Custom picker cell view
  - UIToolbar + UIBarButtonItems
  - UILabels
  - UIButtons
  - Custom DotsView
  - UIImageViews

Flurry Analytics

- Added the Flurry library to SpinCalc
- Can track number of users.
- Could track event statistics.
- About 250 KB

```
[FlurryAPI startSession:@"P…S"];
```


Lessons Learned

- Constant evolution
- Beta testers necessary
  - Crucial details/features suggested by testers
- Keep list of features to add
- Redesigned graphics multiple times
  - Splash screen
  - About screen
  - Main screen
  - TV screens

Lesson Learned: Graphics

- Spent more time tweaking the look-and-feel than on the code.

