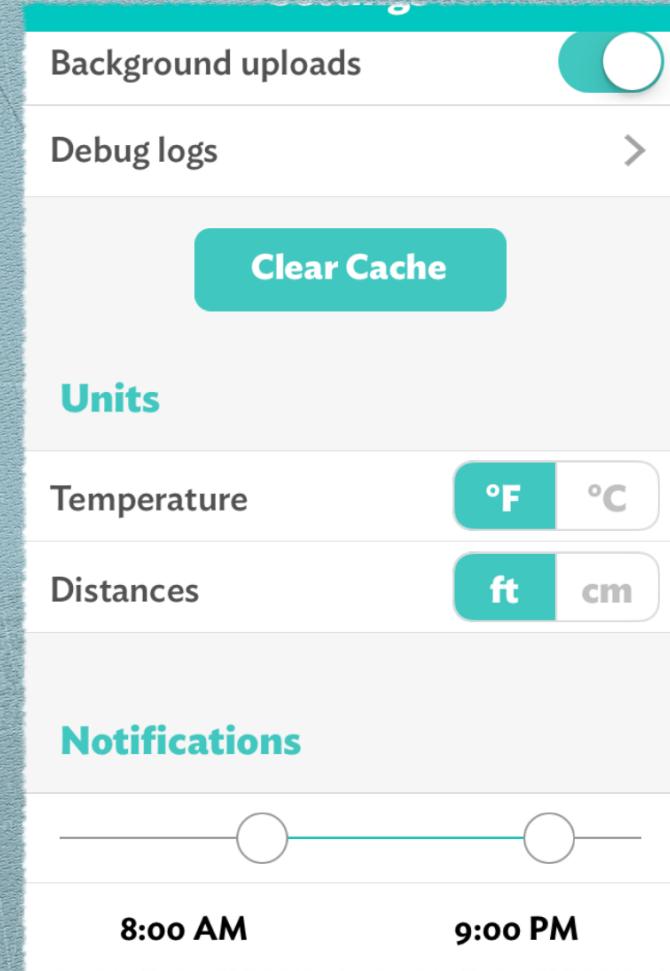
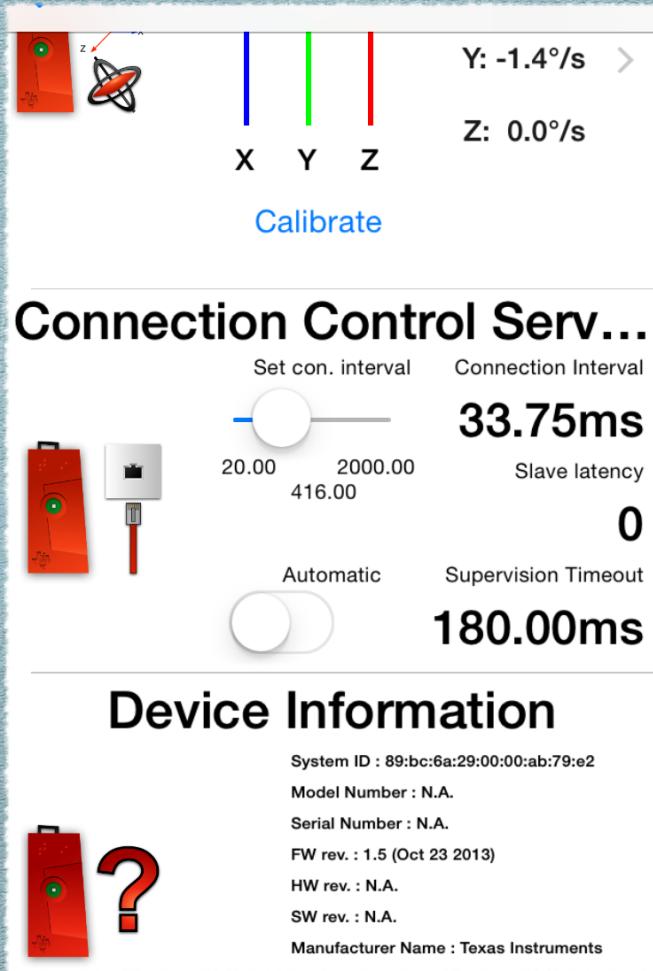


Do We Overuse UICollectionView & UITableView?



TableViews? Really??

There is nothing “tabular” about either model being presented.

When is efficient inefficient?

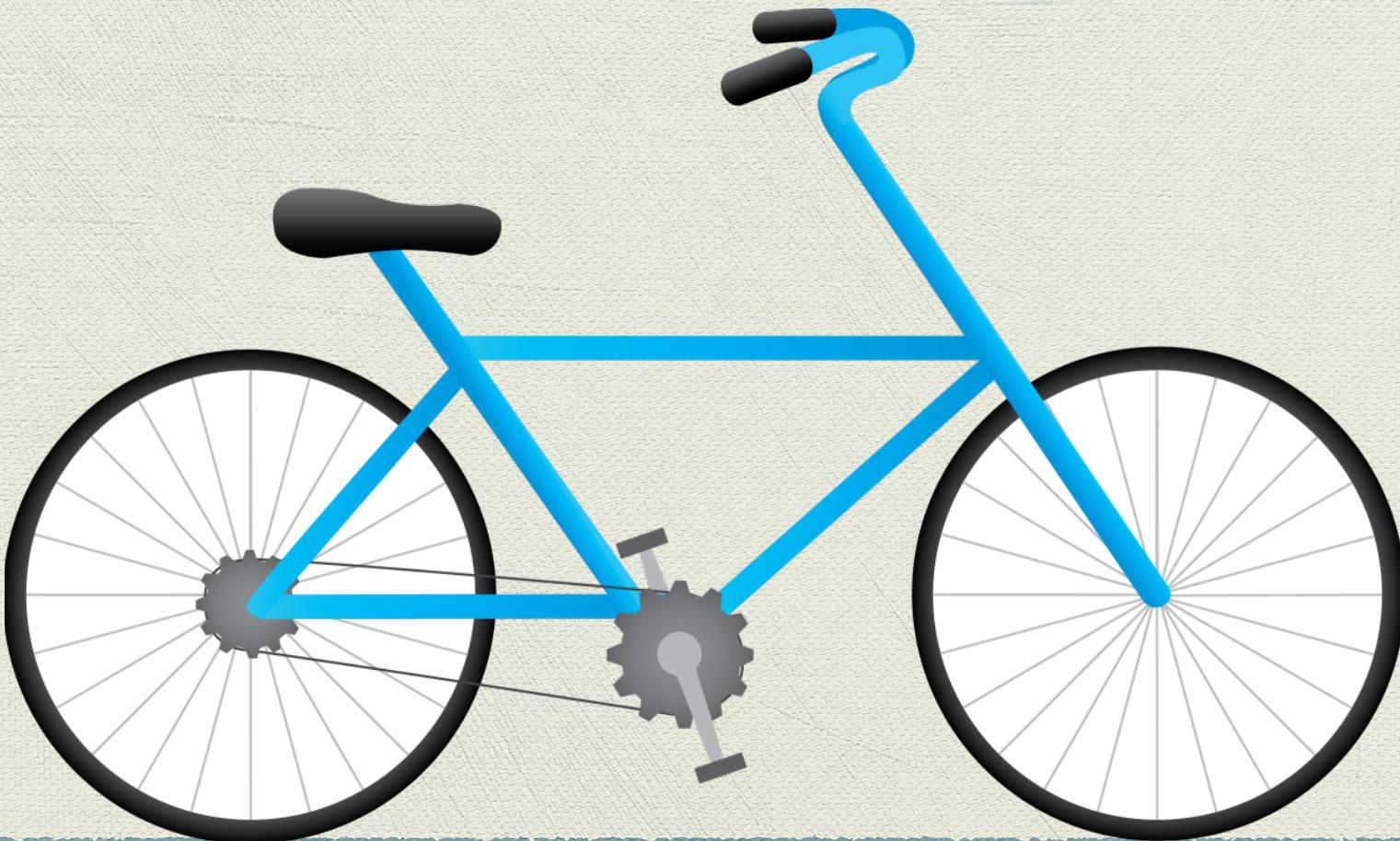
UIView:

Using plain UIView and drawRect:



UILabel:

One step up from drawRect: for text

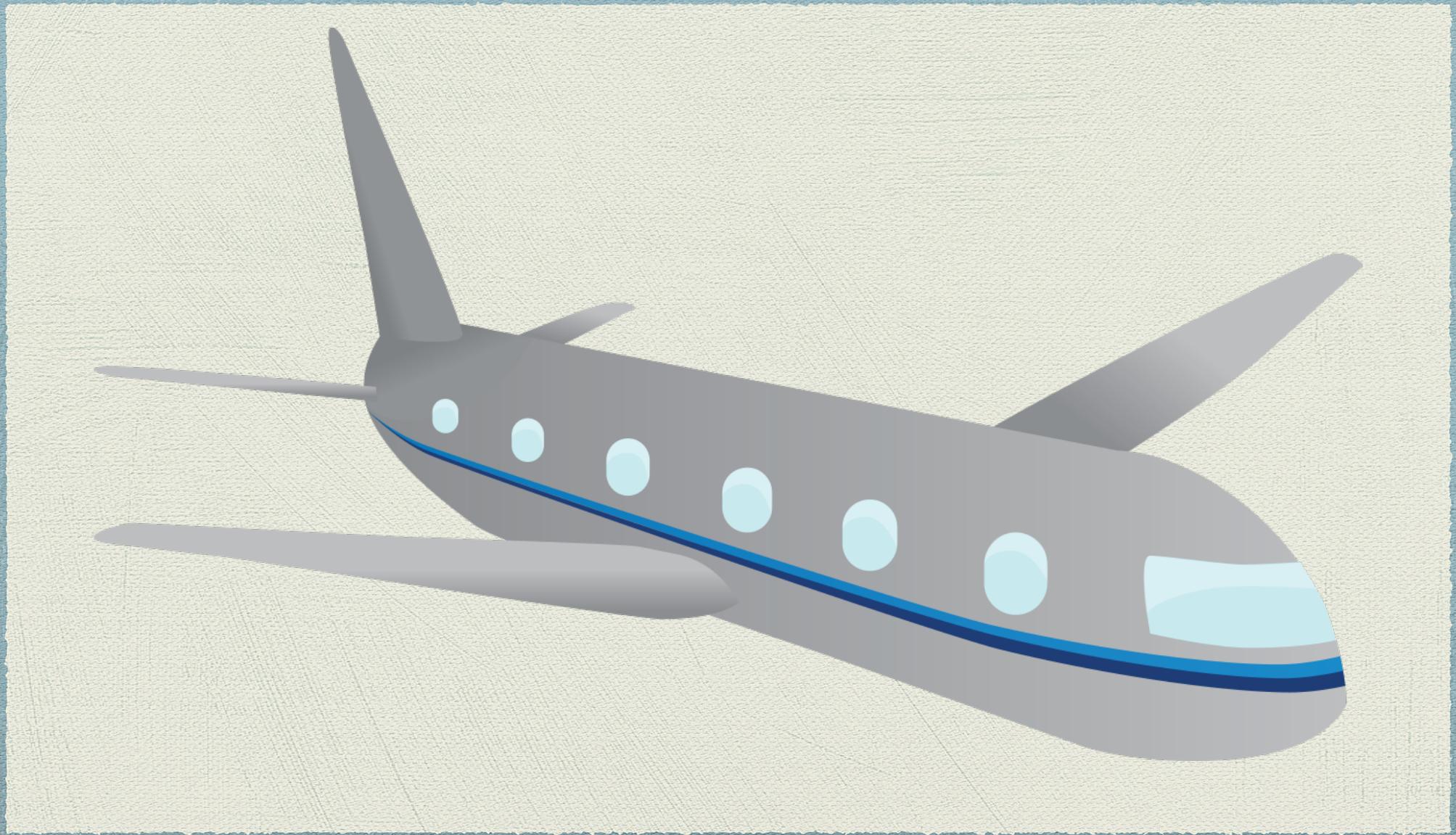


Pick one....



UICollectionView:

The Airbus of views



What if they all look the same?
How do you pick the appropriate view?



UICollection UITable pros & cons

Pros

- ◆ Very efficient for large lists
- ◆ Built by Apple
- ◆ Ready to scroll
- ◆ View does not know about model

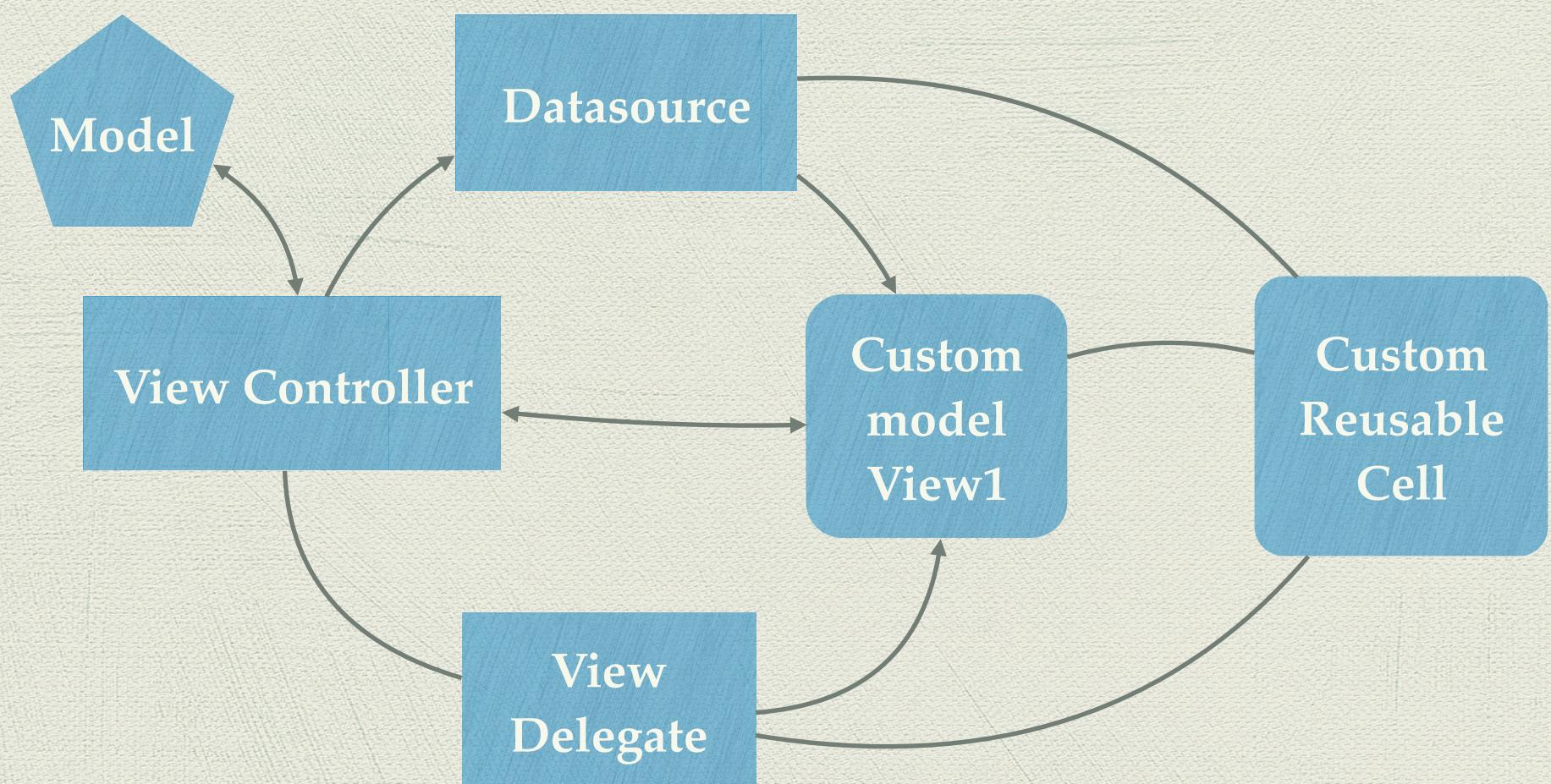
Cons

- ◆ Lots of moving parts
- ◆ Based on ScrollView
- ◆ Encourages Controller to modify model
- ◆ View does not know about model
- ◆ Poor view re-use with IB

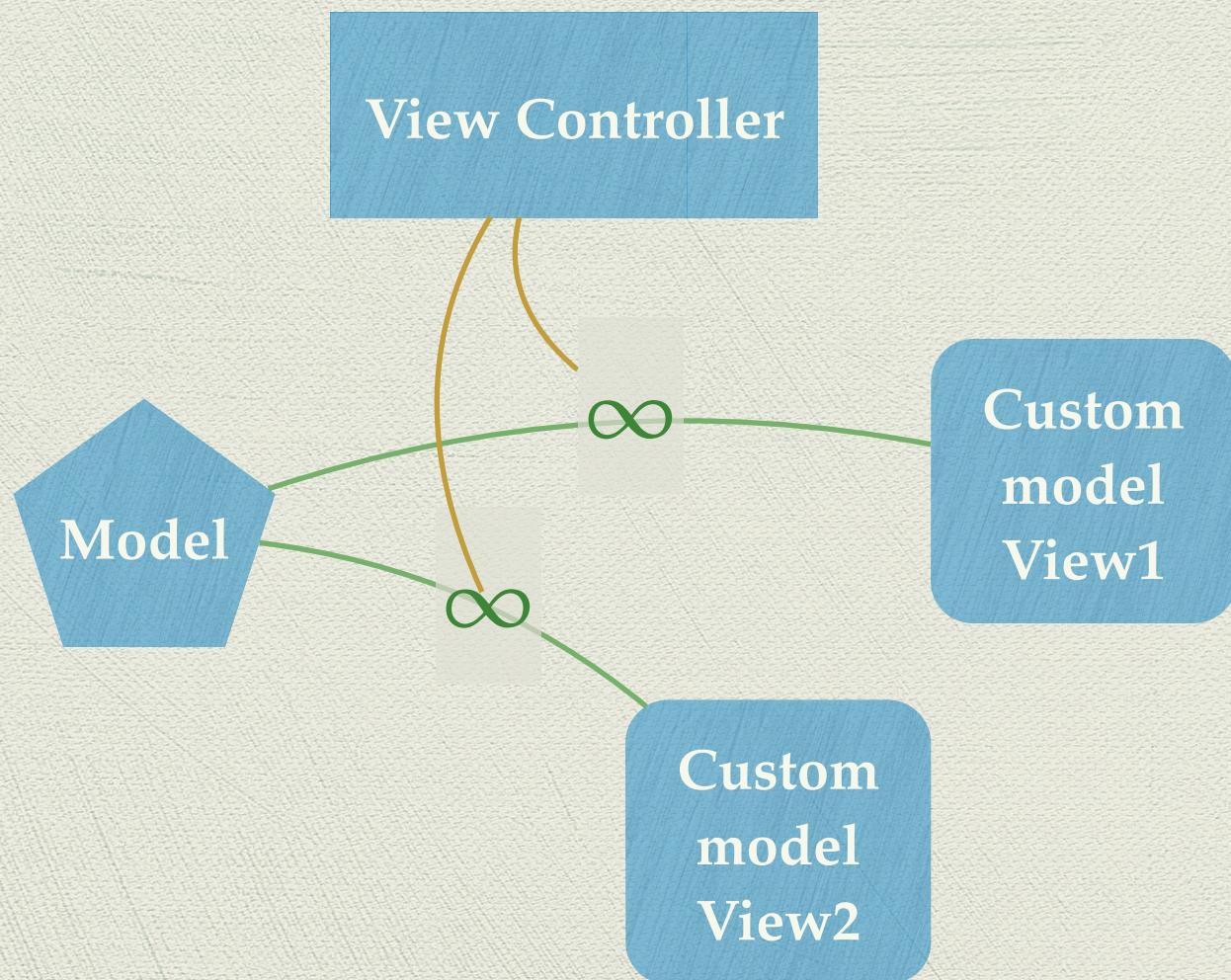
Model View Controller?

Part 2

Cocoa Model Controller View



Smalltalk MVC



Smalltalk MVC pros & cons

Pros

- ◆ View knows about model
- ◆ Less code
- ◆ Fewer moving parts
- ◆ Independent of ScrollView
- ◆ Re-usable views
- ◆ Efficient on a small scale

Cons

- ◆ View knows about model
- ◆ Possibly Less efficient for large lists
- ◆ Blank canvas in IB

IB_Designable

Part 3

IB_DESIGNABLE

+

IBInspectable

+

Autolayout

=

AWESOME! (but buggy)

IB_DESIGNABLE +IBInspectable+Autolayout

Pros

- ◆ Allows seeing the view design in IB!
- ◆ Encourages lightweight Smalltalk style MVC
- ◆ Model Views can easily be used in different places.
- ◆ Less autolayout stress
- ◆ Easy to create tables and flows

Cons

- ◆ Still unstable
- ◆ Can't really design view in IB

Demo

<http://vimeo.com/115025604>


```
-(void) calcConstraintConstantsForIndex: (NSUInteger) index {
    NSUInteger itemsPerLine = self.itemsPerLine;
    NSUInteger widthMargin = _justify ? (self.bounds.size.width - (itemsPerLine * _tileWidth)) / (itemsPerLine-1) : _tileMargin;
    NSUInteger lineNumber = floorf((float)index/(float)itemsPerLine);
    NSInteger hOffset = self.outlineMargin + (_tileWidth + widthMargin) * (index - lineNumber*itemsPerLine);
    NSInteger vOffset = (lineNumber==0 && _showOutline) ? self.outlineMargin : lineNumber*(_tileWidth+_tileMargin);
    MDBListConstraints* listItemConstraints = self.itemConstraints[index];
    listItemConstraints.hConstraint.constant = hOffset;
    listItemConstraints.vConstraint.constant = vOffset;
}
```

IB_DESIGNABLE tips

- ◆ IB_DESIGNABLE compiler info at end of Report Navigator tool list.
- ◆ Recommend you don't have an IB tab and IB_DESIGNABLE class tab open simultaneously
- ◆ Look for 100% “IBDesignablesAgentCocoaTouch” process and Kill if Xcode goes spin crazy

Model View Controller Links

- ◆ Technical Note TN2154 - UIScrollView And Autolayout
developer.apple.com/library/ios/technotes/tn2154/_index.html
- ◆ Excellent summary of MVC called Whats a Controller Anyway by Kyle Brown
<http://c2.com/cgi/wiki?WhatsControllerAnyway>
- ◆ Model View Controller History by Mitchell Model
<http://c2.com/cgi/wiki?ModelViewControllerHistory>
- ◆ Model View Controller background information by Randy Stafford
<http://c2.com/cgi/wiki?ModelModelViewController>