Layout and Animation Techniques For WatchKit

Session 216

Miguel Sanchez WatchKit Engineer
Tom Witkin WatchKit Engineer
Agenda
Agenda

Layout Fundamentals
Agenda

Layout Fundamentals

Using Groups
Agenda

Layout Fundamentals
Using Groups
Existing Animations in watchOS 1
Agenda

Layout Fundamentals
Using Groups
Existing Animations in watchOS 1
New Animation API in watchOS 2
Layout Fundamentals
WatchKit Layout Model
WatchKit Layout Model

Same model as WatchKit in watchOS 1
WatchKit Layout Model

Same model as WatchKit in watchOS 1
Different from UIKit and AppKit
WatchKit Layout Model

Same model as WatchKit in watchOS 1
Different from UIKit and AppKit
Flow-based layout
WatchKit Layout Model

Flow-based layout

- **Group**: A container that manages the layout of other items.
- **Table**: Displays one or more rows of data.
- **Image**: Displays a static or animated image.
- **Separator**: A line for separating content in your interface.
- **Button**: A tappable area with a title and/or image.
- **Switch**: A control for indicating a binary value.
- **Slider**: A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker**: A control for selecting an item from a list.
- **Label**: Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

- Group - A container that manages the layout of other items.
- Table - Displays one or more rows of data.
- Image - Displays a static or animated image.
- Separator - A line for separating content in your interface.
- Button - A tappable area with a title and/or image.
- Switch - A control for indicating a binary value.
- Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
- Picker - A control for selecting an item from a list.
- Label - Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

Group - A container that manages the layout of other items.
Table - Displays one or more rows of data.
Image - Displays a static or animated image.
Separator - A line for separating content in your interface.
Button - A tappable area with a title and/or image.
Switch - A control for indicating a binary value.
Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
Picker - A control for selecting an item from a list.
Label - Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

10:09

Image

- Group - A container that manages the layout of other items.
- Table - Displays one or more rows of data.
- Image - Displays a static or animated image.
- Separator - A line for separating content in your interface.
- Button - A tappable area with a title and/or image.
- Switch - A control for indicating a binary value.
- Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
- Picker - A control for selecting an item from a list.
- Label - Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

Image

- Group: A container that manages the layout of other items.
- Table: Displays one or more rows of data.
- Image: Displays a static or animated image.
- Separator: A line for separating content in your interface.
- Button: A tappable area with a title and/or image.
- Switch: A control for indicating a binary value.
- Slider: A control for selecting a floating-point value from a range of continuous or discrete values.
- Picker: A control for selecting an item from a list.
- Label: Displays a static text string.
WatchKit Layout Model

Flow-based layout

Interface Controller

- Image
- Button

Group - A container that manages the layout of other items.
Table - Displays one or more rows of data.
Image - Displays a static or animated image.
Separator - A line for separating content in your interface.
Button - A tappable area with a title and/or image.
Switch - A control for indicating a binary value.
Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
Picker - A control for selecting an item from a list.
Label - Displays a static text string.
WatchKit Layout Model
Flow-based layout

Interface Controller

Image

Button

Group - A container that manages the layout of other items.
Table - Displays one or more rows of data.
Image - Displays a static or animated image.
Separator - A line for separating content in your interface.
Button - A tappable area with a title and/or image.
Switch - A control for indicating a binary value.
Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
Picker - A control for selecting an item from a list.
Label - Displays a static text string.
WatchKit Layout Model

Groups are containers of elements

- Group - A container that manages the layout of other items.
- Table - Displays one or more rows of data.
- Image - Displays a static or animated image.
- Separator - A line for separating content in your interface.
- Button - A tappable area with a title and/or image.
- Switch - A control for indicating a binary value.
- Slider - A control for selecting a floating-point value from a range of continuous or discrete values.
- Picker - A control for selecting an item from a list.
- Label - Displays a static text string.
WatchKit Layout Model

Groups are containers of elements

- **Group** - A container that manages the layout of other items.
- **Table** - Displays one or more rows of data.
- **Image** - Displays a static or animated image.
- **Separator** - A line for separating content in your interface.
- **Button** - A tappable area with a title and/or image.
- **Switch** - A control for indicating a binary value.
- **Slider** - A control for selecting a floating-point value from a range of continuous or discrete values.
- **Picker** - A control for selecting an item from a list.
- **Label** - Displays a static text string.
WatchKit Layout Model

Groups are containers of elements
WatchKit Layout Programming Model
WatchKit Layout Programming Model

You don’t write object creation code
WatchKit Layout Programming Model

You don’t write object creation code

Fine tuned control of:
WatchKit Layout Programming Model

You don’t write object creation code

Fine tuned control of:

• Layout hierarchy
WatchKit Layout Programming Model

You don’t write object creation code

Fine tuned control of:

• Layout hierarchy
• Alignment and sizing
WatchKit Layout Programming Model

You don’t write object creation code
Fine tuned control of:

• Layout hierarchy
• Alignment and sizing
• Animation
Alignment and Sizing
Alignment and Size
Properties on WKInterfaceObject
Alignment and Size
Properties on WKInterfaceObject

Position
- Horizontal: Left
- Vertical: Center

Size
- Width: Fixed
  - Value: 100
- Height: Fixed
  - Value: 100
Alignment and Size
Properties on WKInterfaceObject

Alignment in containing object
Alignment and Size
Properties on WKInterfaceObject

Alignment in containing object

- Horizontal and vertical
Alignment and Size
Properties on WKInterfaceObject

Alignment in containing object

- Horizontal and vertical
- Left, center, or right
Alignment and Size
Properties on WKInterfaceObject

Alignment in containing object
• Horizontal and vertical
• Left, center, or right

Size
Alignment and Size
Properties on WKInterfaceObject

Alignment in containing object
• Horizontal and vertical
• Left, center, or right

Size
• Width and height heuristic
Alignment and Size
Properties on WKInterfaceObject

Alignment in containing object
• Horizontal and vertical
• Left, center, or right

Size
• Width and height heuristic
• Fixed, relative, or sized to fit
Alignment API
WKInterfaceObject

```swift
func setHorizontalAlignment(WKInterfaceObjectHorizontalAlignment)
func setVerticalAlignment(WKInterfaceObjectVerticalAlignment)

enum WKInterfaceObjectHorizontalAlignment {
    case Left
    case Center
    case Right
}

denum WKInterfaceObjectVerticalAlignment {
    case Top
    case Center
    case Bottom
}
```
Alignment in Containing Element

myObj.setHorizontalAlignment(.Left)
myObj.setVerticalAlignment(.Top)
Alignment in Containing Element

```java
myObj.setHorizontalAlignment(.Center)
myObj.setVerticalAlignment(.Center)
```
Alignment in Containing Element

myObj.setHorizontalAlignment(.Right)
myObj.setVerticalAlignment(.Bottom)
Sizing API

WKInterfaceObject

func setWidth(CGFloat)
func setHeight(CGFloat)

func setRelativeWidth(CGFloat, adjustment : CGFloat)
func setRelativeHeight(CGFloat, adjustment : CGFloat)

func sizeToFitWidth()
func sizeToFitHeight()
Sizing API

WKInterfaceObject

- func setWidth(CGFloat)
- func setHeight(CGFloat)

- func setRelativeWidth(CGFloat, adjustment : CGFloat)
- func setRelativeHeight(CGFloat, adjustment : CGFloat)

- func sizeToFitWidth()
- func sizeToFitHeight()
Fixed Width and Height

myObj.setWidth(100)
myObj.setHeight(100)
Fixed Width and Height

myObj.setWidth(100)
myObj.setHeight(100)

-------------------------------
Interpreting values of zero

watchOS 1 - default to storyboard

watchOS 2 - zero value  NEW
Relative to Container
Relative to Container

myObj.setRelativeWidth(0.75, 0)
Relative to Container

myObj.setRelativeWidth(0.75, 0)
**Relative to Container**

```java
myObj.setRelativeWidth(0.75, 0)
myObj.setRelativeHeight(0.50, 0)
```
Adjustment Adds/Subtracts from Size

```java
myObj.setRelativeWidth(0.75, 30)
```

```java
myObj.setRelativeHeight(0.50, -30)
```
Adjustment Adds/Subtracts from Size

```java
myObj.setRelativeWidth(0.75, 30)
myObj.setRelativeHeight(0.50, -30)
```
Sized-to-Fit Content

```java
myObj.sizeToFitWidth()
myObj.sizeToFitHeight()
```
Sized-to-Fit Content

myObj.sizeToFitWidth()
myObj.sizeToFitHeight()
Sized-to-Fit Content

myObj.sizeToFitWidth()
myObj.sizeToFitHeight()
Group Elements
Fine tuning your layouts
WKInterfaceGroup
WKInterfaceGroup

Container without default content
WKInterfaceGroup

Container without default content
Tool for arranging elements
WKInterfaceGroup

Container without default content

Tool for arranging elements

• Vertical or horizontal flow
WKInterfaceGroup

Container without default content
Tool for arranging elements
• Vertical or horizontal flow
• Nesting (including other groups)
WKInterfaceGroup

Container without default content
Tool for arranging elements
• Vertical or horizontal flow
• Nesting (including other groups)
• Fine control of alignment and sizing
Adding Horizontal Flow
Adding Horizontal Flow
Adding Horizontal Flow
Adding Horizontal Flow
Adding Horizontal Flow
Adding Horizontal Flow
Insets and Spacing
Insets and Spacing

Left Inset (10 points)
Insets and Spacing

Bottom Inset
(10 points)
Insets and Spacing

Top Inset (10 points)
Insets and Spacing

Right Inset (10 points)
Insets and Spacing

Spacing (10 points)
Can Also Be Set at Top Level
Can Also Be Set at Top Level

Left Inset (10 points)
Can Also Be Set at Top Level
Can Also Be Set at Top Level

Spacing (10 points)
Group Nesting for Complex Layouts
Group Nesting for Complex Layouts
Group Nesting for Complex Layouts
Group Nesting for Complex Layouts
Layouts in WKRecipes
Ingredients

- Tortillas
- Fish
- Avocados
- Hot sauce
Let’s Explore Three Layouts

Recipe Layout:
- BBQ Wings
- Fish Tacos
- Pizza
- Salmon

Ingredients Layout:
1. Tortillas
2. Fish
3. Avocados
4. Hot sauce

Servings Layout:
4 servings
<table>
<thead>
<tr>
<th>Recipes</th>
<th>10:09</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBQ Wings</td>
<td></td>
</tr>
<tr>
<td>Fish Tacos</td>
<td></td>
</tr>
<tr>
<td>Pizza</td>
<td></td>
</tr>
<tr>
<td>Salmon</td>
<td></td>
</tr>
</tbody>
</table>
Table Rows

Image

- BBQ Wings
- Fish Tacos
- Pizza
- Salmon
<table>
<thead>
<tr>
<th>Image</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBQ Wings</td>
<td></td>
</tr>
<tr>
<td>Fish Tacos</td>
<td></td>
</tr>
<tr>
<td>Pizza</td>
<td></td>
</tr>
<tr>
<td>Salmon</td>
<td></td>
</tr>
</tbody>
</table>
Layout for Table Row Controller

Image  
Label
Layout for Table Row Controller

- Group with horizontal layout

Image | Label
Insets and Spacing
Insets and Spacing

Left Inset (8 points)
Insets and Spacing

Bottom Inset (8 points)
Insets and Spacing

Top Inset (8 points)
Insets and Spacing

Image  Label

Spacing (8 points)
Alignment
Alignment

Horizontal: Left
Vertical: Center
Sizing
Sizing

Fixed Size
Width 30
Height 30
Sizing

Fixed Size
Width 30
Height 30

Size To Fit
Table Row Layout
Table Row Layout

Groups with horizontal layout
Table Row Layout

Groups with horizontal layout

Fine tune

• Alignment

• Insets and spacing

• Sizing
Layout for Ingredients Controller

- Tortillas
- Fish
- Avocados
- Hot sauce
Layout for Ingredients Controller

- Number inside circle
- Label

Ingredients:
- 10 Tortillas
- 1 Fish
- 2 Avocados
- 5 Hot sauce
Layout for Ingredients Controller
Layout for Ingredients Controller

Group with horizontal layout
Layout for Ingredients Controller

Group with horizontal layout
Layout for Ingredients Controller

Group with horizontal layout
Layout for Ingredients Controller

Nested Group

Label
Groups Can Have Backgrounds
Groups Can Have Backgrounds

Group with blue background and radius of 8
Alignment of Number Label

[Diagram showing a watch interface with an ingredient list and a circle labeled '10' next to a label 'Label'.]
Alignment of Number Label

Centered in group

10 Label
Insets and Spacing
Insets and Spacing

Left Inset: 10
Insets and Spacing

Bottom Inset
Insets and Spacing
Insets and Spacing
Ingredients Controller Layout

- Tortillas
- Fish
- Avocados
- Hot sauce
Ingredients Controller Layout

Nested groups

- Tortillas
- Fish
- Avocados
- Hot sauce
Ingredients Controller Layout

Nested groups
Groups with background
Ingredients Controller Layout

Nested groups

Groups with background
  • Color
  • Images

- Tortillas
- Fish
- Avocados
- Hot sauce
Layout for Servings Controller
Layout for Servings Controller

Circular layout
Layout for Servings Controller

Circular layout

Label
Layout for Servings Controller

4
Three Top-Level Groups
Three Top-Level Groups
Three Top-Level Groups
Three Top-Level Groups
Group Alignment and Sizing
Group Alignment and Sizing

Horizontal: Center
Fixed Size
Group Alignment and Sizing

Horizontal: Center
Fixed Size

Width relative to 100% of container
Second Group Has Nested Groups
Second Group Has Nested Groups
Second Group Has Nested Groups

4
Second Group Has Nested Groups
Second Group Has Nested Groups
Alignment Inside Groups
Alignment Inside Groups
Alignment Inside Groups

Horizontal: Right
Vertical: Top
Alignment Inside Groups

- Horizontal: Right
  Vertical: Top

- Horizontal: Left
  Vertical: Center
Alignment Inside Groups

- Horizontal: Right, Vertical: Top
- Horizontal: Left, Vertical: Center
- Horizontal: Right, Vertical: Bottom
Ingredients Controller Layout
Ingredients Controller Layout

Group nesting for complex layouts
Ingredients Controller Layout

Group nesting for complex layouts
Don’t abuse power of groups
Implications of Complex Layouts
Implications of Complex Layouts

WatchKit has no APIs for direct element creation
Implications of Complex Layouts

WatchKit has no APIs for direct element creation
What you describe in IB is created
Implications of Complex Layouts

WatchKit has no APIs for direct element creation
What you describe in IB is created
Hidden objects
Implications of Complex Layouts

WatchKit has no APIs for direct element creation
What you describe in IB is created
Hidden objects
• Creation cost
Implications of Complex Layouts

WatchKit has no APIs for direct element creation

What you describe in IB is created

Hidden objects

• Creation cost
• Save on layout cost
Using Images in Layouts
Using Images in Layouts

Don’t ignore transfer costs
Using Images in Layouts

Don’t ignore transfer costs

• watchOS 1 apps run extension on iPhone
Using Images in Layouts

Don’t ignore transfer costs

- watchOS 1 apps run extension on iPhone
- watchOS 2 apps need to install extension
Using Images in Layouts

Don’t ignore transfer costs
• watchOS 1 apps run extension on iPhone
• watchOS 2 apps need to install extension
Use appropriate sizes
Using Images in Layouts

Don’t ignore transfer costs

• watchOS 1 apps run extension on iPhone
• watchOS 2 apps need to install extension

Use appropriate sizes

Image slicing can accommodate various sizes
Animations

Tom Witkin
WatchKit Engineer
Existing Methods of Animation
Tables and animated images
Tables
Certain updates already animate
Tables

Certain updates already animate

Insert rows
Tables
Certain updates already animate

Insert rows
Remove rows
Tables

Certain updates already animate

Insert rows
Remove rows
Update row content
Ranking

Alphabetical
Alphabetical Ranking
Tables
Insert and remove rows

[NSIndexSet indexSetWithIndex:0]
\[NSIndexSet indexSetWithIndex:0\]
\begin{verbatim}
[self.recipeTable insertRowsAtIndexes:[NSIndexSet indexSetWithIndex:0]
 withRowType:@"status";]

StatusRowController *rc = [self.recipeTable rowControllerAtIndex:0];
[rc setText:text];

self.statusRowTimer = [NSTimer scheduledTimerWithTimeInterval:2.0 target:self
 selector:@selector(hideStatusRow) userInfo:nil repeats:NO];
\end{verbatim}
Tables

Insert and remove rows

```objective-c
[self.recipeTable insertRowsAtIndexes:[NSIndexSet indexSetWithIndex:0]
withRowType:@"status"];

StatusRowController *rc = [self.recipeTable rowControllerAtIndex:0];
[rc setText:text];

self.statusRowTimer = [NSTimer scheduledTimerWithTimeInterval:2.0 target:self
selector:@selector(hideStatusRow) userInfo:nil repeats:NO];
```
Tables
Insert and remove rows

[self.recipeTable insertRowsAtIndexes:[NSIndexSet indexSetWithIndex:0] withRowType:@"status"];  

StatusRowController *rc = [self.recipeTable rowControllerAtIndex:0]; 
[rc setText:text];

self.statusRowTimer = [NSTimer scheduledTimerWithTimeInterval:2.0 target:self selector:@selector(hideStatusRow) userInfo:nil repeats:NO];
Insert and remove rows

[self.recipeTable insertRowsAtIndexes:[NSIndexSet indexSetWithIndex:0] withRowType:@"status"];

StatusRowController *rc = [self.recipeTable rowControllerAtIndex:0];
[rc setText:text];

self.statusRowTimer = [NSTimer scheduledTimerWithTimeInterval:2.0 target:self selector:@selector(hideStatusRow) userInfo:nil repeats:NO];
Tables
Insert and remove rows
Tables

Insert and remove rows

Insert or remove rows of any type
Tables

Insert and remove rows

Insert or remove rows of any type

To avoid animations

```swift
setRowTypes()
setNumberOfRows(_: withRowType:)
```
Tables

Insert and remove rows

Insert or remove rows of any type

To avoid animations

```swift
setRowTypes()
setNumberOfRows(_: withRowType:)
```
Celebrate with delicious fish tacos....
Tap for more...

- Servings
- Ingredients
Celebrate with delicious fish tacos....
Tap for more...

Servings
Ingredients
Groups as Spacing Elements
Groups as Spacing Elements

Table Cell
Groups as Spacing Elements
Groups as Spacing Elements
Groups as Spacing Elements

shortDescriptionLabel

moreLabel
Groups as Spacing Elements

- shortDescriptionLabel
- moreLabel
- fullDescriptionLabel
Groups as Spacing Elements
Groups as Spacing Elements

fullDescriptionLabel

Hidden Objects

shortDescriptionLabel

moreLabel
DescriptionRowController *rc = [self.table rowControllerAtIndex:0];
[rc.fullDescriptionLabel setHidden:NO];
[rc.shortDescriptionLabel setHidden:YES];
[rc.moreLabel setHidden:YES];
Tables
Reload row content
Tables

Reload row content

Rows reload when content changes in height
Tables

Reload row content

Rows reload when content changes in height
Make sure rows have size to fit height
Tables

Reload row content

Rows reload when content changes in height
Make sure rows have size to fit height
Animated Images
Animated Images

Cycle through a series of images
Animated Images

Cycle through a series of images
Repeat and reverse animations
Animated Images

Cycle through a series of images
Repeat and reverse animations
Optimizing images is important
• Reduce size and number of images
WKInterfacePicker

30%
WKInterfacePicker
Related Session

WatchKit In-Depth, Part 2

WWDC15 Videos
Animation API
Animatable Properties
Animatable Properties

Opacity
Animatable Properties

- Opacity
- Width / height
Animatable Properties

- Opacity
- Width / height
- Alignment
Animatable Properties

- Opacity
- Width / height
- Alignment
- Background color
Animatable Properties

- Opacity
- Width / height
- Alignment
- Background color
- Color / tint color
Animatable Properties

- Opacity
- Width / height
- Alignment
- Background color
- Color / tint color
- Group insets
New API

WKInterfaceController

func animateWithDuration(duration: NSTimeInterval, animations: () -> Void)

func animateWithDuration(duration: NSTimeInterval, animations: () -> Void)
Animation Examples
Celebrate with delicious fish tacos. For the engineers that want to add a bit of spice to their life.
Celebrate with delicious fish tacos. For the engineers that want to add a bit of spice to their life.
want to add a bit of spice to their life.
want to add a bit of spice to their life.
Sequential Animations

for (NSInteger i = 0; i < self.outerGroups.count; i++) {

    WKInterfaceGroup *group = self.outerGroups[i];

    dispatch_time_t time = dispatch_time(DISPATCH_TIME_NOW, (int64_t)delay);
    dispatch_after(time, dispatch_get_main_queue(), ^{

        [self animateWithDuration:duration animations:^{
            [group setAlpha:alpha];
        }];

    });

}
for (NSInteger i = 0; i < self.outerGroups.count; i++) {

    WKInterfaceGroup *group = self.outerGroups[i];

    dispatch_time_t time = dispatch_time(DISPATCH_TIME_NOW, (int64_t)delay);
    dispatch_after(time, dispatch_get_main_queue(), ^{

        [self animateWithDuration:duration animations:^{
            [group setAlpha:alpha];
        }];

    });

}
Sequential Animations

for (NSInteger i = 0; i < self.outerGroups.count; i++) {

    WKInterfaceGroup *group = self.outerGroups[i];

    dispatch_time_t time = dispatch_time(DISPATCH_TIME_NOW, (int64_t)delay);
    dispatch_after(time, dispatch_get_main_queue(), ^{

        [self animateWithDuration:duration animations:^{
            [group setAlpha:alpha];
        }];;

    });

}
Sequential Animations

for (NSInteger i = 0; i < self.outerGroups.count; i++) {

    WKInterfaceGroup *group = self.outerGroups[i];

    dispatch_time_t time = dispatch_time(DISPATCH_TIME_NOW, (int64_t)delay);
    dispatch_after(time, dispatch_get_main_queue(), ^{

        [self animateWithDuration:duration animations:^{
            [group setAlpha:alpha];
        }];

    });

}
New API

WKInterfaceController

- (void)didAppear;
- (void)willDisappear;
New API

WKInterfaceController

- (void)didAppear;
- (void)willDisappear;

*Coming in a future watchOS 2 seed*
Sequential Animations
Sequential Animations
Sequential Animations

Use `didAppear`, not `willActivate`
Sequential Animations

Use `-didAppear`, not `-willActivate`

Stagger animations using timers or GCD
Sequential Animations

Use `didAppear`, not `willActivate`

Stagger animations using timers or GCD

- Interface controller must be active
Sequential Animations

Use **-didAppear**, not **-willActivate**

Stagger animations using timers or GCD

- Interface controller must be active
- Keep total duration short
Sequential Animations

Use `-didAppear`, not `-willActivate`

Stagger animations using timers or GCD

- Interface controller must be active
- Keep total duration short

Set initial animation values in storyboard
Sequential Animations

Use `-didAppear`, not `-willActivate`

Stagger animations using timers or GCD

- Interface controller must be active
- Keep total duration short

Set initial animation values in storyboard
want to add a bit of spice to their life.

Servings

Ingredients

Notes
want to add a bit of spice to their life.
Groups as Spacing Elements
Groups as Spacing Elements
Groups as Spacing Elements

numberBackground
Groups as Spacing Elements

numberBackground

textContent
Groups as Spacing Elements

- numberBackground
- spacerGroup
- textLabel
Groups as Spacing Elements

spacerGroup
Groups as Spacing Elements
Groups as Spacing Elements
Groups as Spacing Elements

numberBackground

textLabel
Groups as Spacing Elements

[self animateWithDuration:0.3 animations:^{

    for (NSInteger i = 0 ; i < self.instructionsTable numberOfRows; i++) {

        IngredientRowController *rowController = [self.instructionsTable rowControllerAtIndex:i];

        [rowController numberBackground setAlpha:1.0];
        [rowController.textLabel setAlpha:1.0];
        [rowController.spacerGroup setWidth:0.0];

    }

}];
Groups as Spacing Elements

[self animateWithDuration:0.3 animations:^{
    for (NSInteger i = 0 ; i < self.instructionsTable.numberOfLines; i++) {
        IngredientRowController *rowController = [self.instructionsTable rowControllerAtIndex:i];

        [rowController.numberBackground setAlpha:1.0];
        [rowController.textLabel setAlpha:1.0];
        [rowController.spacerGroup setWidth:0.0];
    }
}];
Groups as Spacing Elements

[self animateWithDuration:0.3 animations:^{

    for (NSInteger i = 0 ; i < self.instructionsTable.numberOfRows; i++) {

        IngredientRowController *rowController = [self.instructionsTable rowControllerAtIndex:i];

        [rowController.numberBackground setAlpha:1.0];
        [rowController.textLabel setAlpha:1.0];
        [rowController.spacerGroup setWidth:0.0];

    }

}];
Groups as Spacing Elements

```
[self animateWithDuration:0.3 animations:^{

    for (NSInteger i = 0 ; i < self.instructionsTable.numberOfRows; i++) {

        IngredientRowController *rowController = [self.instructionsTable rowControllerAtIndex:i];

        [rowController.numberBackground setAlpha:1.0];
        [rowController.textLabel setAlpha:1.0];
        [rowController.spacerGroup setWidth:0.0];

    }

}];
```
Groups as Spacing Elements
Groups as Spacing Elements
Groups as Spacing Elements

Invisible spacer groups
Groups as Spacing Elements

Invisible spacer groups
Adjust width, height, or alignment
Groups as Spacing Elements

Invisible spacer groups
Adjust width, height, or alignment
Animating will re-layout entire interface
Groups as Spacing Elements

Invisible spacer groups
Adjust width, height, or alignment
Animating will re-layout entire interface
want to add a bit of spice to their life.
want to add a bit of spice to their life.

Servings

Ingredients

Notes
This is tasty.

Watch out! This is spicy.

I need to make this again.
Watch out! This is spicy.
Watch out! This is spicy.
Complex Interface Transitions
Complex Interface Transitions
Complex Interface Transitions

textBubble
Complex Interface Transitions
Complex Interface Transitions

textContainer
Complex Interface Transitions
Complex Interface Transitions
Complex Interface Transitions

[self animateWithDuration:duration animations:^{
    [self.textBubble setRelativeWidth:1.0 withAdjustment:0.0];
    [self.textBubble sizeToFitHeight];
    [self.textBubble setAlpha:1.0];

    [self.confirmationButton setAlpha:1.0];

    [self.textContainer sizeToFitHeight];
}];
Complex Interface Transitions

```swift
[self animateWithDuration:duration animations:^{
    [self.textBubble setRelativeWidth:1.0 withAdjustment:0.0];
    [self.textBubble sizeToFitHeight];
    [self.textBubble setAlpha:1.0];

    [self.confirmationButton setAlpha:1.0];

    [self.textContainer sizeToFitHeight];
}
];
```
Complex Interface Transitions

[self animateWithDuration:duration animations:^{

    [self.textBubble setRelativeWidth:1.0 withAdjustment:0.0];
    [self.textBubble sizeToFitHeight];
    [self.textBubble setAlpha:1.0];

    [self.confirmationButton setAlpha:1.0];

    [self.textContainer sizeToFitHeight];

}];
[self animateWithDuration:duration animations:^{
    [self.textBubble setRelativeWidth:1.0 withAdjustment:0.0];
    [self.textBubble sizeToFitHeight];
    [self.textBubble setAlpha:1.0];

    [self.confirmationButton setAlpha:1.0];

    [self.textContainer sizeToFitHeight];
}];
Complex Interface Transitions

```objc
[self animateWithDuration:duration animations:^{
    [self.textBubble setRelativeWidth:1.0 withAdjustment:0.0];
    [self.textBubble sizeToFitHeight];
    [self.textBubble setAlpha:1.0];

    [self.confirmationButton setAlpha:1.0];

    [self.textContainer sizeToFitHeight];
}];
```
Watch out! This is spicy.
Watch out! This is spicy.
Complex Interface Transitions
Complex Interface Transitions
Complex Interface Transitions

textContainer

confirmationContainer
Complex Interface Transitions

textContainer

confirmationContainer
Complex Interface Transitions

textContainer

confirmationContainer
Complex Interface Transitions

[self animateWithDuration:duration animations:^{

    [self.textContainer setAlpha:0.0];
    [self.textContainer setHeight:0.0];

    [self.confirmationContainer setRelativeHeight:1.0 withAdjustment:0.0];

}];
Complex Interface Transitions

[self animateWithDuration:duration animations:^{
    [self.textContainer setAlpha:0.0];
    [self.textContainer setHeight:0.0];
    [self.confirmationContainer setRelativeHeight:1.0 withAdjustment:0.0];
}];
Complex Interface Transitions

[self animateWithDuration:duration animations:^{
    [self.textContainer setAlpha:0.0];
    [self.textContainer setHeight:0.0];

    [self.confirmationContainer setRelativeHeight:1.0 withAdjustment:0.0];

}];
Complex Interface Transitions

```swift
[self animateWithDuration:duration animations:^{

    [self.textContainer setAlpha:0.0];
    [self.textContainer setHeight:0.0];

    [self.confirmationContainer setRelativeHeight:1.0 withAdjustment:0.0];

};
```
Animation
A few notes
Animation

A few notes

Any update that affects sizing can animate layout
Animation

A few notes

Any update that affects sizing can animate layout

• Example—WKInterfaceLabel text
Animation

A few notes

Any update that affects sizing can animate layout

• Example—WKInterfaceLabel text

Concurrent animations and complex layouts affect performance
Animation

A few notes

Any update that affects sizing can animate layout

• Example—WKInterfaceLabel text

Concurrent animations and complex layouts affect performance

• Test on hardware!
Animation

A few notes

Any update that affects sizing can animate layout
• Example—WKInterfaceLabel text

Concurrent animations and complex layouts affect performance
• Test on hardware!

API functions within apps, not Glances or dynamic notifications
Animation

A few more notes
Animation

A few more notes

Use with restraint
Animation

A few more notes

Use with restraint

Should never be the focus
Animation

A few more notes

Use with restraint
Should never be the focus
Keep duration short
<table>
<thead>
<tr>
<th><strong>Related Session</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Designing with Animation</strong></td>
</tr>
</tbody>
</table>
Summary
Summary

Layout
Summary

Layout

• Specified at design time
Summary

Layout

• Specified at design time
• Flow-based
Summary

Layout

• Specified at design time
• Flow-based
• Groups are powerful
Summary

Layout
- Specified at design time
- Flow-based
- Groups are powerful

Animation
Summary

Layout

• Specified at design time
• Flow-based
• Groups are powerful

Animation

• Add liveliness and feedback to your app
Summary

Layout

• Specified at design time
• Flow-based
• Groups are powerful

Animation

• Add liveliness and feedback to your app
• Tables and images can already animate
Summary

Layout
• Specified at design time
• Flow-based
• Groups are powerful

Animation
• Add liveliness and feedback to your app
• Tables and images can already animate
• New API in watchOS 2
More Information

Documentation
watchOS 2 Transition Guide
WatchKit Programming Guide

Sample Code
WKRecipes
WatchKit Catalog
http://developer.apple.com/watchOS

Technical Support
Apple Developer Forums
Developer Technical Support

General Inquiries
Jake Behrens, watchOS Frameworks Evangelist
behrens@apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing WatchKit for watchOS 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Building Watch Apps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WatchKit In-Depth, Part 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WatchKit In-Depth, Part 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designing for Apple Watch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WatchKit Tips and Tricks</td>
<td>Presidio</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>Apple Watch Design Tips and Tricks</td>
<td>Presidio</td>
<td>Friday 3:30PM</td>
</tr>
</tbody>
</table>
## Labs

<table>
<thead>
<tr>
<th>Lab</th>
<th>Frameworks Lab</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>WatchKit Layout and Animation Lab</td>
<td>B</td>
<td>Thursday 3:30PM</td>
</tr>
<tr>
<td>WatchKit and ClockKit Complications Lab</td>
<td>A</td>
<td>Friday 1:30PM</td>
</tr>
</tbody>
</table>