Making Apps Adaptive

Part 1

Session 222

Kevin Cathey Interface Builder Engineer
Brent Shank Interface Builder Engineer
Takeaway
The system is going to do most of the work so you don’t have to.
300+

Combinations
Takeaway

The system is going to do most of the work so you don’t have to.
The system is going to do most of the work so you don’t have to.
PART 1
What traits are
What traits are Size classes
What traits are

How traits work

Size classes
Traits
horizontalSizeClass
[horizontalSizeClass = Compact]
[horizontalSizeClass = Compact,
verticalSizeClass = Regular,
displayGamut = P3]
Trait Examples
Trait Examples

Size Classes
Trait Examples

- Size Classes
- Dynamic Type
Trait Examples

Size Classes

Dynamic Type

Layout Direction
Trait Examples

- **Size Classes**
- **Dynamic Type**
- **Layout Direction**
Trait Examples

- **LAYOUT**
  - Size Classes
  - Dynamic Type
  - Layout Direction

- **APPEARANCE**
Trait Examples

- **Layout**
  - Size Classes
  - Dynamic Type
  - Layout Direction

- **Appearance**
  - Display Gamut
Trait Examples

Layout
- Size Classes
- Dynamic Type
- Layout Direction

Appearance
- Display Gamut
- Interface Style
Trait Examples

Layout
- Size Classes
- Dynamic Type
- Layout Direction

Appearance
- Display Gamut
- Interface Style

Capabilities
- 3D Touch
Trait Examples

- **LAYOUT**
  - Size Classes
  - Dynamic Type
  - Layout Direction

- **APPEARANCE**
  - Display Gamut
  - Interface Style

- **CAPABILITIES**
  - 3D Touch
Trait Examples

Size Classes
Size Classes
Takeaway
Base layout on available space, not device, orientation, or adaptation
Reacting to Available Space
Reacting to Available Space

Coarse Changes
Reacting to Available Space

Coarse Changes

Fine Changes
Reacting to Available Space

Coarse Changes

Fine Changes
“All problems in computer science can be solved by another level of indirection”

David Wheeler
Takeaway

Size classes express experience
Here's a great photo Emily took on Friday.

It really captures my Frisbee prowess.

You know you dropped it, right?

That's what I meant.

If nothing else, the dinosaur socks look terrific. Haha.

I figured you'd like those.

We need to hang out again, soon. Don't be extinct, OK?
Hold on tight: The universe is expanding faster than we thought.

You’ve probably already heard that the universe is expanding. It’s been doing so since the Big Bang — about 13.7 billion years ago — so you’ve had plenty of time to get in the loop. And, according to new research, it’s accelerating, even speeding up.
The Specifics of Size Classes
The Specifics of Size Classes
The Specifics of Size Classes

horizontalSizeClass

verticalSizeClass
The Specifics of Size Classes

horizontalSizeClass

verticalSizeClass

Compact  Regular
The Specifics of Size Classes

- **Compact**
  - horizontalSizeClass: wC hC
  - verticalSizeClass: wC hR

- **Regular**
  - horizontalSizeClass: wR hC
  - verticalSizeClass: wR hR
The Specifics of Size Classes

horizontalSizeClass

Compact

wC hC

wR hC

Regular

wR hR

verticalSizeClass

Compact

Regular
The Specifics of Size Classes

horizontalSizeClass

Compact wC hC Regular wR hC

verticalSizeClass

Compact Regular
The Specifics of Size Classes

horizontalSizeClass

Compact

wR hC

verticalSizeClass

Regular

Compact

Regular
The Specifics of Size Classes

**horizontalSizeClass**

- Compact
- Regular

**verticalSizeClass**

- Compact
- Regular
How Does This Help Me?
How Does This Help Me?

Only think about four combinations
How Does This Help Me?

Only think about four combinations

• But most commonly just two
How Does This Help Me?

Only think about four combinations

- But most commonly just two
- Width is most common
How Does This Help Me?

Only think about four combinations

• But most commonly just two
• Width is most common

System decides what combination applies
How Does This Help Me?

Only think about four combinations

• But most commonly just two
• Width is most common

System decides what combination applies

• Size class can change dynamically
How Does This Help Me?

Only think about four combinations
- But most commonly just two
- Width is most common

System decides what combination applies
- Size class can change dynamically
- If you use size classes, system can do the work for you
Dynamic Size Class Changes
View Controller structure
Dynamic Size Class Changes
View Controller structure
Dynamic Size Class Changes

Presentations
Dynamic Size Class Changes
Presentations
Dynamic Size Class Changes

Presentations
Fine Grain Changes
Fine Grain Changes

Use Auto Layout to specify changes within a size class
Fine Grain Changes

Use Auto Layout to specify changes within a size class

Medium grain changes?

Making Apps Adaptive, Part 2

Presidio

Friday 9:00AM
What traits are

Size classes
Propagating Changes

UITraitEnvironment
Propagating Changes

UITraitEnvironment

Screen
Propagating Changes

UITraitEnvironment

Screen
Propagating Changes

UITraitEnvironment

Screen → Window

TC
Propagating Changes

UITraitEnvironment

Screen → Window → View Controller

TC
Propagating Changes

UITraitEnvironment

Screen → Window → View Controller → View

TC
Propagating Changes

UIKitEnvironment

Screen → Window → View Controller → View

traitCollectionDidChange(:)

TC
Propagating Changes

UITraitEnvironment

Screen → Window → View Controller → View

traitCollectionDidChange(:)

TC
Propagating Changes

Traits can change during propagation
Propagating Changes

Traits can change during propagation
Traits can change during propagation.
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
override func traitCollectionDidChange(_ previousTraits:UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()  
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)
    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
            case .compact:
                setupConstraintsForCompactEnvironment()
            case .unspecified: fallthrough
            case .regular:
                setupConstraintsForRegularEnvironment()
        }
    }
}
traitCollectionDidChange(:)
traitCollectionDidChange(
)

Called for each `UITraitEnvironment`
traitCollectionDidChange( )

Called for each UITraitEnvironment

Override and check for specific trait changes
traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

Some systems react to `traitCollectionDidChange(:)` for you
traitCollectionDidChange(:)

Called for each `UITraitEnvironment`
Override and check for specific trait changes
Some systems react to `traitCollectionDidChange(:)` for you
- Interface Builder
traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

Some systems react to `traitCollectionDidChange(:)` for you

- Interface Builder
- Asset catalog
traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

Some systems react to `traitCollectionDidChange:(:)` for you

- Interface Builder
- Asset catalog
- `UIAppearance`
Takeaways
Takeaways

Traits describe environment
Takeaways

Traits describe environment

• Layout, appearance, capabilities
Takeaways

Traits describe environment

- Layout, appearance, capabilities

Override `traitCollectionDidChange:` to react to trait changes
Takeaways

Traits describe environment

• Layout, appearance, capabilities

Override `traitCollectionDidChange:` to react to trait changes

Size classes describe experience
Takeaways

Traits describe environment
- Layout, appearance, capabilities

Override `traitCollectionDidChange:` to react to trait changes

Size classes describe experience

System is going to do most of the work for you
Demo

Brent Shank
Recap

Building adaptive apps in Interface Builder
Recap
Building adaptive apps in Interface Builder
Recap

Building adaptive apps in Interface Builder
Recap

Building adaptive apps in Interface Builder
Recap

Building adaptive apps in Interface Builder
Summary
Summary

Traits describe environment

Override `traitCollectionDidChange:` to react to trait changes

Size classes describe experience
Summary

Traits describe environment

Override `traitCollectionDidChange:` to react to trait changes

Size classes describe experience

Interface Builder lets you customize on size class and preview on specific configurations
The system is going to do most of the work so you don’t have to.
<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Making Apps Adaptive, Part 2</td>
<td>Presidio</td>
<td>Friday 9:00AM</td>
</tr>
<tr>
<td>What’s New In Auto Layout</td>
<td>Presidio</td>
<td>Friday 3:00PM</td>
</tr>
<tr>
<td>Labs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Interface Builder and Auto Layout Lab</td>
<td>Developer Tools Lab B</td>
<td>Thursday 3:00PM</td>
</tr>
<tr>
<td>Interface Builder and Auto Layout Lab</td>
<td>Developer Tools Lab C</td>
<td>Friday 9:00AM</td>
</tr>
</tbody>
</table>