iOS Mobile Development
Today

Finish Animation Demo
Less tippy, guided drops.

Autolayout
How to make device autorotation easy(er).
And make your View Controller work in different environments (i.e. with different bounds).

Autolayout Demo
Making Attributor autorotate properly.
More Dropit
Less tippy!
Guiding the fall of drops.
If time permits, gridding using collision delegate (if not, will post code).

What to look for today ...
UIDynamicItemBehavior (basically physics configuration)
UIAttachmentBehavior
Adding an action block to a behavior
Observing the behavior of items (elapsed animation time, linear velocity, etc.)
UICollisionBehavior's collisionDelegate
Autolayout

Setting UIView frames using **rules** rather than **numbers**
Why? Because many things affect the size of the area available to put views …
  - Rotation
  - 4 inch versus 3.5 inch iPhone
  - Embedding Controller’s Views inside other Controllers (tab bars, navigation controllers, etc.)
We need these rules to put the views in their place no matter what bounds are available.
We call these rules “constraints”.
There is a very powerful API (NSLayoutConstraint) for doing this, but …

We almost always set up these rules in Xcode 5 graphically
So this is all best shown with some screen shots …
Let’s start with two objects, Thing 1 and Thing 2. They are UILabels, but they could be any UIView.

They have been dragged out here without using the blue guidelines.
It'd be great to get a Preview of what this will look like when we run in various autorotations!

Turns out you can do exactly that in Xcode 5 using its Preview feature.

Think of Preview as just another “document” in your project.

You can open it up and put it wherever you want (Assistant, main editing window, separate window, etc.).
Click here to bring up a mini-navigator menu.

A cool trick is to hold down CTRL and SHIFT while clicking on a file to open ...
... a little window will appear asking you where you want to put this file.
Let's put it in the Assistant Editor.
Thing 1

Assistent Editor with Preview.

Thing 2

Thing 1

Thing 2

View Controller
Preview lets you pick the orientation ...
Uh oh! No Thing 2!
Let's rotate back to Portrait.
You can also pick tall vs. short iPhone.
Thing 1 and Thing 2 are both staying stuck to the origin (upper left) and not adapting to the changes in size of their superview.
Close Assistant Editor.
It is also possible to preview Landscape mode in Xcode while editing. It’s not exactly the same layout as running it, but it’s pretty close.

Just select a scene ...
... then, in its Attributes Inspector, choose Orientation Landscape.
Thing 2 is exactly where it was before (relative to the upper left origin). But that's now off-screen.
Let’s go back to Portrait.

“Inferred” means inferred from the context of the Controller (e.g. with tab bars, navigation controllers, etc. shown). By default, Portrait is inferred.
Whew, Thing 2 is indeed still there!
Let's say we want Thing 1 and Thing 2 to stick to their nearby corner (i.e. to stick to that corner no matter where the corner moves to).

We can communicate that to Xcode by dragging to that corner and letting the blue guidelines appear.
Ditto for Thing 2.
Now let's try Landscape again.
Still doesn't work because the blue guidelines are not enough. We have to tell iOS that we want the blue guidelines to be used to create some “constraints” on our layout.
How do we tell Xcode to invent these constraints which will keep our views in the spots implied by the blue guidelines?

Using this little button here ...
... to ask Xcode 5 to suggest constraints.

The “Suggested” constraints are usually very good as long as you use blue guidelines. Always think twice before varying from the Suggested guidelines (maybe even go back and redo blue guidelines?).

The top part of this menu works on an individual view whereas the bottom half works on all the views in the Controller's View.
Nothing looks any different...

But let's click on Thing 1 to see what happened.
And also click on the Size Inspector. That's where all constraints are shown for a view.

This first constraint constrains Thing 1's left (leading) edge to its superview's leading edge (separated by the "default" distance).

This second constraint constrains it to the default distance from the top of its superview.

Xcode knew to add these particular constraints because we used the blue guidelines!
Let's look at Thing 2 ...

Thing 2 has similar constraints, but to the right and bottom edges.

You can actually see the constraints as little blue beams.

Blue (vs other colors) constraints mean “constraints with no apparent problems.”

These beams are selectable and inspectable and you can delete a constraint by hitting delete (so be careful).
Let's check out Landscape ...
Victory!

Thing 1

Thing 2
Let's see what happens if we don't use blue guidelines ...

Excellent.

Let's see what happens if we don't use blue guidelines ...

Excellent.

Thing 1

Thing 2
Here's a “Bad Thing” that was dragged out and sized without the blue guidelines.

It’s supposed to be in the middle of the View but, again, no blue guidelines were used, so it’s a little off.
Let's try to set its constraints to the Suggested constraints.
Xcode tried its best, but these constraints are very bad because they all have "magic numbers" in them (e.g. 62, 89, 68, 198).

It is usually the wrong thing to have a constraint with a magic number in it. Especially if text is involved.
Also, if we try Landscape ...

Bad Thing
... the Bad Thing will not stay anywhere near the “center”.
Okay, back to Portrait.
If a view has bad constraints, you can clear them out at any time using this menu item.
Let's add some constraints to Bad Thing in a different way (i.e. not using blue guidelines and Suggested constraints).

One way to do that is with this button which is used to line up a view with other views or with its superview.
We're going to pick both the Horizontal and Vertical Centering options ("in Container" means in our superview). If you pick 2 (or more) views at once (using shift-click), you can also align them in all these ways.
Clicking here adds the 2 new constraints.
It added them!

Notice that they are drawn in yellow. This is because they don’t match what is currently showing in the scene.
That fact is also reported here ...
We have not talked much about the Document Outline, but it is awesome! It shows everything (views, gestures, constraints, etc.) in your storyboard in outline form. You can select objects here and also ctrl-drag to/from them!

Let’s click on this!

Click here to show the Document Outline.
Yellow problems are generally mismatches between what’s showing in the scene and what the constraints would do. Click on the yellow triangle to resolve a problem. The dashed yellow line shows what the constraints think this view’s frame should be.
Here are the choices to resolve the mismatch.

Since we're happy with our constraints...

...we'll choose to Update Frame to change the storyboard to match the constraints.
Bingo!

No more yellow constraints.
Okay, Landscape again.

Okay, Landscape again.

Bad Thing
And back.
Thing 1

Bad Thing

Thing 2
You can click on a constraint directly.

And change things about it. We're not going to get into that level of detail at this point, though.
If you hit DELETE, a selected constraint will be removed!

That has caused a serious problem here, though. You can tell because of this red circle in the Document Outline.

Let's click on that to see what's up ...
"Need constraints for: X position"

Indeed, there is no way for the autolayout system to know where to put Bad Thing horizontally now.
Luckily, we can just click on this red circle ...

... and Xcode will offer to fix it for us!
The constraint lines are back to being blue (not yellow or red).
What if we change our minds and want Bad Thing to sit on top of Thing 2?

We can just pick it up and drag it to where we want with blue guidelines.
However, this will NOT change the constraints.
We could clear constraints here ... or we can delete them individually. Here's another way to do that.
Now we want to constraint Bad Thing to stay on top of Thing 2.

Let's do that yet a third way (i.e. not with blue guidelines/Suggested and not with a menu at the bottom).
If you want a view to be constrained by another view’s size or position, just ctrl-drag between them.
You will then be asked how you want them constrained. You can pick multiple ways.
Here we'll keep the two views a fixed distance apart (constrained Vertical Spacing), and ...
Keep their right edges aligned at all times.
Notice new constraints.
Let's try Landscape now...
Yay!

Thing 1

Yay!

Bad Thing

Thing 2
... and back.

Bad Thing

Thing 1

Thing 2
In addition to ctrl-dragging between two views, you can ctrl-drag from a view to its superview ...
Or even a view to itself (if you want to constrain its width or height, for example).
This is all just the tip of the iceberg for Autolayout, but hopefully it will get you started!

And we’ve definitely covered everything you should need for your homework.
Demo

Attributor Autorotation

Since we dragged to blue guidelines, it’s mostly going to be automatic.
But there are a couple of things to fix up.
And we’ll make a couple of changes too.