What's New in UIKit Dynamics and Visual Effects

Session 229

Michael Turner UIKit Engineer
David Duncan UIKit Engineer
# Recommended Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with UIKit Dynamics</td>
<td>WWDC13</td>
</tr>
<tr>
<td>Advanced Techniques with UIKit Dynamics</td>
<td>WWDC13</td>
</tr>
<tr>
<td>Creating Custom iOS User Interfaces</td>
<td>WWDC14</td>
</tr>
</tbody>
</table>
Agenda
What we will cover
Agenda
What we will cover

A basic overview
Agenda
What we will cover

A basic overview
What’s new in UIKit Dynamics
Agenda

What we will cover

A basic overview
What’s new in UIKit Dynamics
Utilizing UIVisualEffectView in your app
Agenda
What we will cover

A basic overview
What’s new in UIKit Dynamics
Utilizing UIVisualEffectView in your app
Best practices for working with UIKit Dynamics and Auto Layout
UIKit Dynamics
UIKit Dynamics

2D Physics-inspired animation system
UIKit Dynamics

2D Physics-inspired animation system

Composable and declarative
UIKit Dynamics

2D Physics-inspired animation system
Composable and declarative
Not a replacement for Core Animation or UIView animations
UIKit Dynamics
Sliding example

What's New in UIKit Dynamics & Visual Effects
9:41 AM
UIKit Dynamics

Sliding example
UIKit Dynamics
Sliding example
UIKit Dynamics
Sliding example

Define the coordinate system
UIKit Dynamics

Sliding example

Define the coordinate system

Provide the overall context for animation
UIKit Dynamics

Sliding example

Define the coordinate system
Provide the overall context for animation
Keep track of behaviors
UIKit Dynamics

Sliding example

Define the coordinate system
Provide the overall context for animation
Keep track of behaviors
UIKit Dynamics

Sliding example

Define the coordinate system
Provide the overall context for animation
Keep track of behaviors
UIDynamicItems are associated with one or more behaviors
UIKit Dynamics

What’s new
UIKit Dynamics

What’s new

Support for non-rectangular collision bounds
UIKit Dynamics

What’s new

Support for non-rectangular collision bounds

UIDynamicItemGroup
UIKit Dynamics

What’s new

Support for non-rectangular collision bounds

UIDynamicItemGroup

UIFieldBehavior—models vector force fields
UIKit Dynamics
What’s new

Support for non-rectangular collision bounds
UIDynamicItemGroup
UIFieldBehavior—models vector force fields
UIDynamicItemBehavior
UIKit Dynamics

What’s new

Support for non-rectangular collision bounds
UIDynamicItemGroup
UIFieldBehavior—models vector force fields
UIDynamicItemBehavior
UISnapBehavior
UIKit Dynamics

What’s new

Support for non-rectangular collision bounds
UIDynamicItemGroup
UIFieldBehavior—models vector force fields
UIDynamicItemBehavior
UISnapBehavior
UIAttachmentBehavior
UIKit Dynamics
What’s new

Support for non-rectangular collision bounds
UIDynamicItemGroup
UIFieldBehavior—models vector force fields
UIDynamicItemBehavior
UISnapBehavior
UIAttachmentBehavior

New ways to debug dynamic animations
Collision Bounds
UIDynamicItem
enum UIDynamicItemCollisionBoundsType : UInt {
}

Collision Bounds
UIDynamicItem
enum UIDynamicItemCollisionBoundsType : UInt {
    case Rectangle
}
Collision Bounds

UIDynamicItem

```swift
enum UIDynamicItemCollisionBoundsType : UInt {
    case Rectangle
    case Ellipse
}
```
enum UIDynamicItemCollisionBoundsType : UInt {
    case Rectangle
    case Ellipse
    case Path
}
Collision Bounds

UIDynamicItem

enum UIDynamicItemCollisionBoundsType : UInt {
    case Rectangle
    case Ellipse
    case Path
}

protocol UIDynamicItem : NSObjectProtocol {
    var center: CGPoint { get set }
    var bounds: CGRect { get }
    var transform: CGAffineTransform { get set }
}
enum UIDynamicItemCollisionBoundsType : UInt {
    case Rectangle
    case Ellipse
    case Path
}

protocol UIDynamicItem : NSObjectProtocol {
    var center: CGPoint { get set }
    var bounds: CGRect { get }
    var transform: CGAffineTransform { get set }
    optional var collisionBoundsType: UIDynamicItemCollisionBoundsType { get }
    optional var collisionBoundingPath: UIBezierPath { get }
}
enum UIDynamicItemCollisionBoundsType : UInt {
    case Rectangle
    case Ellipse
    case Path
}

protocol UIDynamicItem : NSObjectProtocol {
    var center: CGPoint { get set }
    var bounds: CGRect { get }
    var transform: CGAffineTransform { get set }
    
    optional var collisionBoundsType: UIDynamicItemCollisionBoundsType { get }
    optional var collisionBoundingPath: UIBezierPath { get }
}
UIDynamicItemCollisionBoundsType

Path

Convex
Counter-clockwise wound
Non-self intersecting

Convex

Concave
UIDynamicItemGroup
UIDynamicItemGroup

Makes multiple dynamic items behave as one
UIDynamicItemGroup

Makes multiple dynamic items behave as one
Preserves the individual collision bounds
UIDynamicItemGroup

Makes multiple dynamic items behave as one
Preserves the individual collision bounds
Dynamic items in a group must not be added to behaviors individually
UIDynamicItemGroup

Makes multiple dynamic items behave as one
Preserves the individual collision bounds
Dynamic items in a group must not be added to behaviors individually
A group cannot contain other groups
UIDynamicItemGroup

- Makes multiple dynamic items behave as one
- Preserves the individual collision bounds
- Dynamic items in a group must not be added to behaviors individually
- A group cannot contain other groups
- Concave or other complex shapes are possible
UIKit Dynamics

Force example
UIKit Dynamics

Force example
UIKit Dynamics

Force example
A force is modeled as a vector
UIKit Dynamics

Force example

A force is modeled as a vector

• Length and direction
A force is modeled as a vector

• Length and direction

UIPushBehavior applies a force
UIKit Dynamics
Force example

A force is modeled as a vector
- Length and direction

UIPushBehavior applies a force
- Continuous
A force is modeled as a vector
• Length and direction

UIPushBehavior applies a force
• Continuous
• Instantaneous
A force is modeled as a vector
• Length and direction

UIPushBehavior applies a force
• Continuous
• Instantaneous
UIKit Dynamics

Force example
UIKit Dynamics

Force example

Vertical motion over time of the sliding view
UIKit Dynamics

Force example

Vertical motion over time of the sliding view
Initial Impulse applied

- $y$ axis
- Time axis
UIKit Dynamics

Force example

Vertical motion over time of the sliding view

Initial Impulse applied

Gravity causes the view’s velocity to slow and eventually reverse direction
Physics Fields

Linear Gravity as a field

Linear Gravity Field
Physics Fields

Linear Gravity as a field

A function that assigns a vector to each point in a domain
UIFieldBehavior
UIFieldBehavior can be added to a region of your view.
UIFieldBehavior can be added to a region of your view. The field is evaluated at each point within the region.
UIFieldBehavior can be added to a region of your view.
The field is evaluated at each point within the region.
Resulting forces are applied by the animator.
UIFieldBehavior can be added to a region of your view. The field is evaluated at each point within the region. Resulting forces are applied by the animator. UIGravityBehavior is a field already!
UIFieldBehavior can be added to a region of your view.
The field is evaluated at each point within the region.
Resulting forces are applied by the animator.
UIGravityBehavior is a field already.
Simplified physics, well-tuned for performance;
not useful for building interstellar space stations.
UIFieldBehavior

- Linear Gravity
- Radial Gravity
- Noise
- Custom

Drag and Velocity, Vortex, Turbulence, Spring, Electric and Magnetic
UIFieldBehavior
Linear Gravity
UIFieldBehavior

Linear Gravity

Field properties

var `region`: UIRegion
var `direction`: CGVector
var `strength`: CGFloat
UIFieldBehavior
Radial Gravity
UIFieldBehavior

Radial Gravity

Field properties

```swift
var position: CGPoint
var region: UIRegion
var strength: CGFloat
var falloff: CGFloat
var minimumRadius: CGFloat
```
UIFieldBehavior

Noise
UIFieldBehavior

Noise

Field properties

var region: UIRegion
var strength: CGFloat
var smoothness: CGFloat
var animationSpeed: CGFloat
UIFieldBehavior
Custom
UIFieldBehavior

Custom

Field Sample

- position
- velocity
- mass
- charge
- time
Demo
UIKit Dynamics—StickyCorners
Debugging UIKit Dynamics
Debugging UIKit Dynamics

Overlay to show Physics
Debugging UIKit Dynamics

Overlay to show Physics
Visualize fields, collision bounds, and attachments
Debugging UIKit Dynamics

Overlay to show Physics
Visualize fields, collision bounds, and attachments
Not API, but accessible in lldb
Debugging UIKit Dynamics

Overlay to show Physics
Visualize fields, collision bounds, and attachments
Not API, but accessible in lldb

dbgEnabled
Debugging UIKit Dynamics

Overlay to show Physics
Visualize fields, collision bounds, and attachments
Not API, but accessible in lldb

debugEnabled
debugInterval
Debugging UIKit Dynamics

Overlay to show Physics
Visualize fields, collision bounds, and attachments
Not API, but accessible in llDb

debugEnabled
debugInterval
debugAnimationSpeed
UIDynamicItemBehavior
UIDynamicItemBehavior

Customize physical properties
UIDynamicItemBehavior

Customize physical properties
Applied to one or more items
UIDynamicItemBehavior

Customize physical properties

Applied to one or more items

var `elasticity`: CGFloat
var `friction`: CGFloat
var `density`: CGFloat
var `resistance`: CGFloat
var `angularResistance`: CGFloat
UIDynamicItemBehavior

Customize physical properties

Applied to one or more items

```swift
var charge: CGFloat
var anchored: Bool
```
UIAttachmentBehavior

Distance attachment
UIAttachmentBehavior

Distance attachment
Customizable Damping and Frequency
UIAttachmentBehavior
UIAttachmentBehavior

Limit Attachment
UIAttachmentBehavior

Limit Attachment
Behaves like a rope between items
UIAttachmentBehavior

Limit Attachment
Behaves like a rope between items
Attachment point offset from each item’s center
UIAttachmentBehavior
UIAttachmentBehavior

Fixed Attachment
UIAttachmentBehavior

Fixed Attachment
Items positioned with respect to the attachment anchor point
UIAttachmentBehavior

Fixed Attachment
Items positioned with respect to the attachment anchor point
No relative movement
UIAttachmentBehavior

Pin Attachment
UIAttachmentBehavior

Pin Attachment

Items positioned with respect to the attachment anchor point
UIAttachmentBehavior

Pin Attachment
Items positioned with respect to the attachment anchor point
Specify a rotatable range
Pin Attachment
Items positioned with respect to the attachment anchor point
Specify a rotatable range
UIAttachmentBehavior
UIAttachmentBehavior

Sliding Attachment
UIAttachmentBehavior

Sliding Attachment

Items positioned with respect to the attachment anchor point
UIAttachmentBehavior

Sliding Attachment
Items positioned with respect to the attachment anchor point
Allows only relative translation along a specified axis
UIAttachmentBehavior

**Sliding Attachment**

Items positioned with respect to the attachment anchor point.

Allows only relative translation along a specified axis.

Prevents relative rotation of the dynamic items.
Sliding Attachment
Items positioned with respect to the attachment anchor point
Allows only relative translation along a specified axis
Prevents relative rotation of the dynamic items
Specify a translatable range
Sliding Attachment
Items positioned with respect to the attachment anchor point
Allows only relative translation along a specified axis
Prevents relative rotation of the dynamic items
Specify a translatable range
UISnapBehavior
UISnapBehavior

Snap a view in place
UISnapBehavior

Snap a view in place

Customizable damping
UISnapBehavior

Snap a view in place
Customizable damping
Customizable snapPoint
UISnapBehavior

Snap a view in place
Customizable damping
Customizable snapPoint
Visual Effects
Adding style to your app

David Duncan
UIKit Engineer
UIVisualEffectView
Stylish Overlays
enum UIBlurEffectStyle : Int {
    case ExtraLight
    case Light
    case Dark
}

let blurEffect = UIBlurEffect(style: .ExtraLight)

let blurView = UIVisualEffectView(effect: blurEffect)
enum UIBlurEffectStyle : Int {
    case ExtraLight
    case Light
    case Dark
}

let blurEffect = UIBlurEffect(style: .ExtraLight)

let blurView = UIVisualEffectView(effect: blurEffect)
enum UIBlurEffectStyle : Int {
    case ExtraLight
    case Light
    case Dark
}

let blurEffect = UIBlurEffect(style: .ExtraLight)

let blurView = UIVisualEffectView(effect: blurEffect)
enum UIBlurEffectStyle : Int {
    case ExtraLight
    case Light
    case Dark
}

let blurEffect = UIBlurEffect(style: .ExtraLight)

let blurView = UIVisualEffectView(effect: blurEffect)
let vibrancyEffect = UIVibrancyEffect(forBlurEffect: blurEffect)

let vibrancyView = UIVisualEffectView(effect: vibrancyEffect)

blurView.contentView.addSubview(vibrancyView)

vibrancyView.contentView.addSubview(label)
let vibrancyEffect = UIVibrancyEffect(forBlurEffect: blurEffect)

let vibrancyView = UIVisualEffectView(effect: vibrancyEffect)

blurView.contentView.addSubview(vibrancyView)

vibrancyView.contentView.addSubview(label)
let vibrancyEffect = UIVibrancyEffect(forBlurEffect: blurEffect)

let vibrancyView = UIVisualEffectView(effect:vibrancyEffect)

blurView.contentView.addSubview(vibrancyView)

vibrancyView.contentView.addSubview(label)
let vibrancyEffect = UIVibrancyEffect(forBlurEffect: blurEffect)

let vibrancyView = UIVisualEffectView(effect: vibrancyEffect)

blurView.contentView.addSubview(vibrancyView)

vibrancyView.contentView.addSubview(label)
let vibrancyEffect = UIVibrancyEffect(forBlurEffect: blurEffect)

let vibrancyView = UIVisualEffectView(effect: vibrancyEffect)

blurView.contentView.addSubview(vibrancyView)

vibrancyView.contentView.addSubview(label)
UIVisualEffectView
And action!

Iceland 001.jpg
UIVisualEffectView

And action!

Bounds

Iceland 001.jpg (2500.0 x 1668.0)
UIVisualEffectView
And action!

Bounds
Effect
UIVisualEffectView

Anatomy of an effect
UIVisualEffectView

Anatomy of an effect
UIVisualEffectView

Anatomy of an effect
UIVisualEffectView
Anatomy of an effect
UIVisualEffectView

Anatomy of an effect

Capture Area

Apply Effect
UIVisualEffectView

Anatomy of an effect

Capture Area

Apply Effect
UIVisualEffectView

Anatomy of an effect
UIVisualEffectView

Anatomy of an effect

Capture Area  Apply Effect  Copy Back  Offscreen Pass
UIVisualEffectView

Offscreen Passes

Alpha
Masking
Blur
Vibrancy
Snapshotting
UIVisualEffectView

Offscreen Passes

Alpha
Masking
Blur
Vibrancy

Snapshotting

UIView.snapshotViewAfterScreenUpdates(afterUpdates:)
UIView.drawViewHierarchyInRect(rect:, afterScreenUpdates:)
UIScreen.snapshotViewAfterScreenUpdates()
UIVisualEffectView

Offscreen Passes

Alpha
Masking
Blur
Vibrancy

Snapshotting

UITableView.snapshotViewAfterScreenUpdates(afterUpdates:)
UIView.drawViewHierarchyInRect(rect:, afterScreenUpdates:)
UIScreen.snapshotViewAfterScreenUpdates()
UIVisualEffectView
Anatomy of a broken effect
UIVisualEffectView
Anatomy of a broken effect
UIVisualEffectView
Anatomy of a broken effect

Capture Area
UIVisualEffectView
Anatomy of a broken effect
UIVisualEffectView
Fixing broken effects
UIVisualEffectView
Fixing broken effects

For help in lldb
- (NSString *)_ whatsWrongWithThisEffect
UIVisualEffectView

Fixing broken effects

For help in lldb
- (NSString *)_whatsWrongWithThisEffect

(llldb) po [myEffectView _whatsWrongWithThisEffect]
UIVisualEffectView

Fixing broken effects

For help in lldb
– (NSString *)_whatsWrongWithThisEffect

(llldb) po [myEffectView _whatsWrongWithThisEffect]

ISSUE: One or more masking superviews have been found.

<UIView: 0x7fa030518810; frame = (0 0; 0 0); layer = <CALayer: 0x7fa03050b190>>

UIVisualEffectView cannot be masked, you will need to isolate this effect view into a hierarchy that is not masked or not use masks.
UIVisualEffectView
Fixing broken effects

Rearrange view hierarchy
• Effective for Alpha and Masking

Root
  Container
    Blur
      Content

apply alpha or masking
UIVisualEffectView
Fixing broken effects

Rearrange view hierarchy
• Effective for Alpha and Masking

Root

Blur

Container

Content

apply alpha or masking
UIVisualEffectView

Fixing broken effects

Rearrange view hierarchy

• Effective for Alpha and Masking

Root

Blur

Container

Content

apply alpha or masking
UIVisualEffectView
Fixing broken effects

Mask views individually

Container
Content
Blur
Content

mask added
UIVisualEffectView

Fixing broken effects

Mask views individually
UIVisualEffectView

Fixing broken effects

Mask views individually

Mask added
Fixing broken effects

Snapshot the window or screen

UIVisualEffectView

Screen → Window → Content → View Snapshot requested → Content → Blur
UIVisualEffectView
Fixing broken effects

Snapshot the window or screen
UIVisualEffectView
Fixing broken effects

Snapshot the window or screen

Screen
  ↓
Window
    ↓
Content
      ↓
Content
        ↓
Blur
UIVisualEffectView

Fixing broken effects

Snapshot the window or screen

Screen
→ Screen Snapshot requested

Window

Content

Content

Blur
UIKit Dynamics and Auto Layout
UIKit Dynamics and Auto Layout

Follow via Auto Layout

UIKit Dynamics outside

• `dynamicsView.translatesAutoresizingMaskIntoConstraints = true`

Auto Layout inside

• `innerView.leadingAnchor.constraintEqualToAnchor(dynamicsView.leadingAnchor)`
UIKit Dynamics and Auto Layout
Follow via Auto Layout
UIKit Dynamics and Auto Layout

Follow via Auto Layout

Lola 011.jpg
UIKit Dynamics and Auto Layout
Custom UIDynamicItem
UIKit Dynamics and Auto Layout

Custom UIDynamicItem

Subclass NSObject
UIKit Dynamics and Auto Layout

Custom UIDynamicItem

Subclass NSObject
Conform to UIDynamicItem
UIKit Dynamics and Auto Layout

Custom UIDynamicItem

Subclass NSObject
Conform to UIDynamicItem
Provide .bounds
UIKit Dynamics and Auto Layout

Custom UIDynamicItem

Subclass NSObject
Conform to UIDynamicItem
Provide .bounds
Update constraints when .center and .transform change
Demo
UIKit Dynamics and Auto Layout
Focus on the user experience
Always consider performance impact when adding advanced effects
## Related Sessions and Labs

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started with Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Tuesday 4:30PM</td>
</tr>
<tr>
<td>Multitasking Essentials for Media-Based Apps on iPad in iOS 9</td>
<td>Pacific Heights</td>
<td>Wednesday 2:30PM</td>
</tr>
<tr>
<td>Optimizing Your App for Multitasking on iPad in iOS 9</td>
<td>Presidio</td>
<td>Wednesday 3:30PM</td>
</tr>
<tr>
<td>Mysteries of Auto Layout, Part 1</td>
<td>Presidio</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>Mysteries of Auto Layout, Part 2</td>
<td>Presidio</td>
<td>Thursday 1:30PM</td>
</tr>
<tr>
<td>Building Responsive and Efficient Apps with GCD</td>
<td>Nob Hill</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>UIKit and UIKit Dynamics Lab</td>
<td>Frameworks Lab C</td>
<td>Friday 11:00AM</td>
</tr>
</tbody>
</table>
More Information

Documentation
UIDynamicAnimator Class Reference
UIFieldBehavior Class Reference
UIDynamicBehavior Class Reference
UIAttachmentBehavior Class Reference

Sample Code
StickyCorners

Technical Support
Apple Developer Forums
Developer Technical Support

Curt Rothert
App Frameworks Evangelist
rothert@apple.com

http://developer.apple.com/library