

# What's New in UIKit Dynamics and Visual Effects

Session 229

Michael Turner UIKit Engineer

David Duncan UIKit Engineer

# Recommended Sessions

---

Getting Started with UIKit Dynamics

WWDC13

---

Advanced Techniques with UIKit Dynamics

WWDC13

---

Creating Custom iOS User Interfaces

WWDC14

---

# 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



# 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

UIDynamicAnimator





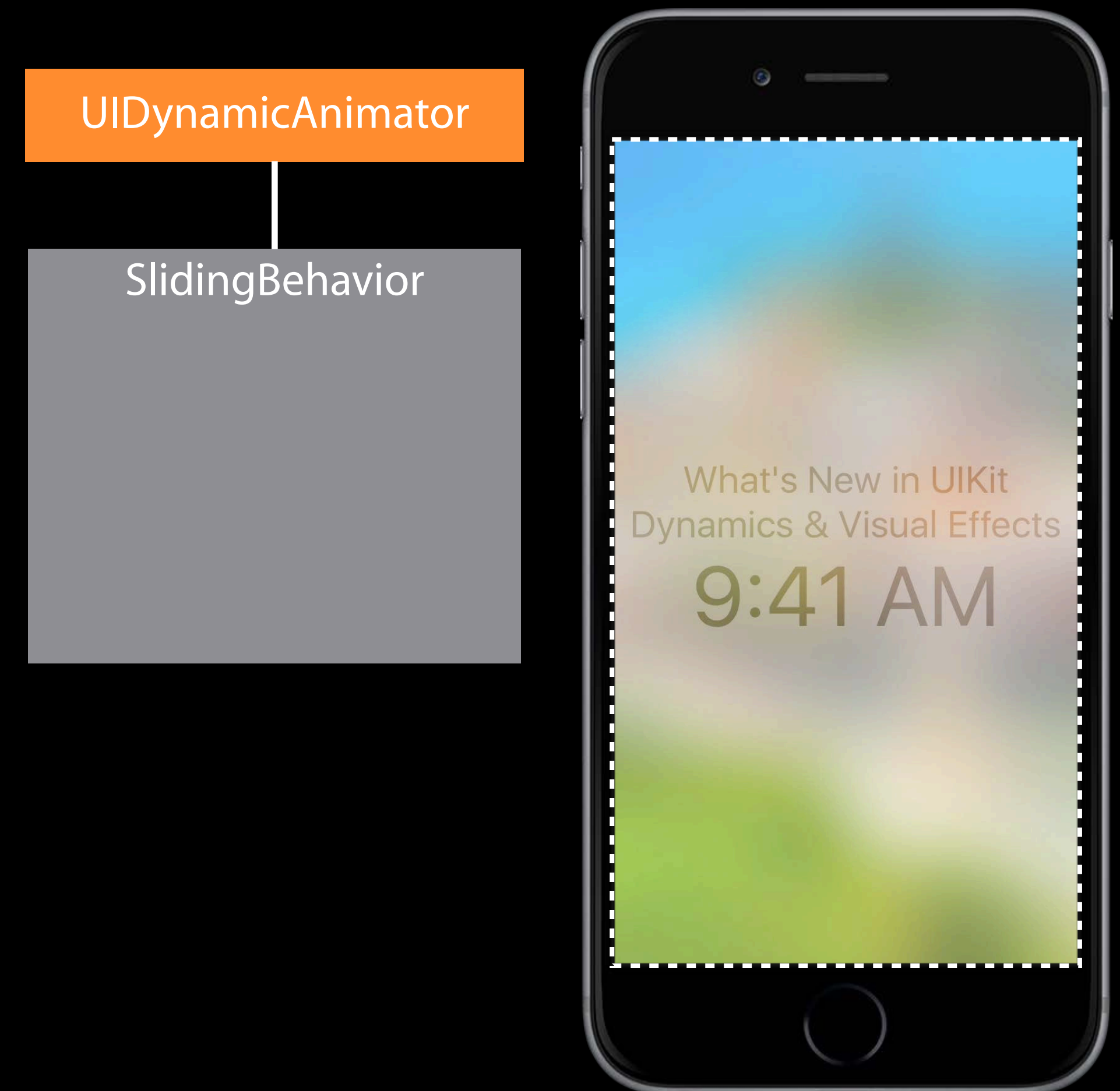
# UIKit Dynamics

## Sliding example

Define the coordinate system

Provide the overall context for animation

Keep track of behaviors



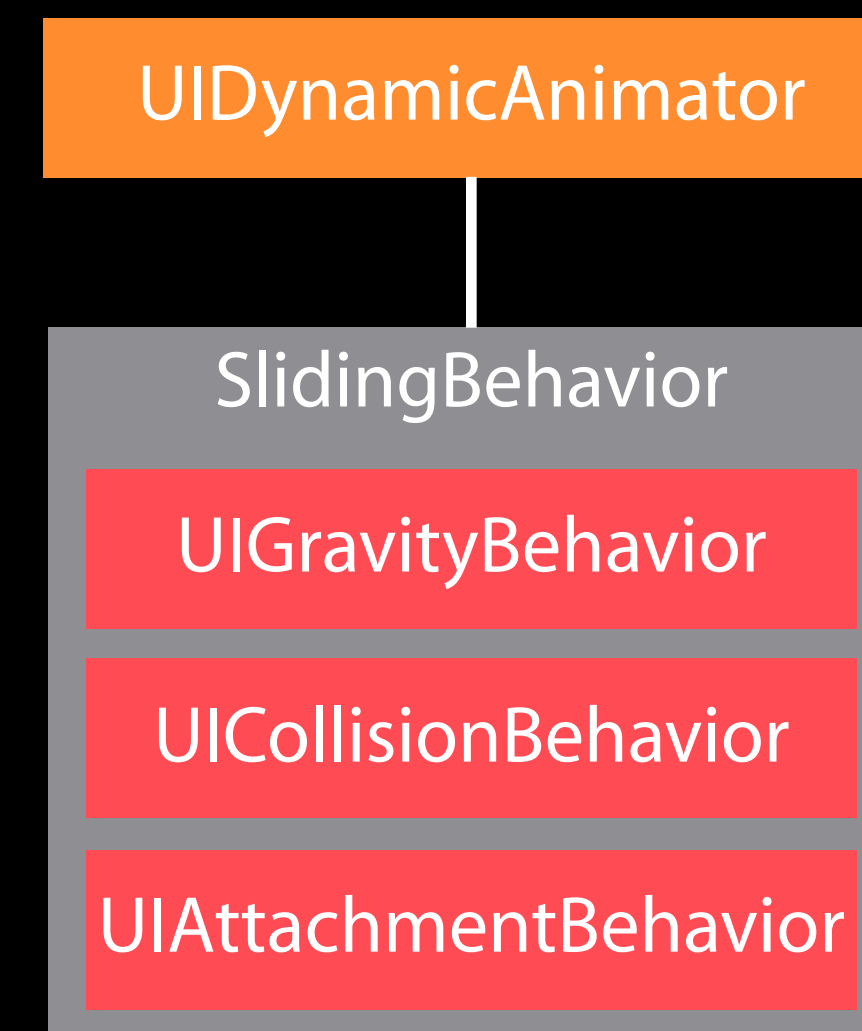
# UIKit Dynamics

## Sliding example

Define the coordinate system

Provide the overall context for animation

Keep track of behaviors



# UIKit Dynamics

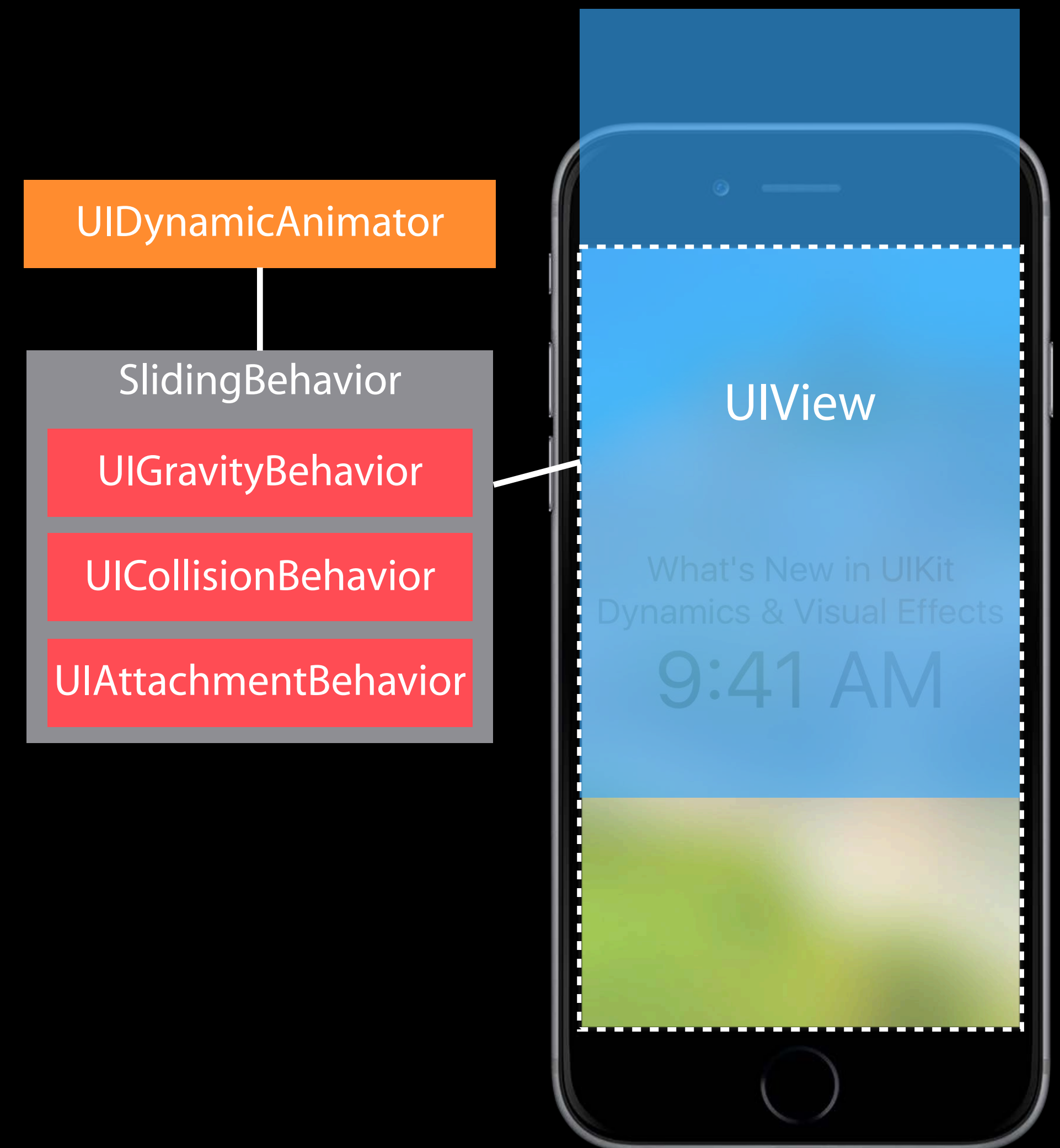
## Sliding example

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

NEW

# Collision Bounds

# UIDynamicItem

NEW

```
enum UIDynamicItemCollisionBoundsType : UInt {
```

}

# Collision Bounds

NEW

UIDynamicItem

```
enum UIDynamicItemCollisionBoundsType : UInt {  
    case Rectangle  
  
}
```

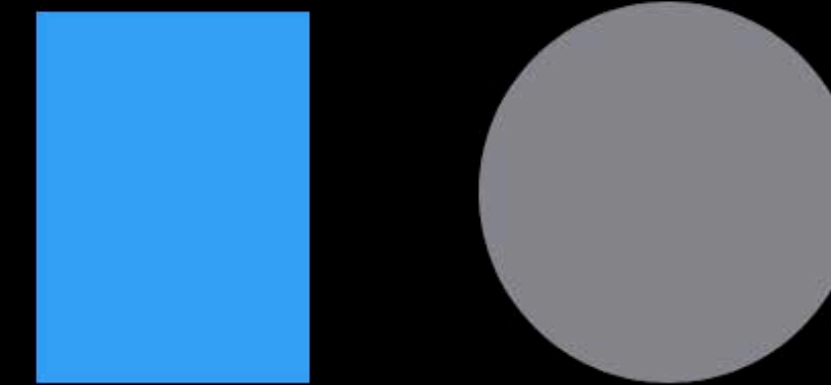


# Collision Bounds

NEW

UIDynamicItem

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



# Collision Bounds

NEW

UIDynamicItem

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



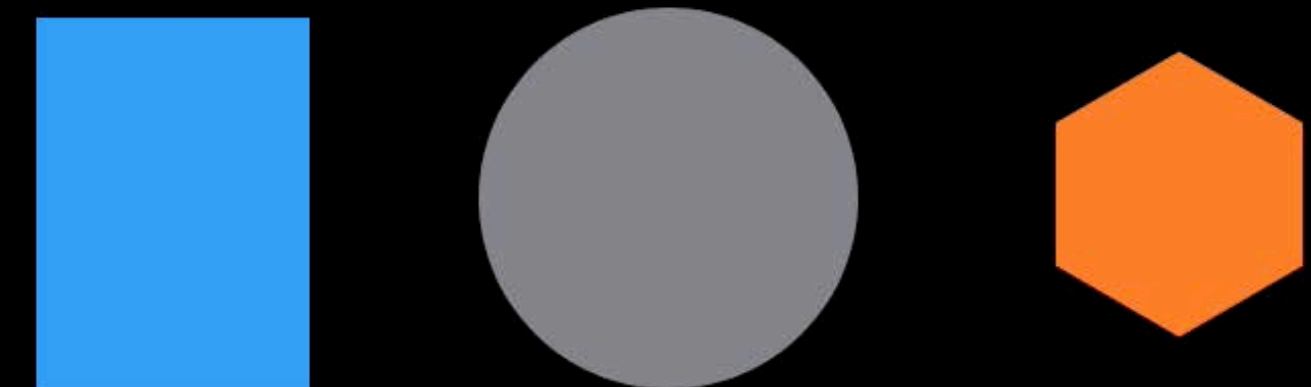


# Collision Bounds

NEW

UIDynamicItem

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



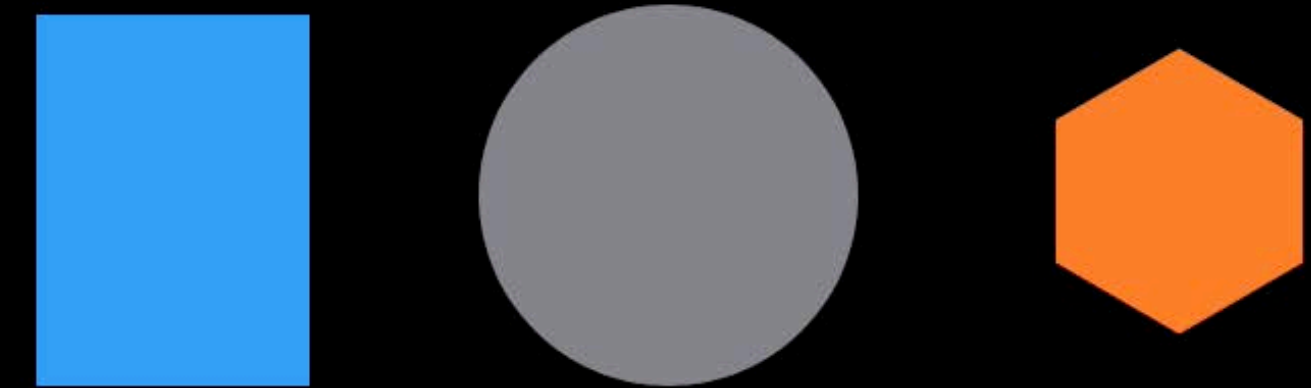
```
protocol UIDynamicItem : NSObjectProtocol {  
    var center: CGPoint { get set }  
    var bounds: CGRect { get }  
    var transform: CGAffineTransform { get set }  
}
```

# Collision Bounds

NEW

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 }
```

```
    optional var collisionBoundsType: UIDynamicItemCollisionBoundsType { get }  
    optional var collisionBoundingPath: UIBezierPath { get }
```

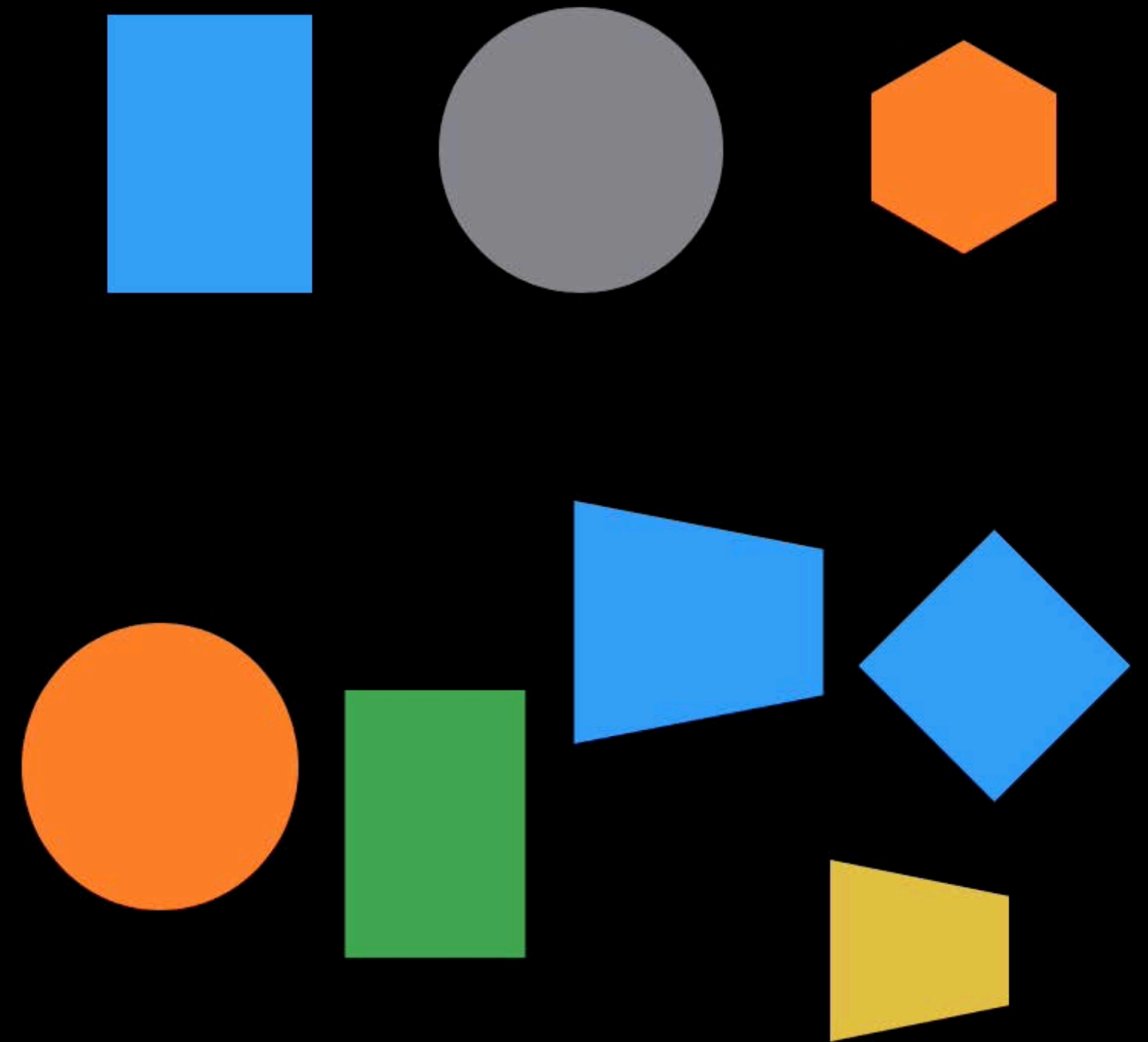
```
}
```

# Collision Bounds

NEW

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 }  
  
    optional var collisionBoundsType: UIDynamicItemCollisionBoundsType { get }  
    optional var collisionBoundingPath: UIBezierPath { get }  
}
```



# UIDynamicItemCollisionBoundsType

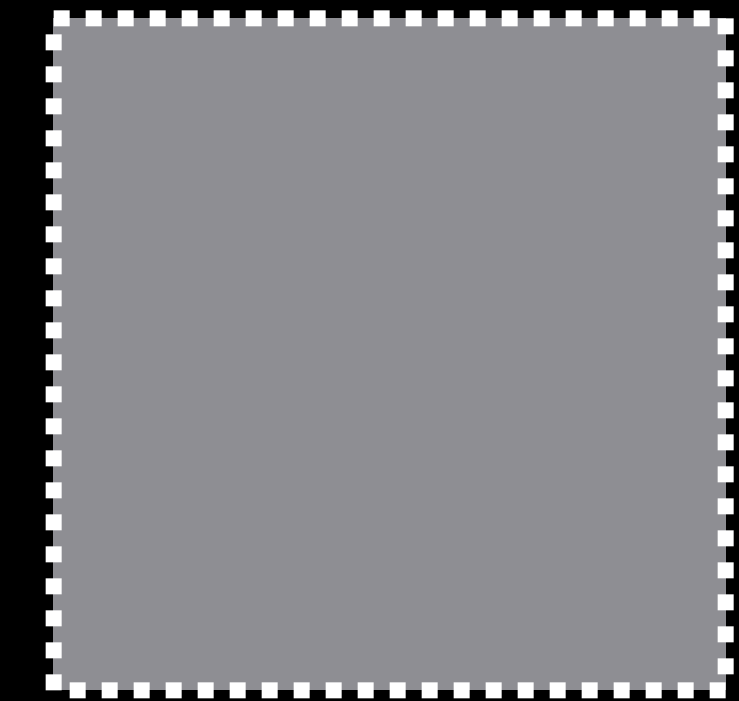
Path

Convex

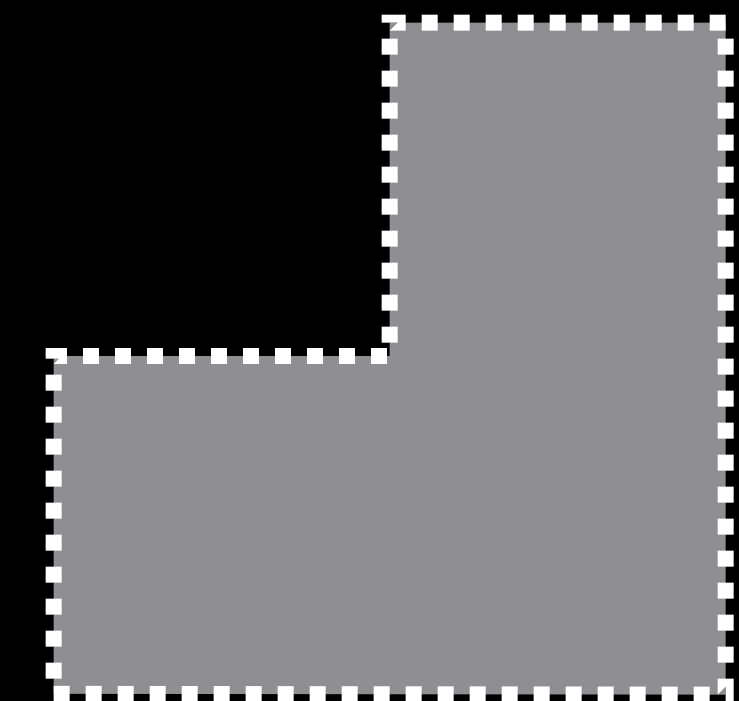
Counter-clockwise wound

Non-self intersecting

Convex

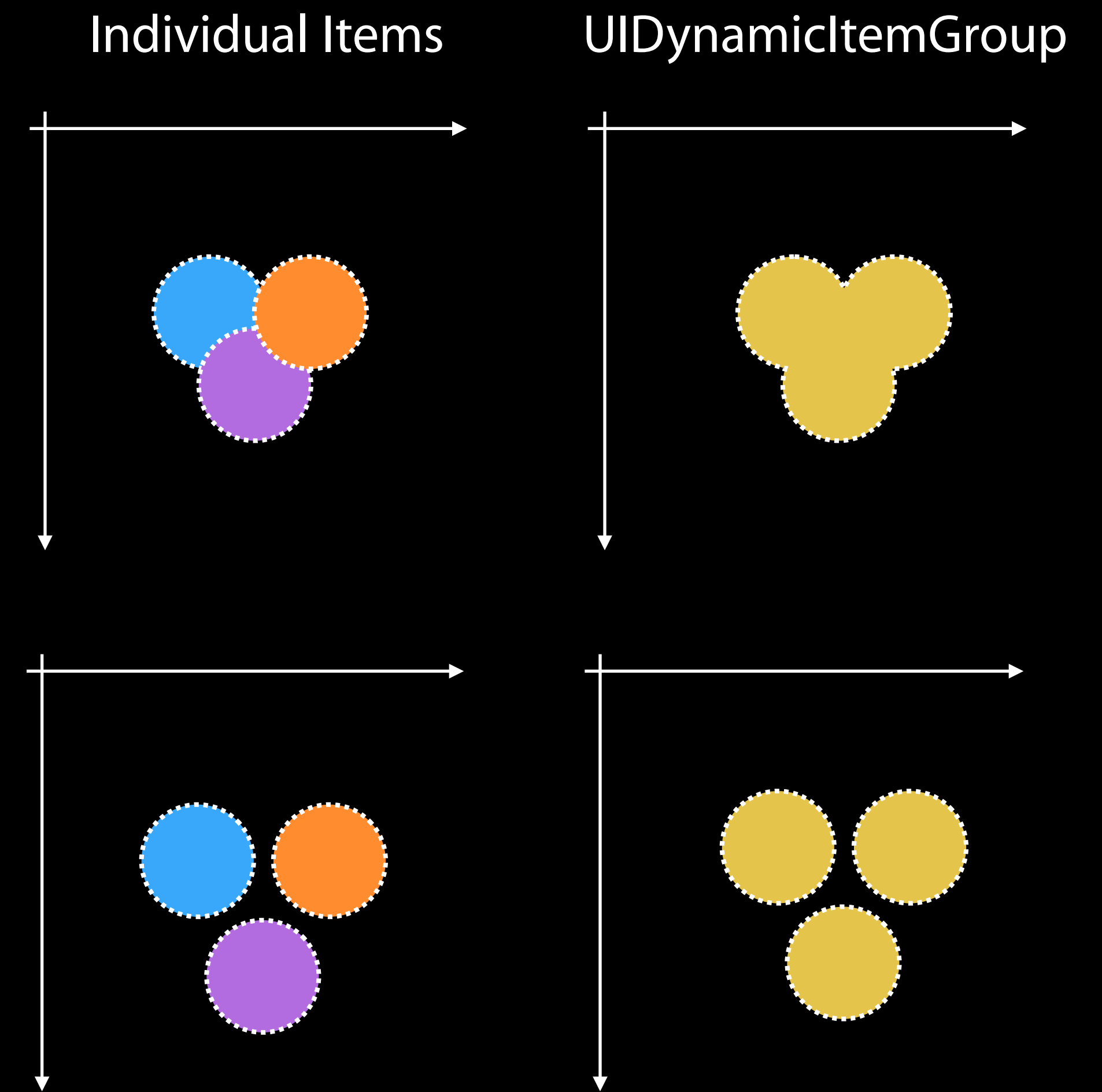


Concave



# UIDynamicItemGroup

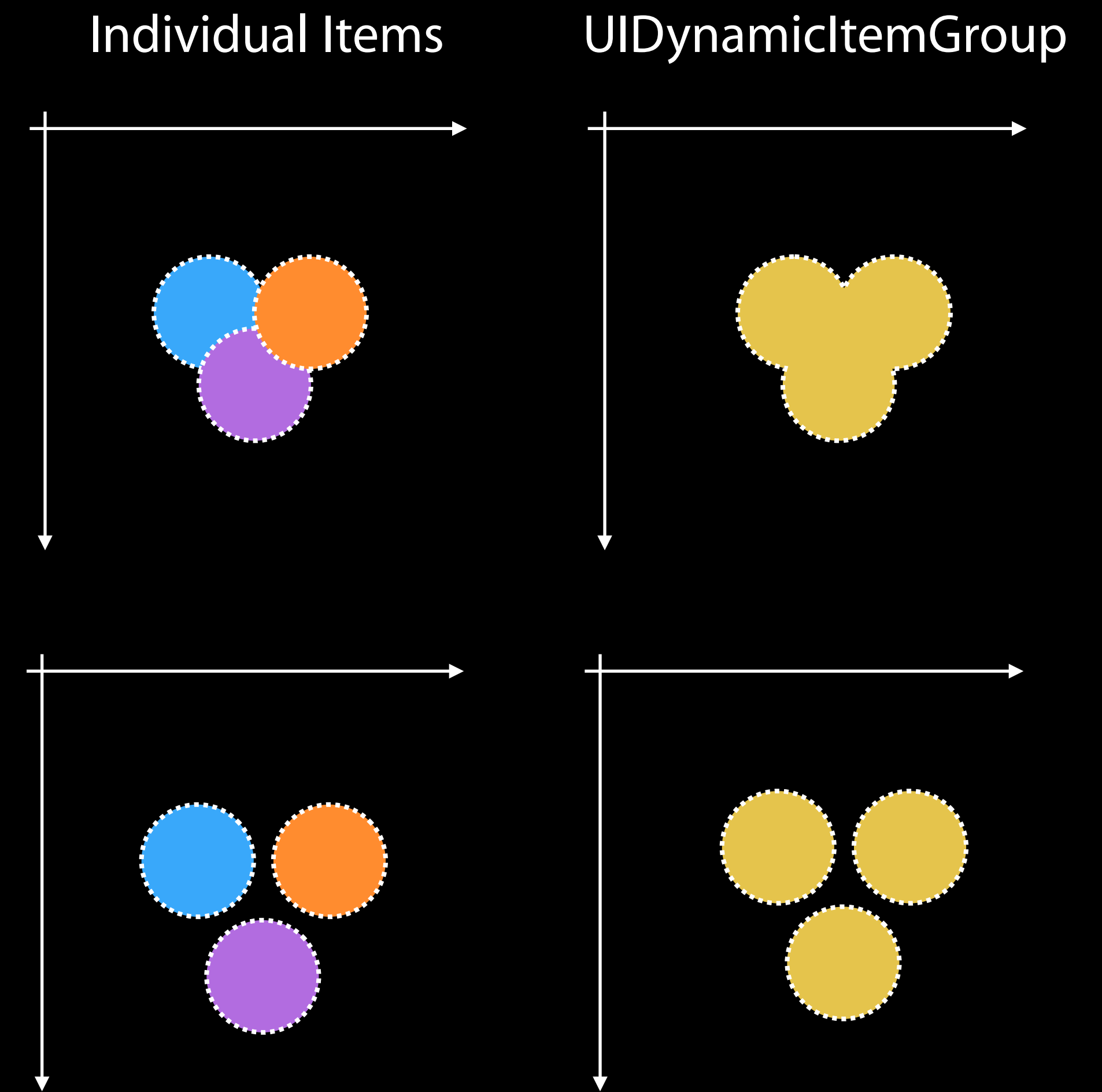
NEW



# UIDynamicItemGroup

NEW

Makes multiple dynamic items behave as one

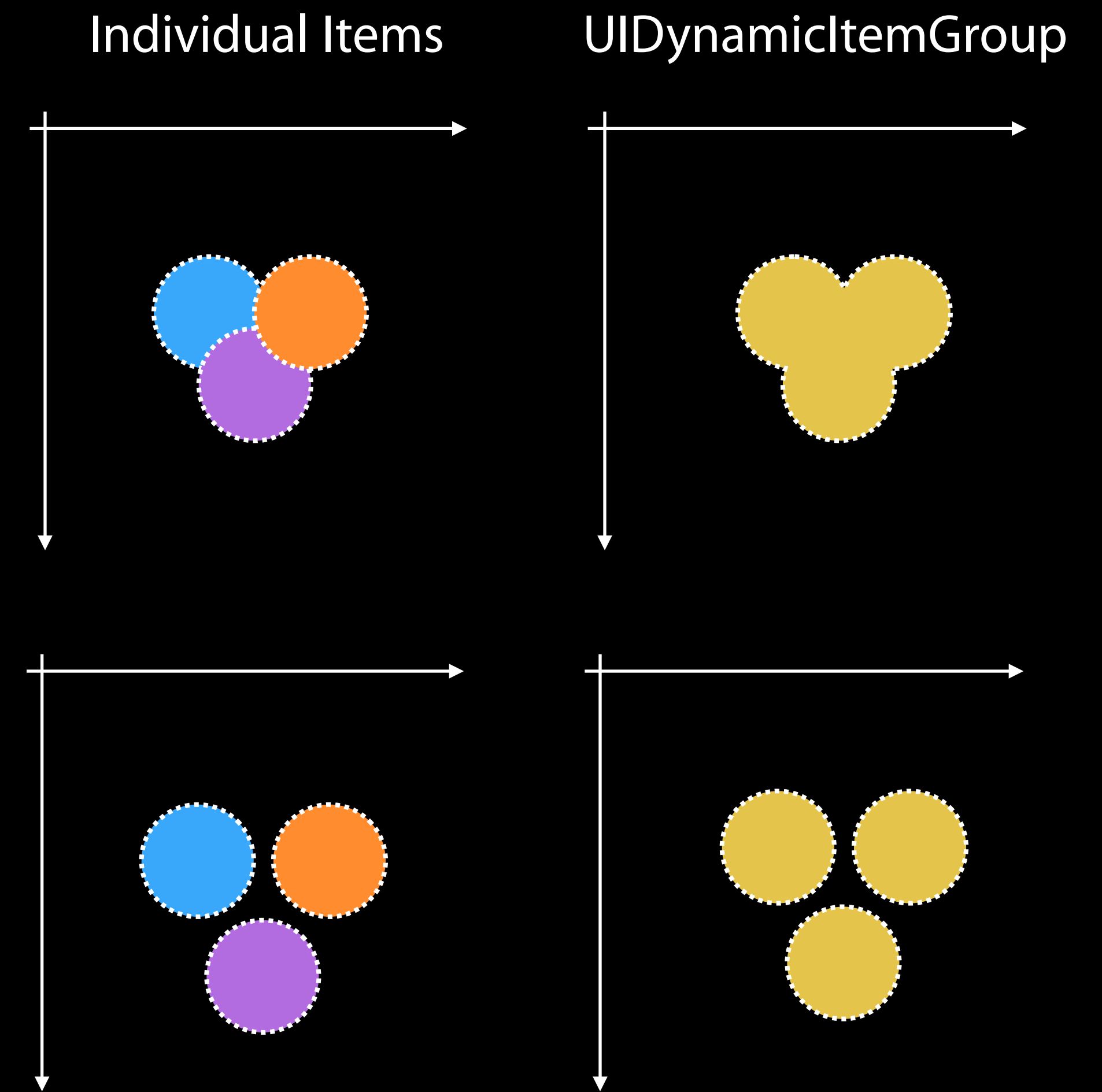


# UIDynamicItemGroup

NEW

Makes multiple dynamic items behave as one

Preserves the individual collision bounds



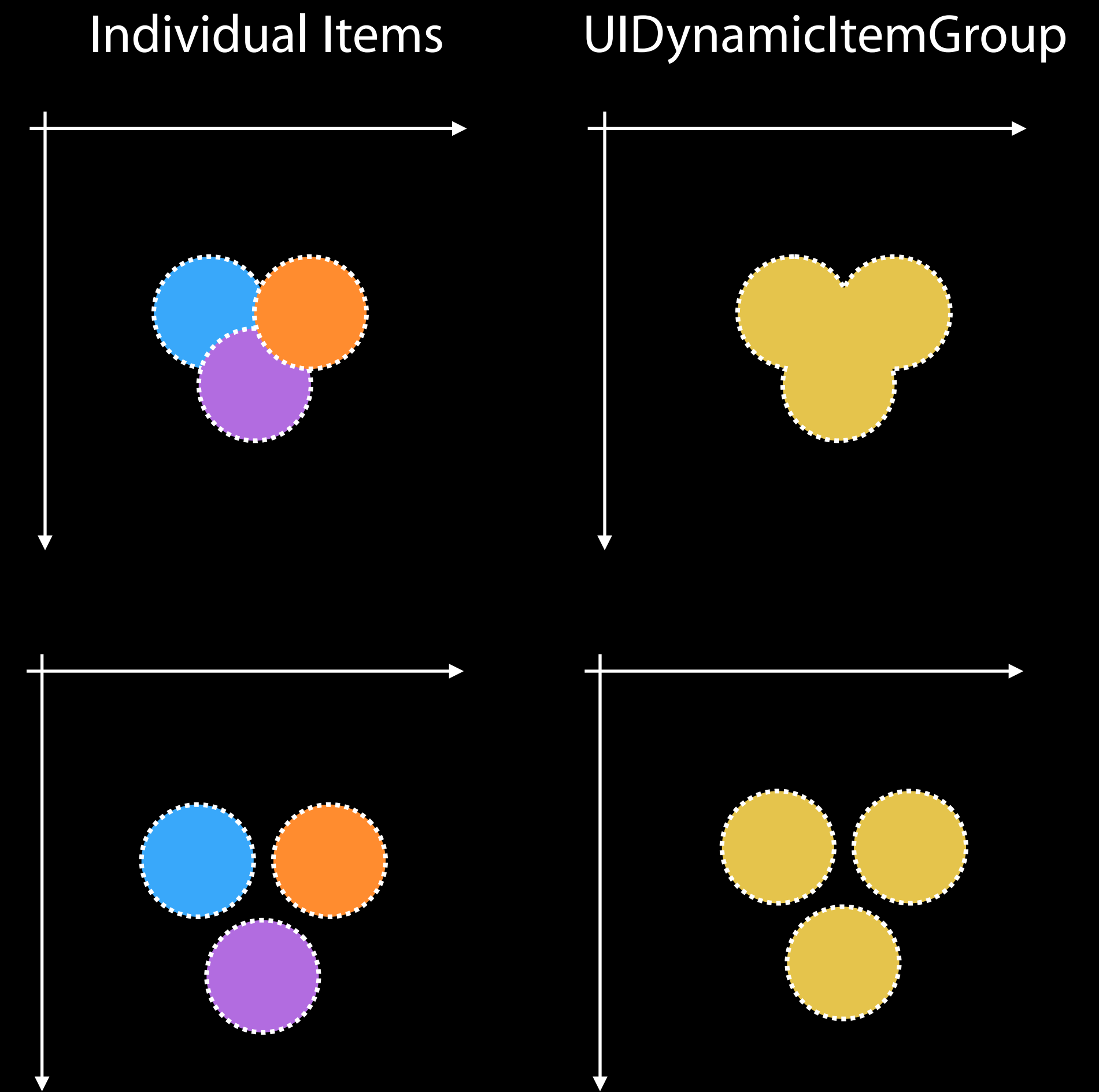
# UIDynamicItemGroup

NEW

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

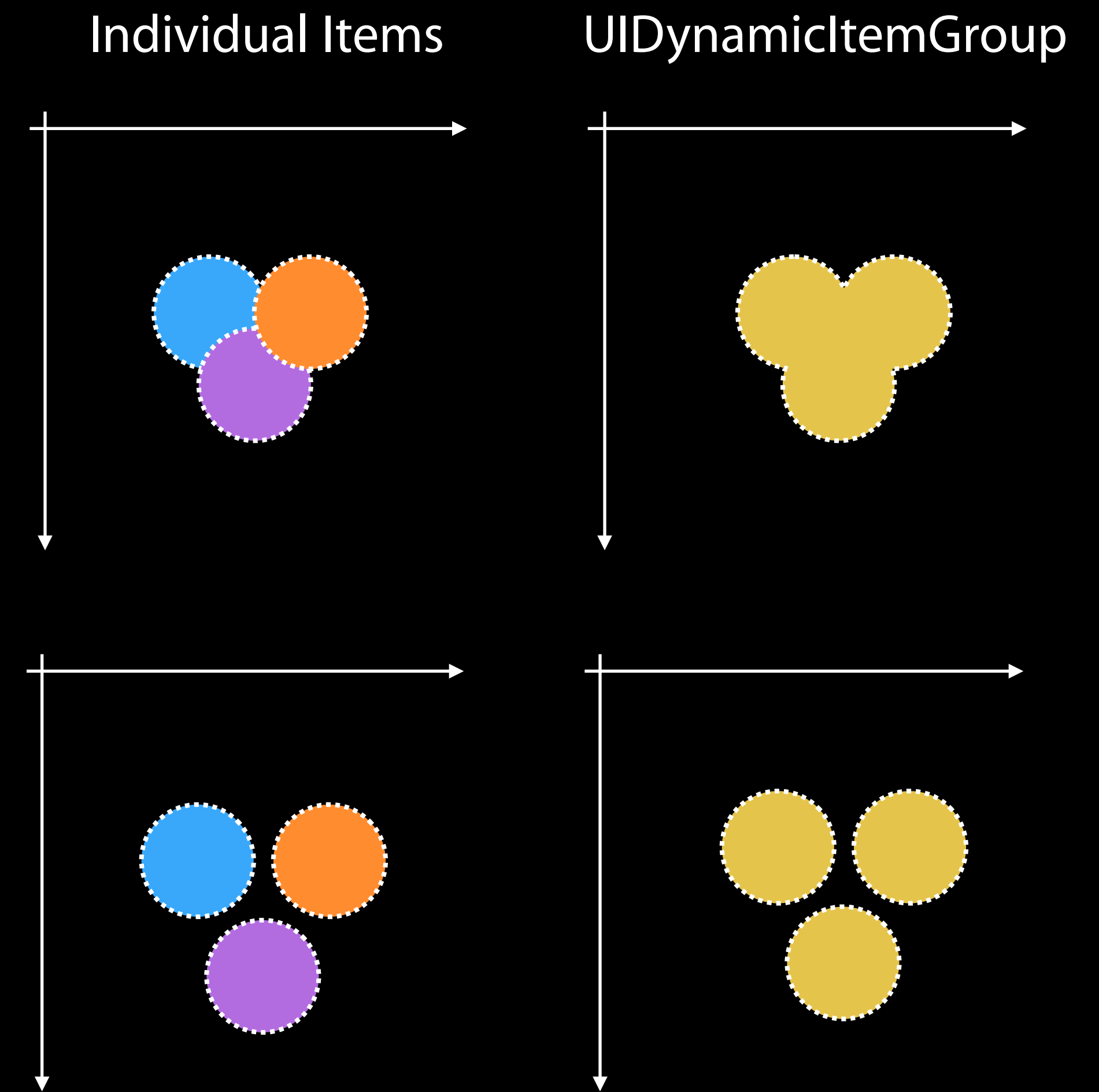
NEW

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

NEW

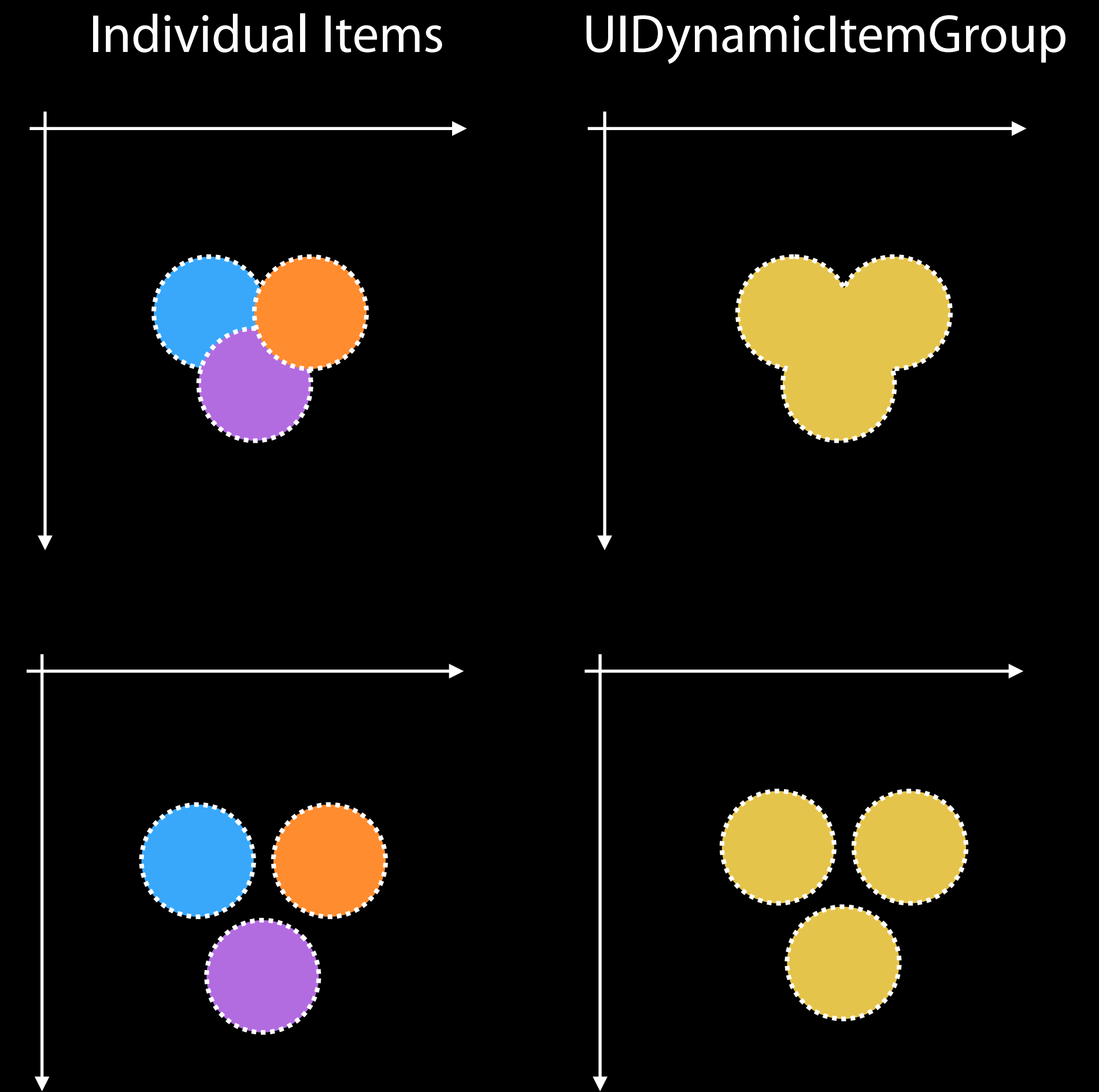
Makes multiple dynamic items behave as one

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

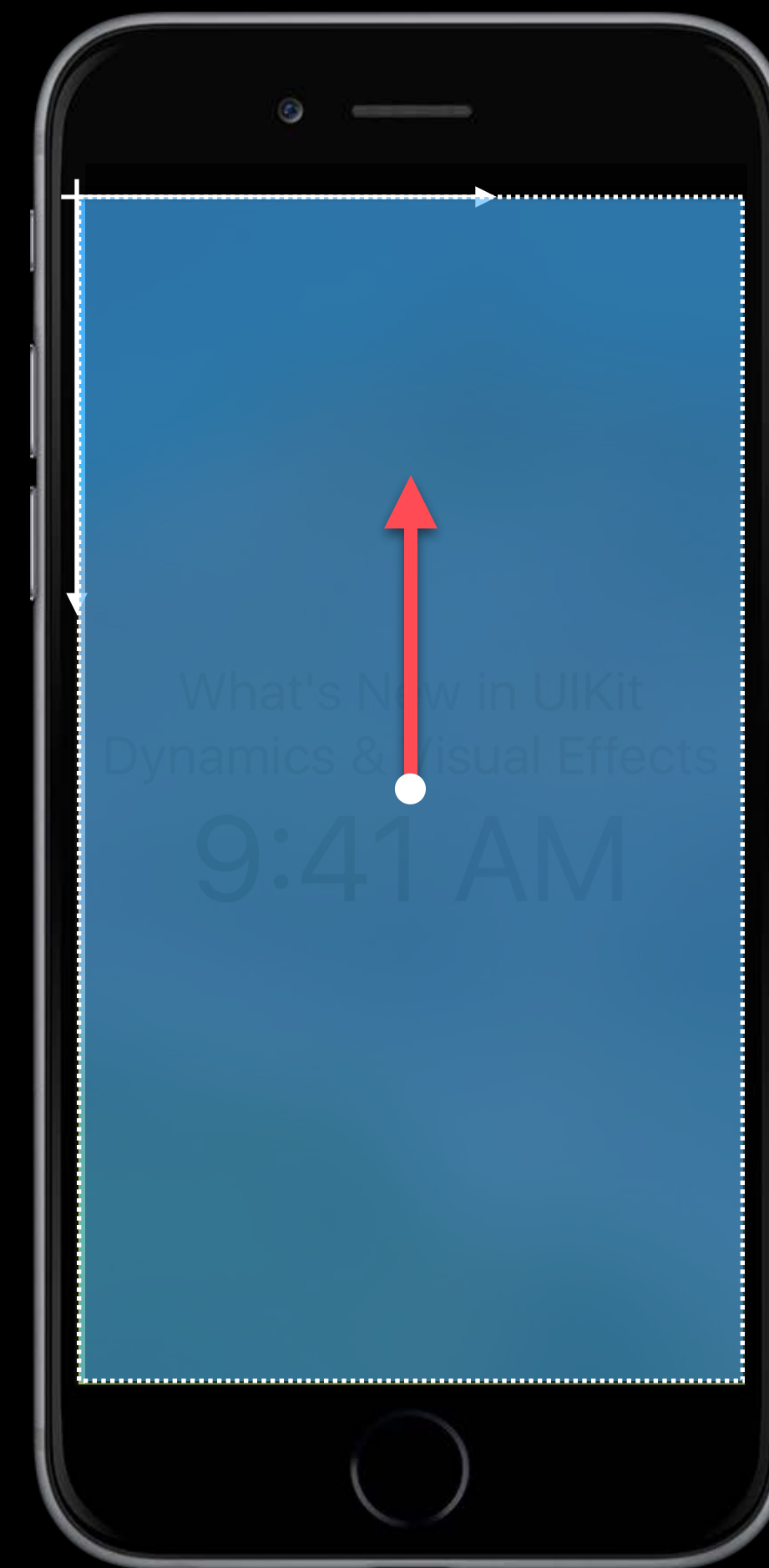
Force example



# UIKit Dynamics

## Force example

A force is modeled as a vector

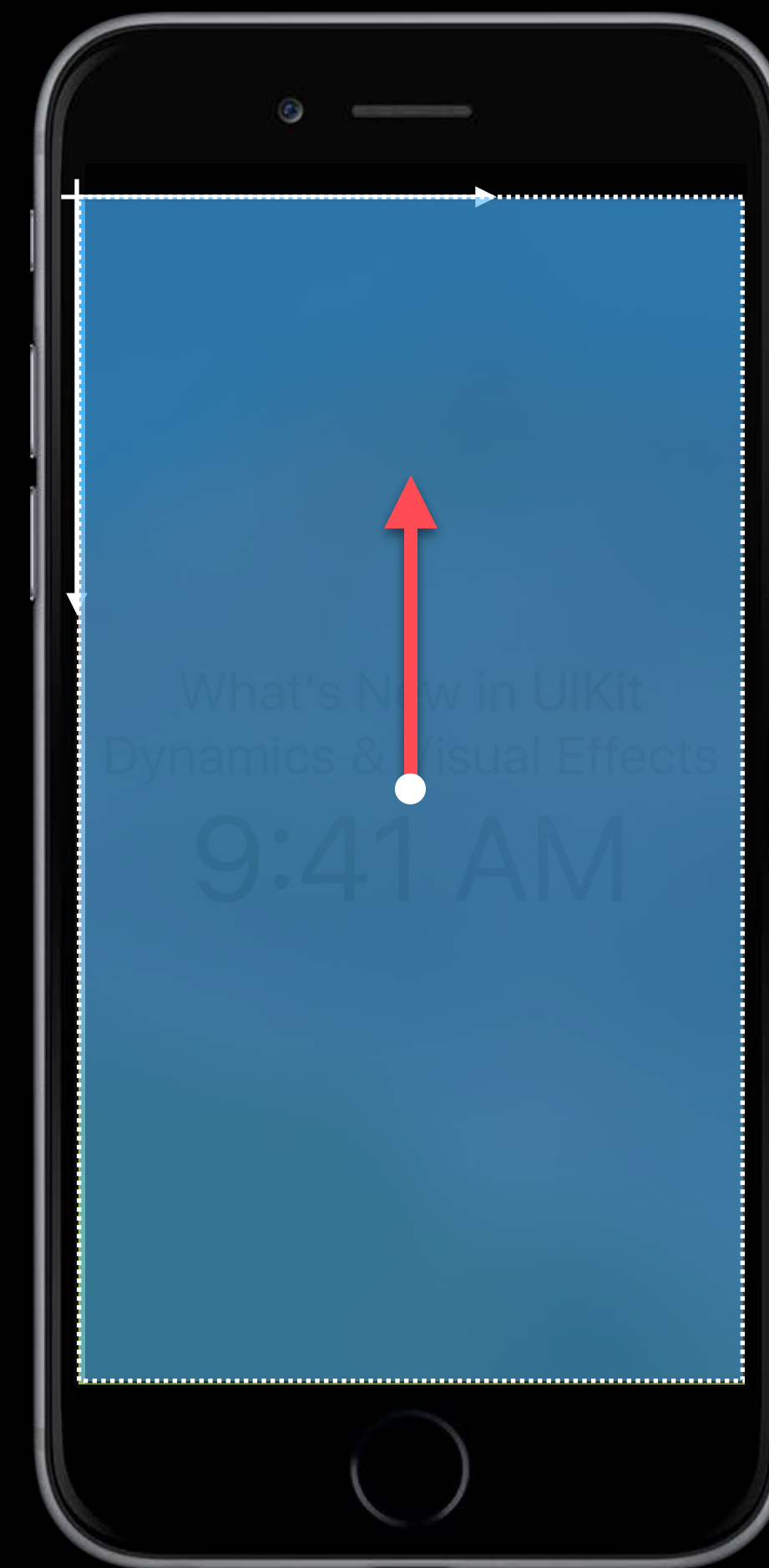


# UIKit Dynamics

## Force example

A force is modeled as a vector

- Length and direction



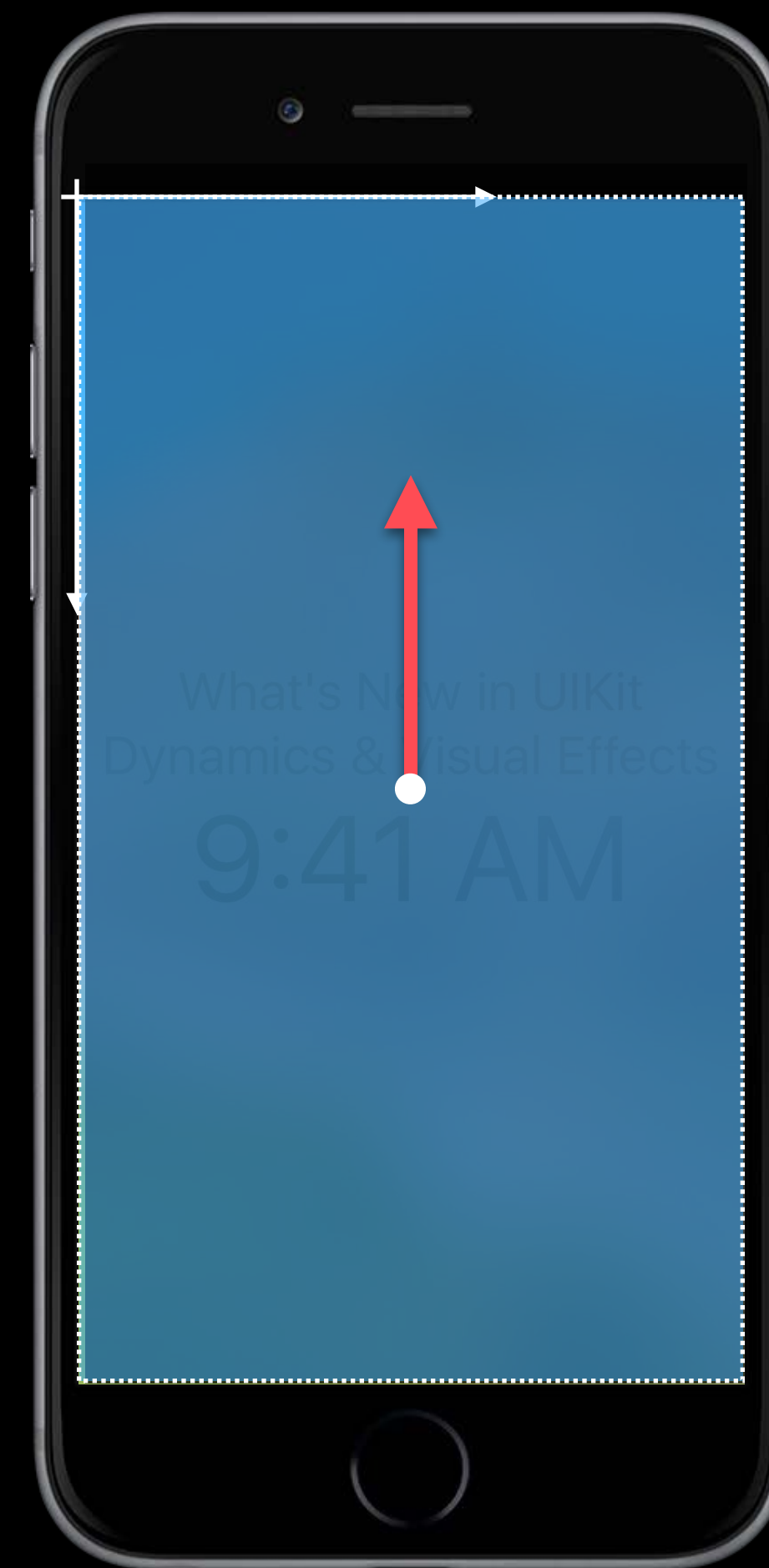
# UIKit Dynamics

## Force example

A force is modeled as a vector

- Length and direction

UIPushBehavior applies a force





# UIKit Dynamics

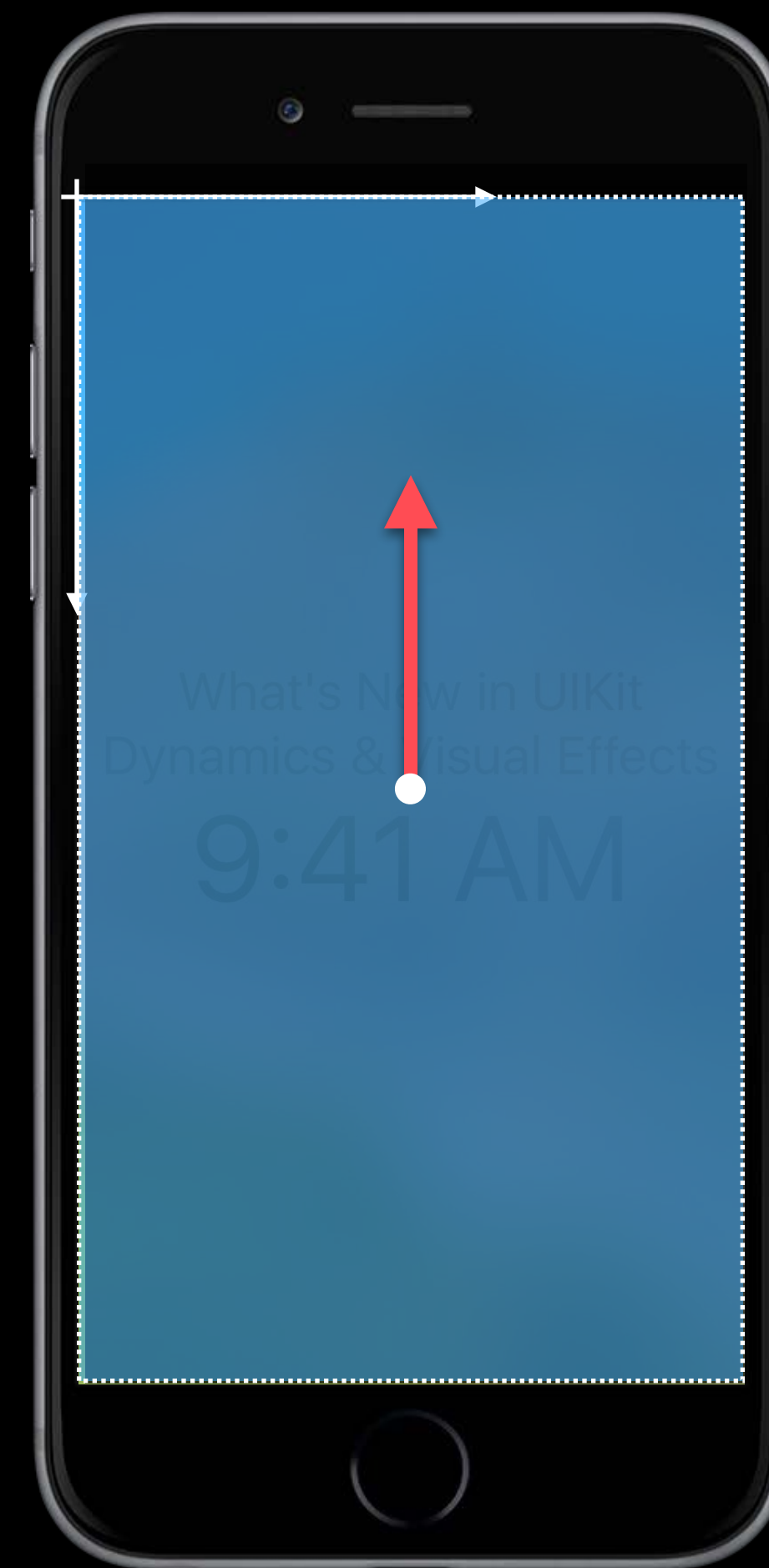
## Force example

A force is modeled as a vector

- Length and direction

UIPushBehavior applies a force

- Continuous



# UIKit Dynamics

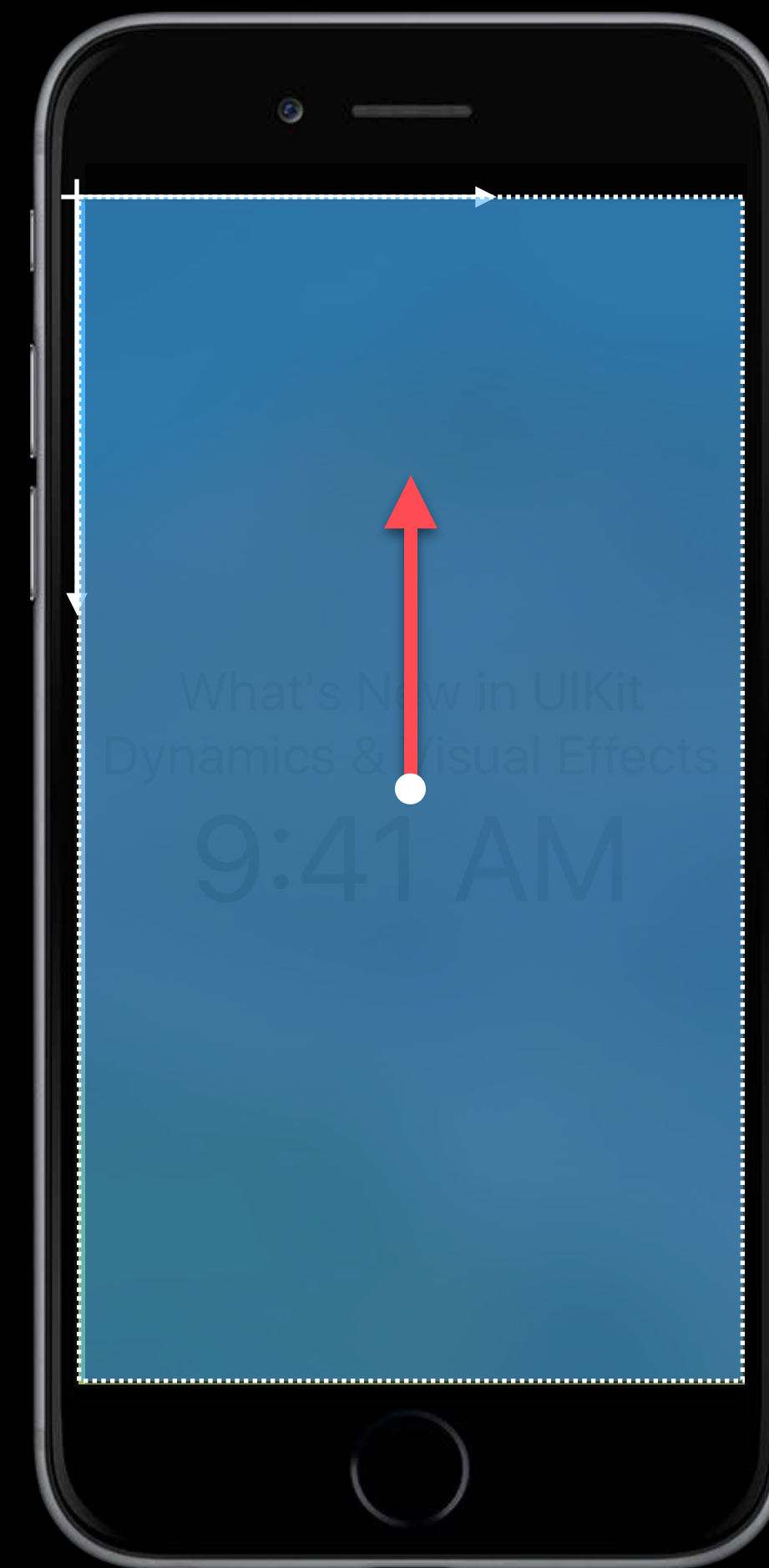
## Force example

A force is modeled as a vector

- Length and direction

UIPushBehavior applies a force

- Continuous
- Instantaneous



# UIKit Dynamics

## Force example

A force is modeled as a vector

- Length and direction

UIPushBehavior applies a force

- Continuous
- Instantaneous



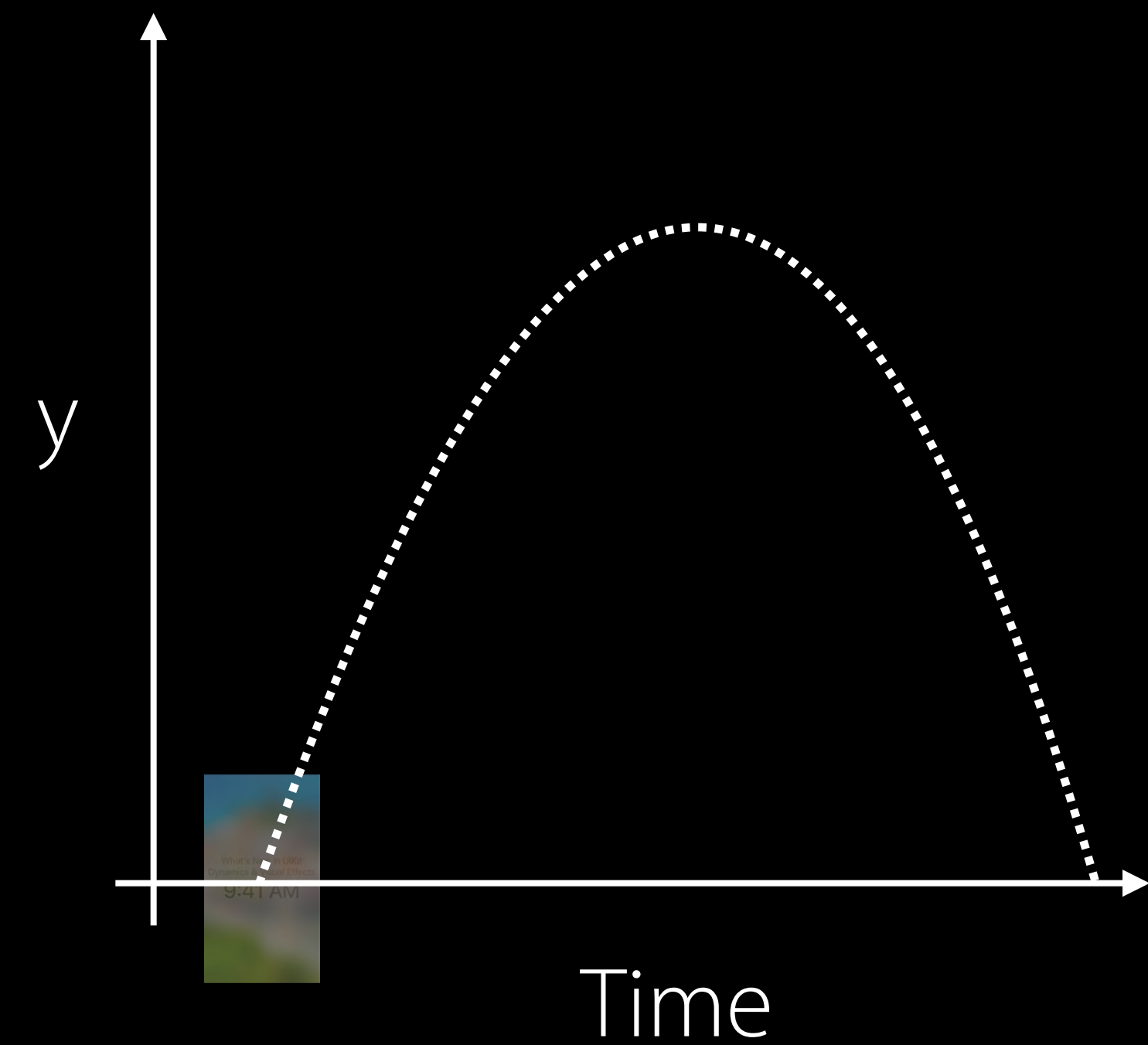
# UIKit Dynamics

Force example

# UIKit Dynamics

## Force example

Vertical motion over time of the sliding view

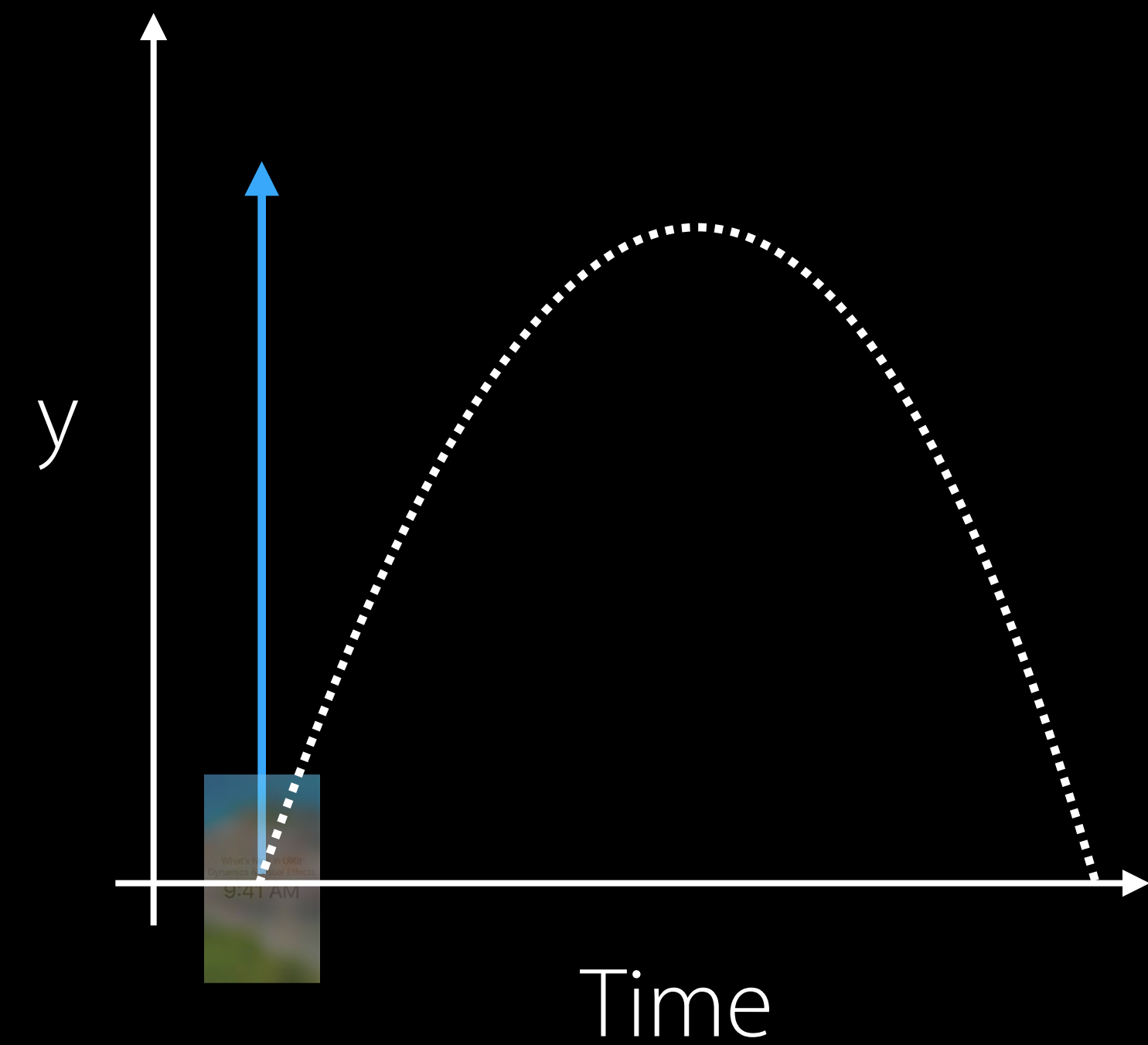


# UIKit Dynamics

## Force example

Vertical motion over time of the sliding view

Initial Impulse applied



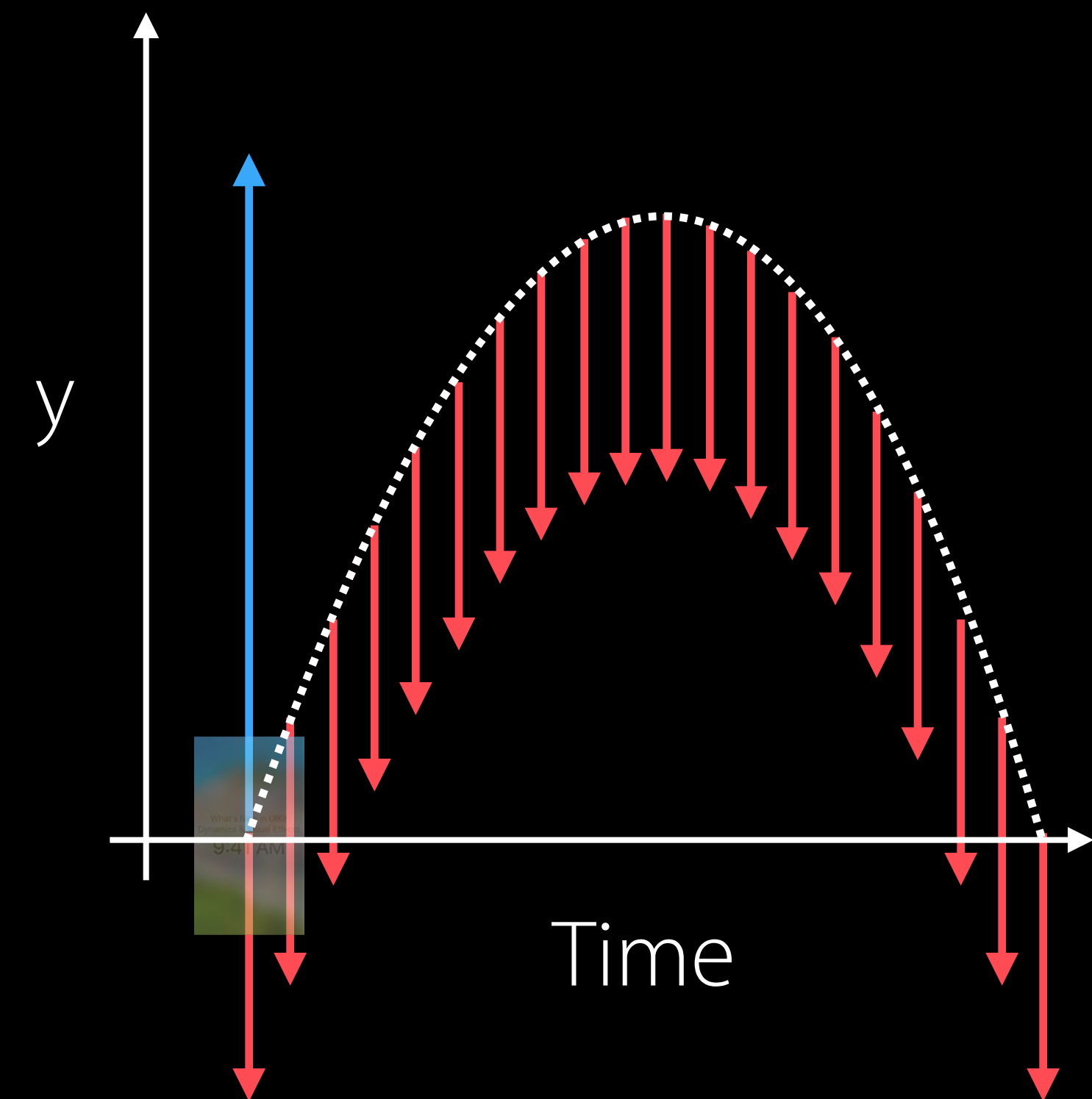
# UIKit Dynamics

## Force example

Vertical motion over time of the sliding view

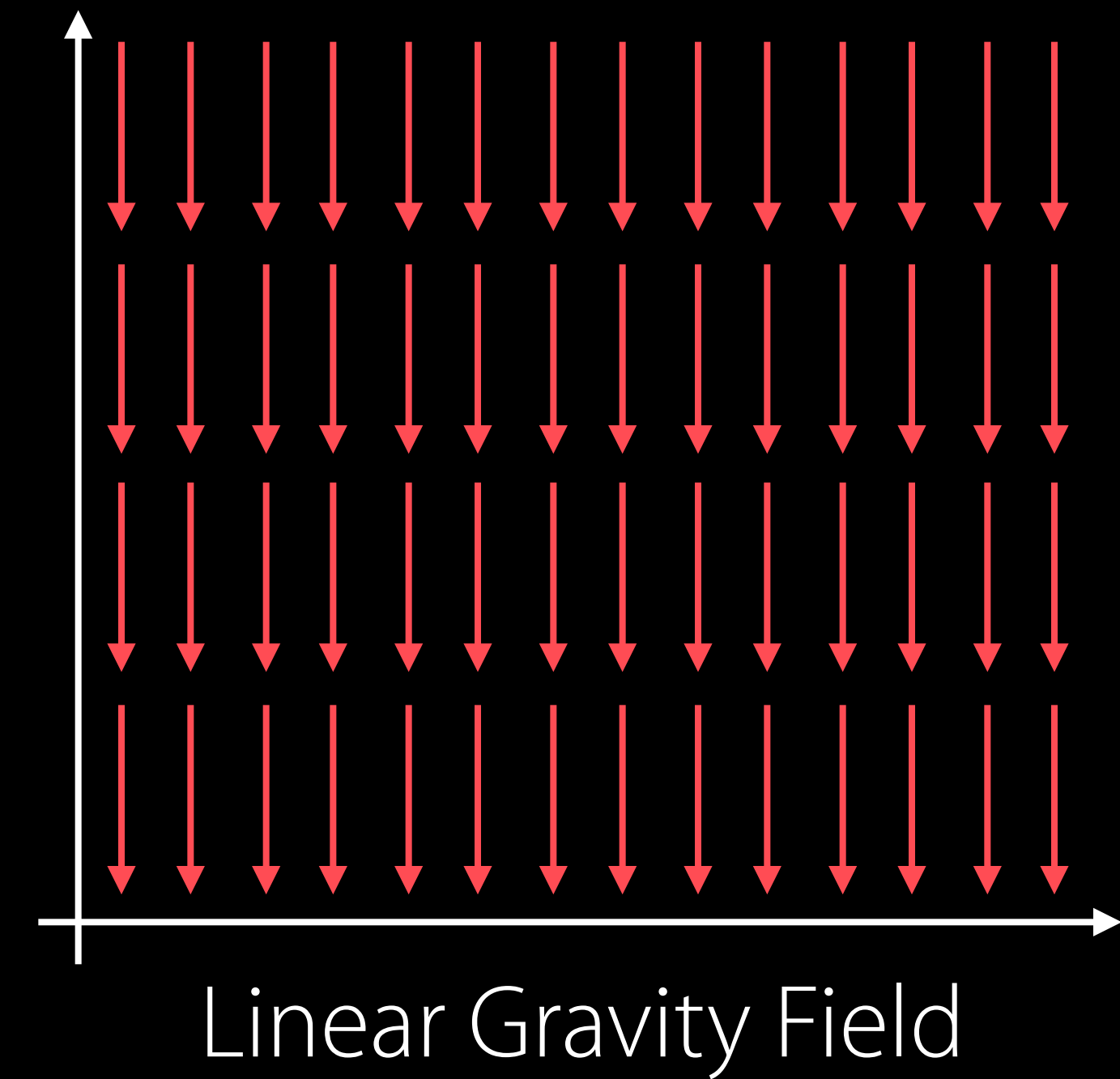
Initial Impulse applied

Gravity causes the view's velocity to slow and eventually reverse direction



# Physics Fields

## Linear Gravity as a field

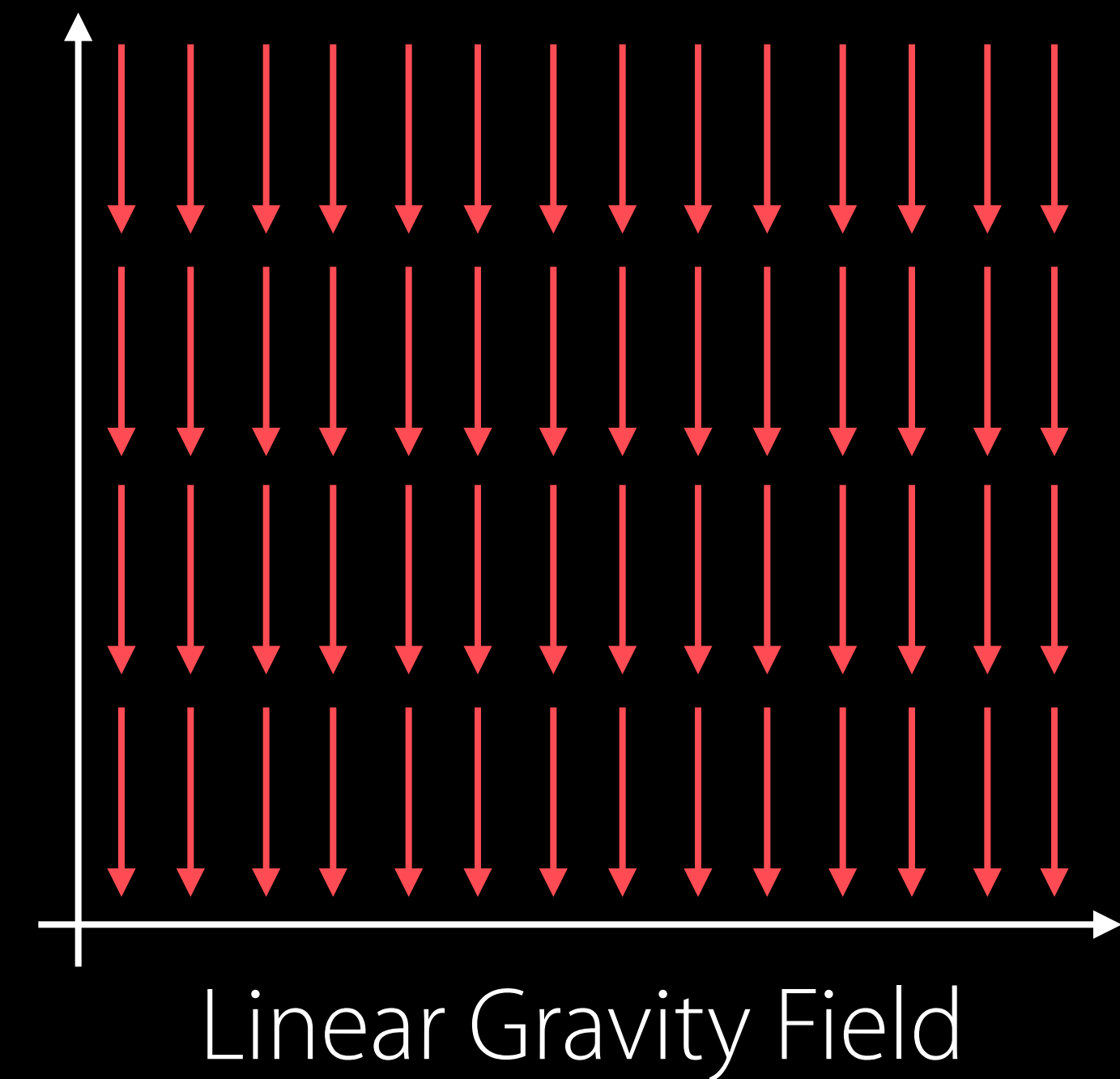




# Physics Fields

## Linear Gravity as a field

A function that assigns a vector to each point in a domain



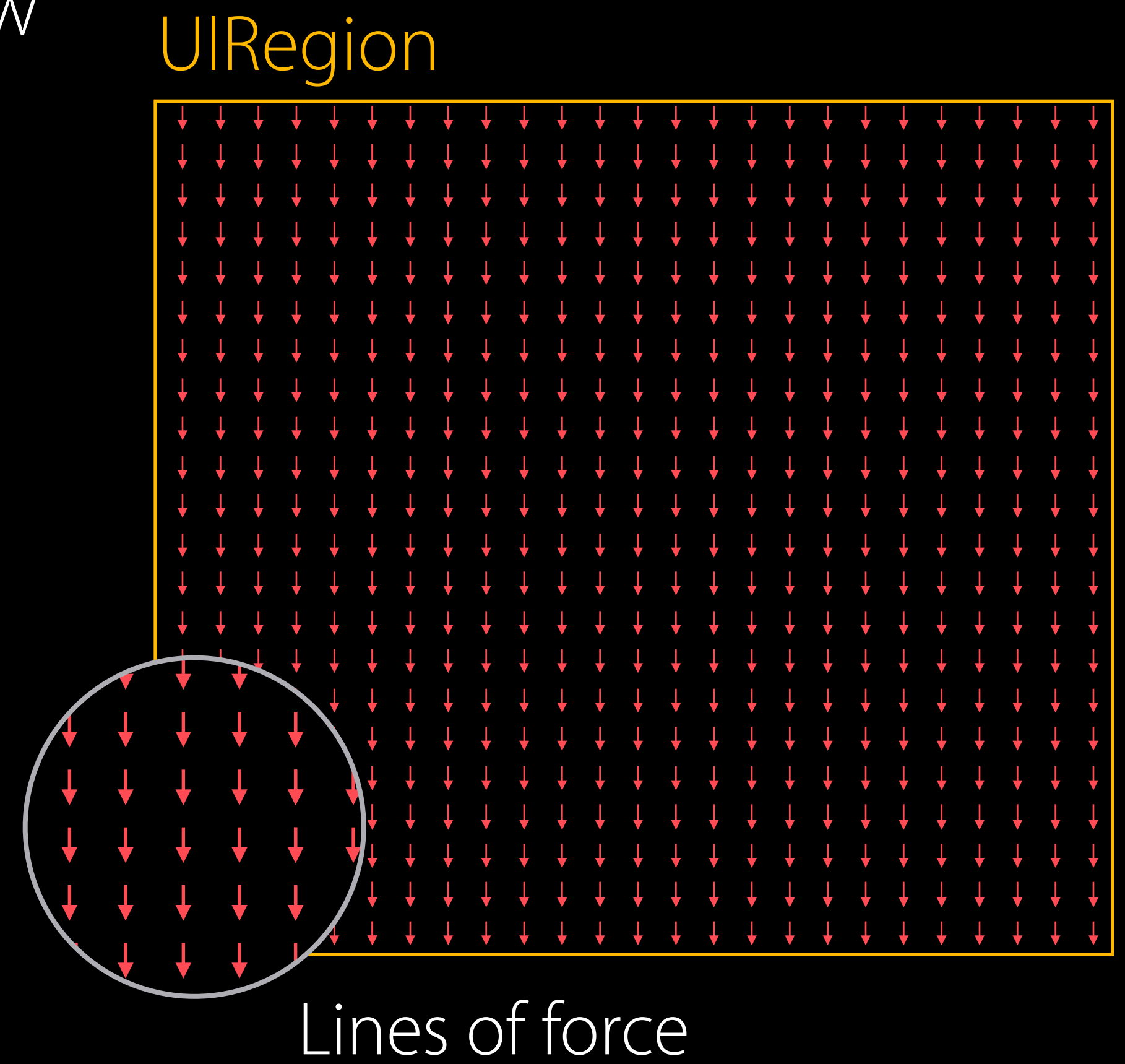
# UITextFieldBehavior

NEW

# UIFieldBehavior

NEW

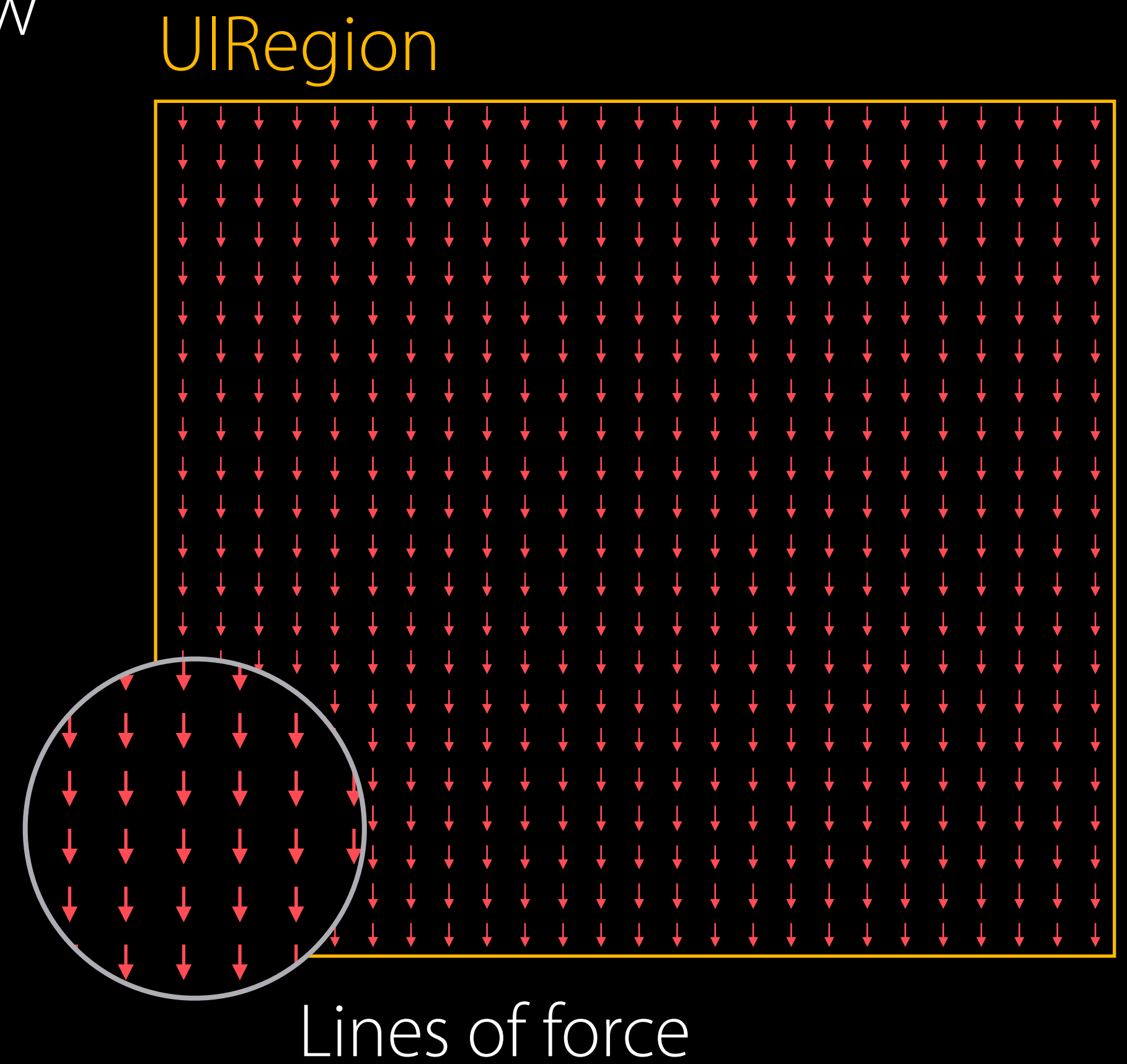
UIFieldBehavior can be added to a **region** of your view



# UIFieldBehavior

NEW

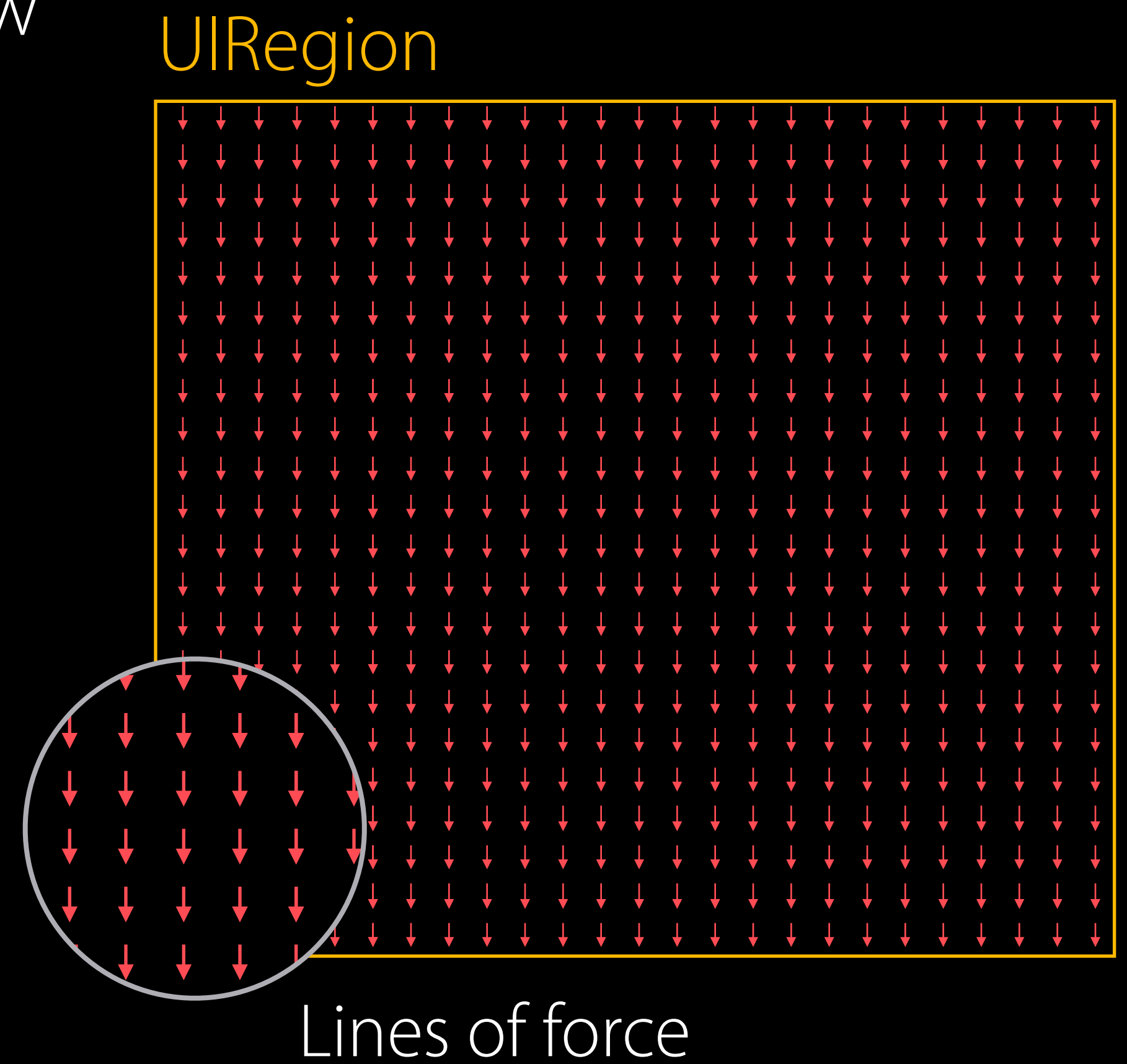
UIFieldBehavior can be added to a **region** of your view  
The field is evaluated at each point within the region



# UIFieldBehavior

NEW

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

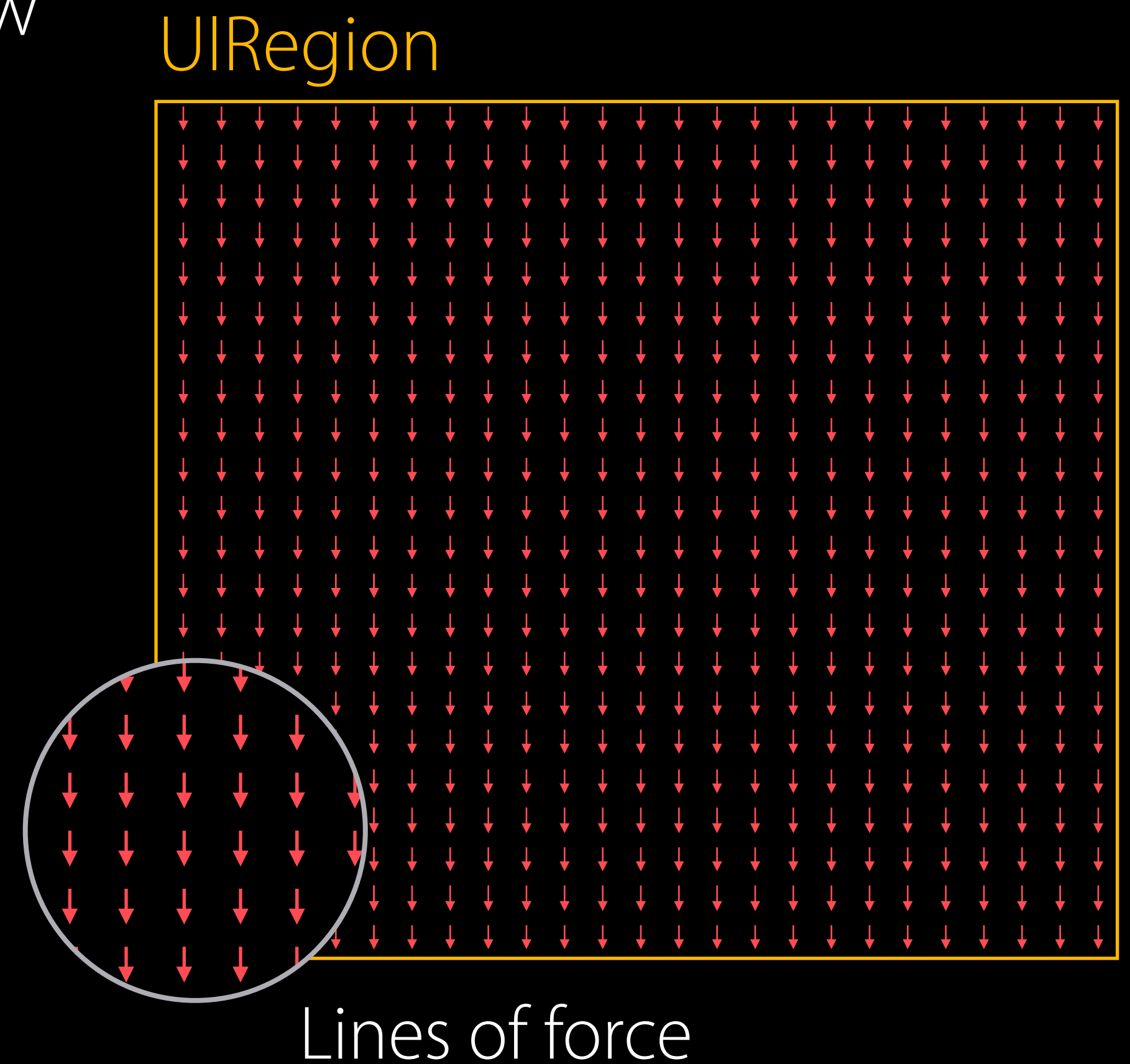
NEW

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

NEW

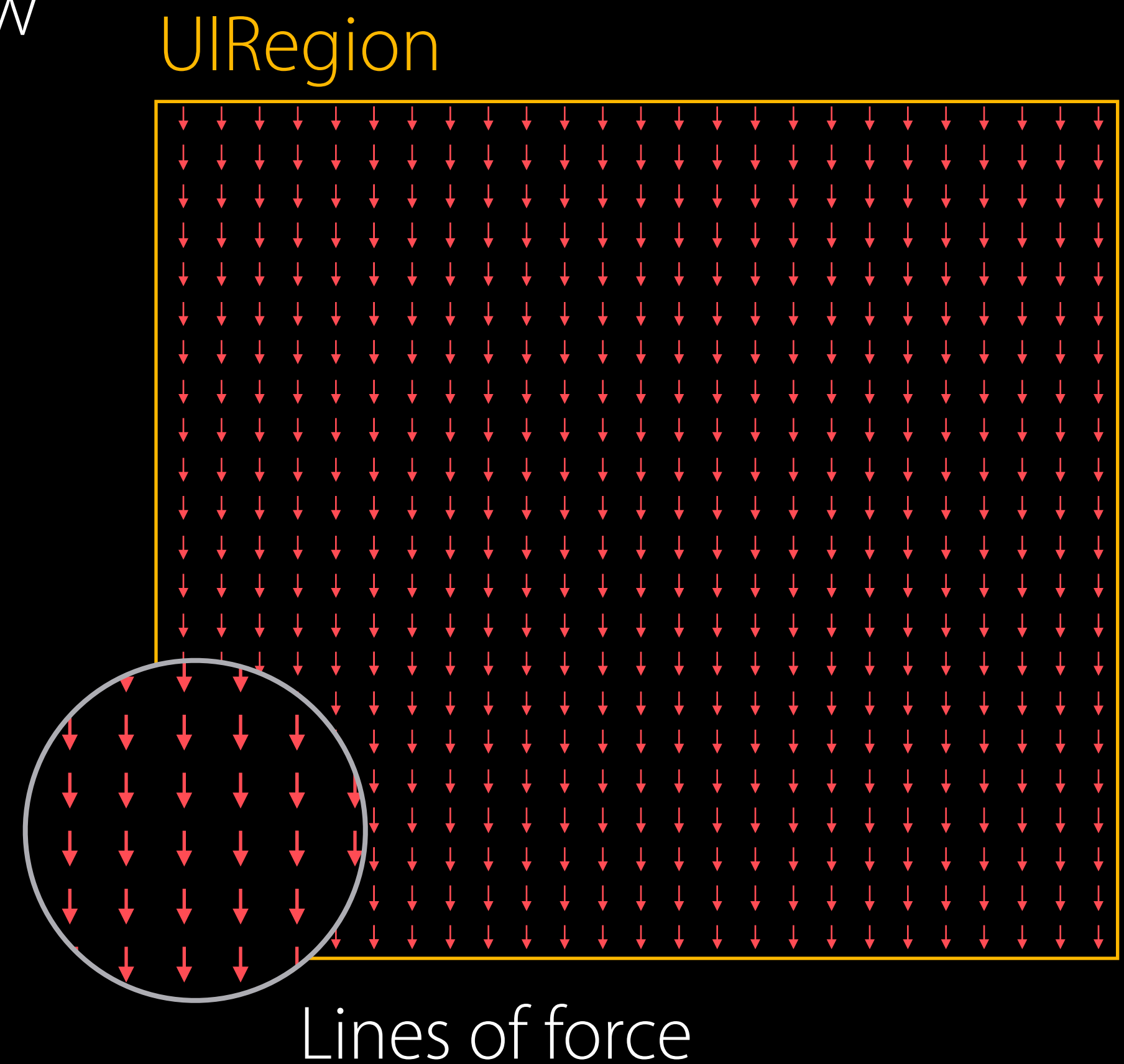
UIFieldBehavior can be added to a **region** of your view

The field is evaluated at each point within the region

Resulting forces are applied by the animator

UIGravityBehavior is a field already!

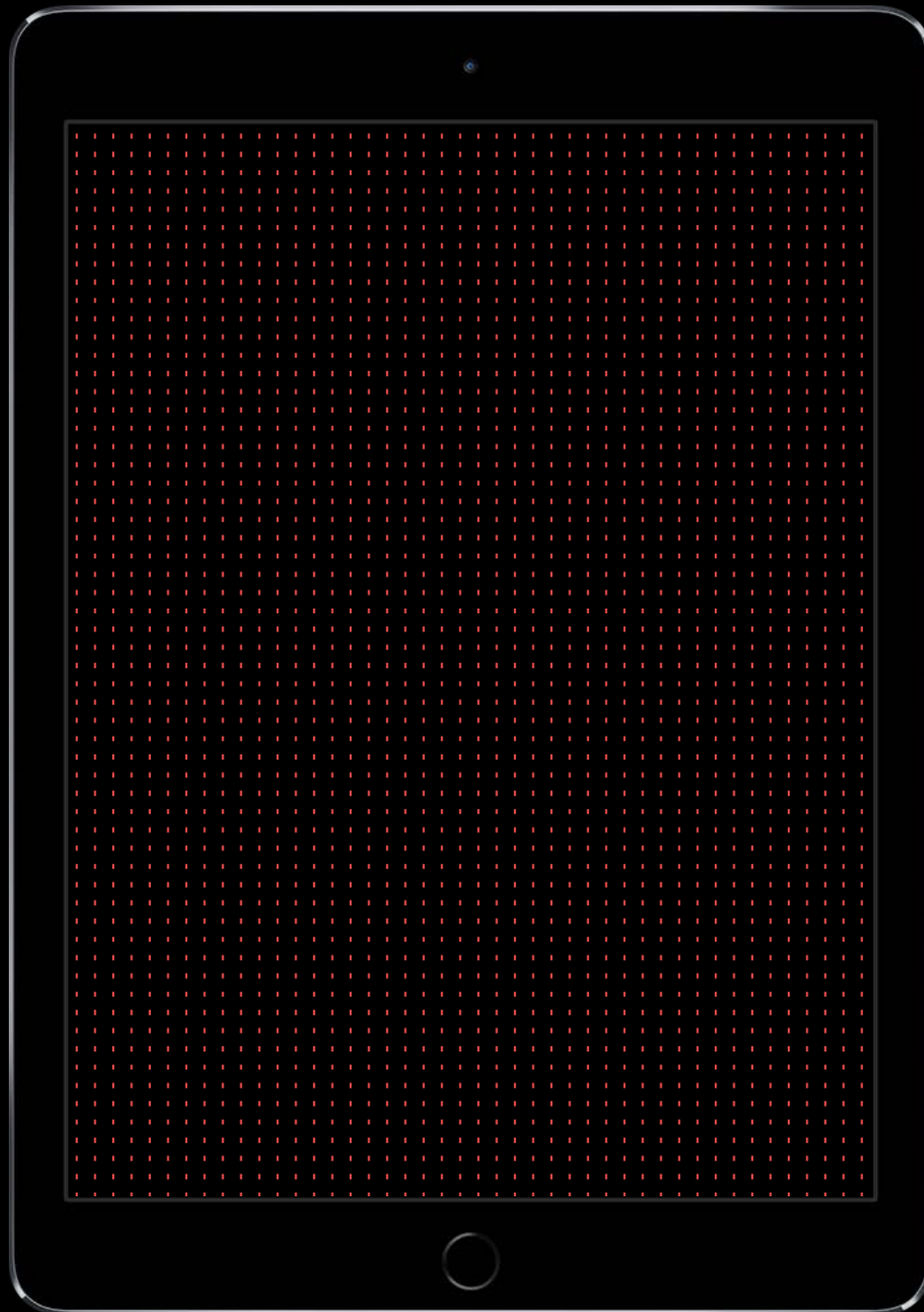
Simplified physics, well-tuned for performance;  
not useful for building interstellar space stations



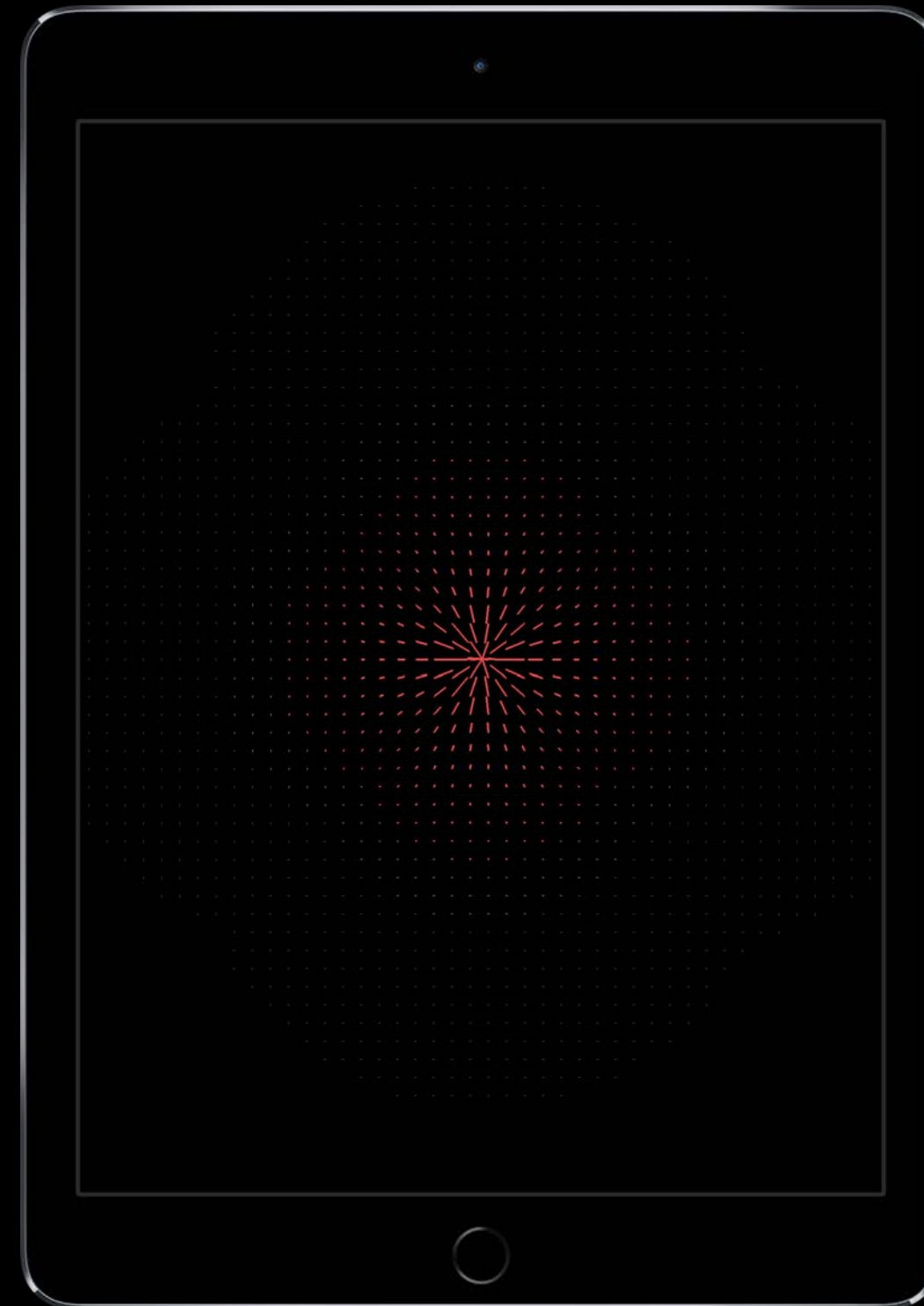
# UIFieldBehavior

NEW

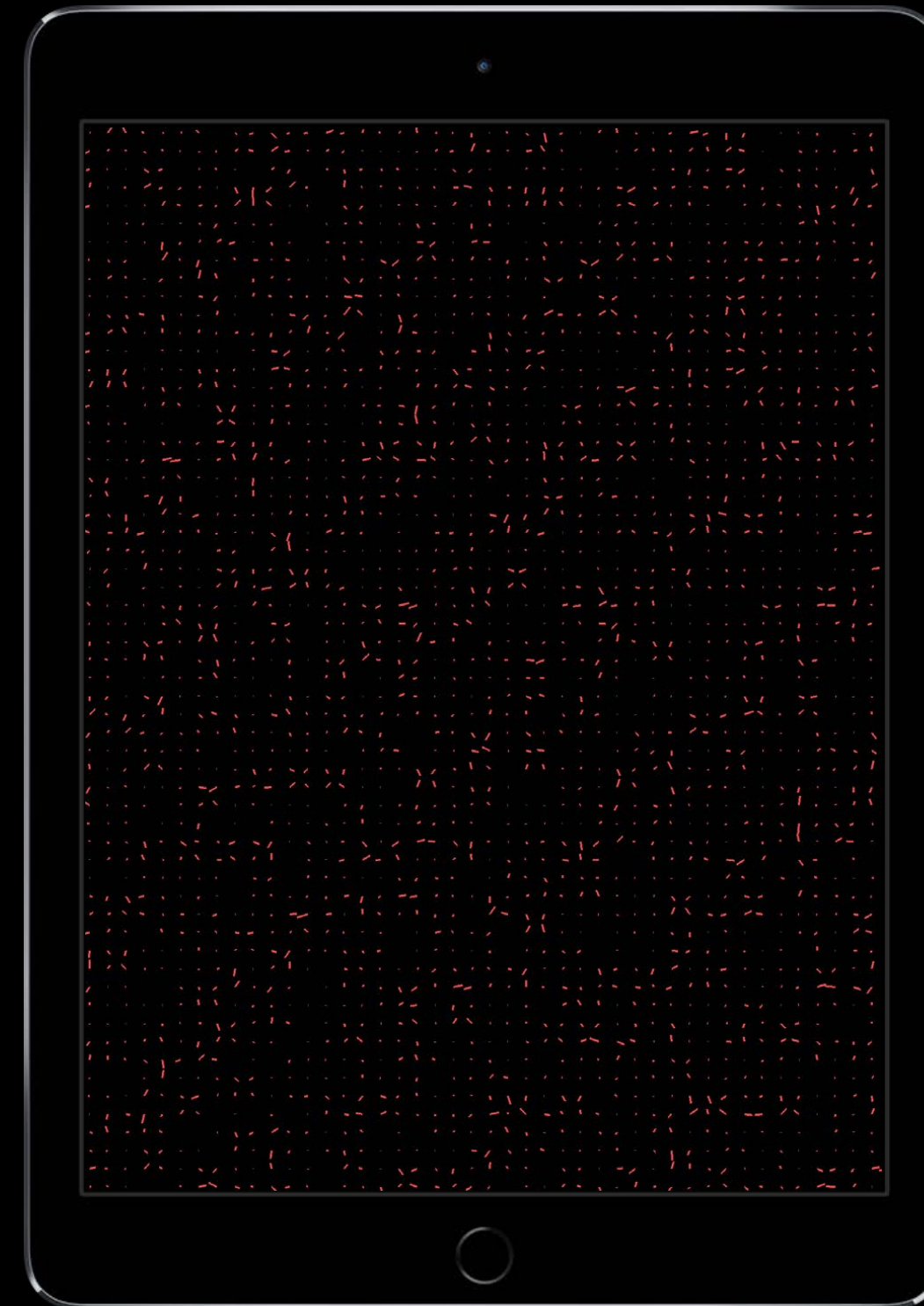
Linear Gravity



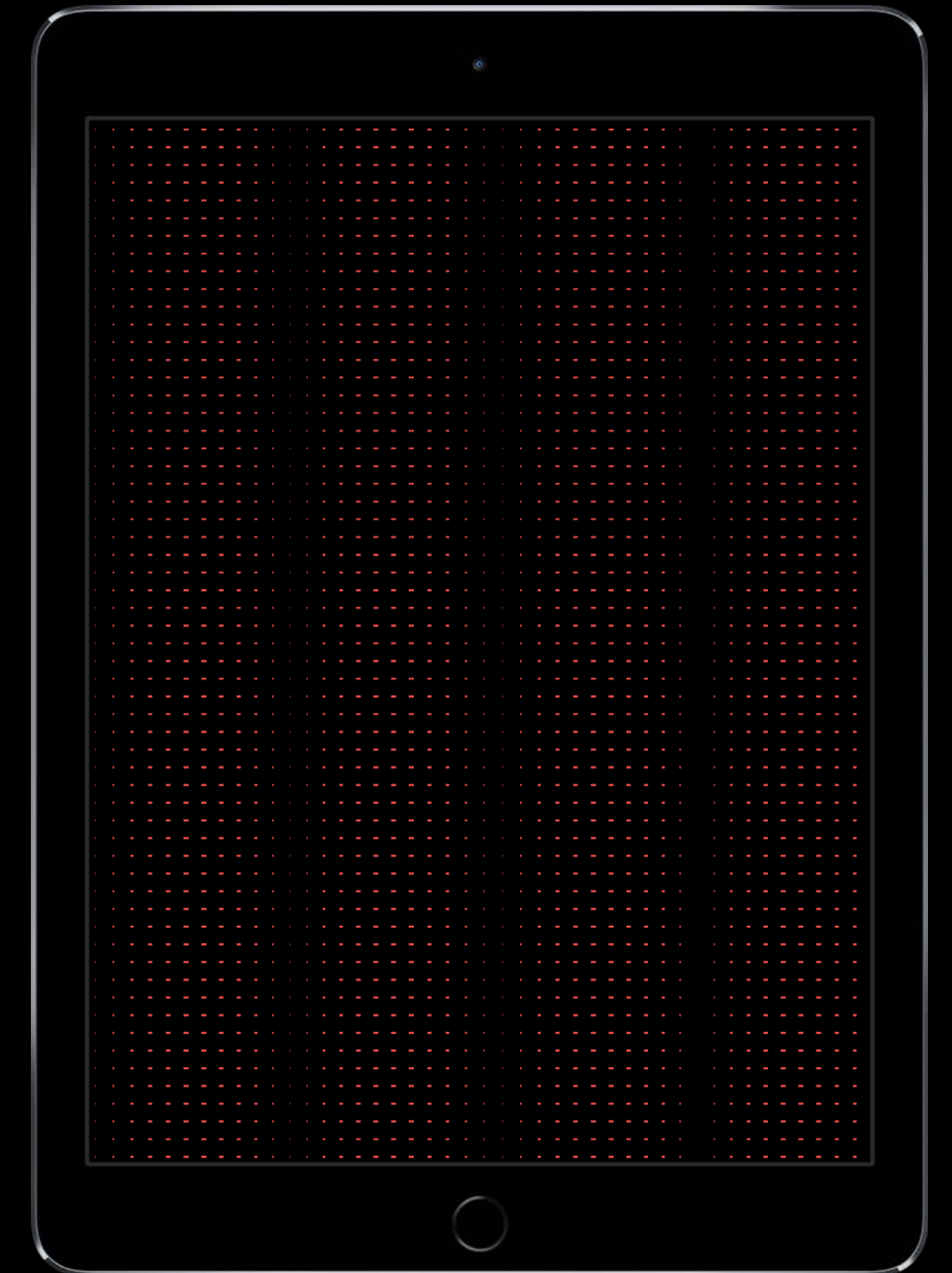
Radial Gravity



Noise



Custom



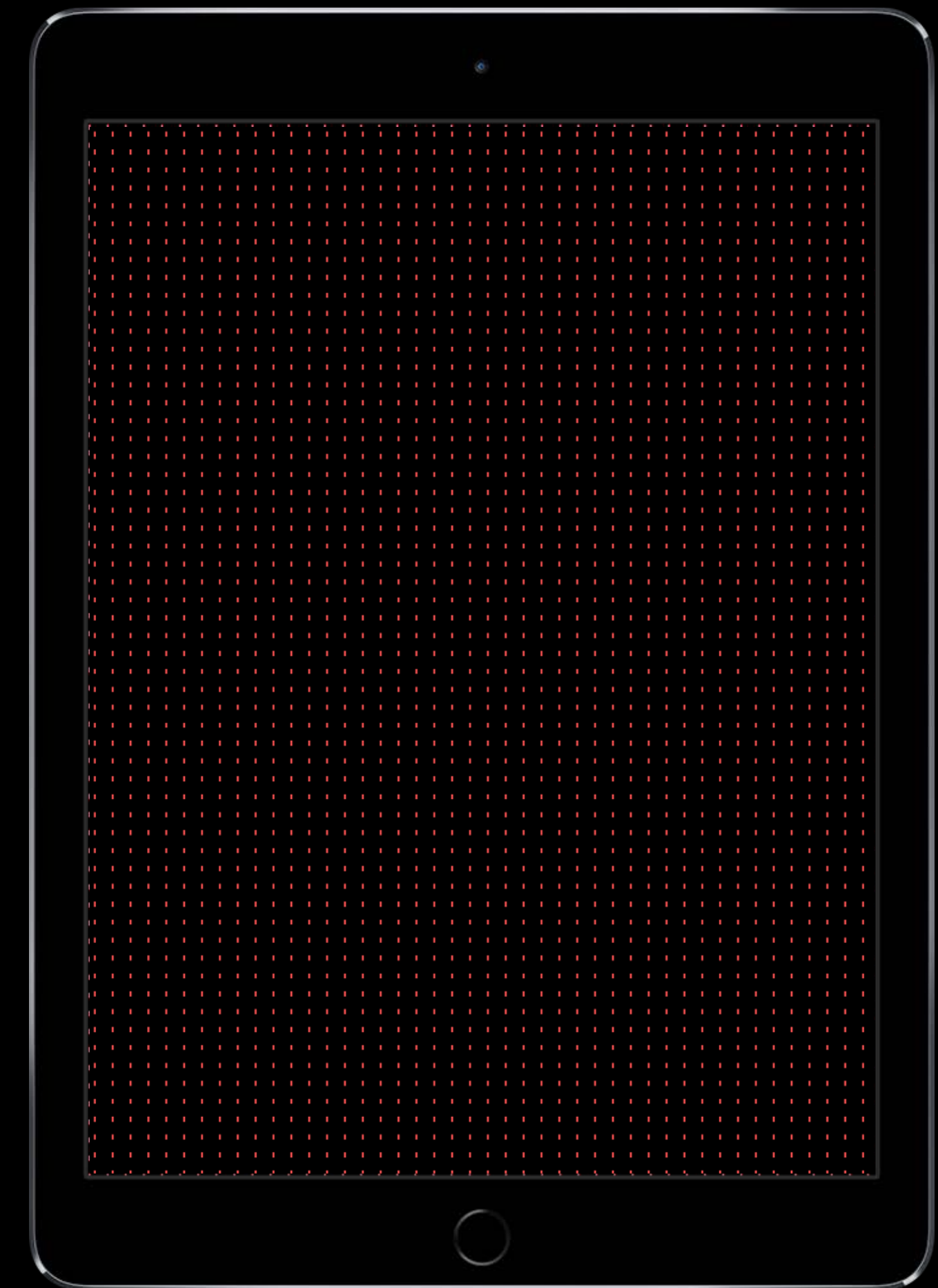
Drag and Velocity, Vortex, Turbulence, Spring, Electric and Magnetic



# UIFieldBehavior

Linear Gravity

NEW



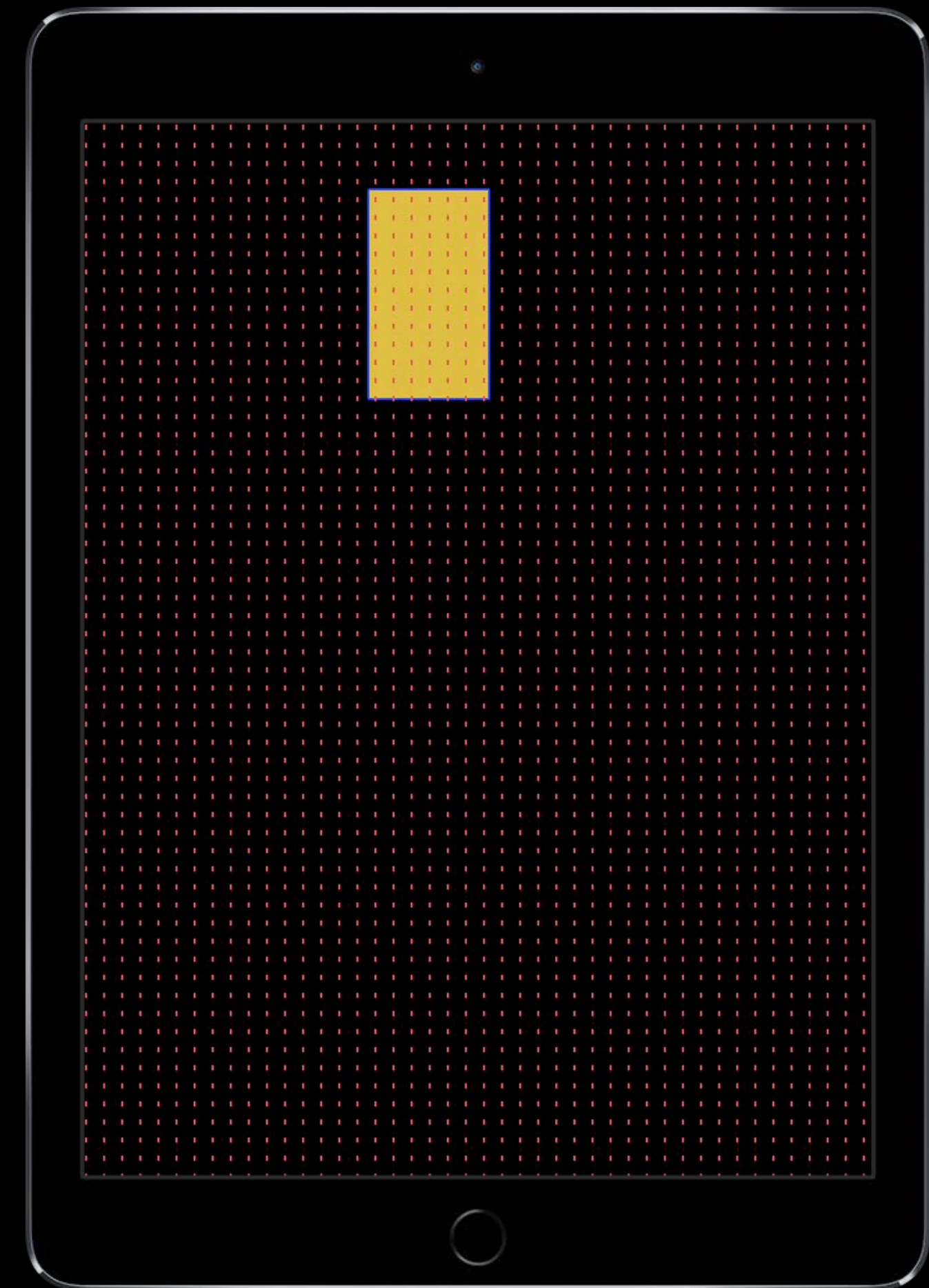
# UIFieldBehavior

## Linear Gravity

NEW

Field properties

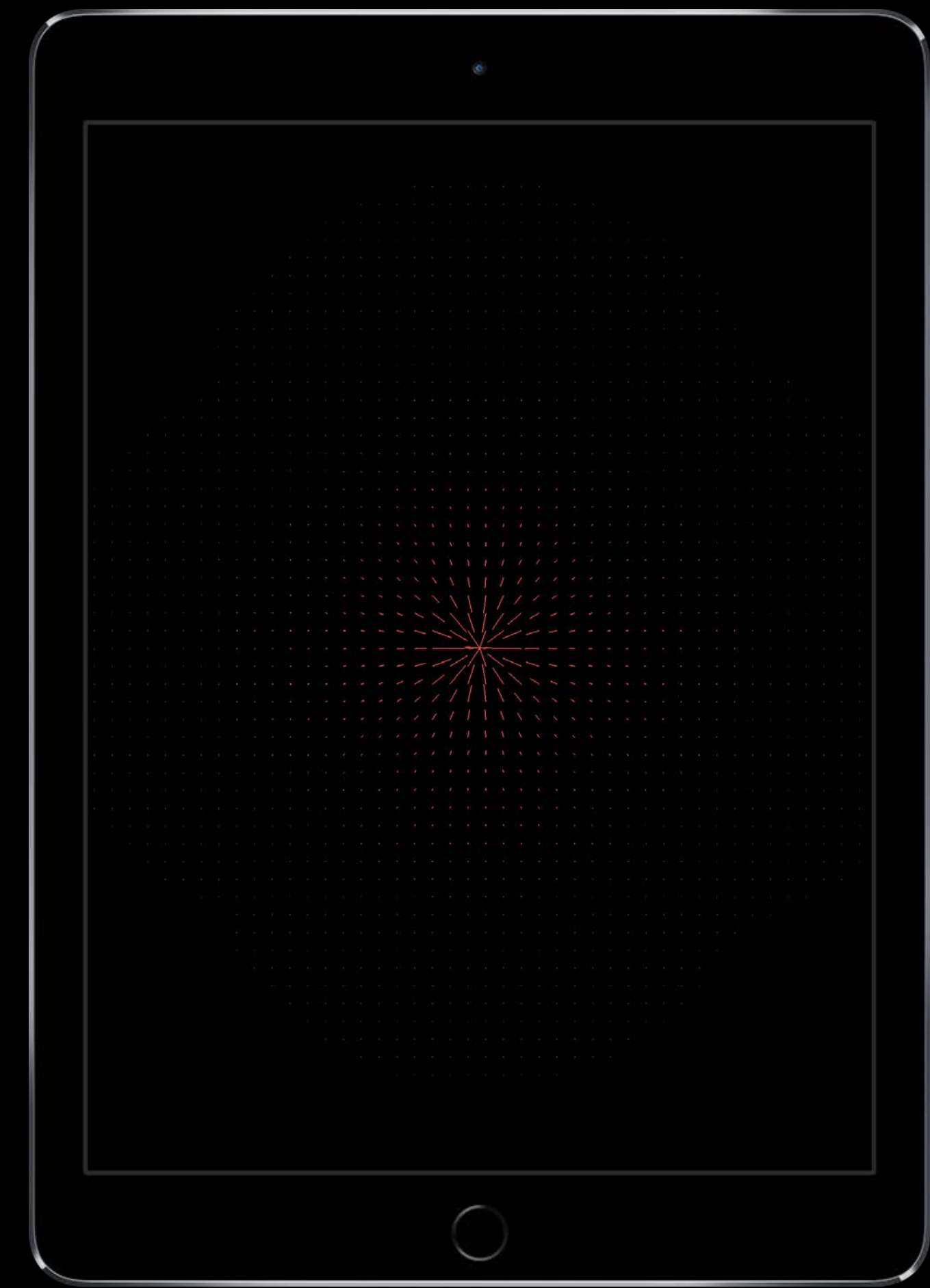
```
var region: UIRRegion  
var direction: CGVector  
var strength: CGFloat
```



# UIFieldBehavior

## Radial Gravity

NEW



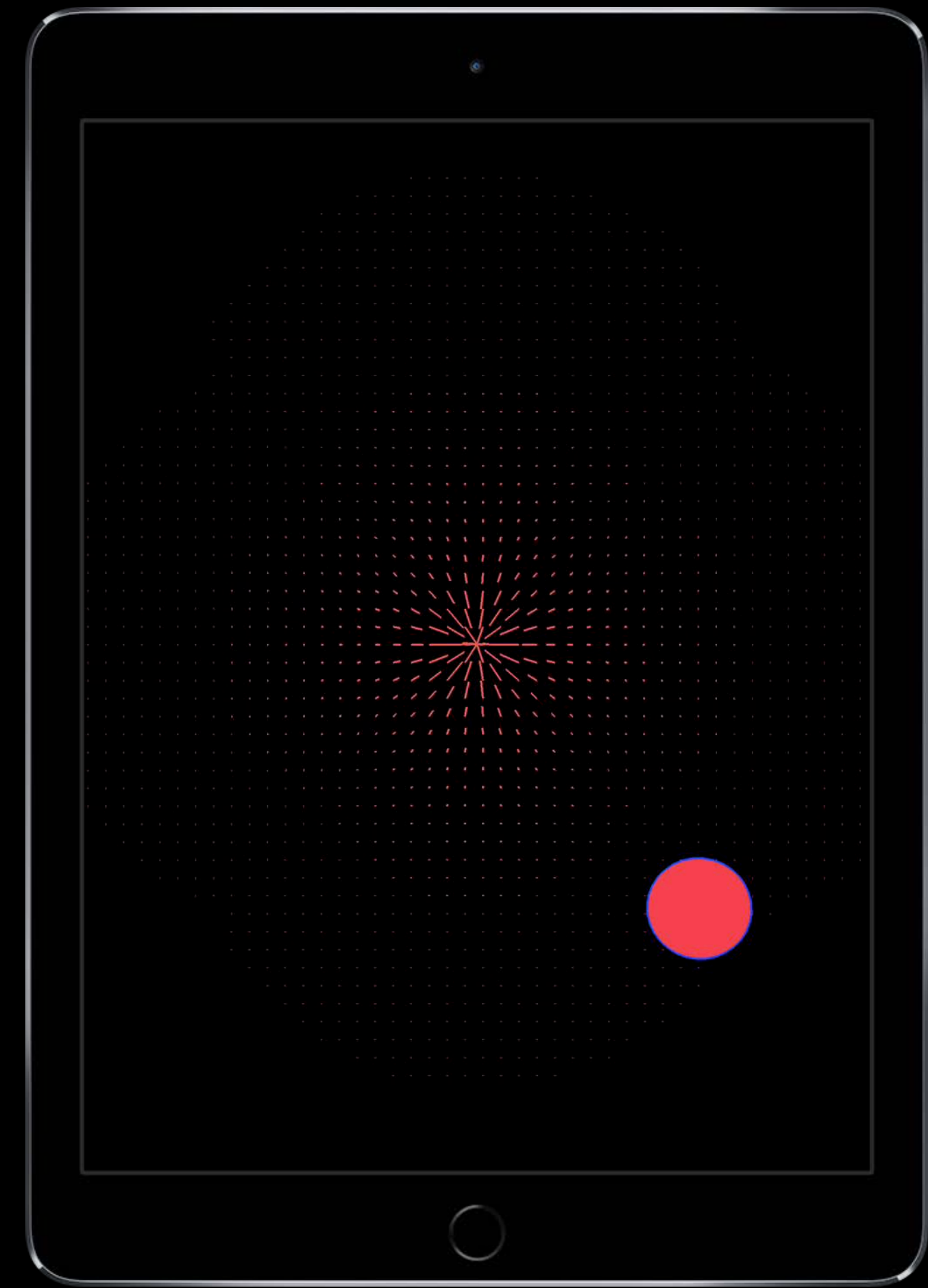
# UIFieldBehavior

## Radial Gravity

NEW

Field properties

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

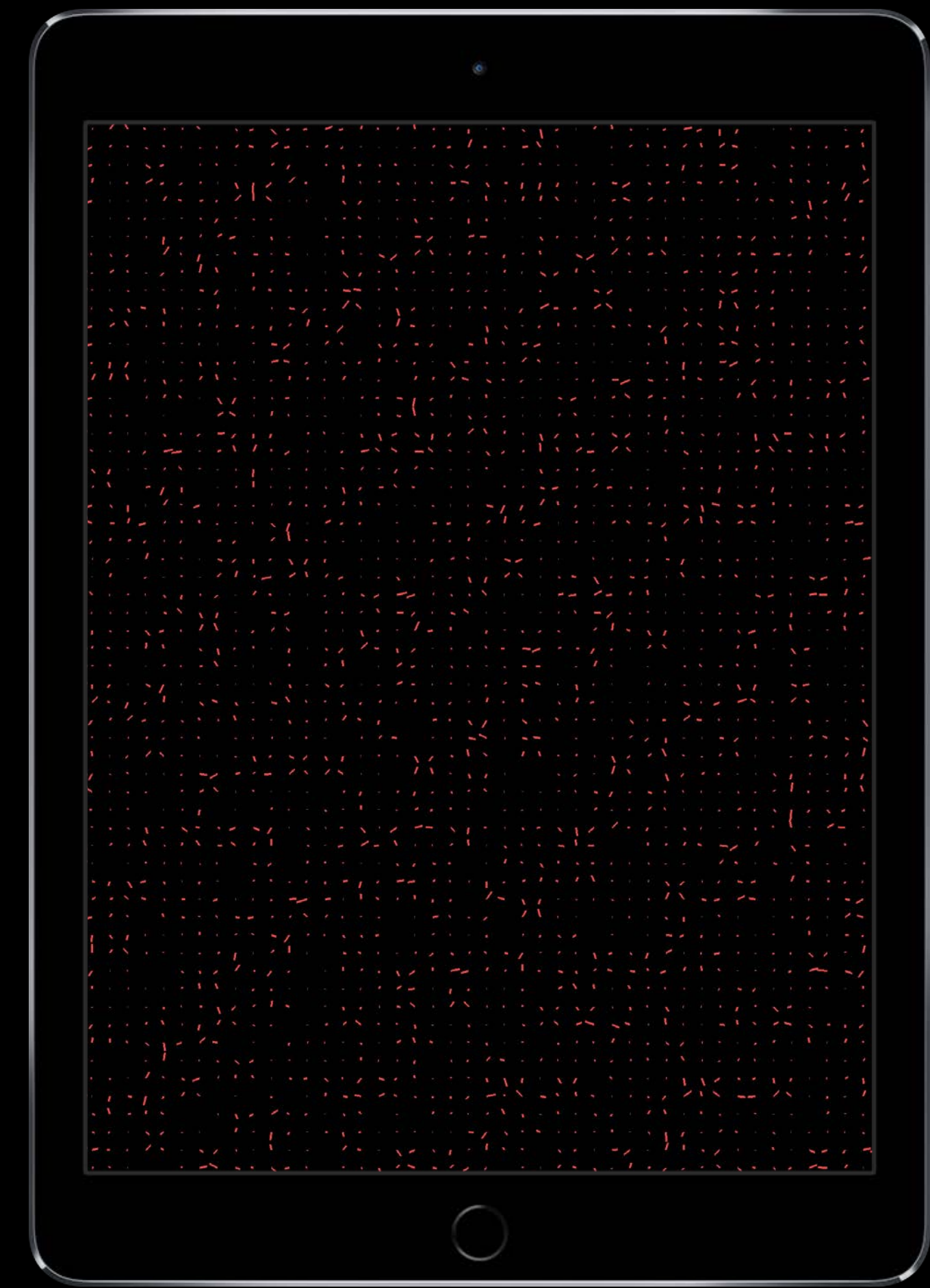




# UIFieldBehavior

Noise

NEW



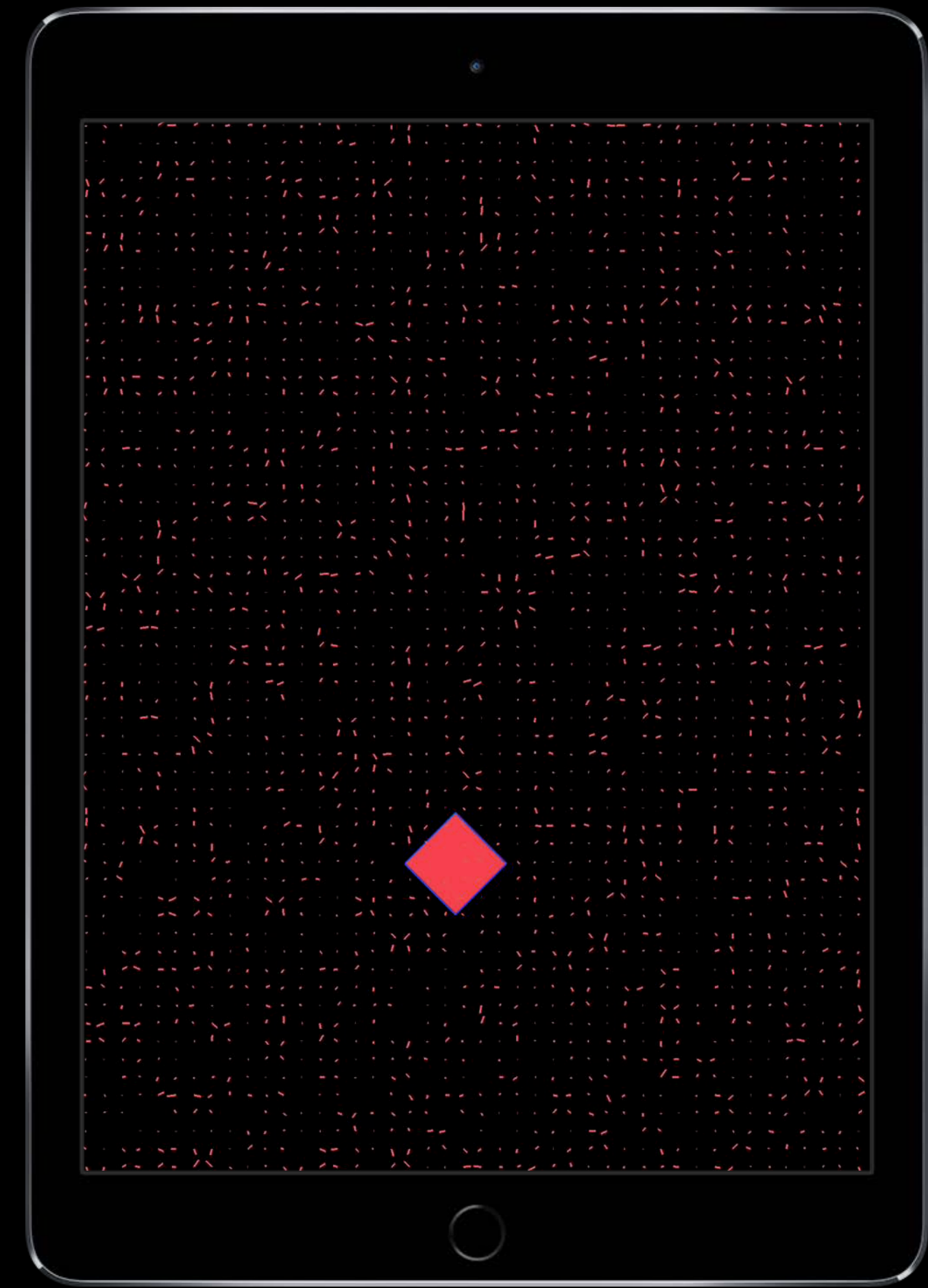
# UIFieldBehavior

NEW

## Noise

### Field properties

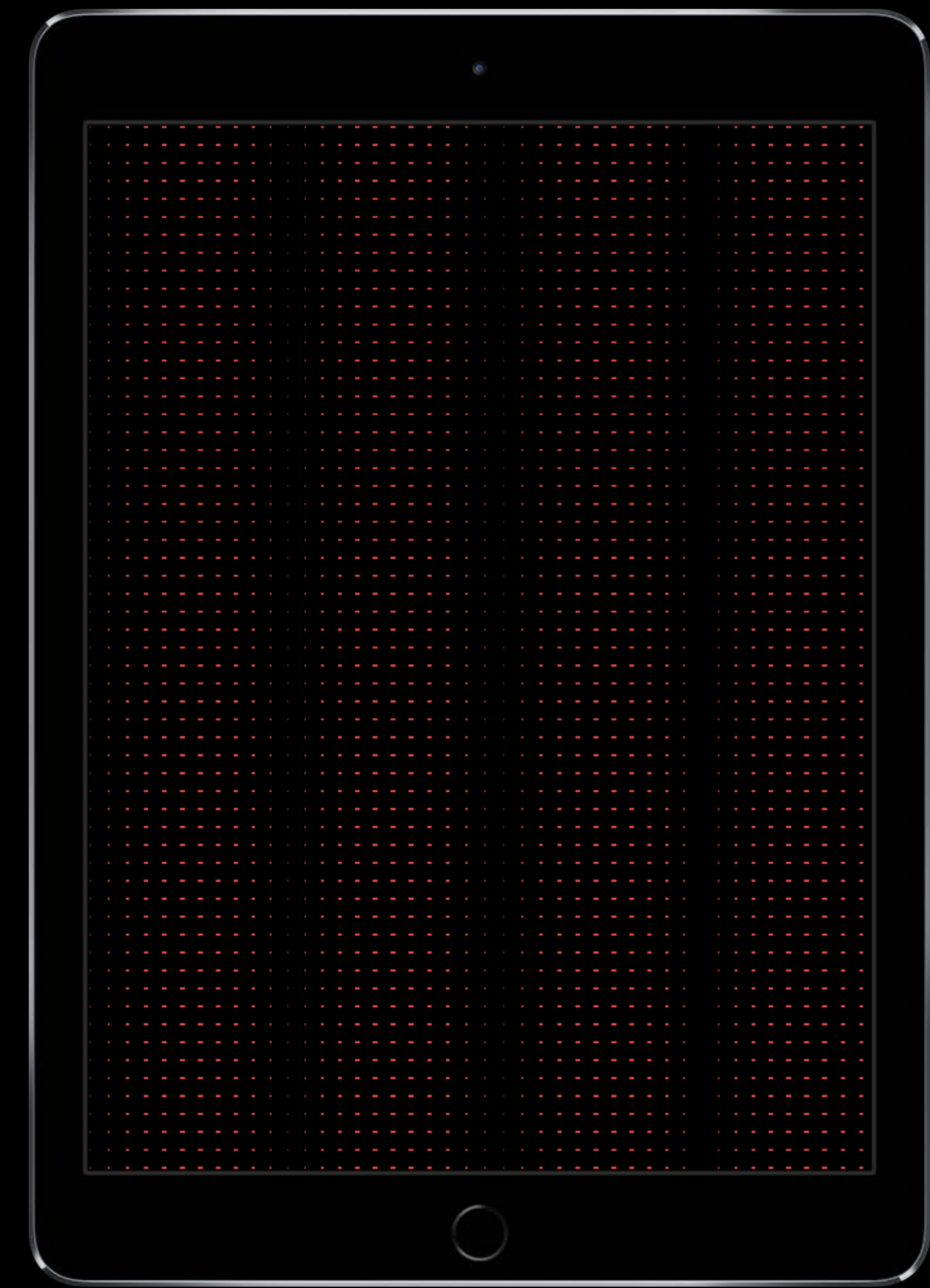
```
var region: UIRRegion  
var strength: CGFloat  
var smoothness: CGFloat  
var animationSpeed: CGFloat
```



# UITextFieldBehavior

Custom

NEW





# UIFieldBehavior

Custom

NEW

Field Sample

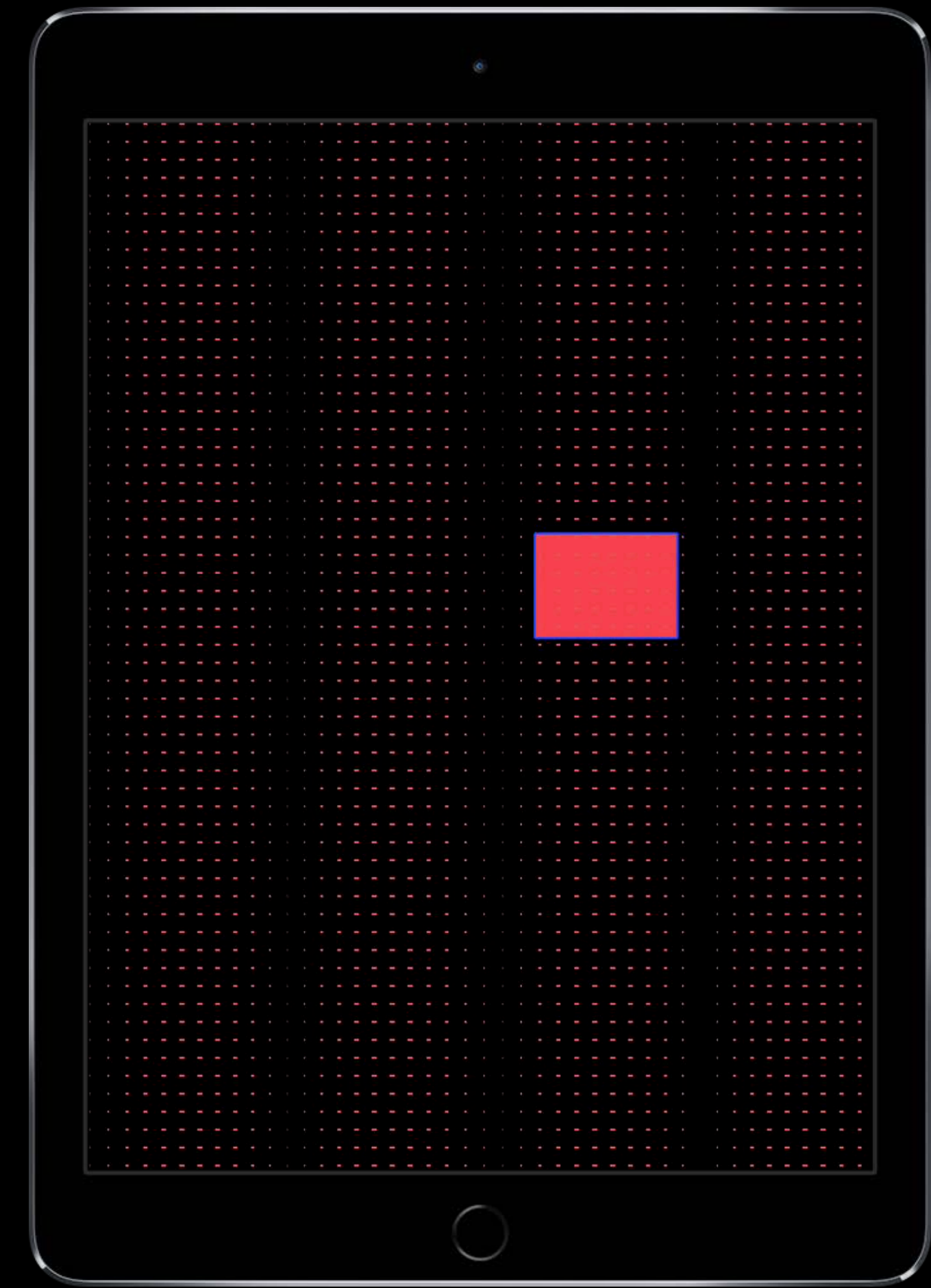
position

velocity

mass

charge

time





*Demo*

UIKit Dynamics—StickyCorners

# Debugging UIKit Dynamics

NEW



# Debugging UIKit Dynamics

NEW

Overlay to show Physics



# Debugging UIKit Dynamics

NEW

Overlay to show Physics

Visualize fields, collision bounds,  
and attachments



# Debugging UIKit Dynamics

NEW

Overlay to show Physics

Visualize fields, collision bounds,  
and attachments

Not API, but accessible in lldb



# Debugging UIKit Dynamics

NEW

Overlay to show Physics

Visualize fields, collision bounds,  
and attachments

Not API, but accessible in lldb

`debugEnabled`



# Debugging UIKit Dynamics

NEW

Overlay to show Physics

Visualize fields, collision bounds,  
and attachments

Not API, but accessible in lldb

`debugEnabled`  
`debugInterval`



# Debugging UIKit Dynamics

NEW

Overlay to show Physics

Visualize fields, collision bounds,  
and attachments

Not API, but accessible in lldb

`debugEnabled`

`debugInterval`

`debugAnimationSpeed`





# UIDynamicItemBehavior

NEW

# UIDynamicItemBehavior

NEW

Customize physical properties

# UIDynamicItemBehavior

NEW

Customize physical properties

Applied to one or more items

# UIDynamicItemBehavior

NEW

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

NEW

Customize physical properties

Applied to one or more items

```
var
```

```
var
```

```
var
```

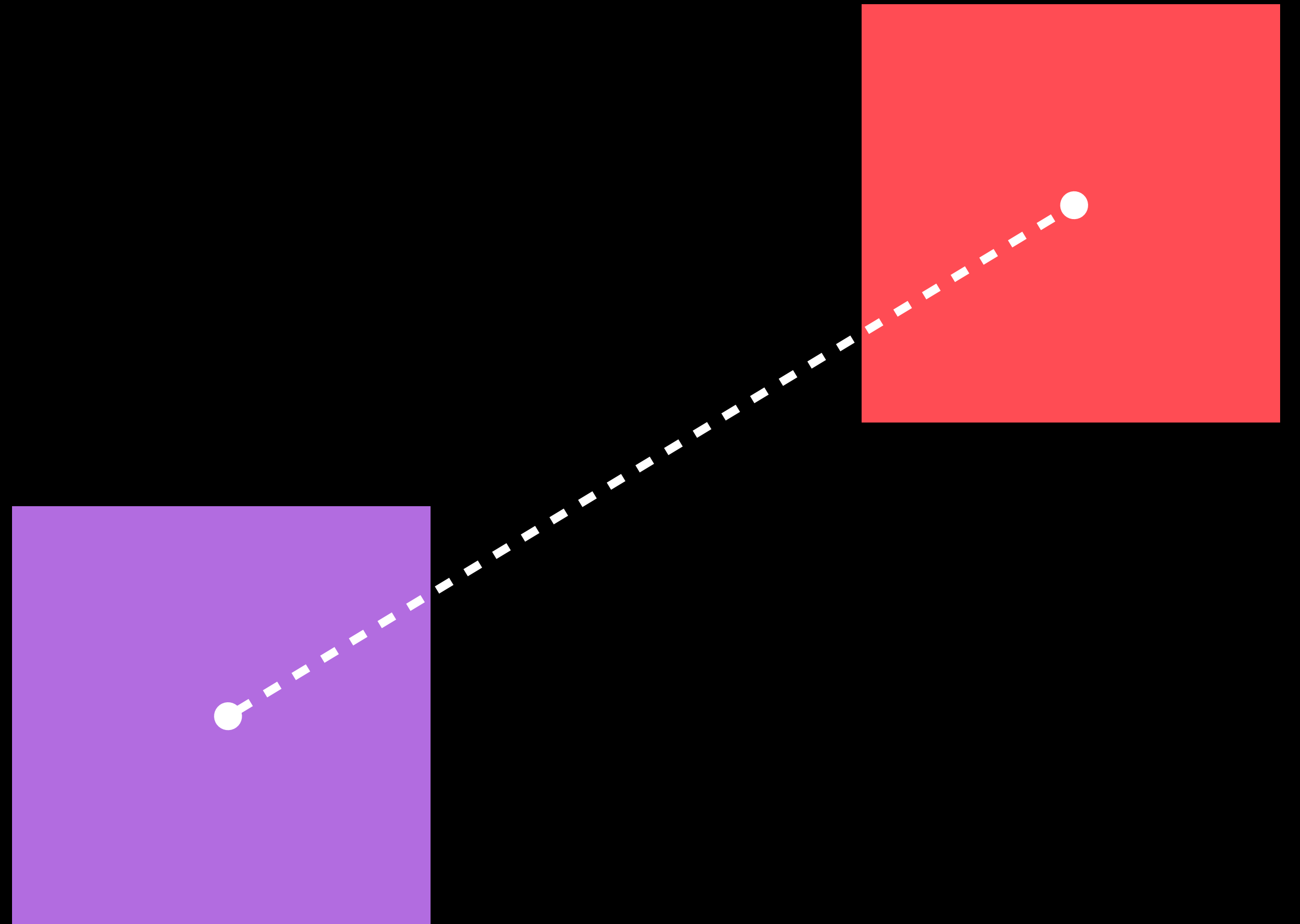
```
var
```

```
var
```

```
var charge: CGFloat
```

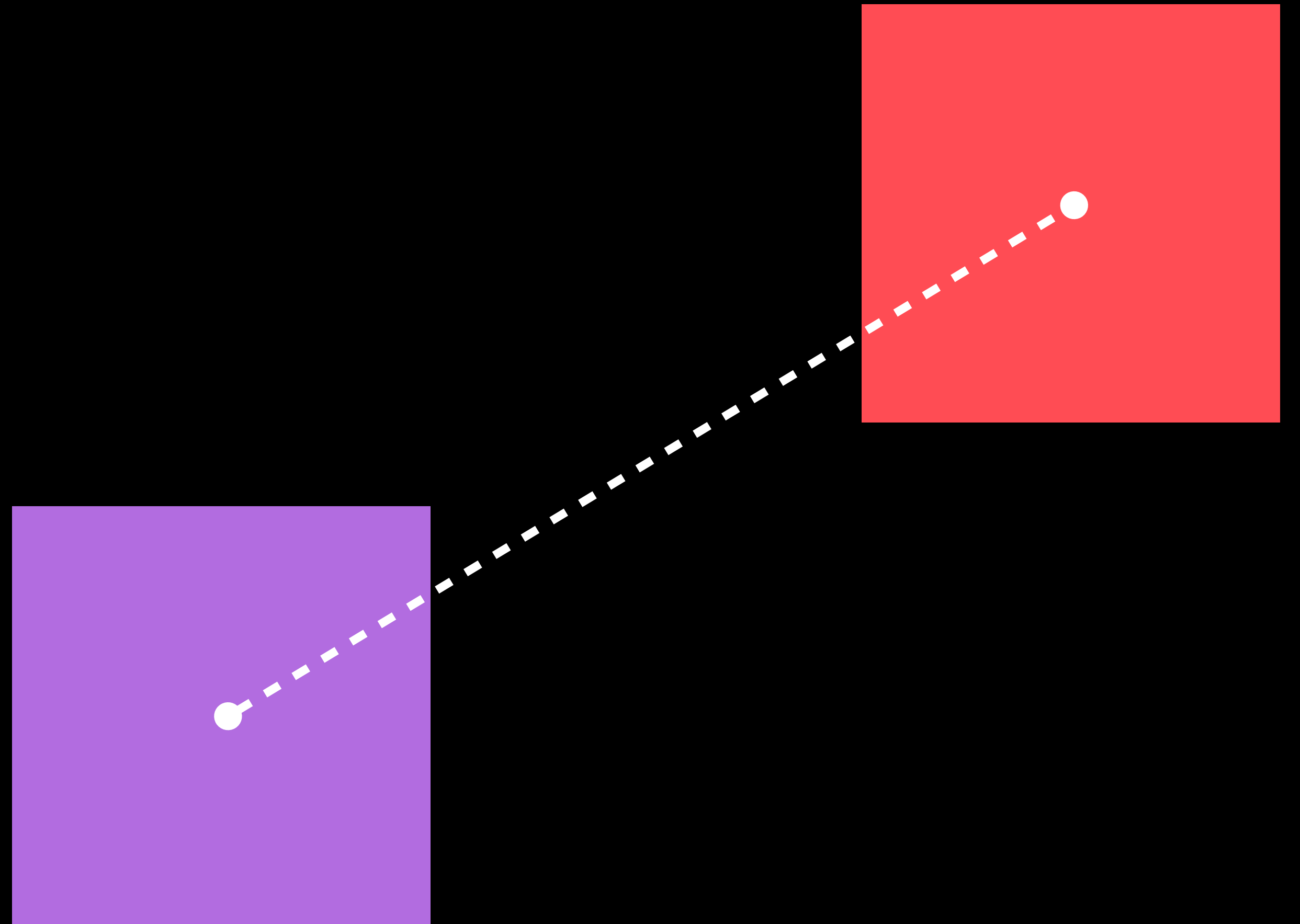
```
var anchored: Bool
```

# UIAttachmentBehavior



# UIAttachmentBehavior

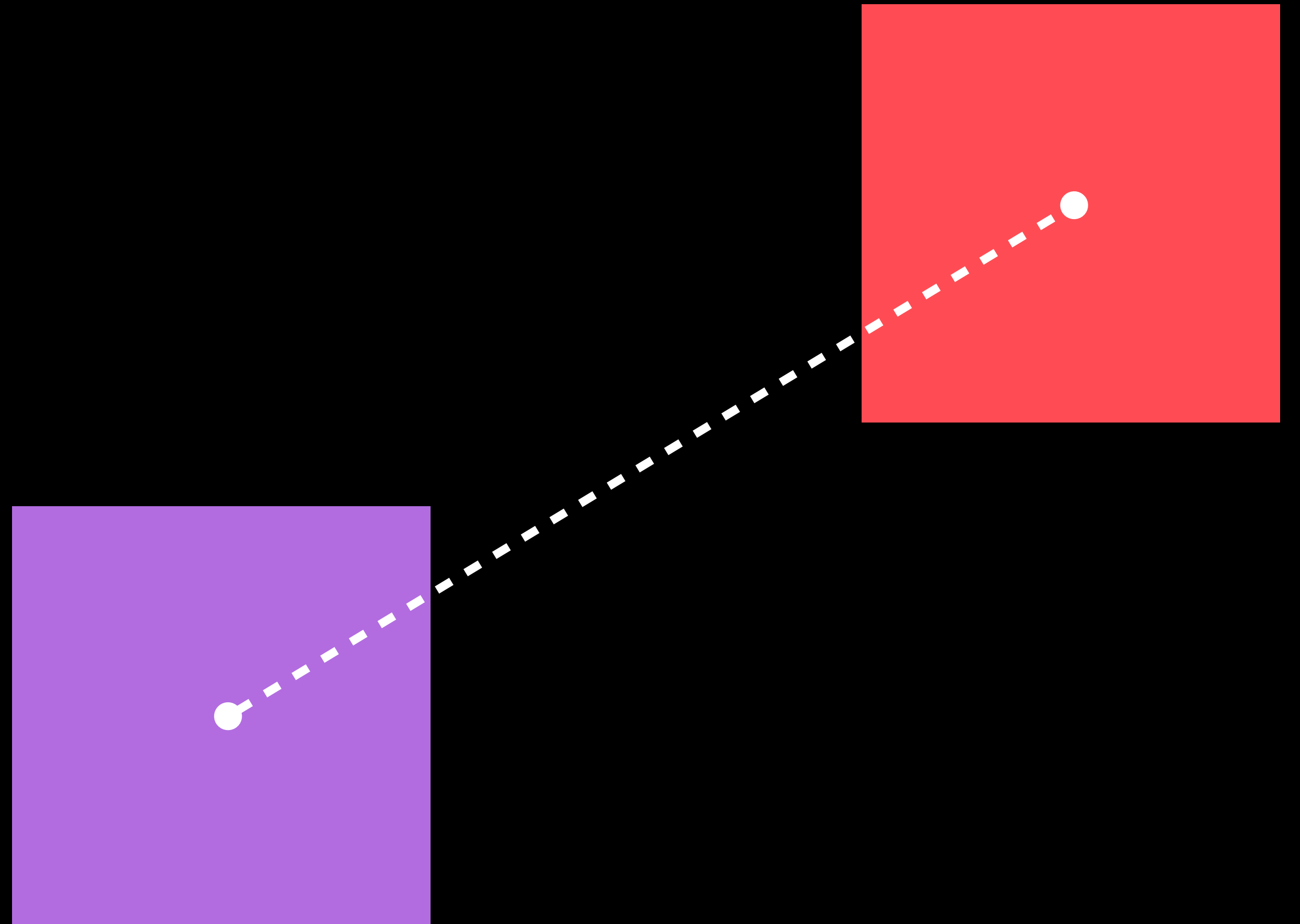
Distance attachment



# UIAttachmentBehavior

Distance attachment

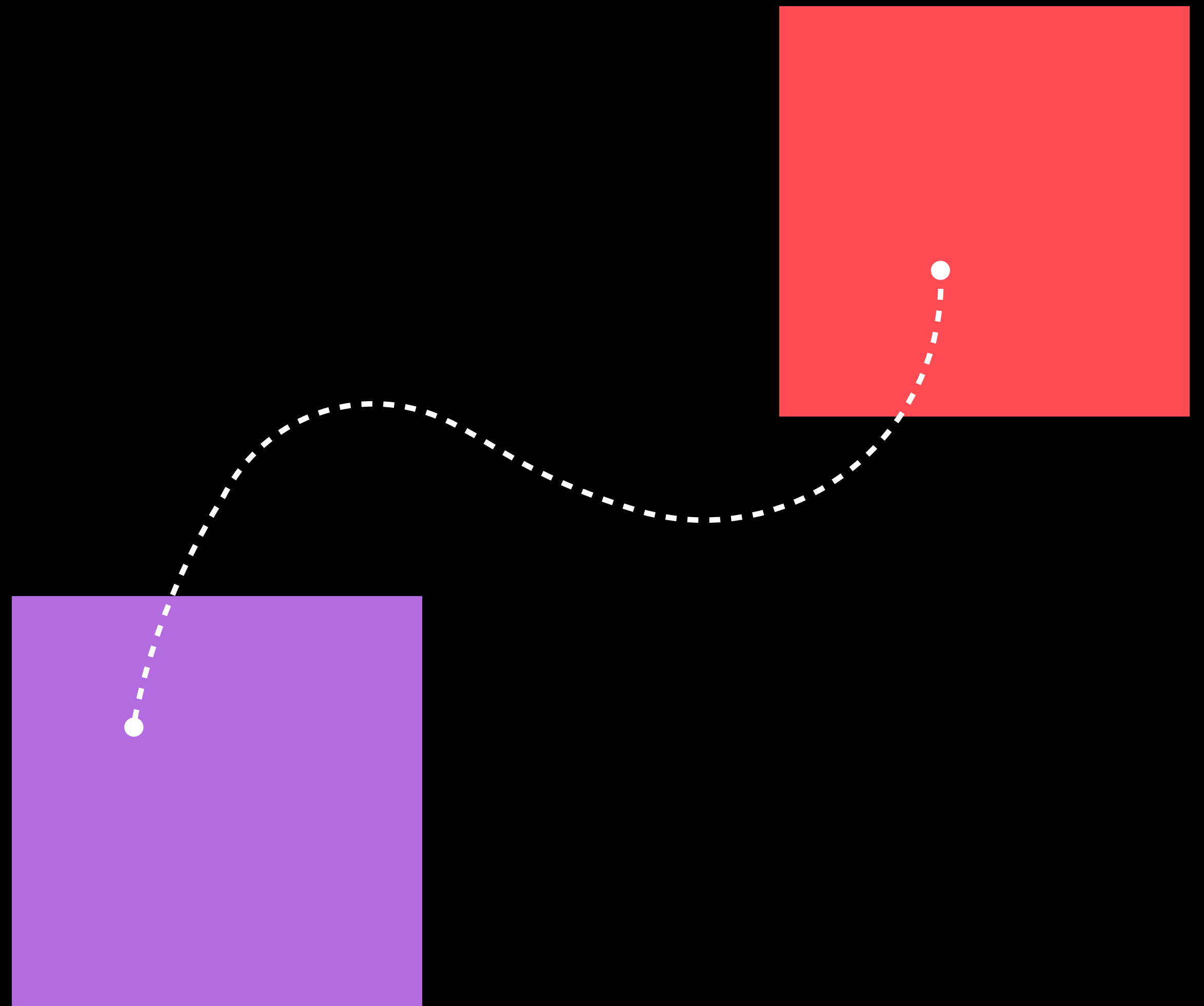
Customizable Damping and Frequency





# UIAttachmentBehavior

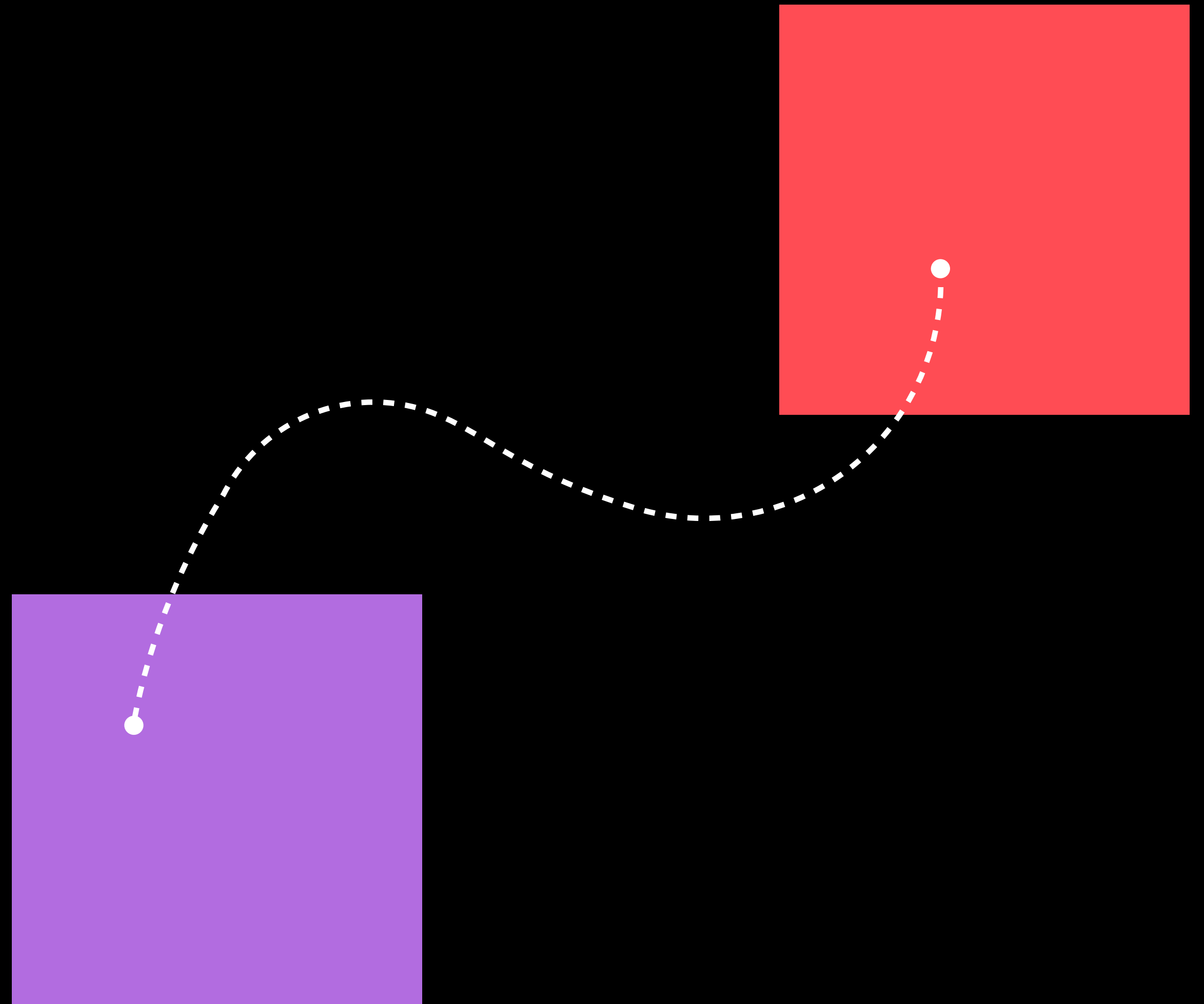
NEW



# UIAttachmentBehavior

NEW

Limit Attachment

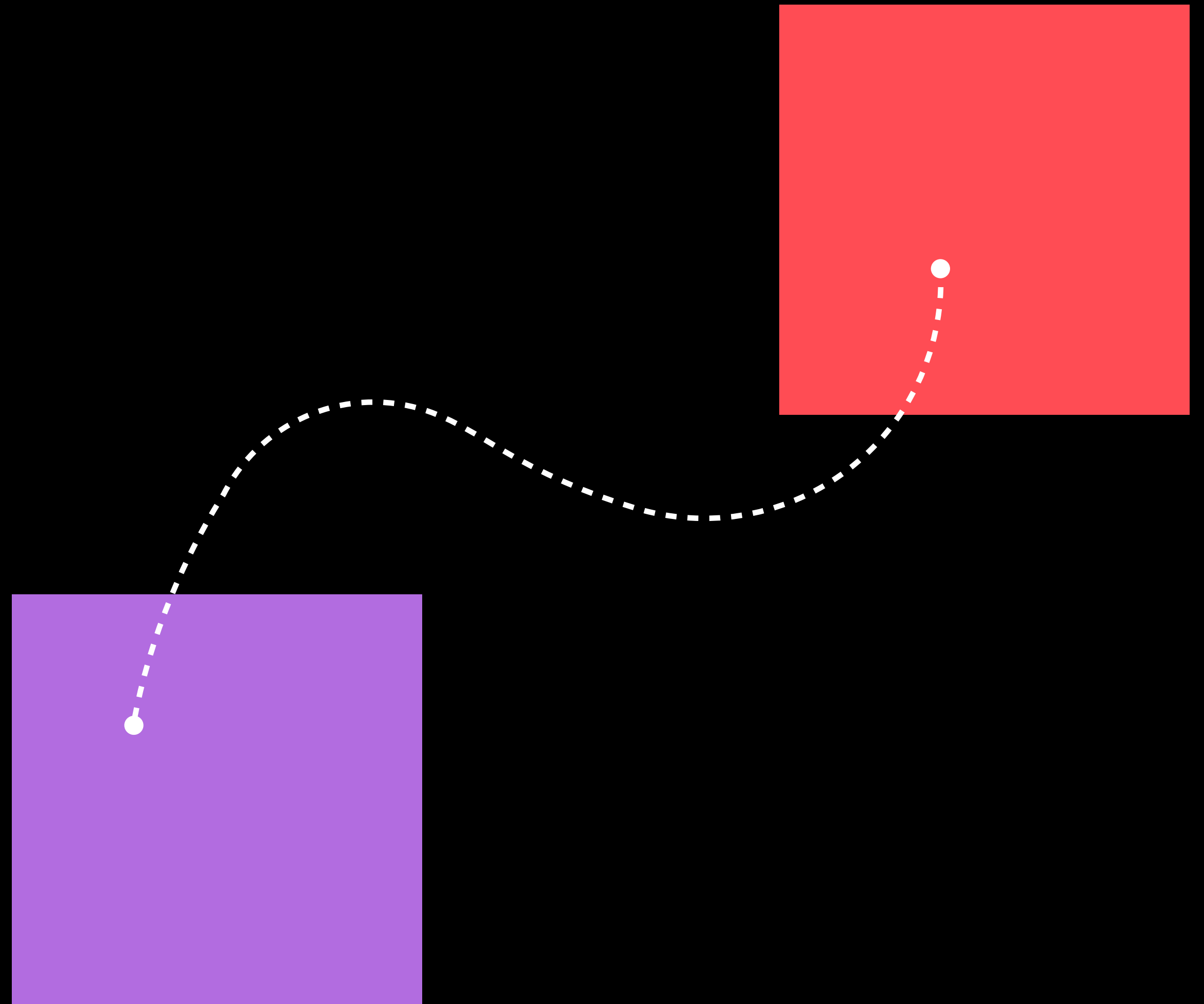


# UIAttachmentBehavior

NEW

Limit Attachment

Behaves like a rope between items



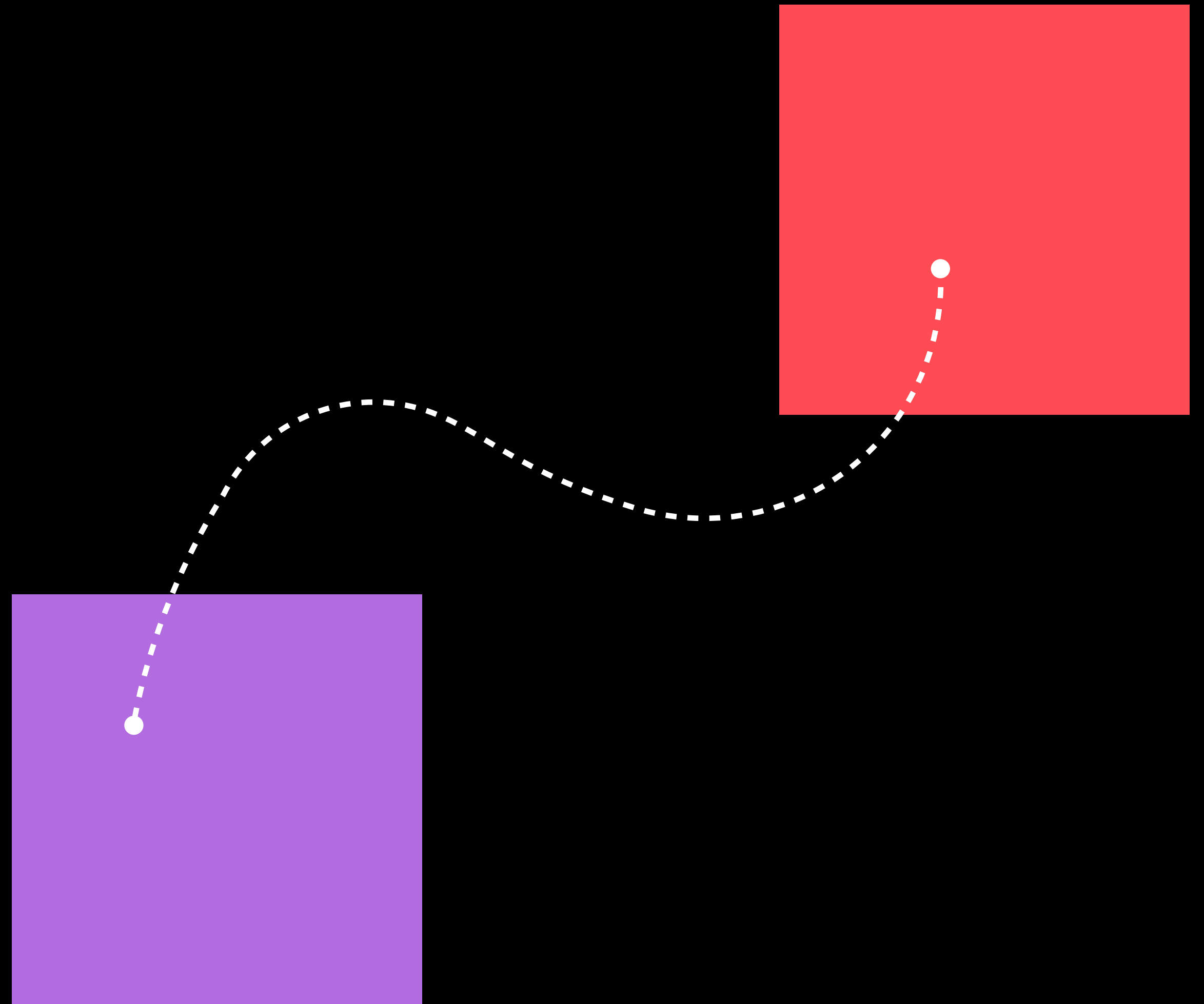
# UIAttachmentBehavior

NEW

Limit Attachment

Behaves like a rope between items

Attachment point offset from each item's center



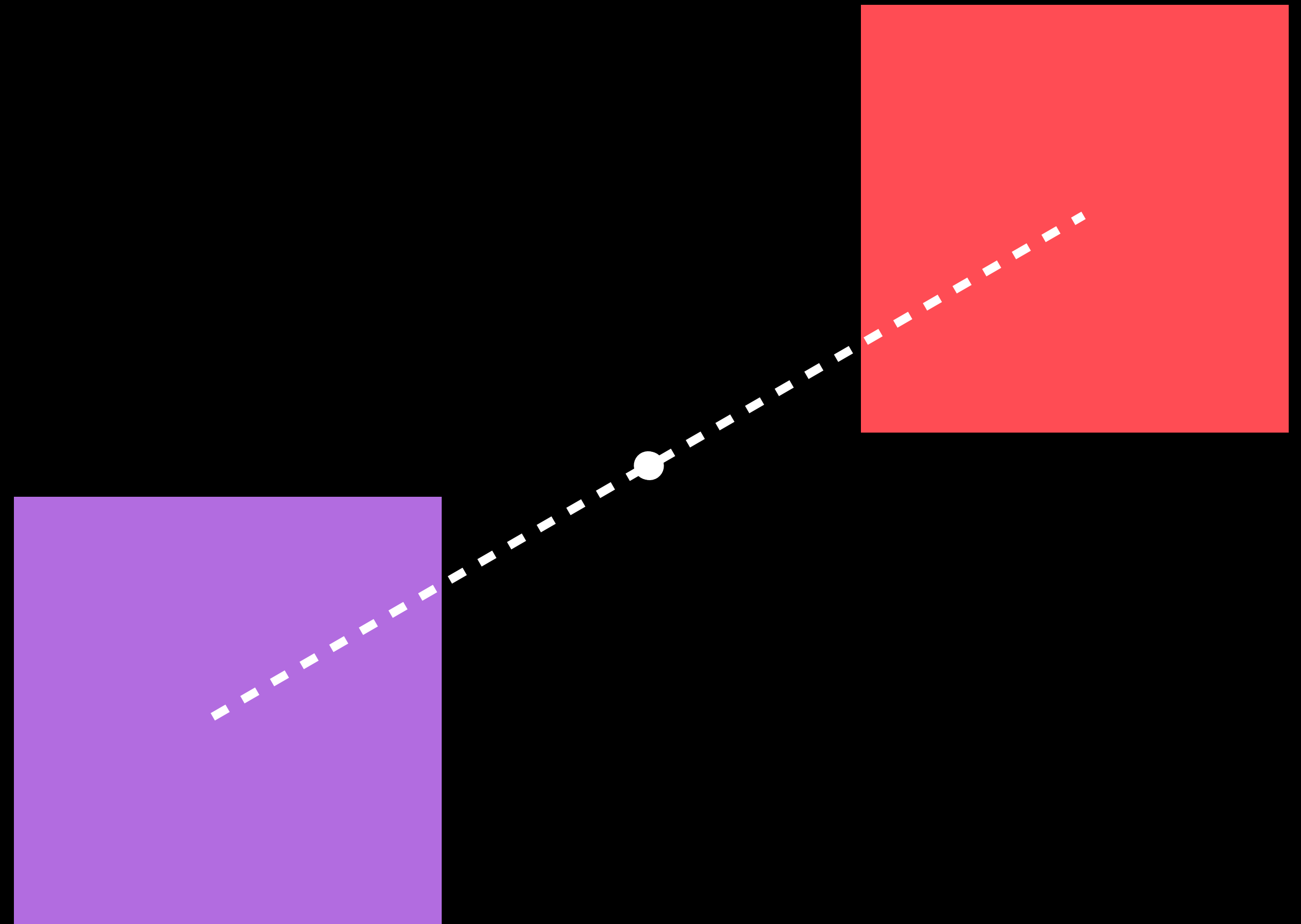
# UIAttachmentBehavior

NEW

# UIAttachmentBehavior

NEW

Fixed Attachment

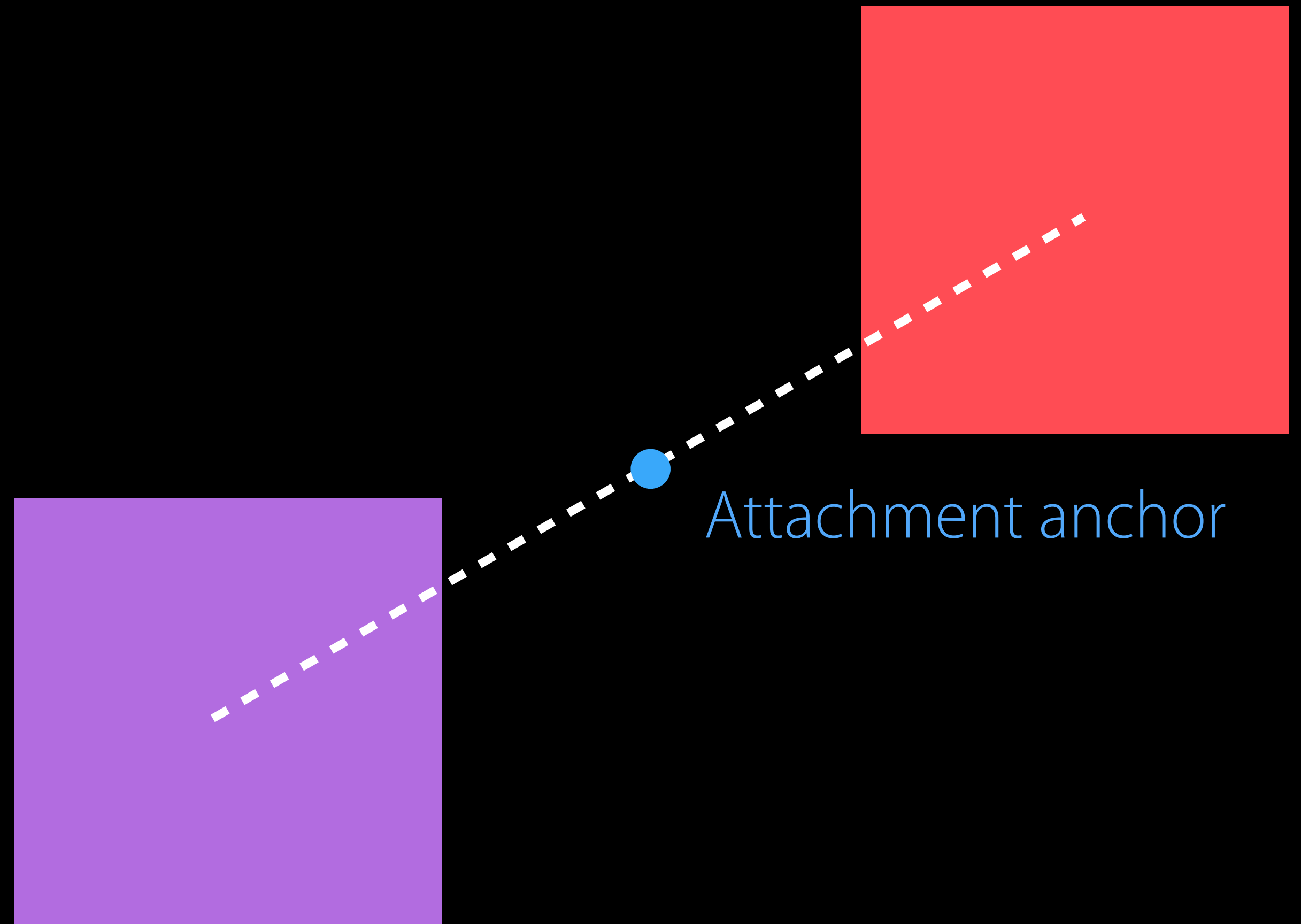


# UIAttachmentBehavior

NEW

## Fixed Attachment

Items positioned with respect  
to the attachment anchor point



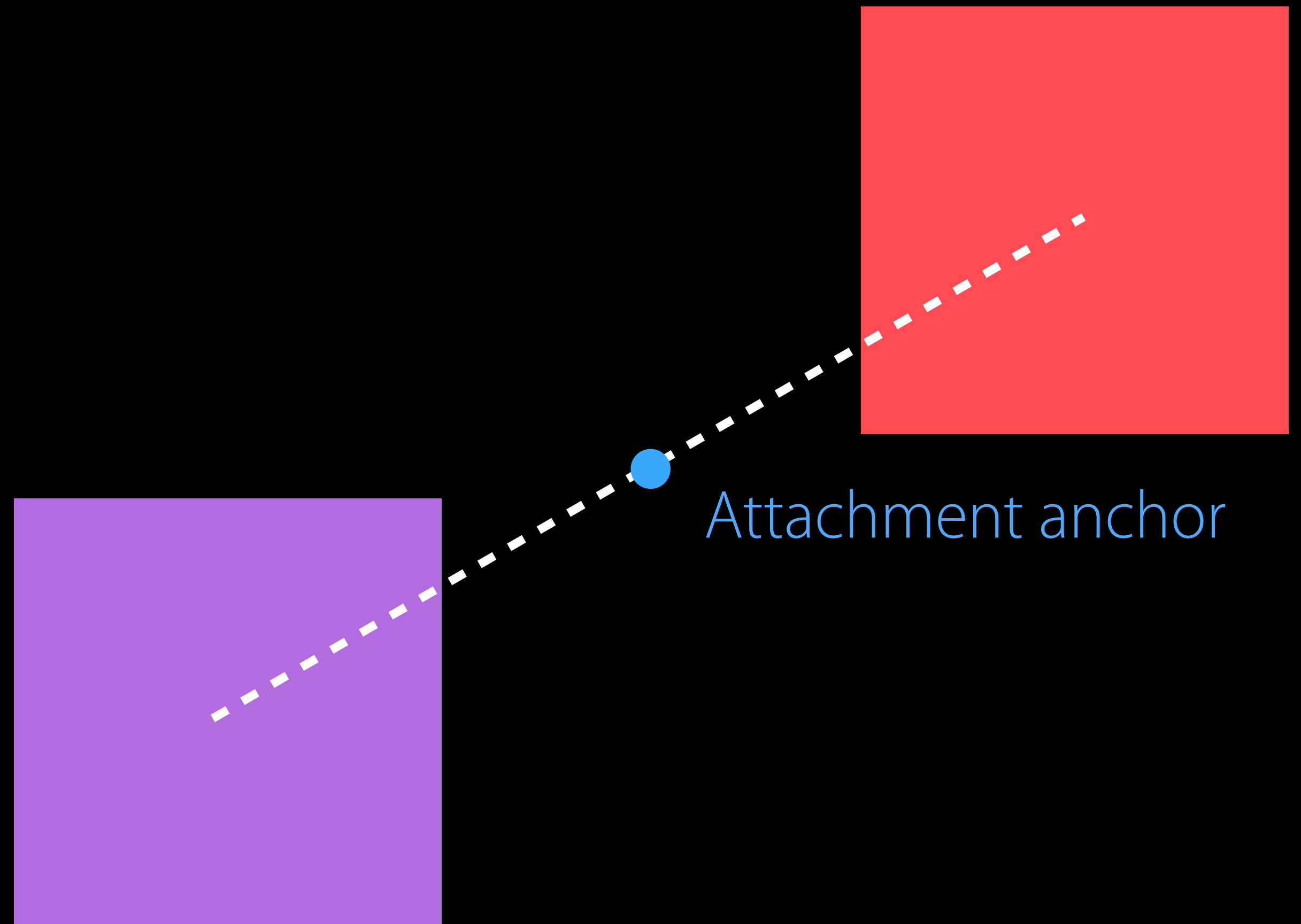
# UIAttachmentBehavior

NEW

## Fixed Attachment

Items positioned with respect  
to the attachment anchor point

No relative movement





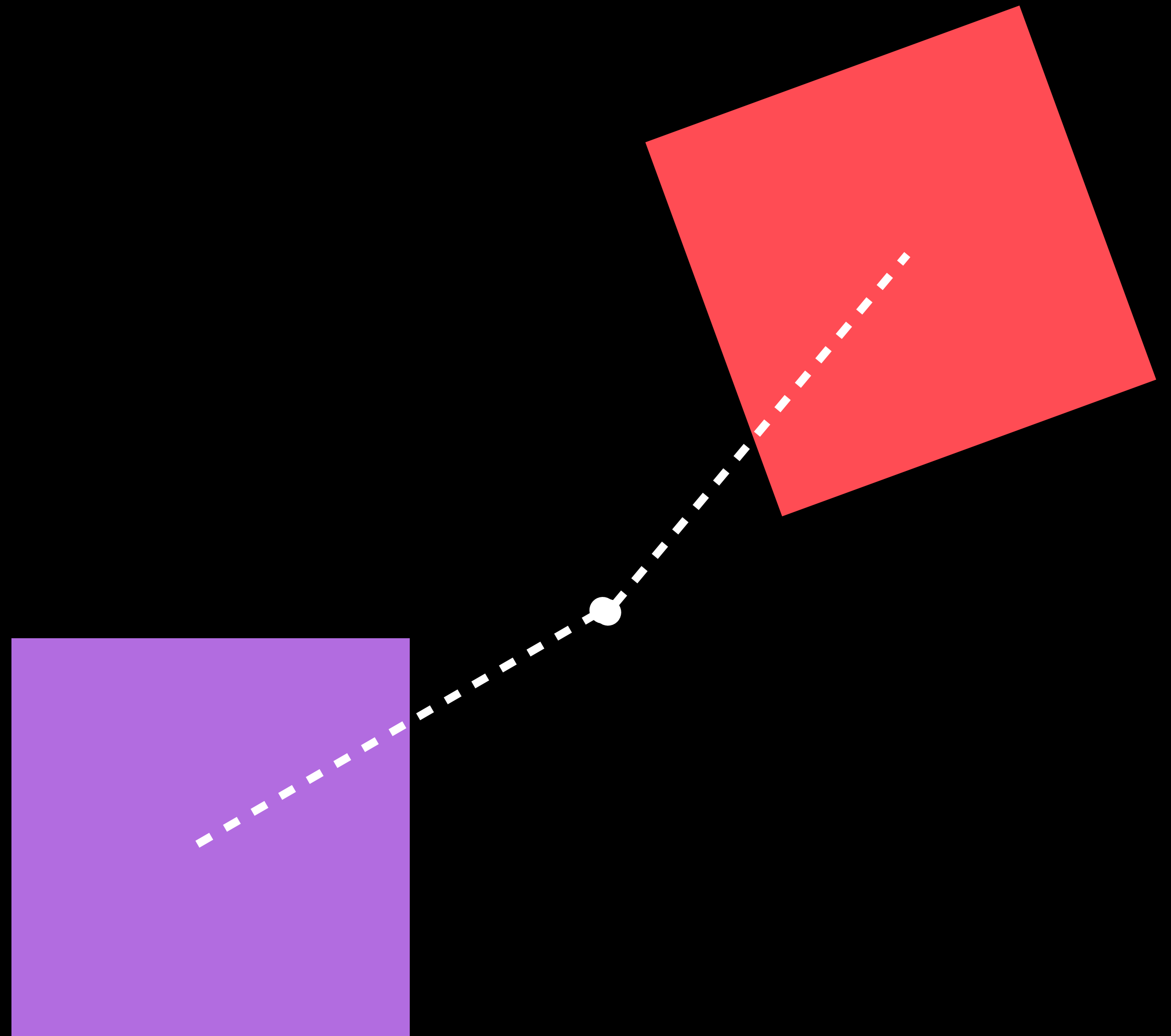
# UIAttachmentBehavior

NEW

# UIAttachmentBehavior

NEW

Pin Attachment

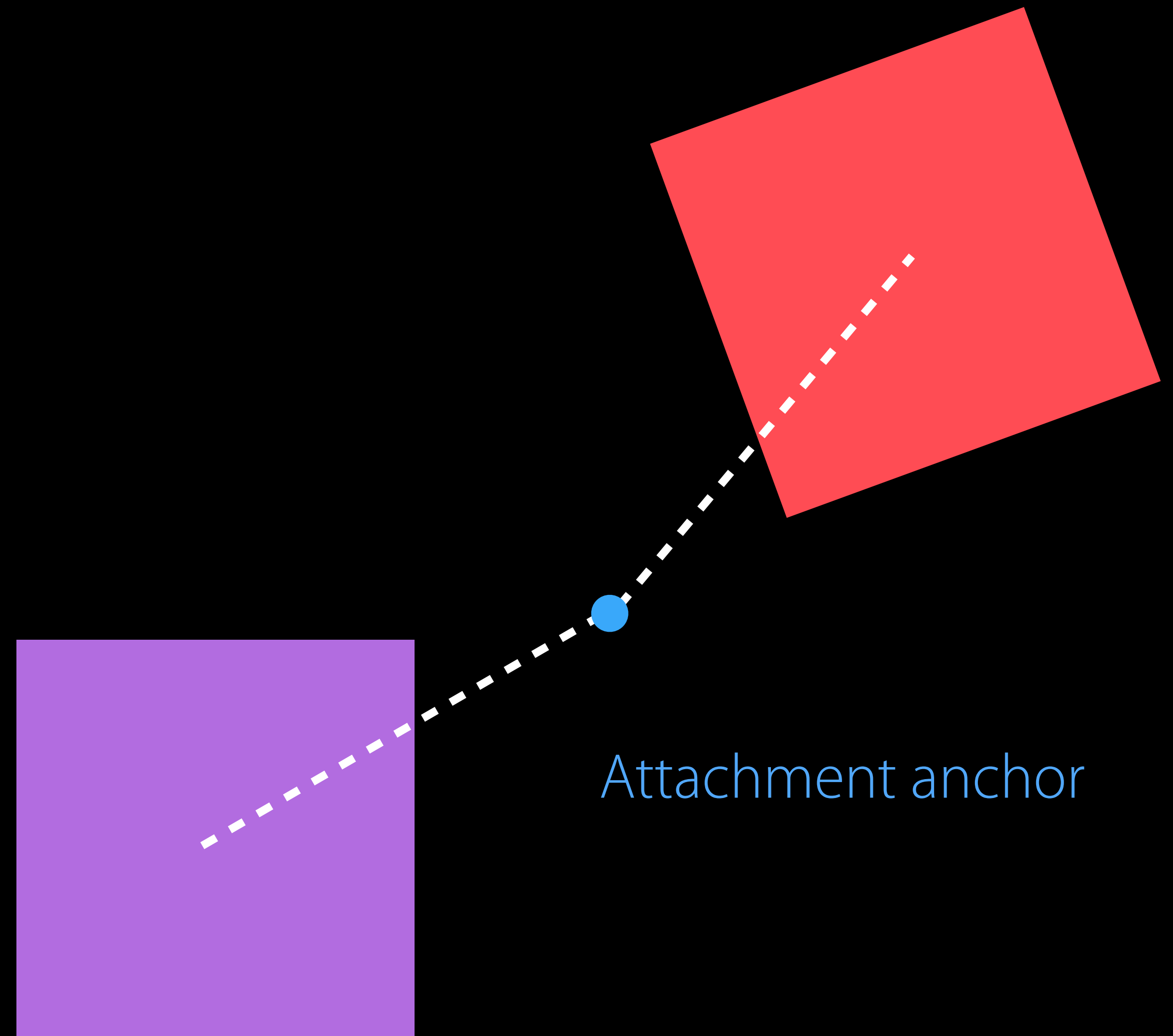


# UIAttachmentBehavior

NEW

## Pin Attachment

Items positioned with respect to the attachment anchor point



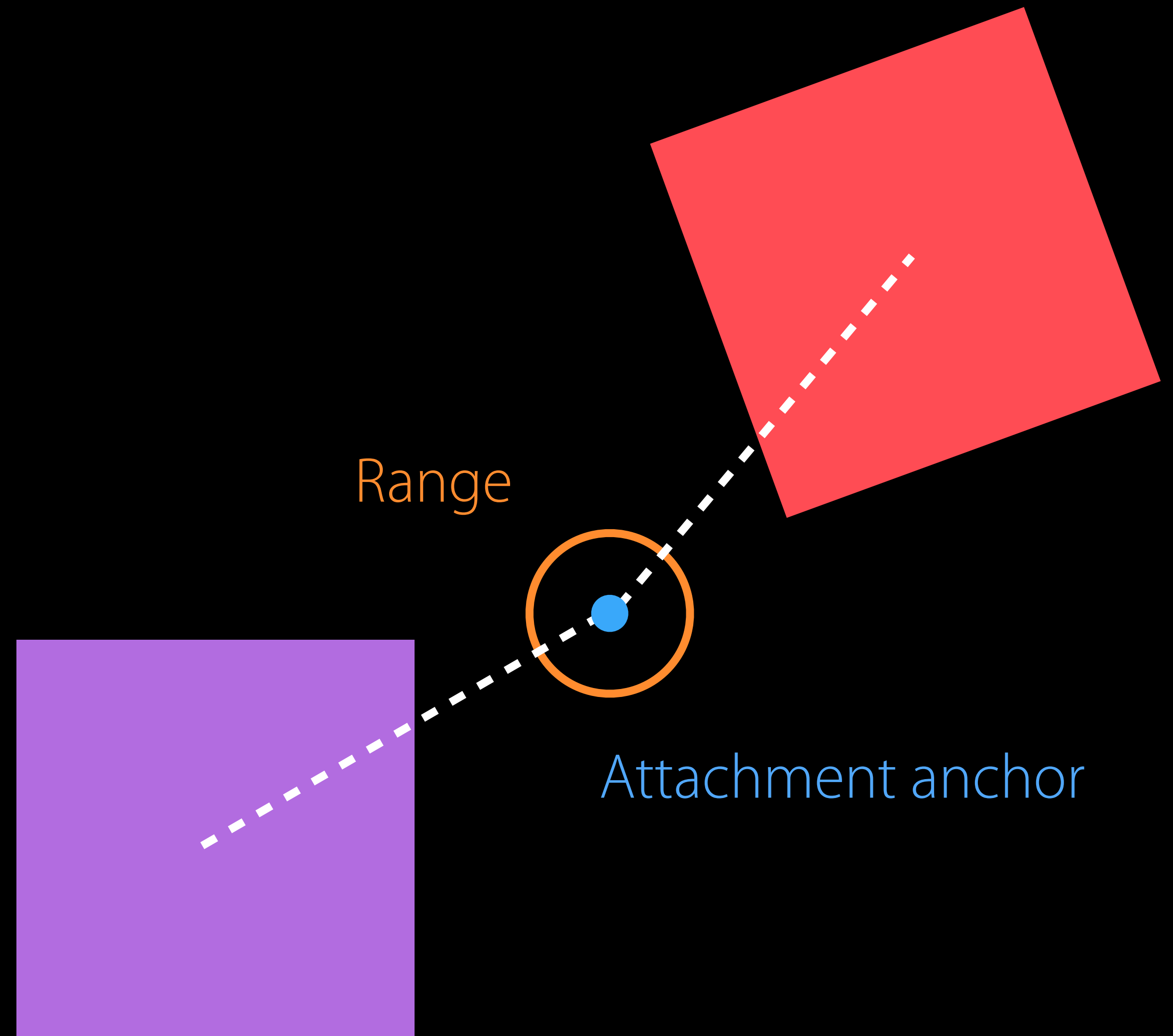
# UIAttachmentBehavior

NEW

## Pin Attachment

Items positioned with respect  
to the attachment anchor point

Specify a rotatable range



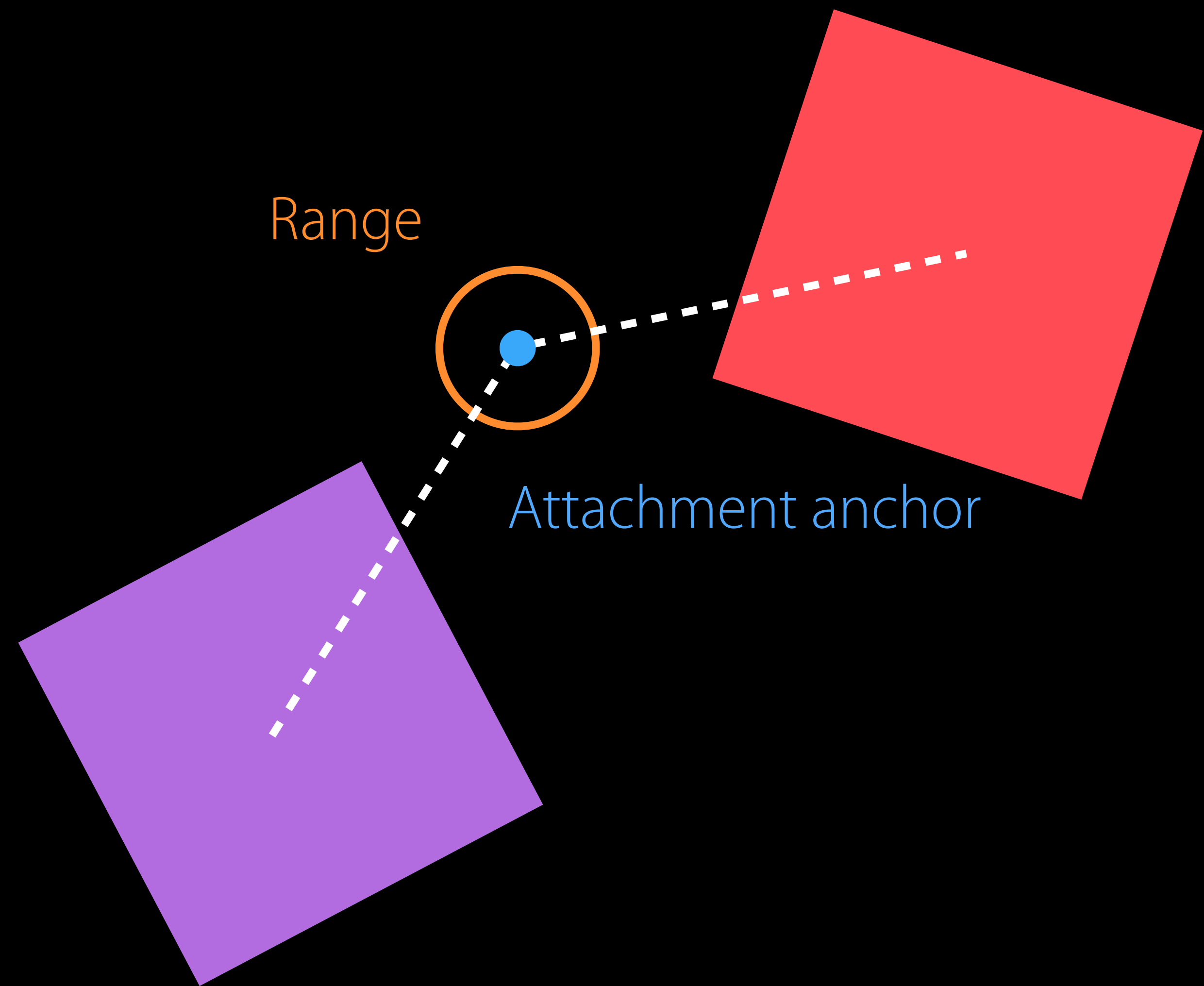
# UIAttachmentBehavior

NEW

## Pin Attachment

Items positioned with respect  
to the attachment anchor point

Specify a rotatable range



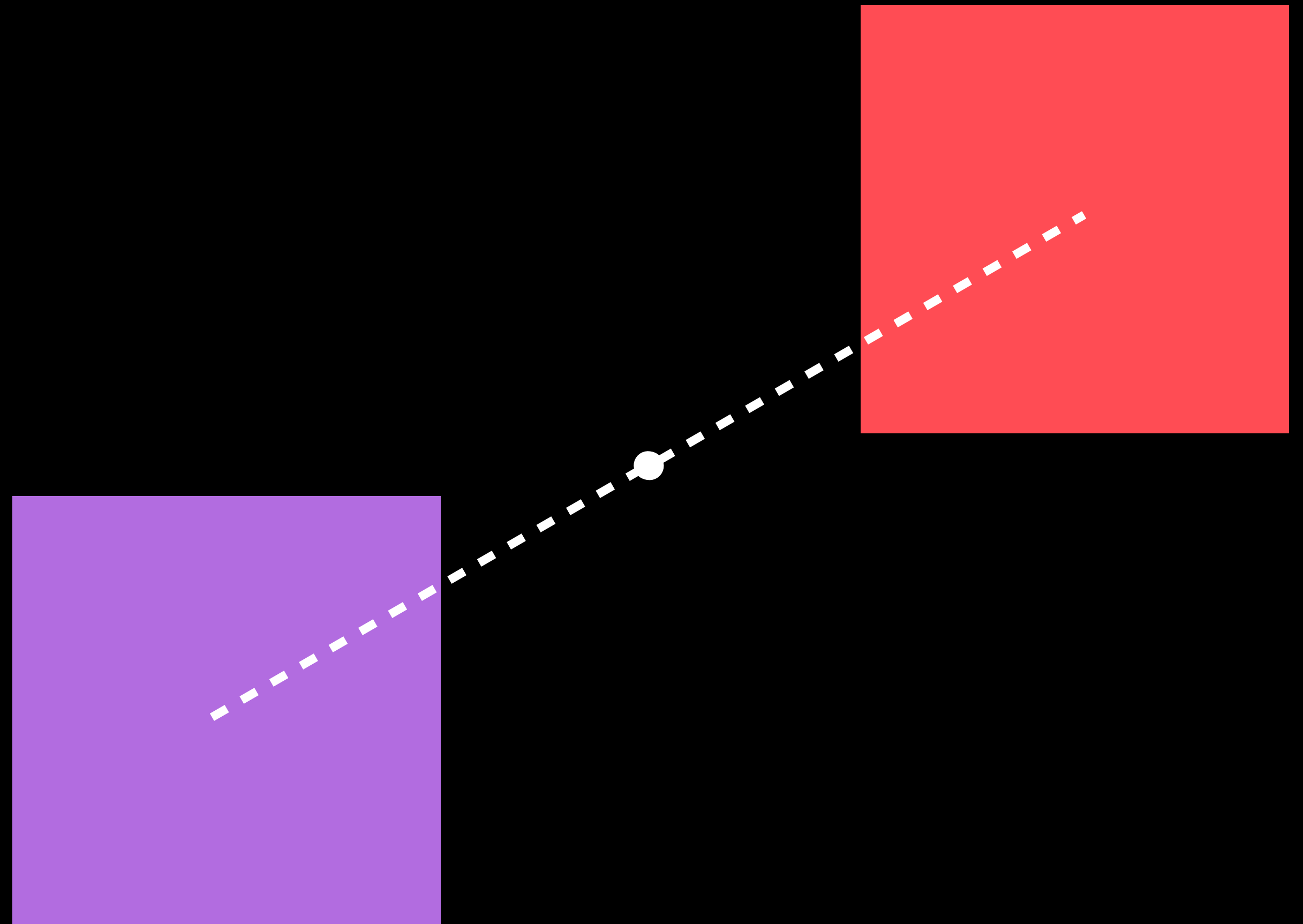
# UIAttachmentBehavior

NEW

# UIAttachmentBehavior

NEW

Sliding Attachment

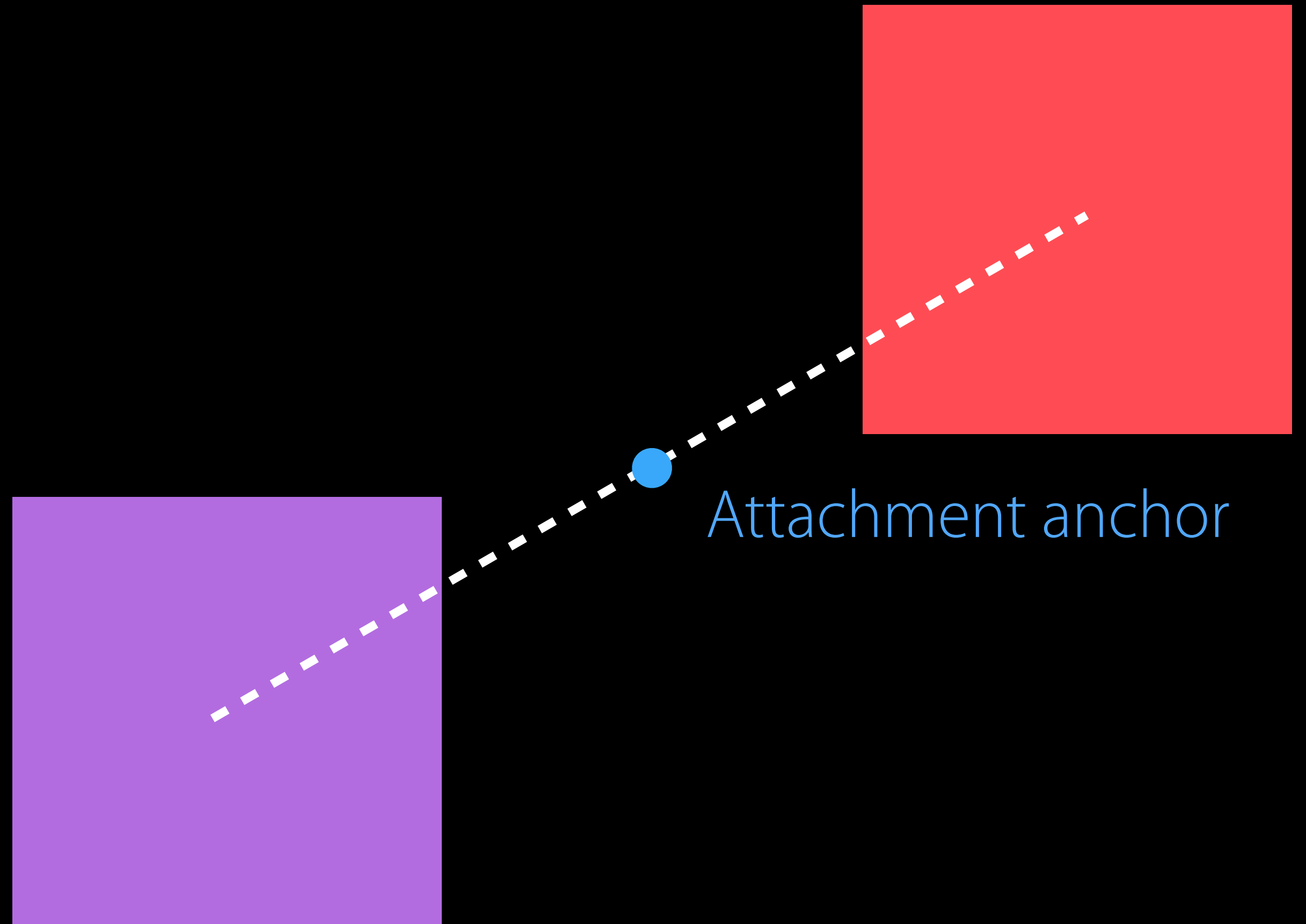


# UIAttachmentBehavior

NEW

## Sliding Attachment

Items positioned with respect to the attachment anchor point





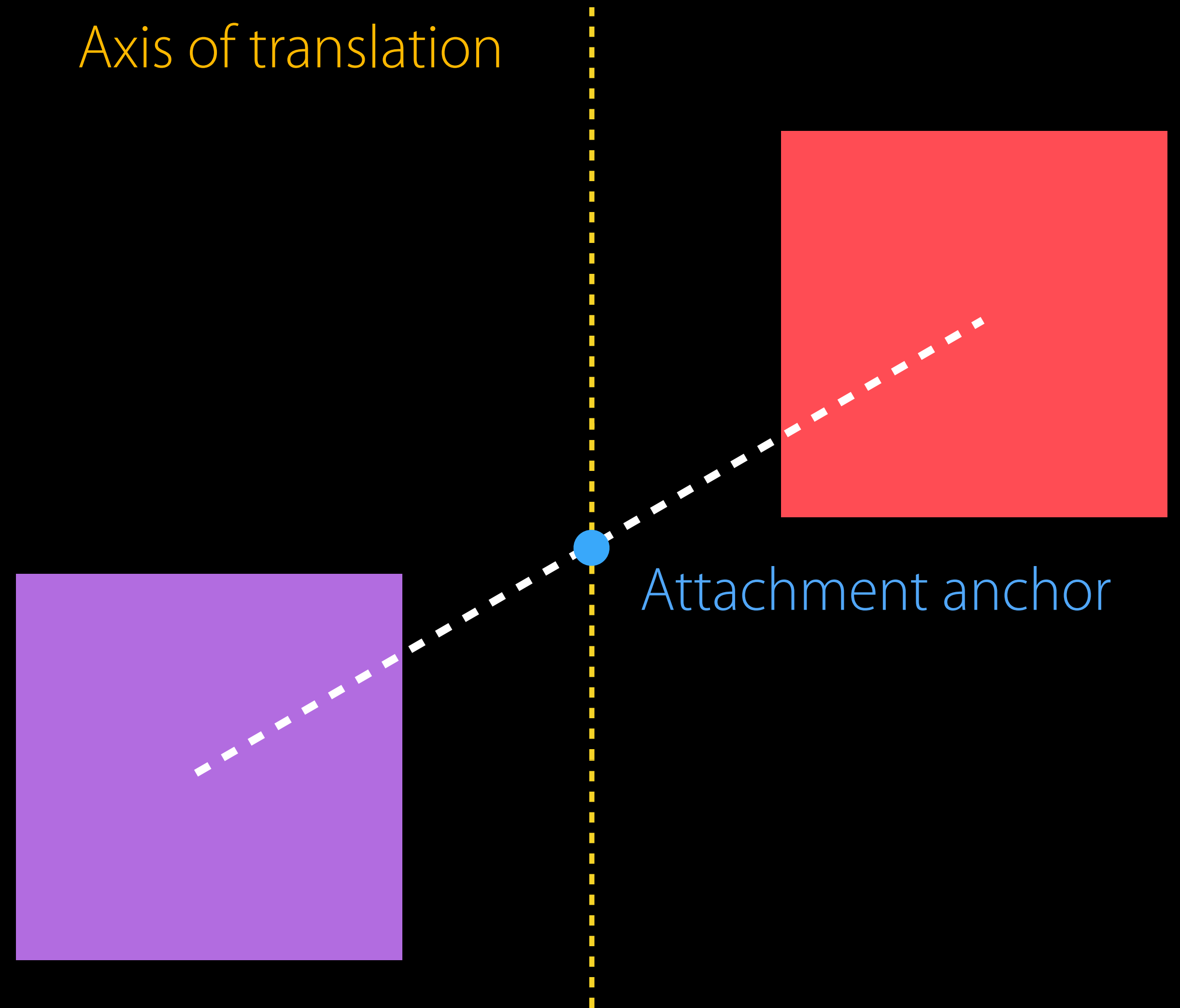
# UIAttachmentBehavior

NEW

## Sliding Attachment

Items positioned with respect to the attachment anchor point

Allows only relative translation along a specified axis



# UIAttachmentBehavior

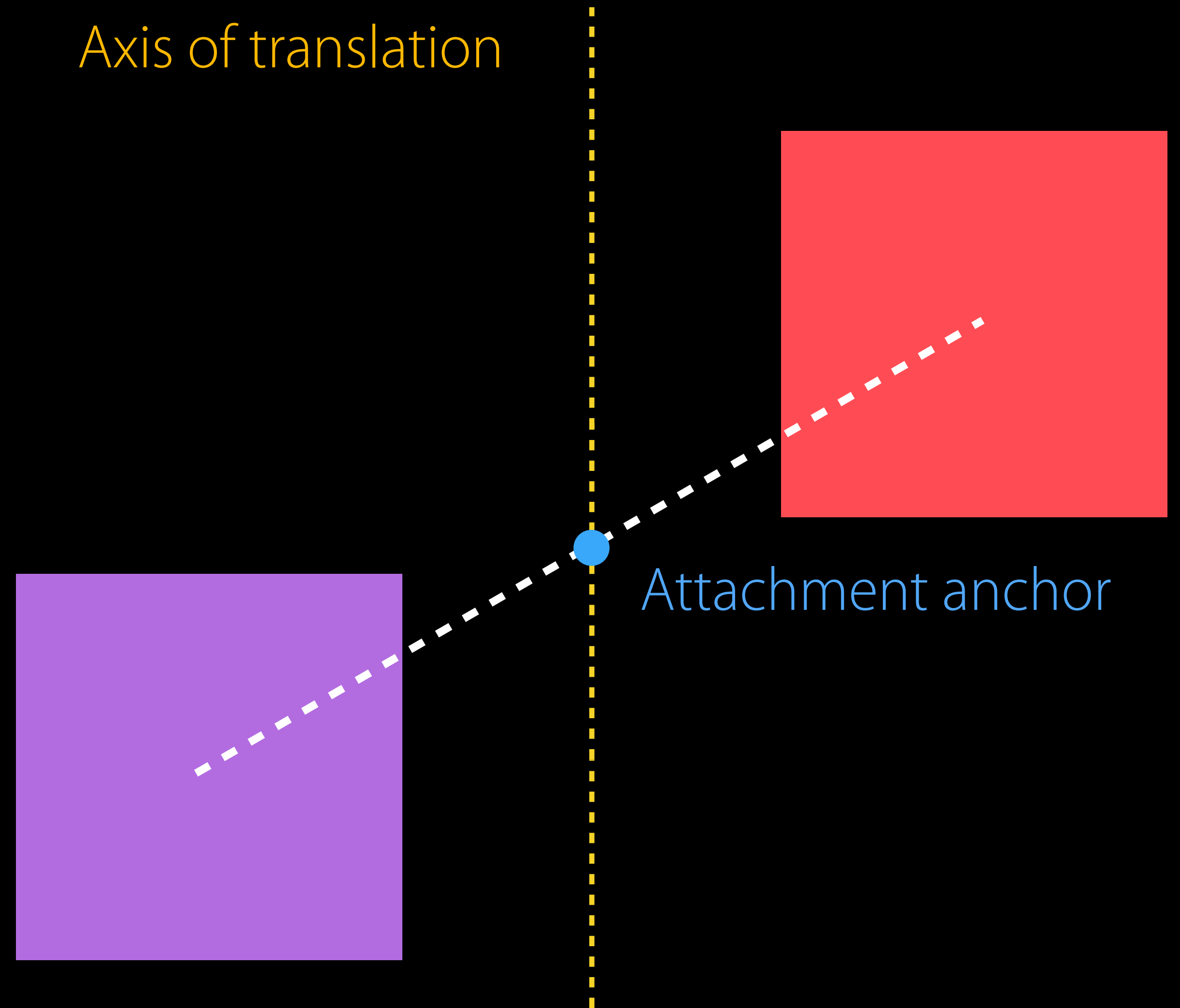
NEW

## 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



# UIAttachmentBehavior

NEW

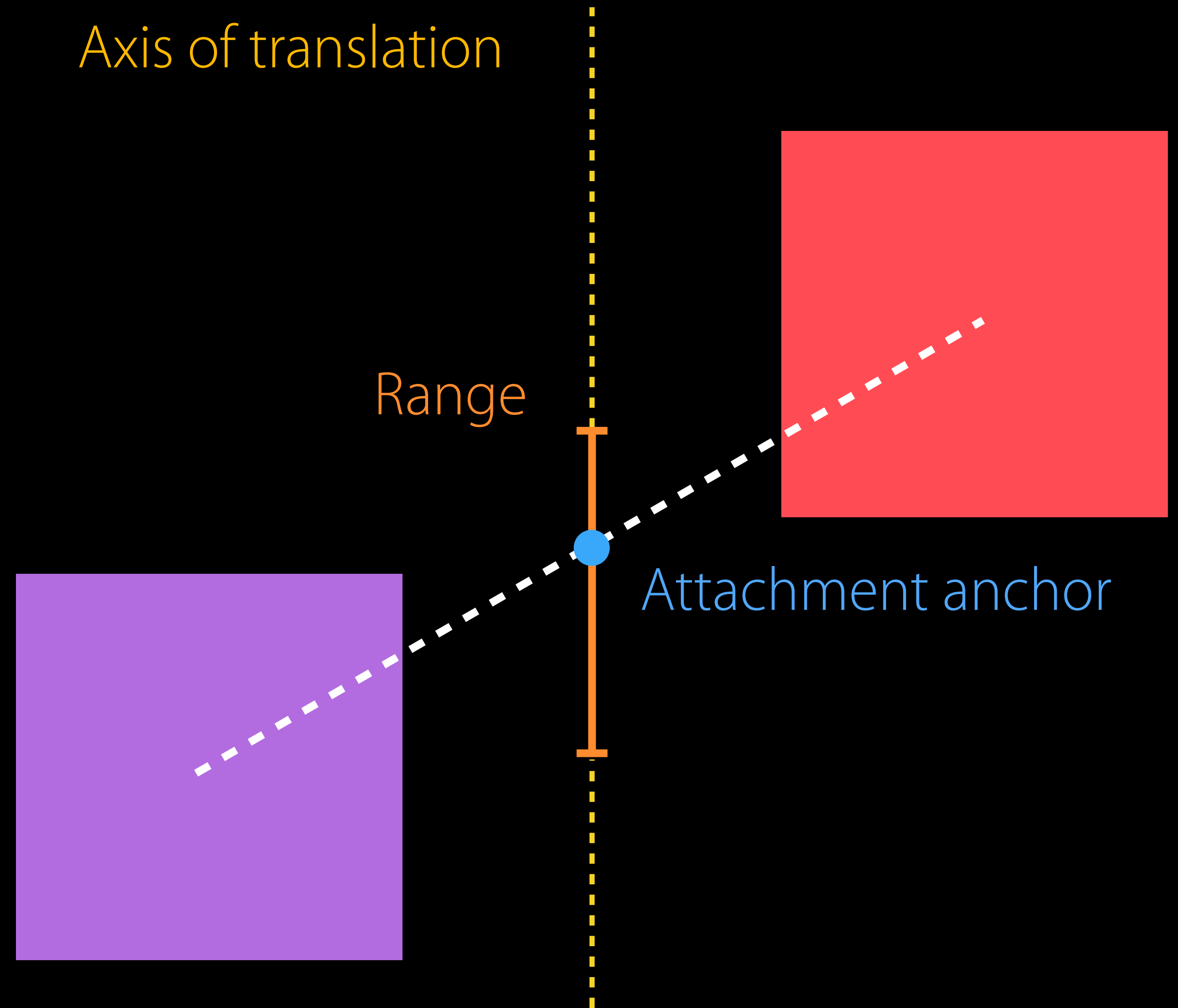
## Sliding Attachment

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



# UIAttachmentBehavior

NEW

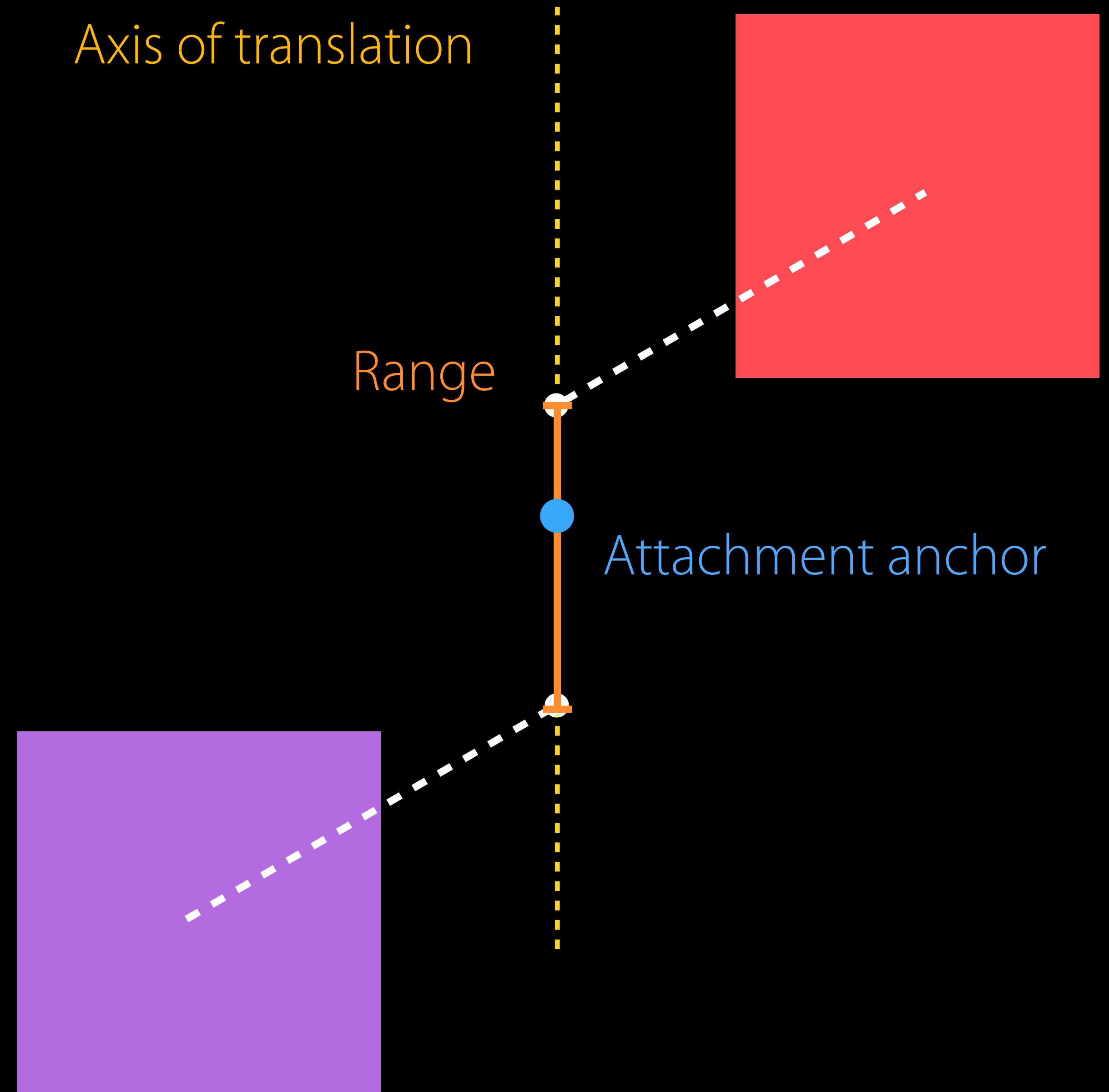
## Sliding Attachment

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





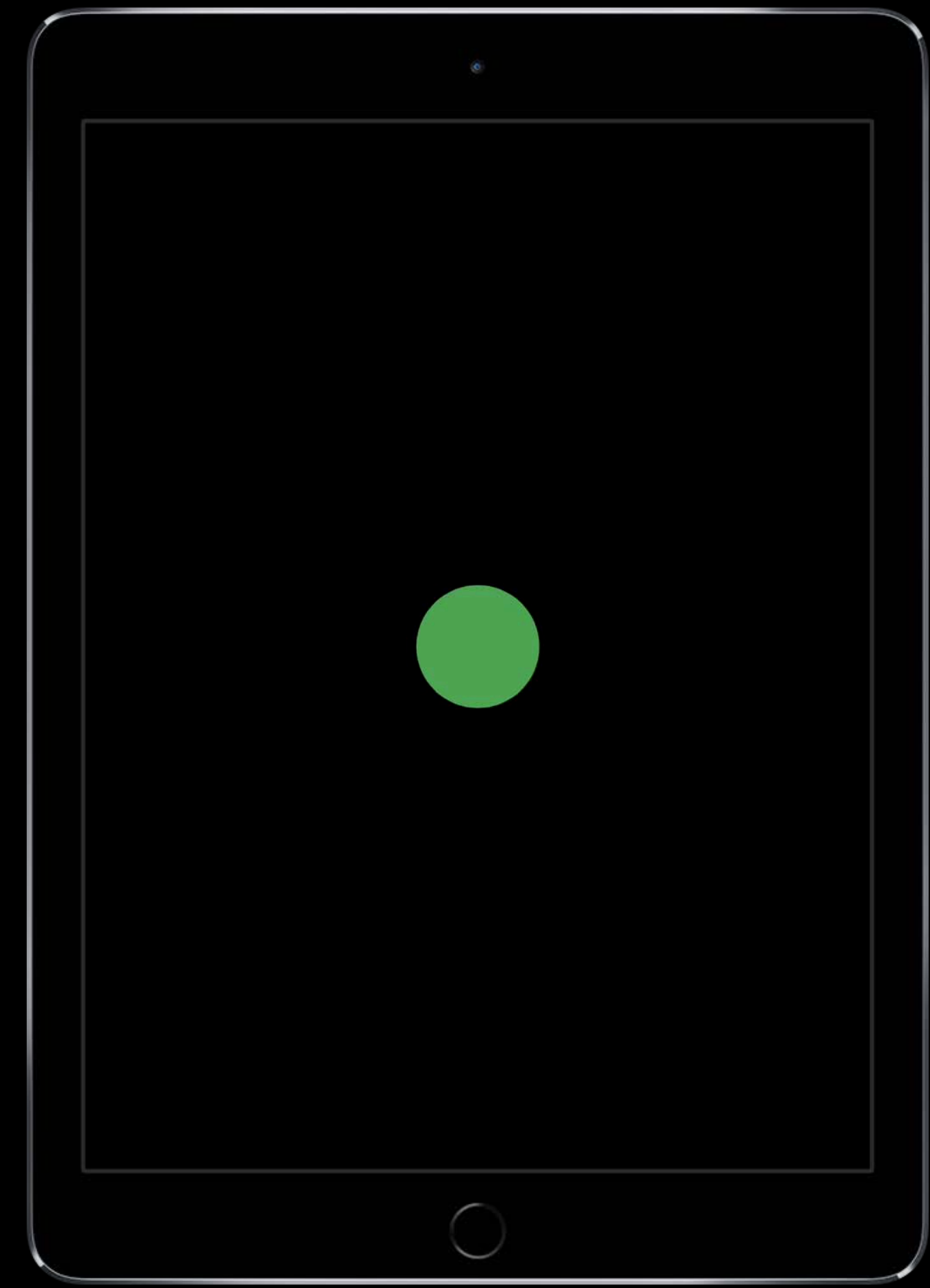
What's New in UIKit  
Dynamics & Visual Effects  
9:41 AM



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

# UISnapBehavior

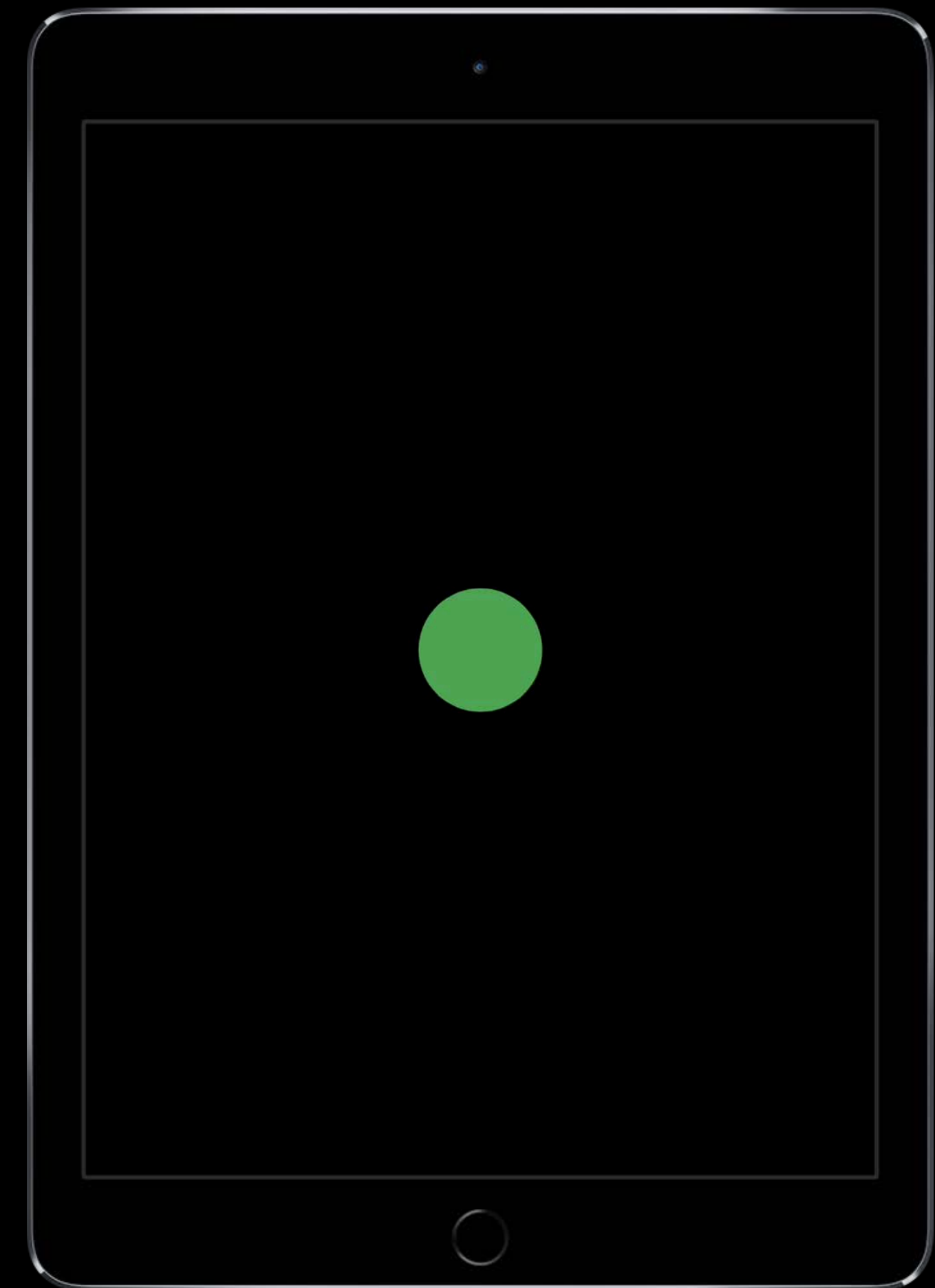
NEW



# UISnapBehavior

NEW

Snap a view in place



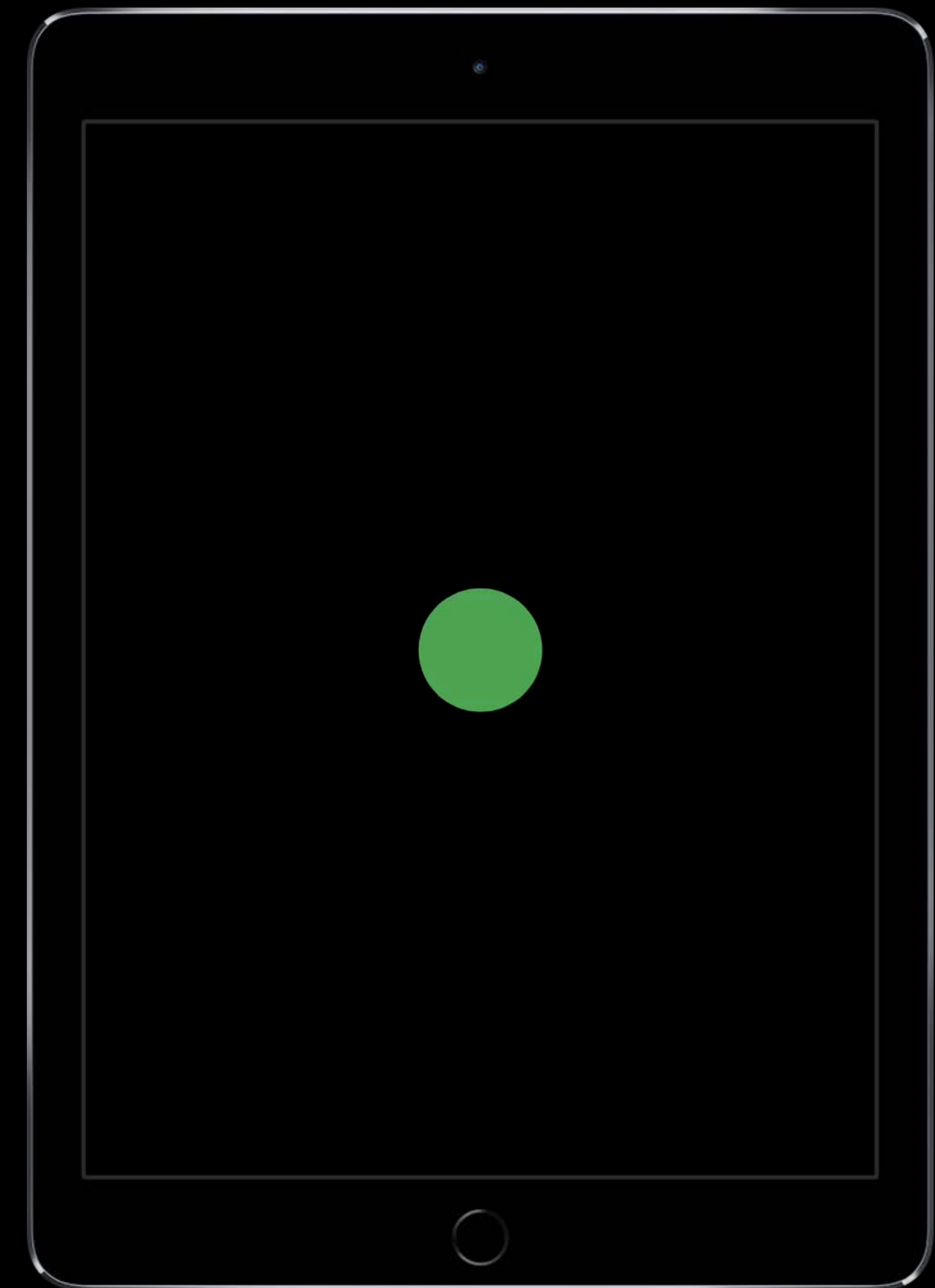


# UISnapBehavior

NEW

Snap a view in place

Customizable damping



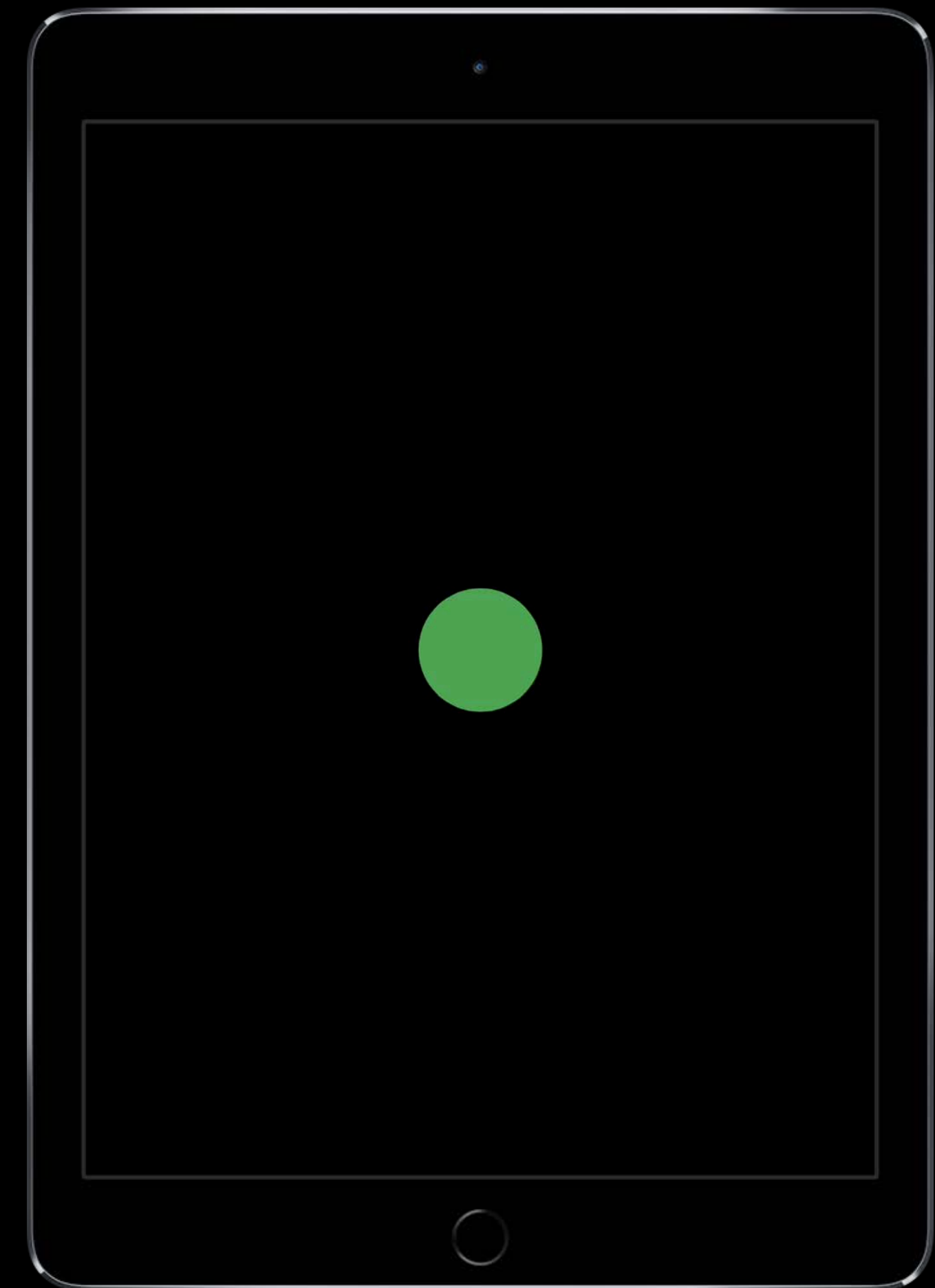
# UISnapBehavior

NEW

Snap a view in place

Customizable damping

Customizable snapPoint



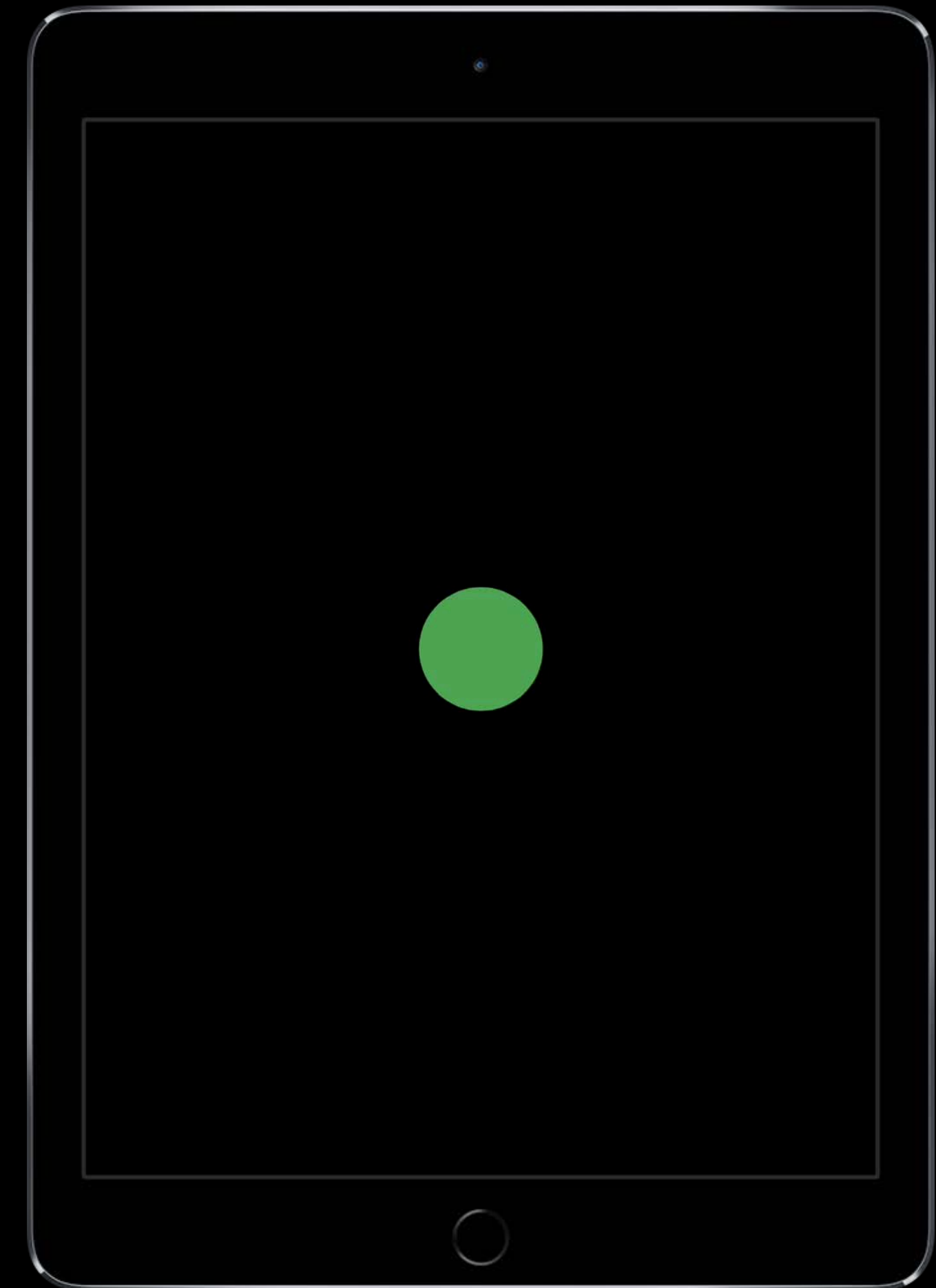
# UISnapBehavior

NEW

Snap a view in place

Customizable damping

Customizable snapPoint



# Visual Effects

Adding style to your app

David Duncan

UIKit Engineer

# UIVisualEffectView

## Stylish Overlays





# UIVisualEffectView

## UIBlurEffect

```
enum UIBlurEffectStyle : Int {  
    case ExtraLight  
    case Light  
    case Dark  
}
```

```
let blurEffect = UIBlurEffect(style: .ExtraLight)
```

```
let blurView = UIVisualEffectView(effect:  
    blurEffect)
```





# UIVisualEffectView

## UIBlurEffect

```
enum UIBlurEffectStyle : Int {  
    case ExtraLight  
    case Light  
    case Dark  
}
```

```
let blurEffect = UIBlurEffect(style: .ExtraLight)
```

```
let blurView = UIVisualEffectView(effect:  
    blurEffect)
```





# UIVisualEffectView

## UIBlurEffect

```
enum UIBlurEffectStyle : Int {  
    case ExtraLight  
    case Light  
    case Dark  
}
```

```
let blurEffect = UIBlurEffect(style: .ExtraLight)
```

```
let blurView = UIVisualEffectView(effect:  
    blurEffect)
```





# UIVisualEffectView

## UIBlurEffect

```
enum UIBlurEffectStyle : Int {  
    case ExtraLight  
    case Light  
    case Dark  
}
```

```
let blurEffect = UIBlurEffect(style: .ExtraLight)
```

```
let blurView = UIVisualEffectView(effect:  
    blurEffect)
```





# UIVisualEffectView

## UIVibrancyEffect

```
let vibrancyEffect = UIVibrancyEffect(forBlurEffect:  
blurEffect)
```

```
let vibrancyView =  
UIVisualEffectView(effect:vibrancyEffect)
```

```
blurView.contentView.addSubview(vibrancyView)
```

```
vibrancyView.contentView.addSubview(label)
```





# UIVisualEffectView

## UIVibrancyEffect

```
let vibrancyEffect = UIVibrancyEffect(forBlurEffect:  
blurEffect)
```

```
let vibrancyView =  
UIVisualEffectView(effect:vibrancyEffect)
```

```
blurView.contentView.addSubview(vibrancyView)
```

```
vibrancyView.contentView.addSubview(label)
```





# UIVisualEffectView

## UIVibrancyEffect

```
let vibrancyEffect = UIVibrancyEffect(forBlurEffect:  
blurEffect)
```

```
let vibrancyView =  
UIVisualEffectView(effect:vibrancyEffect)
```

```
blurView.contentView.addSubview(vibrancyView)
```

```
vibrancyView.contentView.addSubview(label)
```





# UIVisualEffectView

## UIVibrancyEffect

```
let vibrancyEffect = UIVibrancyEffect(forBlurEffect:  
blurEffect)
```

```
let vibrancyView =  
UIVisualEffectView(effect:vibrancyEffect)
```

```
blurView.contentView.addSubview(vibrancyView)
```

```
vibrancyView.contentView.addSubview(label)
```





# UIVisualEffectView

## UIVibrancyEffect

```
let vibrancyEffect = UIVibrancyEffect(forBlurEffect:  
blurEffect)
```

```
let vibrancyView =  
UIVisualEffectView(effect:vibrancyEffect)
```

```
blurView.contentView.addSubview(vibrancyView)
```

```
vibrancyView.contentView.addSubview(label)
```





# UIVisualEffectView

And action!

NEW





# UIVisualEffectView

And action!

NEW

Bounds



Iceland 001.jpg (2500.0 x 1668.0)



# UIVisualEffectView

And action!

Bounds

Effect

NEW



Iceland 001.jpg (2500.0 x 1668.0)



# 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



# UIVisualEffectView

## Anatomy of an effect





# UIVisualEffectView

## Anatomy of an effect





# UIVisualEffectView

## Anatomy of an effect



Offscreen Pass

# UIVisualEffectView

## Offscreen Passes

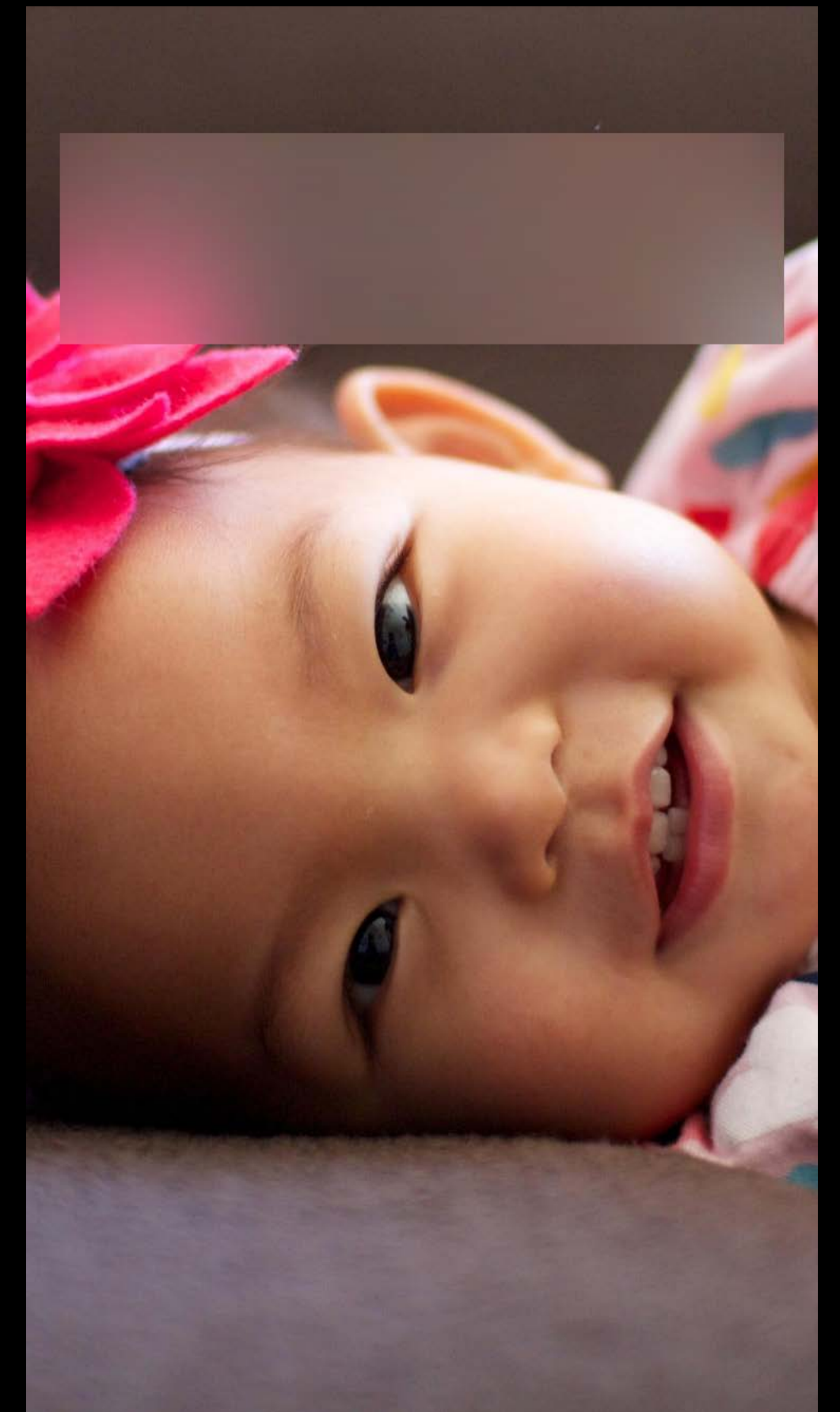
Alpha

Masking

Blur

Vibrancy

Snapshotting



# UIVisualEffectView

## Offscreen Passes

Alpha

Masking

Blur

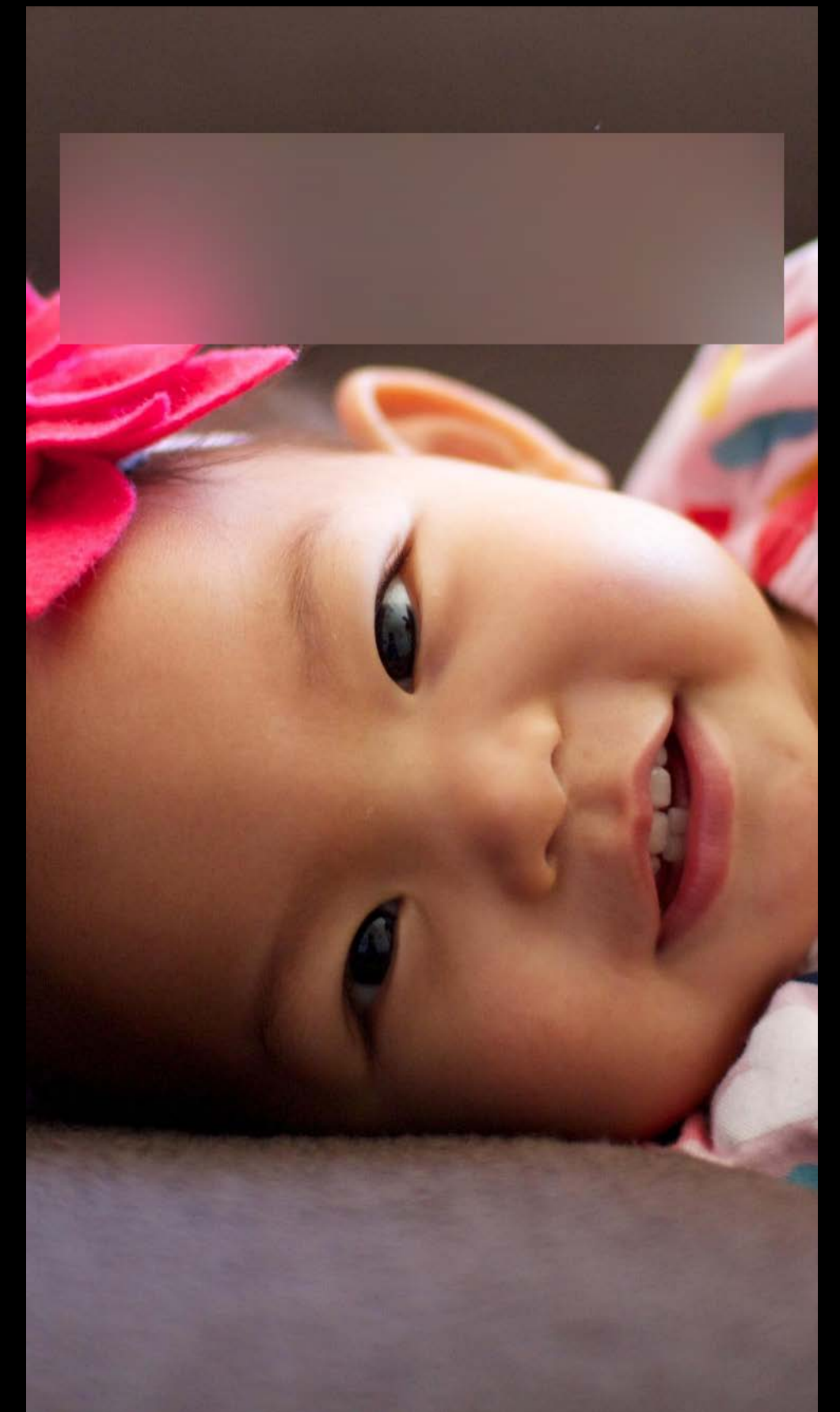
Vibrancy

Snapshotting

```
UIView.snapshotViewAfterScreenUpdates(afterUpdates:)
```

```
UIView.drawViewHierarchyInRect(rect:, afterScreenUpdates:)
```

```
UIScreen.snapshotViewAfterScreenUpdates()
```





# UIVisualEffectView

## Offscreen Passes

Alpha

Masking

Blur

Vibrancy

Snapshotting

```
UIView.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



Capture Area



# UIVisualEffectView

Anatomy of a broken effect



Capture Area

Nothing!

# UIVisualEffectView

Fixing broken effects

# UIVisualEffectView

Fixing broken effects

For help in lldb

– (NSString \*)\_what'sWrongWithThisEffect

# UIVisualEffectView

Fixing broken effects

For help in lldb

– (NSString \*)\_whatWrongWithThisEffect

```
(lldb) po [myEffectView _whatWrongWithThisEffect]
```

# UIVisualEffectView

## Fixing broken effects

For help in lldb

– (NSString \*)\_whatWrongWithThisEffect

```
(lldb) po [myEffectView _whatWrongWithThisEffect]
```

ISSUE: One or more masking superviews have been found.

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

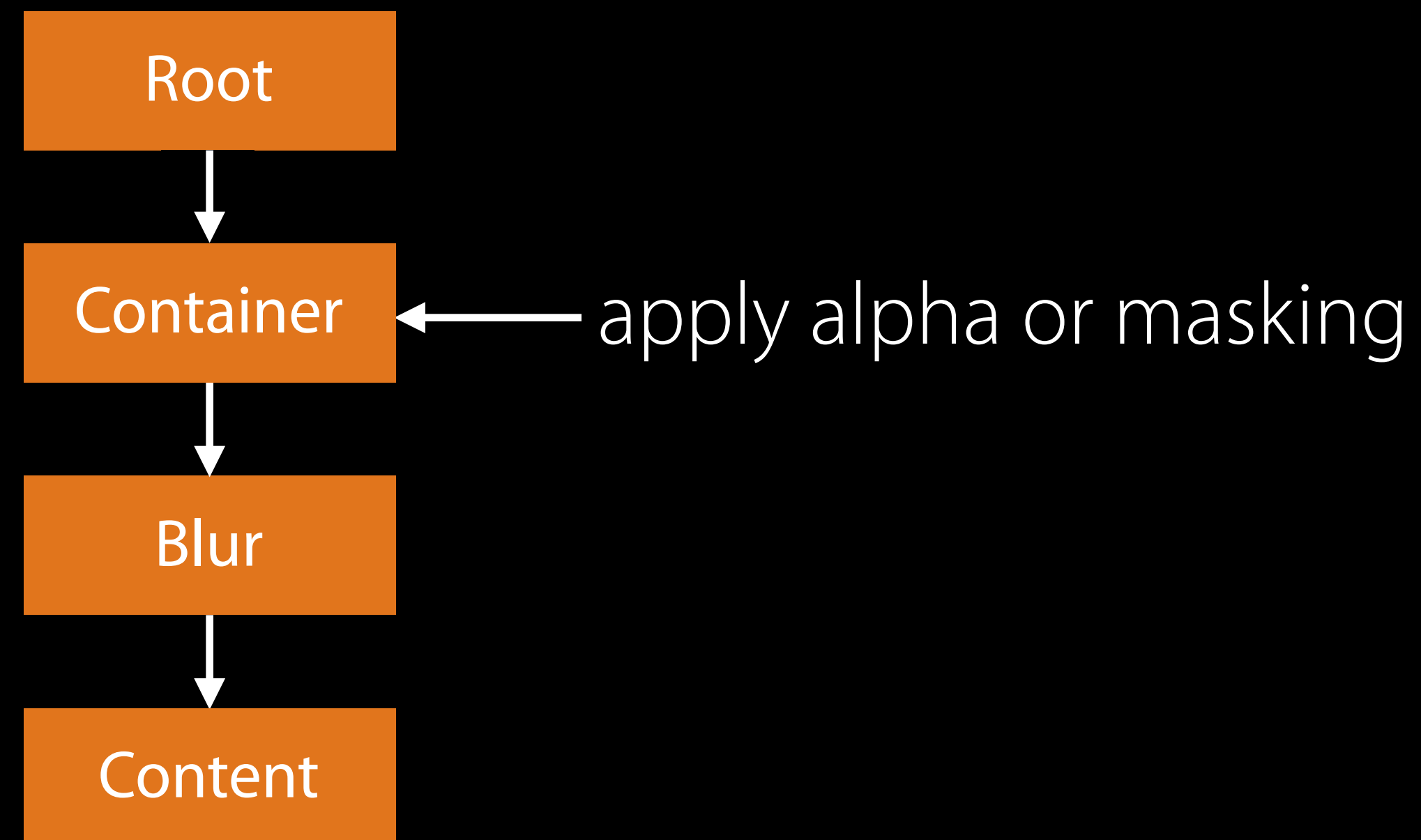
UIVisualEffectView cannot be masked, you will need to isolate this effect view into a hierarchy that is not masked or not use masks.

# UIVisualEffectView

Fixing broken effects

Rearrange view hierarchy

- Effective for Alpha and Masking

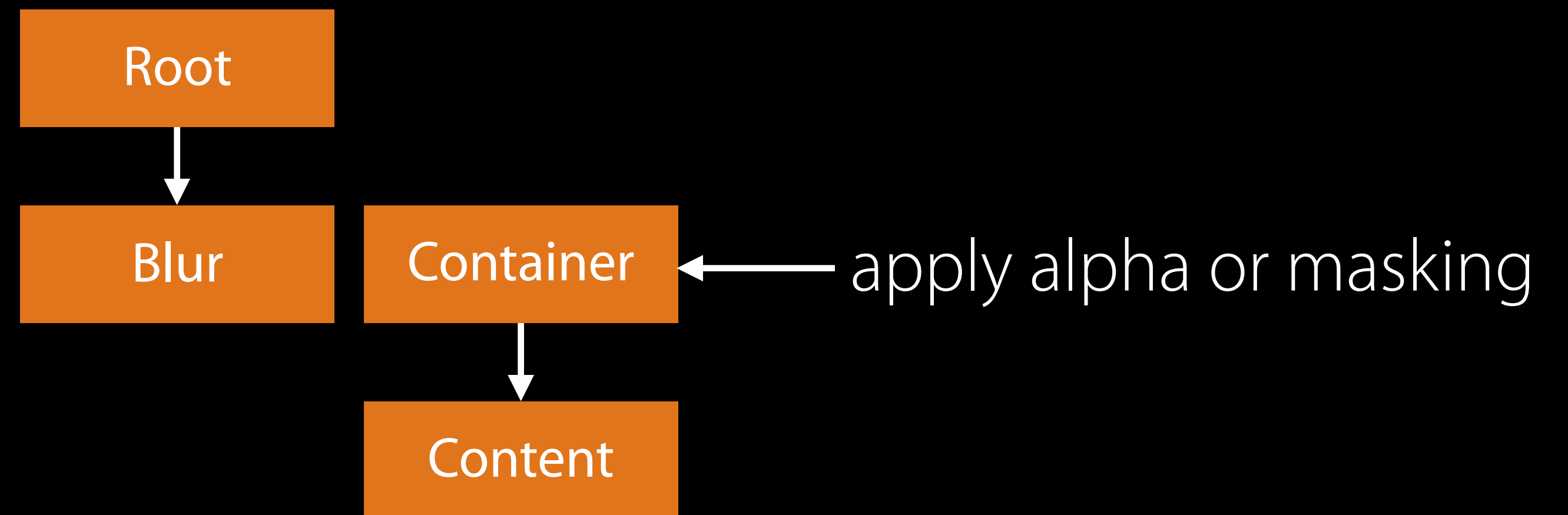


# UIVisualEffectView

Fixing broken effects

Rearrange view hierarchy

- Effective for Alpha and Masking

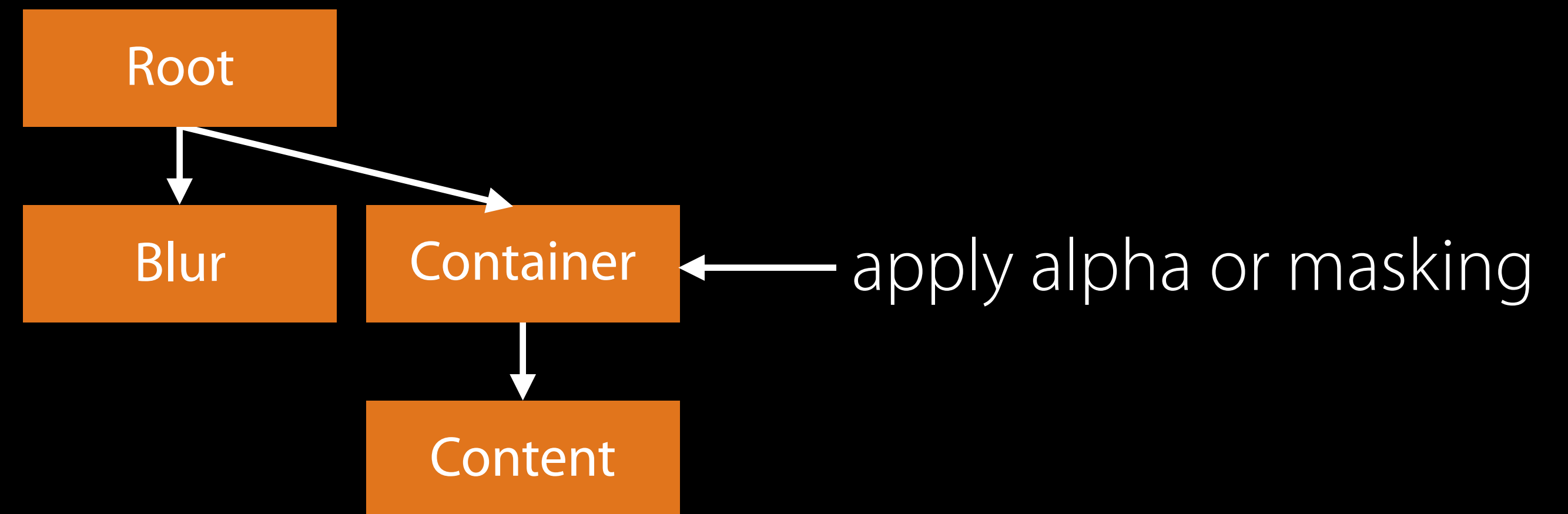


# UIVisualEffectView

## Fixing broken effects

Rearrange view hierarchy

- Effective for Alpha and Masking

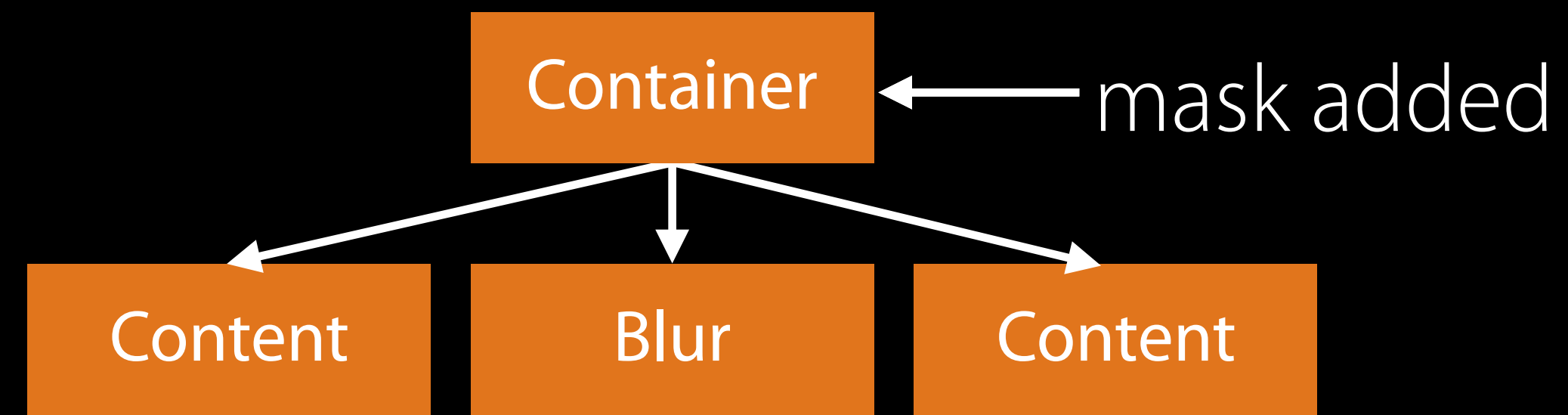




# UIVisualEffectView

Fixing broken effects

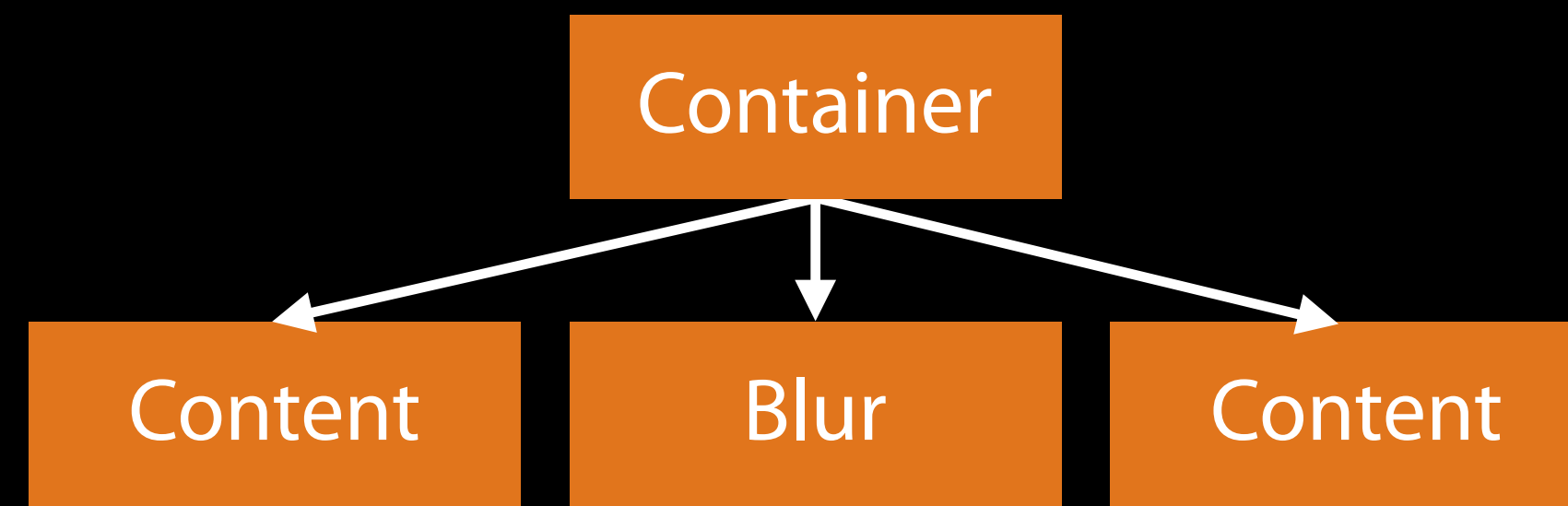
Mask views individually



# UIVisualEffectView

Fixing broken effects

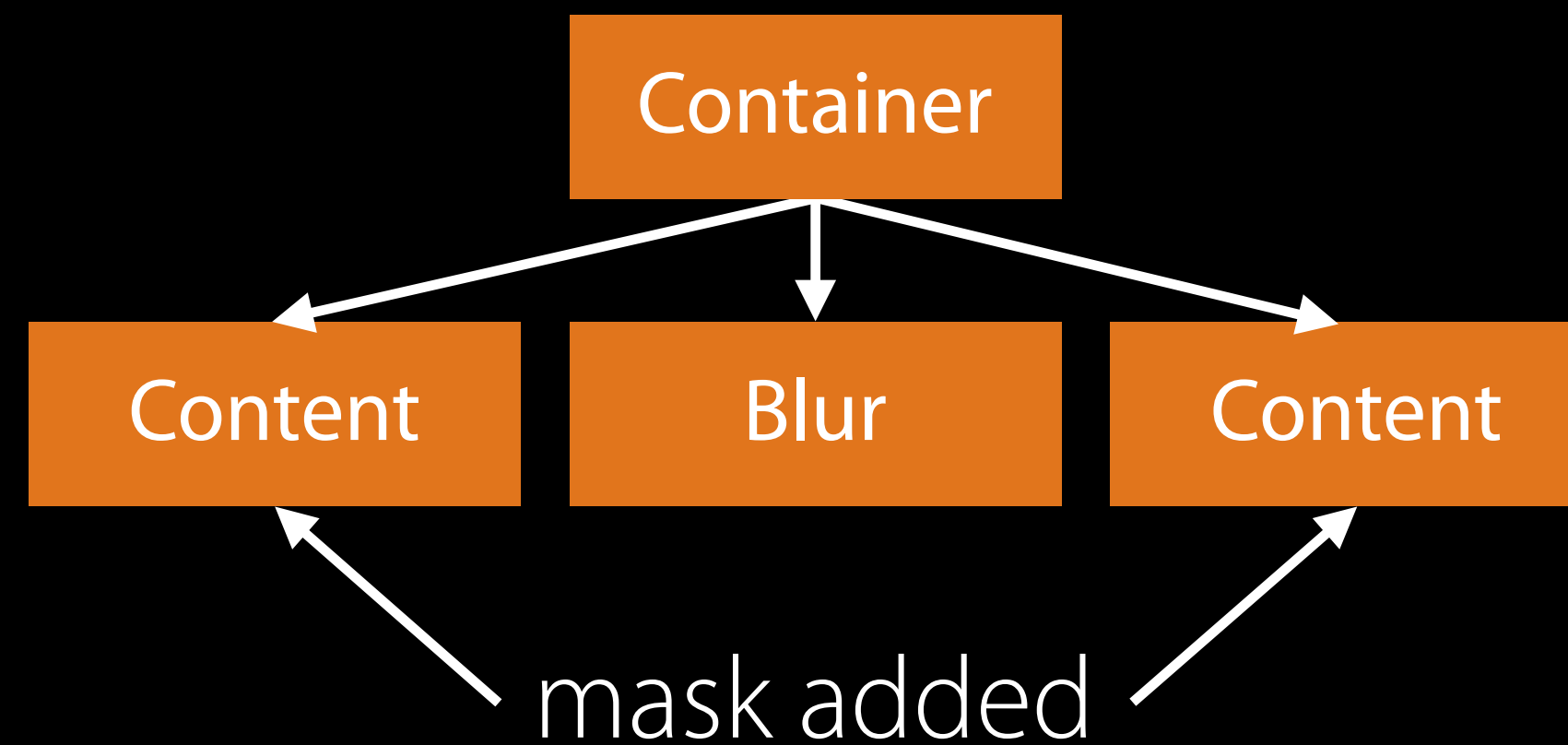
Mask views individually



# UIVisualEffectView

Fixing broken effects

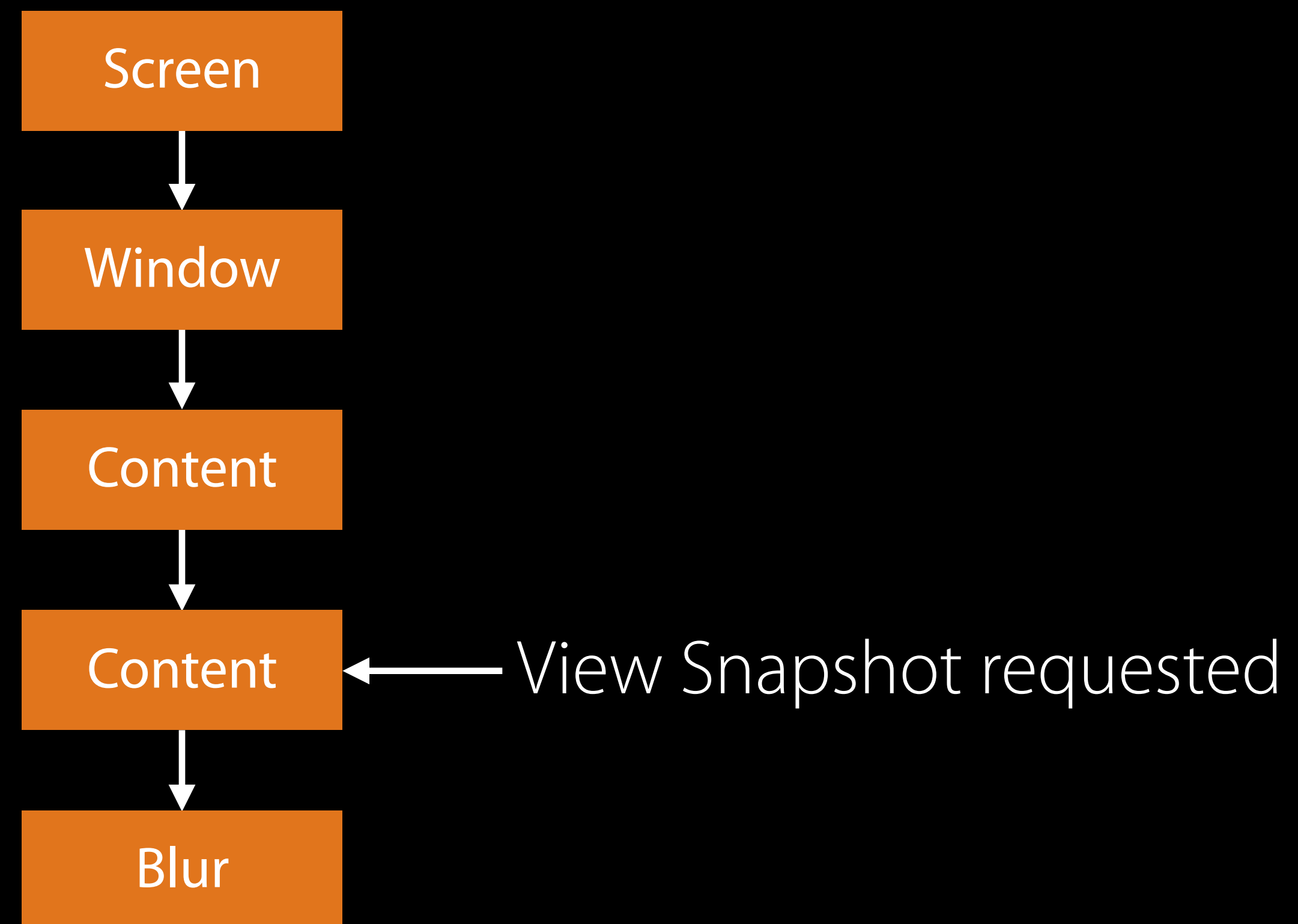
Mask views individually



# UIVisualEffectView

Fixing broken effects

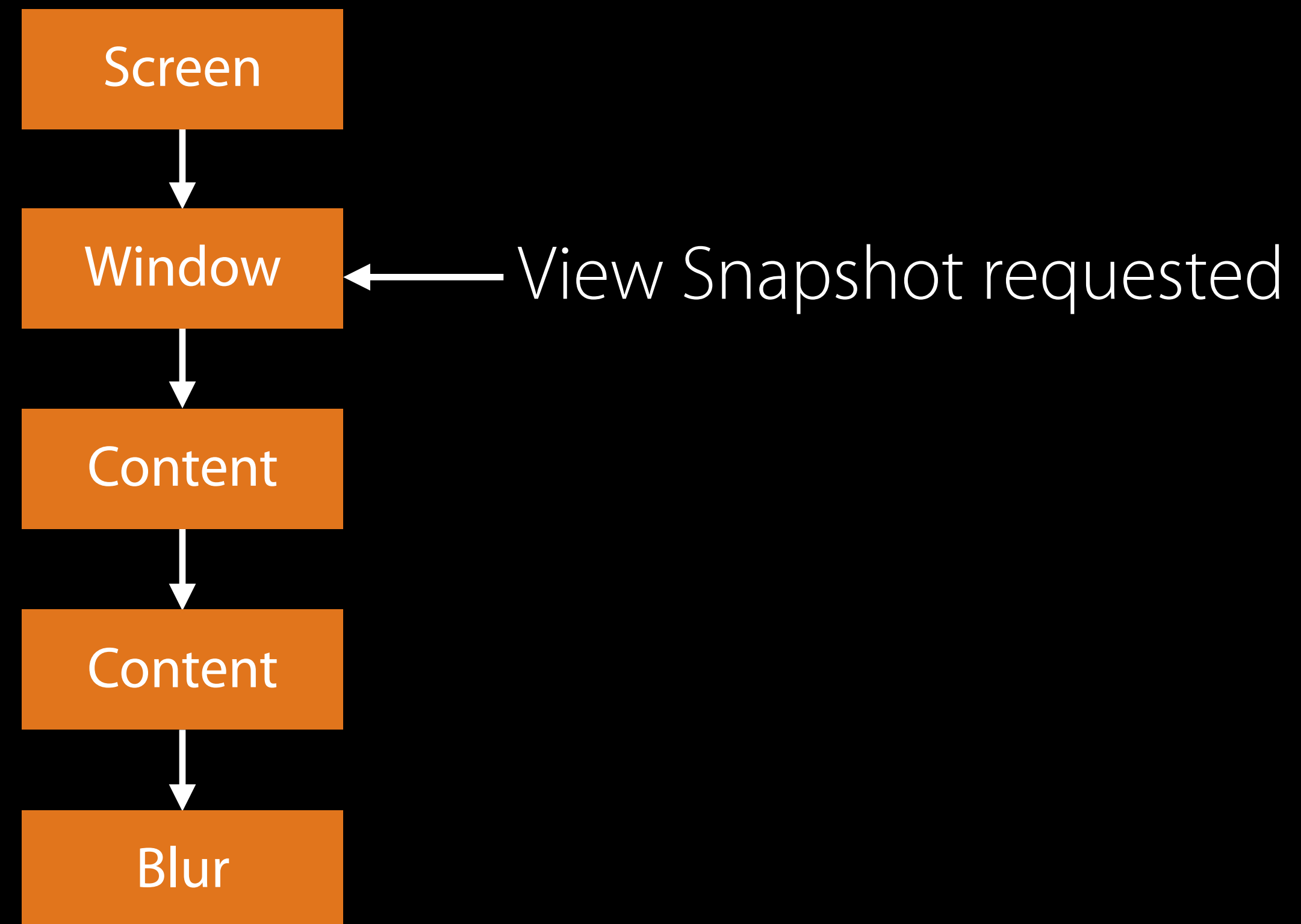
Snapshot the window or screen



# UIVisualEffectView

Fixing broken effects

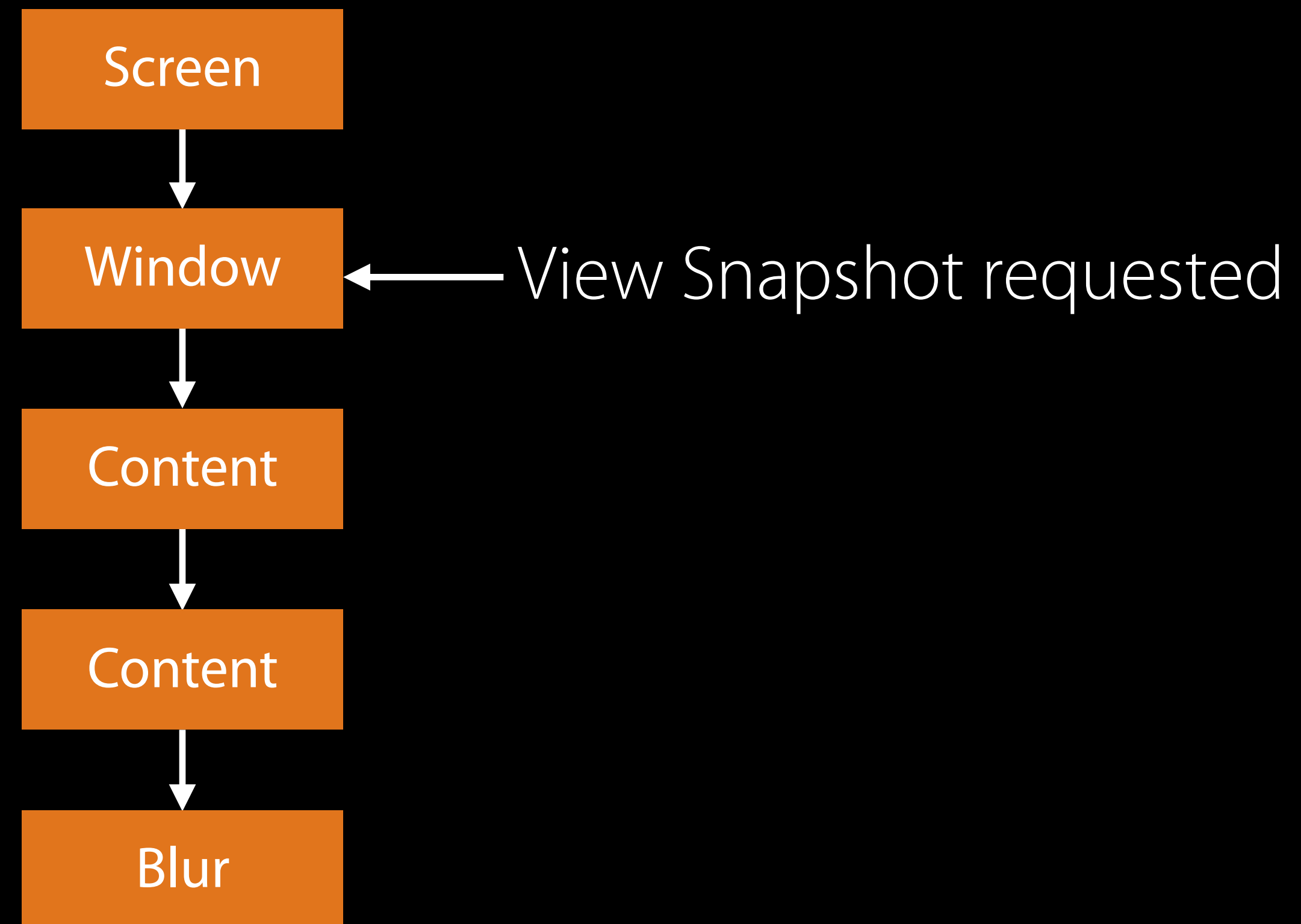
Snapshot the window or screen



# UIVisualEffectView

Fixing broken effects

Snapshot the window or screen

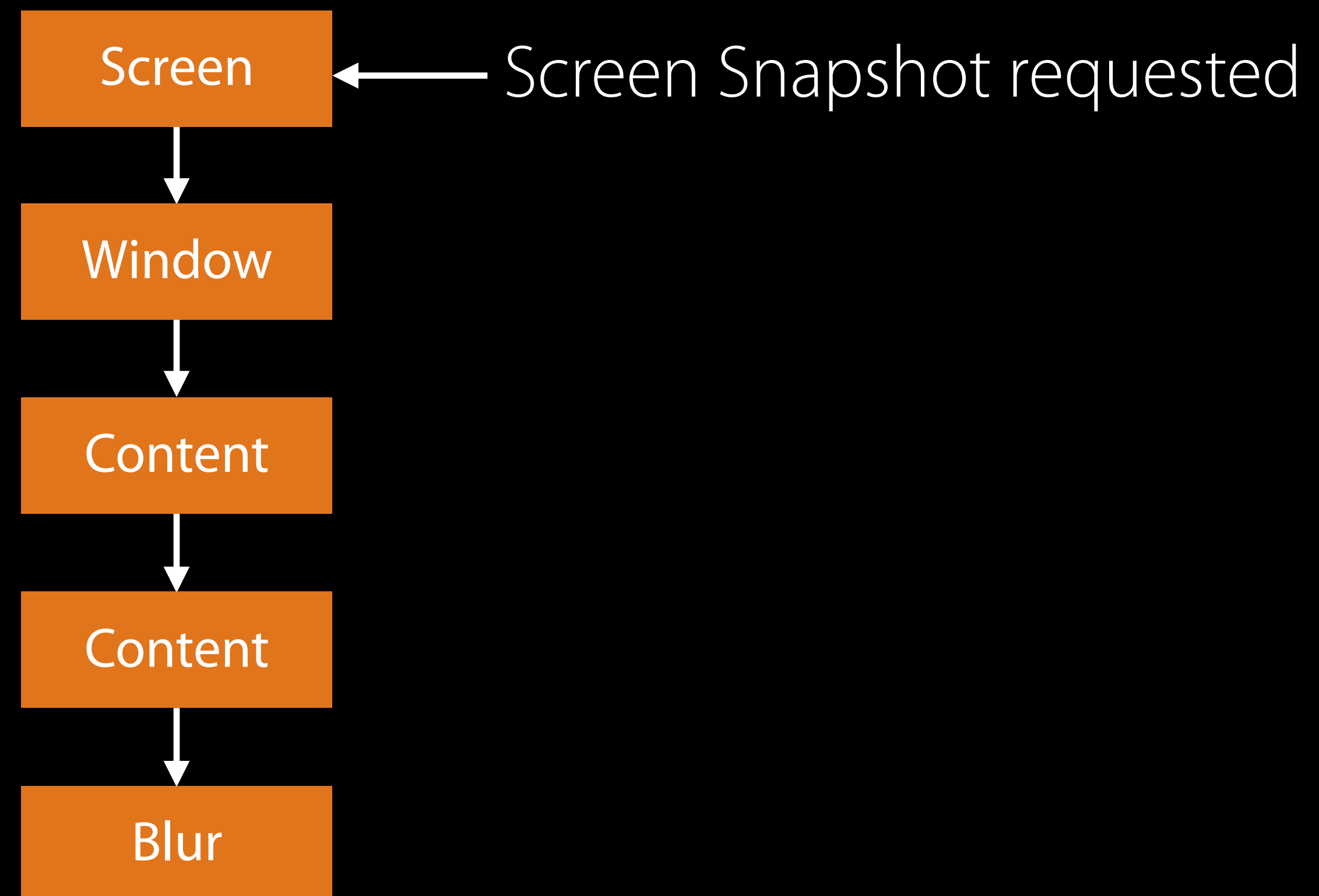




# UIVisualEffectView

Fixing broken effects

Snapshot the window or screen



# 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



Lola 011.jpg

# 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

# Summary

Focus on the user experience

Always consider performance impact when adding advanced effects



# Related Sessions and Labs

Getting Started with Multitasking on iPad in iOS 9	Presidio	Tuesday 4:30PM
Multitasking Essentials for Media-Based Apps on iPad in iOS 9	Pacific Heights	Wednesday 2:30PM
Optimizing Your App for Multitasking on iPad in iOS 9	Presidio	Wednesday 3:30PM
Mysteries of Auto Layout, Part 1	Presidio	Thursday 11:00AM
Mysteries of Auto Layout, Part 2	Presidio	Thursday 1:30PM
Building Responsive and Efficient Apps with GCD	Nob Hill	Friday 10:00AM
UIKit and UIKit Dynamics Lab	Frameworks Lab C	Friday 11:00AM

# More Information

## Documentation

UIDynamicAnimator Class Reference

UIFieldBehavior Class Reference

UIDynamicBehavior Class Reference

UIAttachmentBehavior Class Reference

## Sample Code

StickyCorners

<http://developer.apple.com/library>

## Technical Support

Apple Developer Forums

Developer Technical Support

Curt Rothert

App Frameworks Evangelist

[rothert@apple.com](mailto:rothert@apple.com)

