

# Building Interruptible and Responsive Interactions

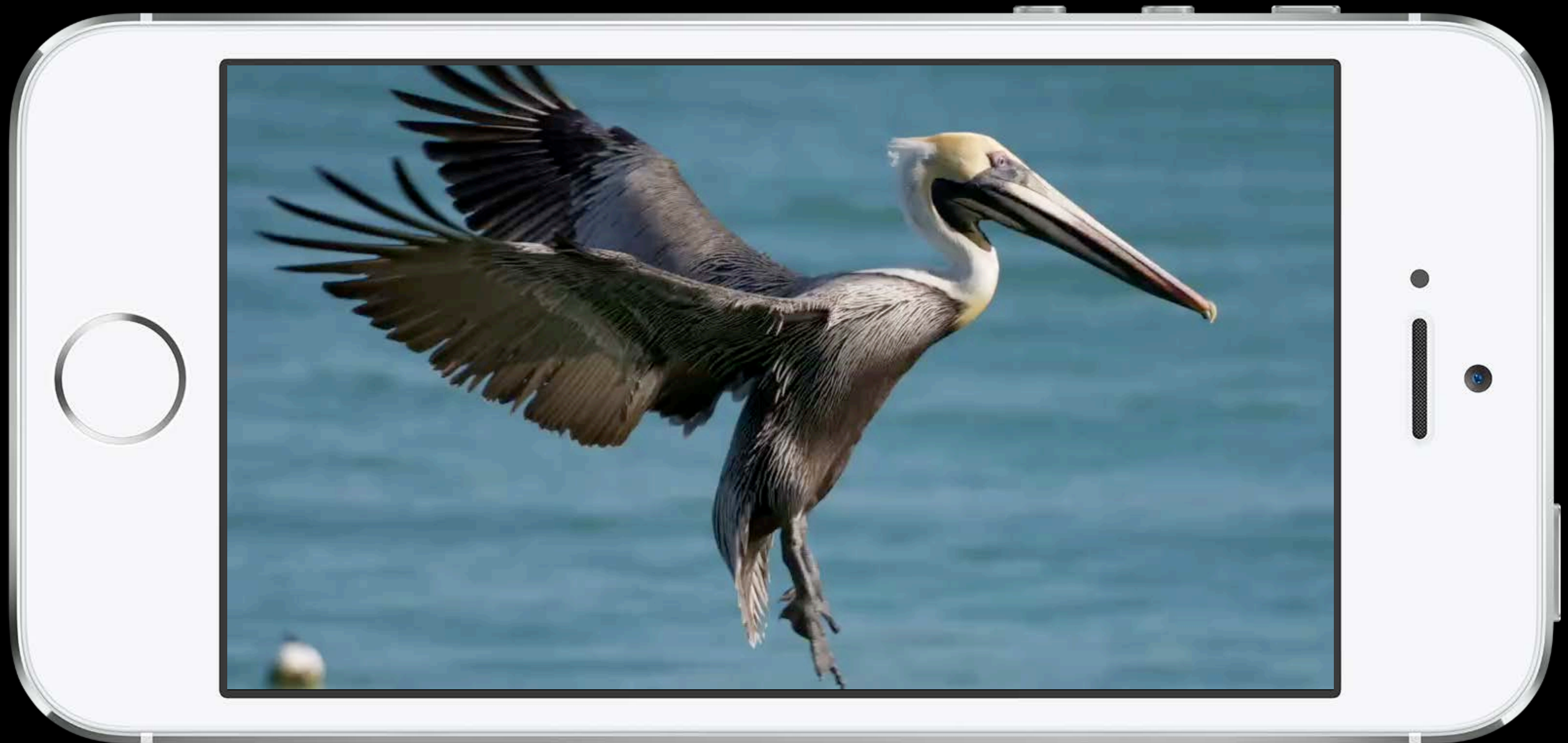
Session 236

Josh Shaffer

iOS Apps and Frameworks Engineer

Andy Matuschak

iOS Apps and Frameworks Engineer





# Transitions

# Gesture to Animation

Gesture to Animation

---

Animation to Animation

Gesture to Animation

---

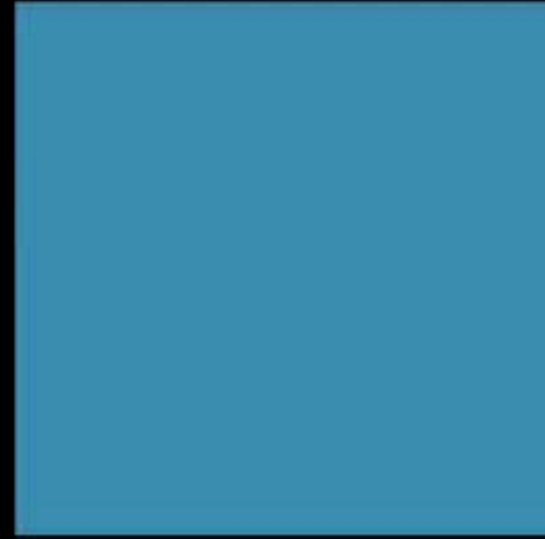
Animation to Animation

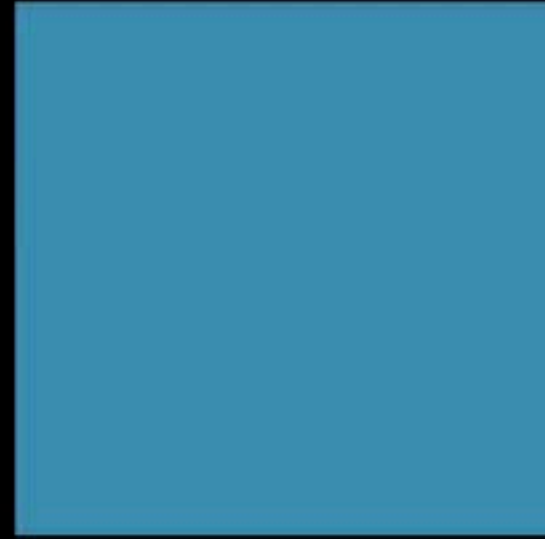
---

Animation to Gesture

# From Gesture to Animation







# Computing Velocity

# Computing Velocity

UIPanGestureRecognizer

```
func velocityInView(view: UIView) -> CGPoint
```

# Computing Velocity

UIPanGestureRecognizer

```
func velocityInView(view: UIView) -> CGPoint
```

UIPinchGestureRecognizer

```
var velocity: CGFloat { get }
```

# Computing Velocity

UIPanGestureRecognizer

```
func velocityInView(view: UIView) -> CGPoint
```

UIPinchGestureRecognizer

```
var velocity: CGFloat { get }
```

UIRotationGestureRecognizer

```
var velocity: CGFloat { get }
```

```
animateWithDuration(_:  
                    delay:  
                    options:  
                    animations:  
                    completion:)
```

```
animateWithDuration(_:  
                    delay:  
usingSpringWithDamping:  
    initialSpringVelocity:  
                    options:  
                    animations:  
                    completion:)
```

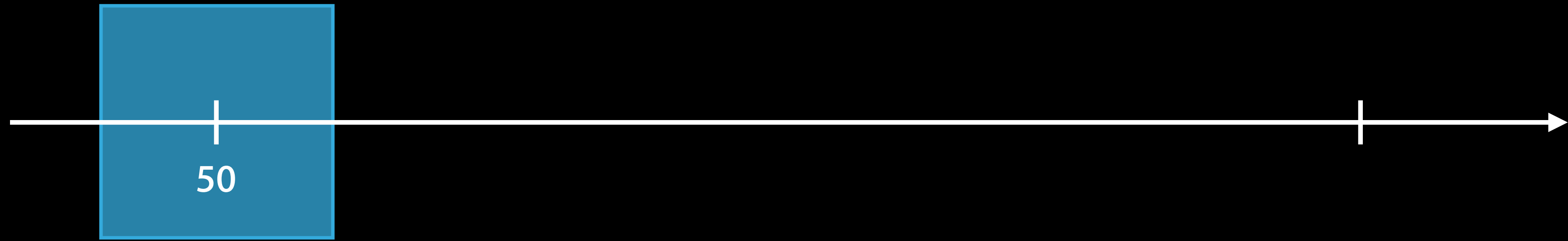


`initialSpringVelocity`

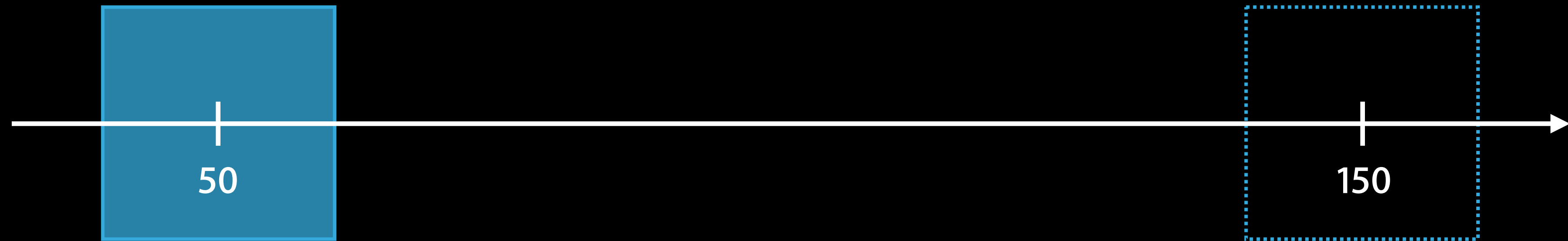
initialSpringVelocity



# initialSpringVelocity



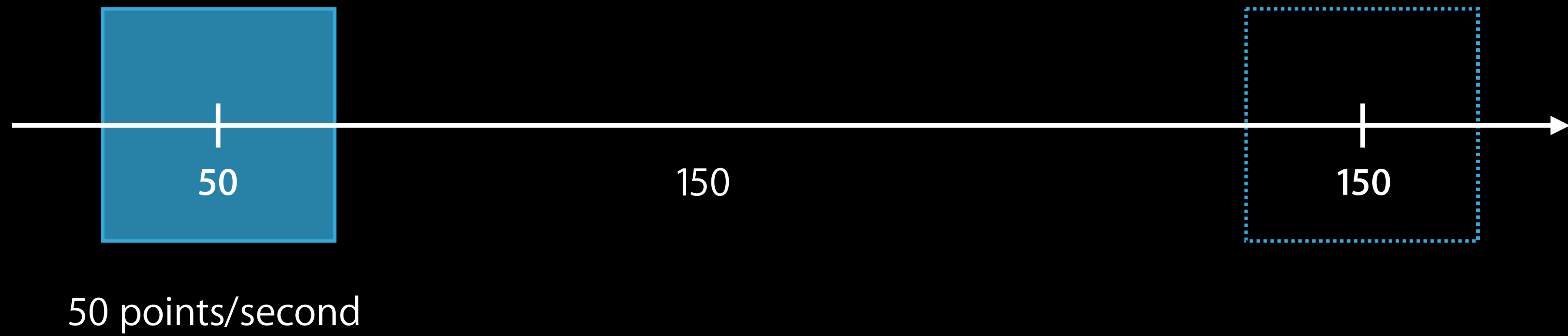
# initialSpringVelocity



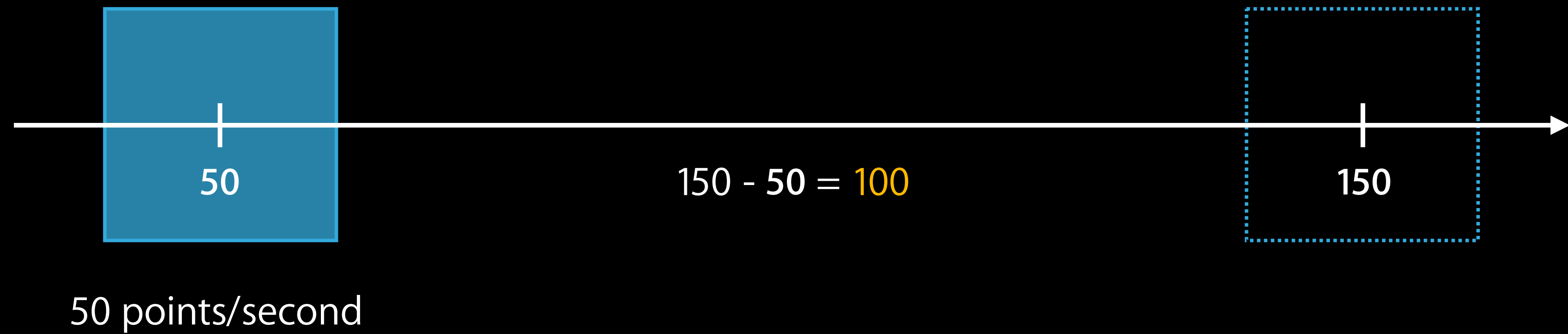
# initialSpringVelocity



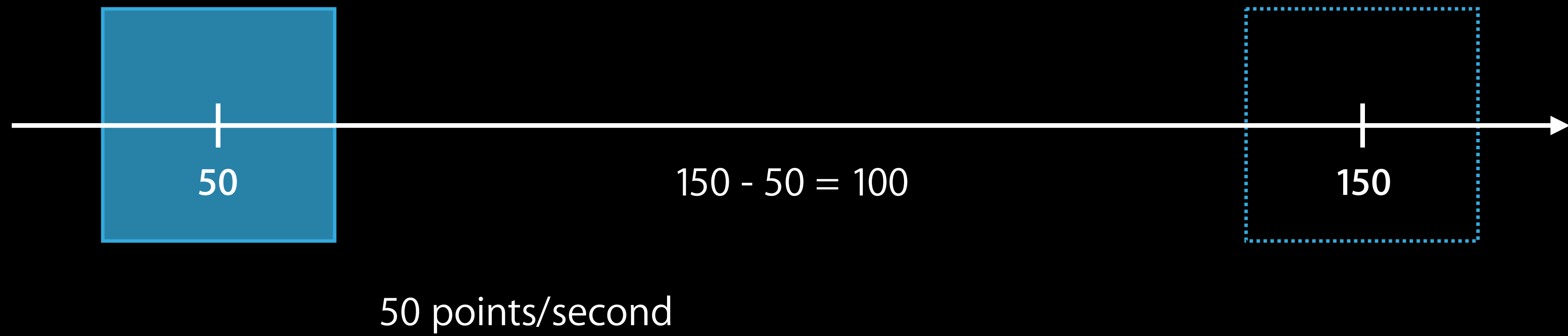
# initialSpringVelocity



# initialSpringVelocity

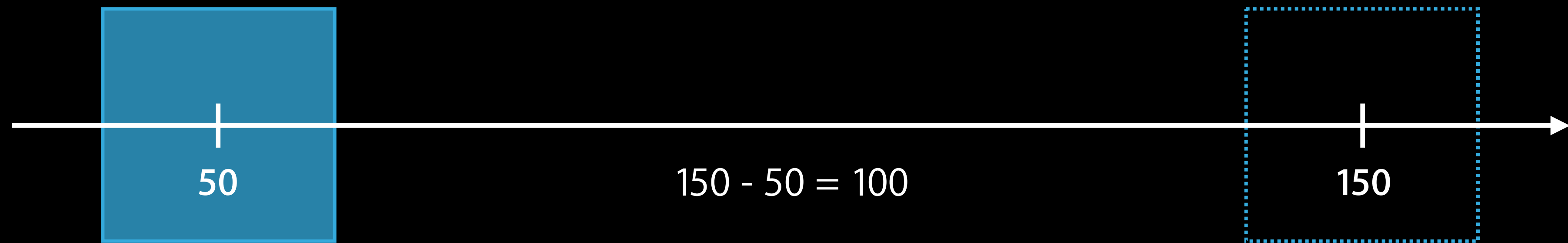


# initialSpringVelocity



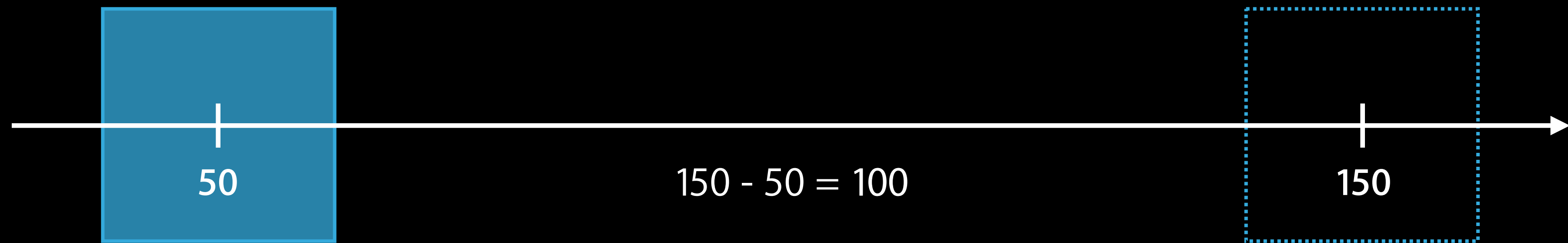


# initialSpringVelocity



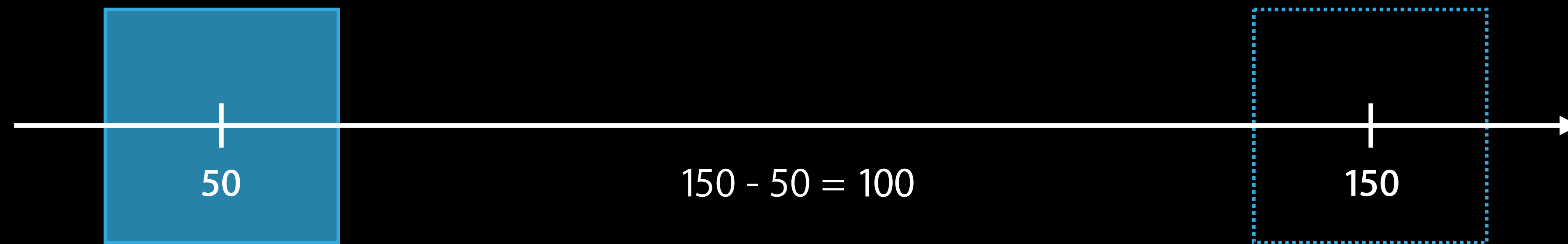
$$50 \text{ points/second} \div 100 \text{ points} = 0.5 \text{ units/second}$$

# initialSpringVelocity



$$50 \text{ points/second} \div 100 \text{ points} = 0.5 \text{ units/second}$$

# initialSpringVelocity



$$50 \text{ points/second} \div 100 \text{ points} = 0.5 \text{ units/second}$$

```
UIView.animateWithDuration( 1.5,  
                           delay: 0.0,  
                           usingSpringWithDamping: 0.7,  
                           initialSpringVelocity: 0.5,  
                           options: nil,  
                           animations: { view.centerX = 150 },  
                           completion: nil)
```







UIDynamicAnimator

# UIDynamicAnimator Setup



# UIDynamicAnimator Setup

```
var dynamicAnimator = UIDynamicAnimator?
```

# UIDynamicAnimator Setup

```
var dynamicAnimator = UIDynamicAnimator?  
let dynamicItemBehavior = UIDynamicItemBehavior(items: nil)
```

# UIDynamicAnimator Setup

```
var dynamicAnimator = UIDynamicAnimator?  
let dynamicItemBehavior = UIDynamicItemBehavior(items: nil)  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
  
}
```

# UIDynamicAnimator Setup

```
var dynamicAnimator = UIDynamicAnimator?  
let dynamicItemBehavior = UIDynamicItemBehavior(items: nil)  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    dynamicAnimator = UIDynamicAnimator(referenceView: view)  
  
}
```

# UIDynamicAnimator Setup

```
var dynamicAnimator = UIDynamicAnimator?  
let dynamicItemBehavior = UIDynamicItemBehavior(items: nil)  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    dynamicAnimator = UIDynamicAnimator(referenceView: view)  
  
    dynamicItemBehavior.resistance = 3.0  
    dynamicItemBehavior.angularResistance = 3.0  
  
}
```

# UIDynamicAnimator Setup

```
var dynamicAnimator = UIDynamicAnimator?  
let dynamicItemBehavior = UIDynamicItemBehavior(items: nil)  
  
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    dynamicAnimator = UIDynamicAnimator(referenceView: view)  
  
    dynamicItemBehavior.resistance = 3.0  
    dynamicItemBehavior.angularResistance = 3.0  
  
    dynamicAnimator!.addBehavior(dynamicItemBehavior)  
}
```

# UIDynamicAnimator Velocity Transfer

# UIDynamicAnimator Velocity Transfer

```
let targetView = panGestureRecognizer.view
```



# UIDynamicAnimator Velocity Transfer

```
let targetView = panGestureRecognizer.view  
switch panGestureRecognizer.state {  
  
  
  
  
  
  
  
  
  
}
```

# UIDynamicAnimator Velocity Transfer

```
let targetView = panGestureRecognizer.view

switch panGestureRecognizer.state {
    case .Ended:

}
}
```

# UIDynamicAnimator Velocity Transfer

```
let targetView = panGestureRecognizer.view

switch panGestureRecognizer.state {
  case .Ended:
    let v = panGestureRecognizer.velocityInView(targetView.superview)

}
```

# UIDynamicAnimator Velocity Transfer

```
let targetView = panGestureRecognizer.view

switch panGestureRecognizer.state {
    case .Ended:
        let v = panGestureRecognizer.velocityInView(targetView.superview)

        dynamicItemBehavior.addLinearVelocity(v, forItem: targetView)
}
```





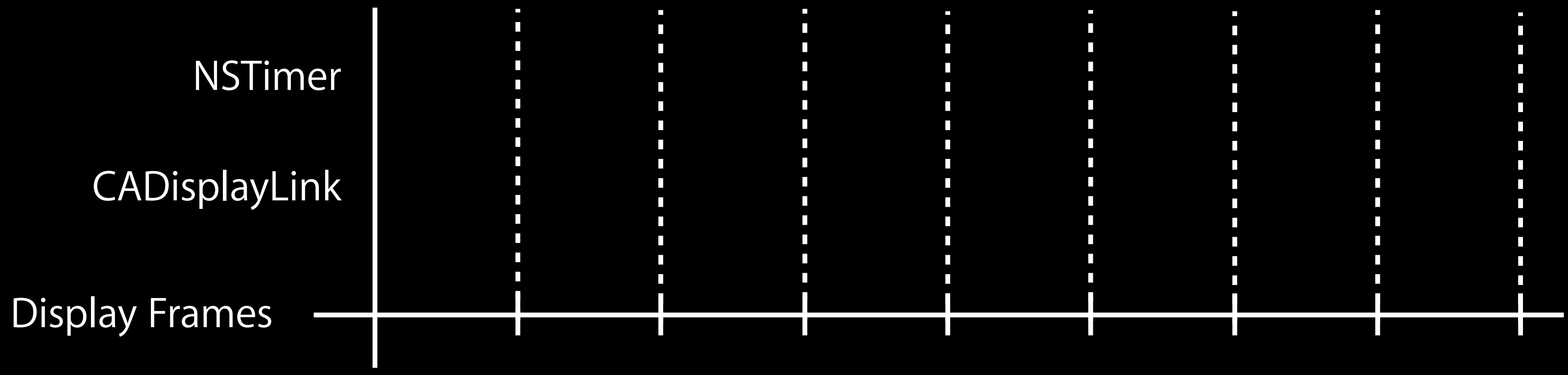
animateWithDuration

UIDynamicAnimator



CADisplayLink

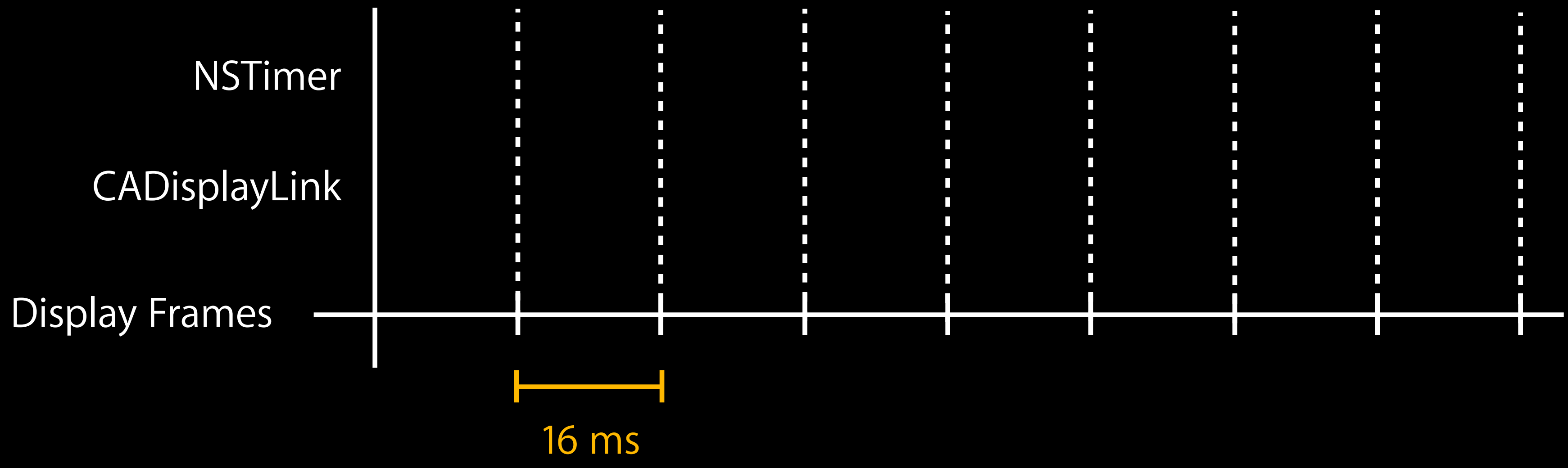
...but what about NSTimer?

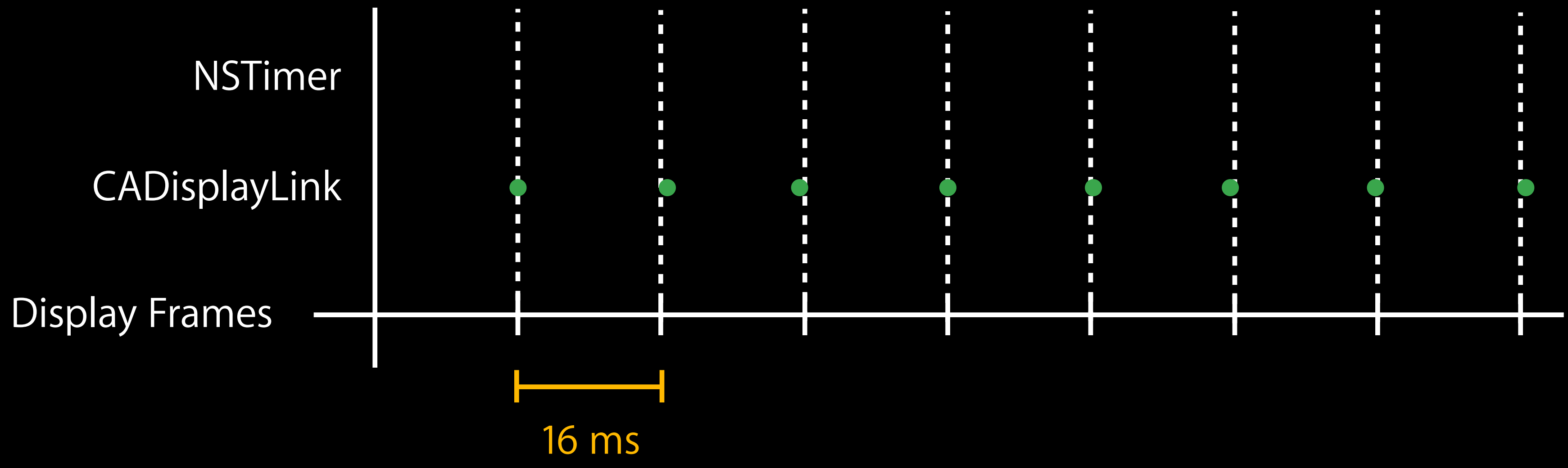


NSTimer

CADisplayLink

Display Frames





NSTimer

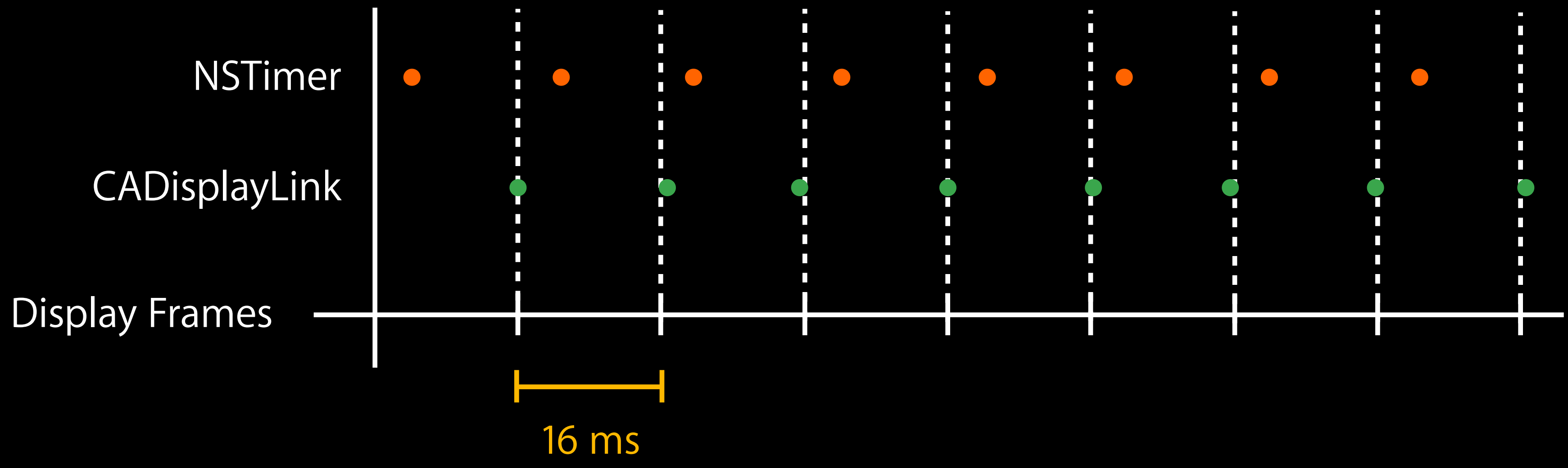
CADisplayLink

Display Frames



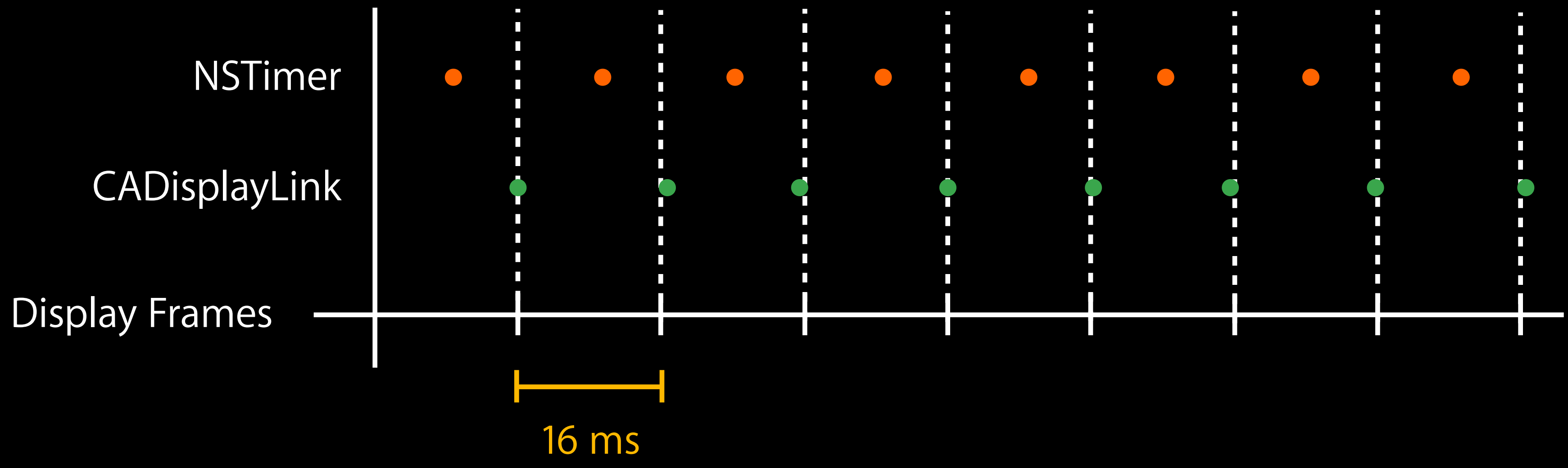
16 ms

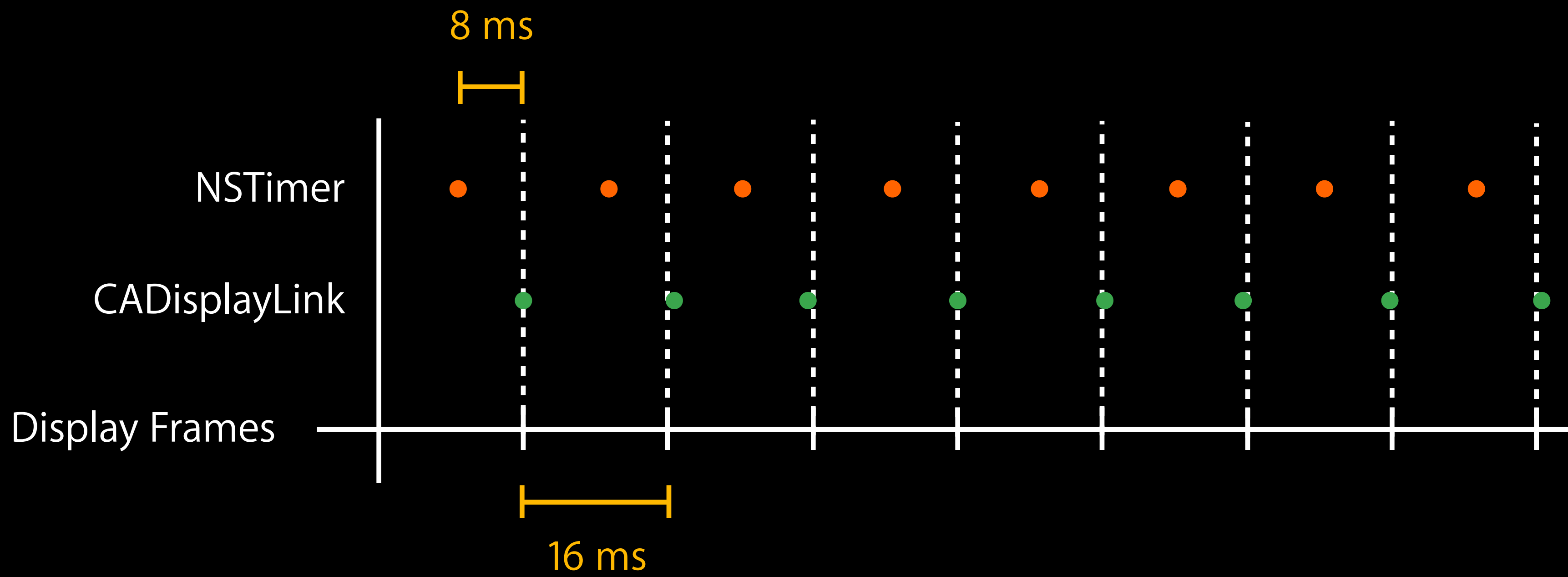


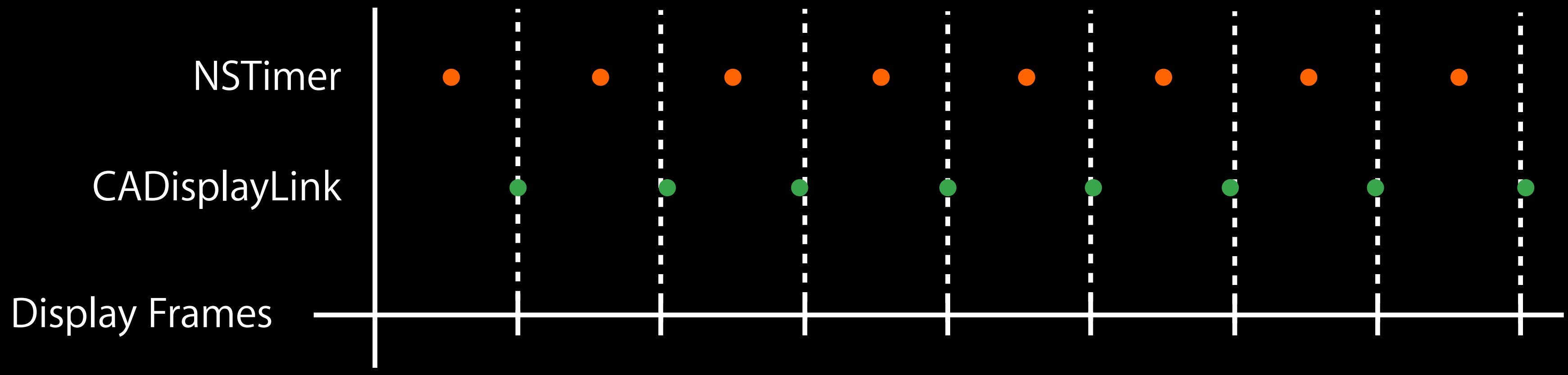




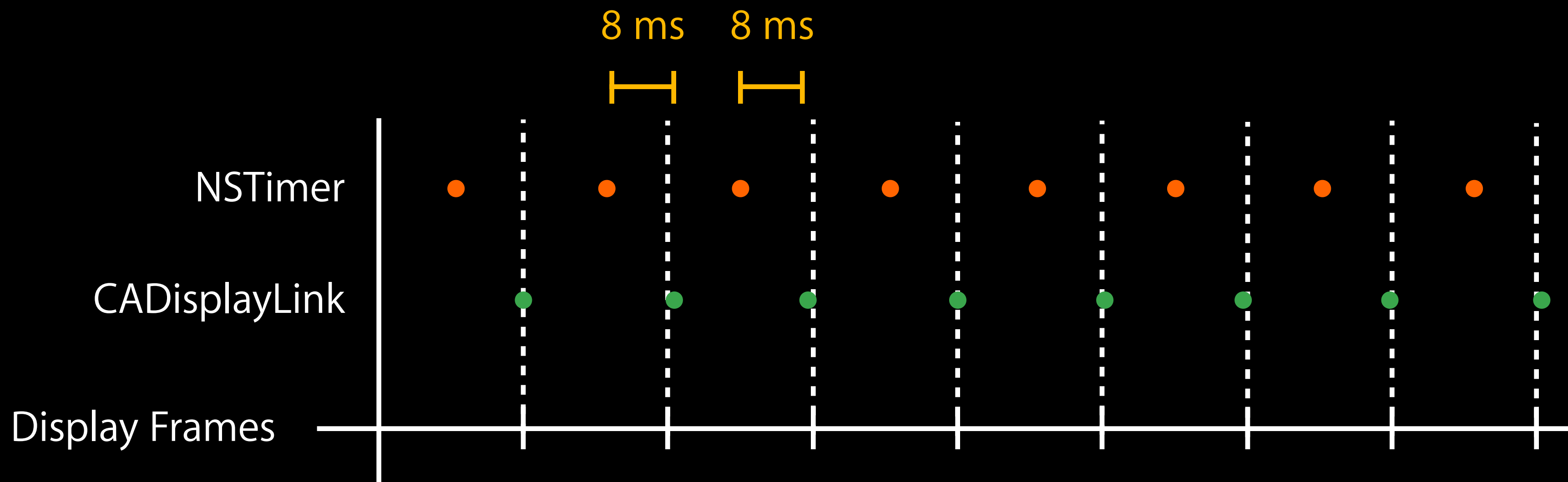


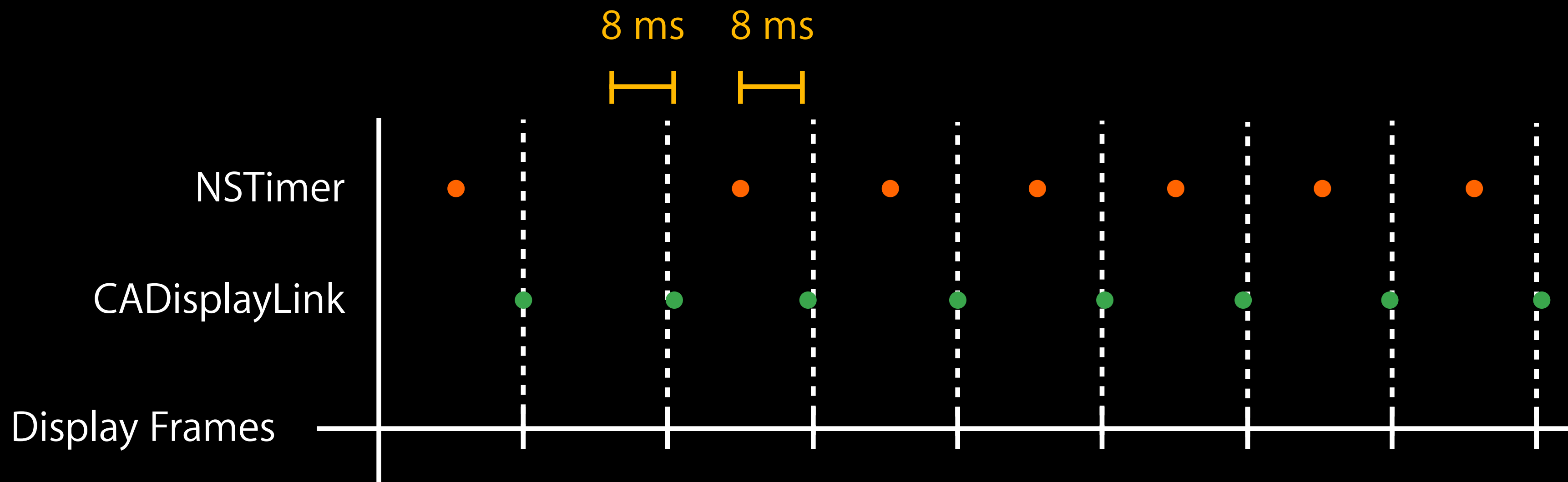


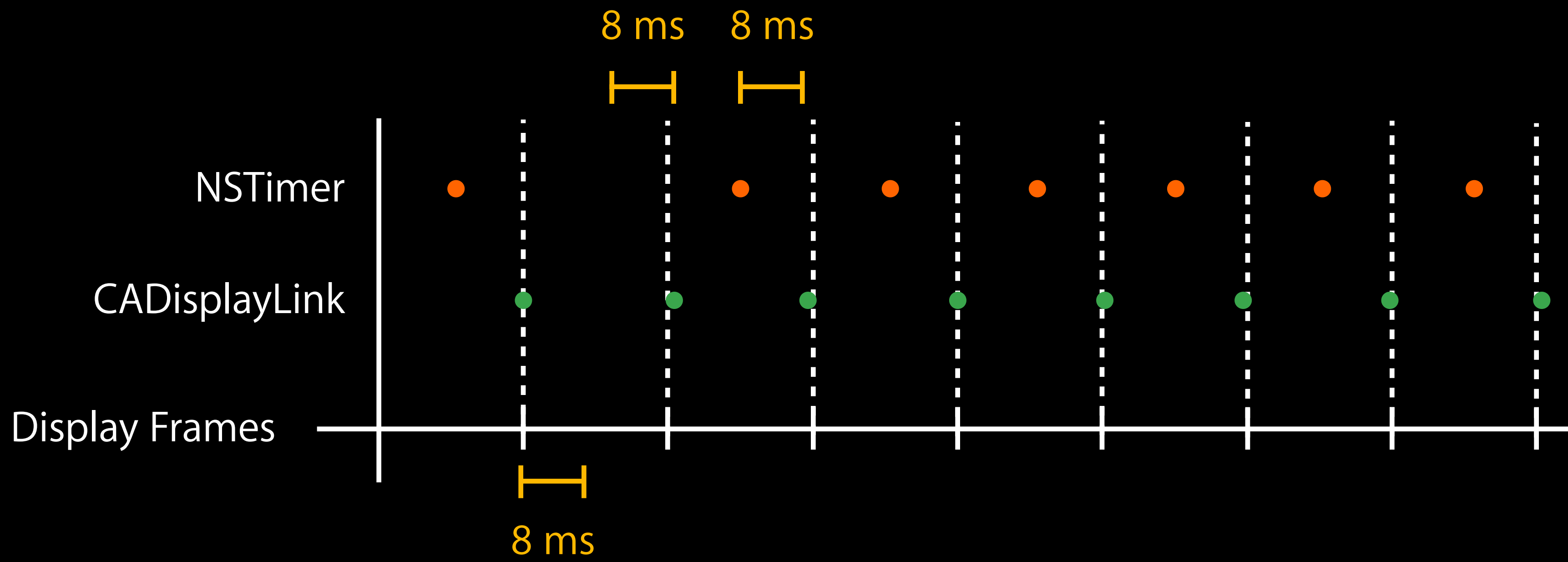


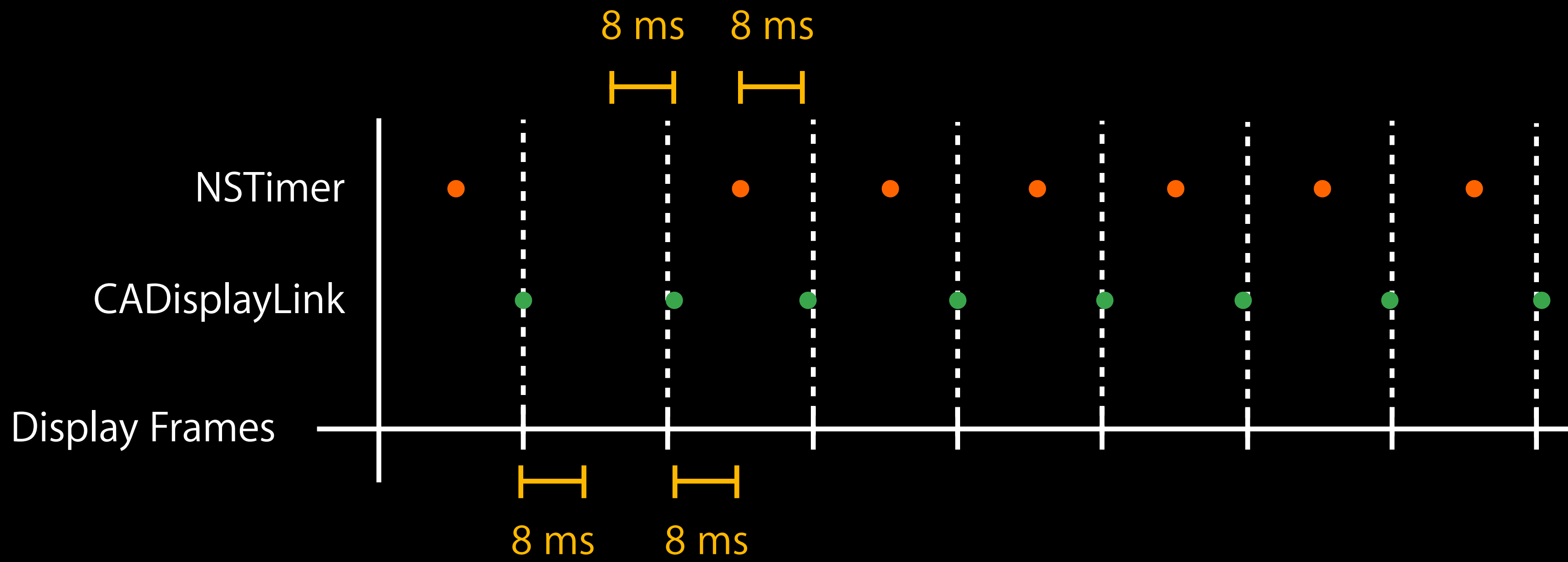










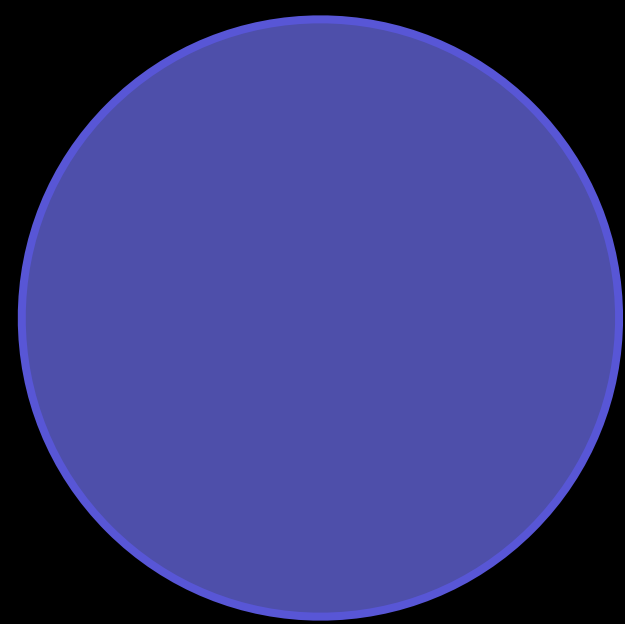




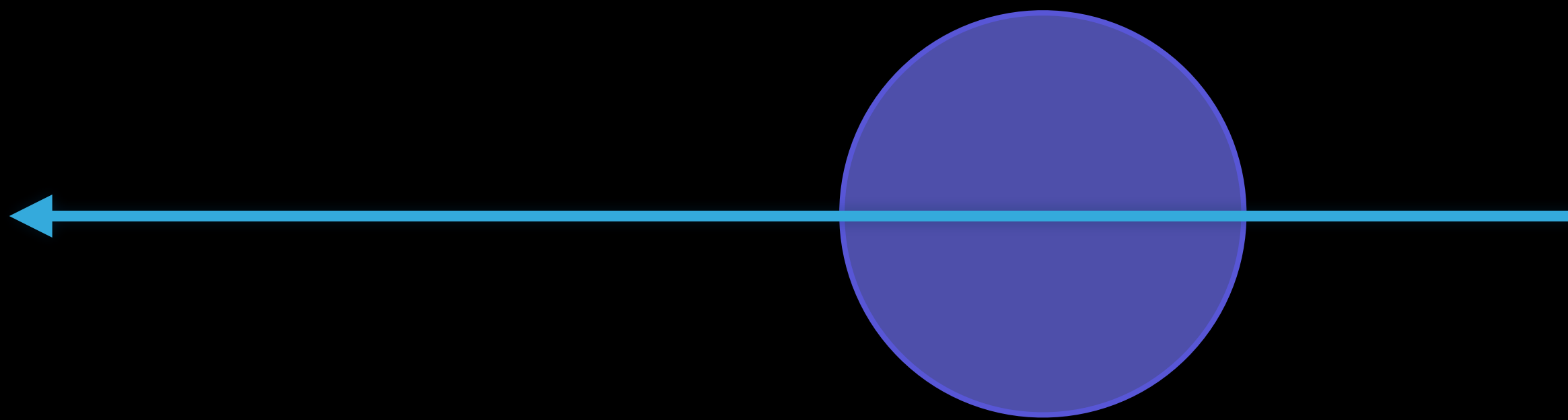
# From Animation to Animation

Andy Matuschak

iOS Apps and Frameworks Engineer

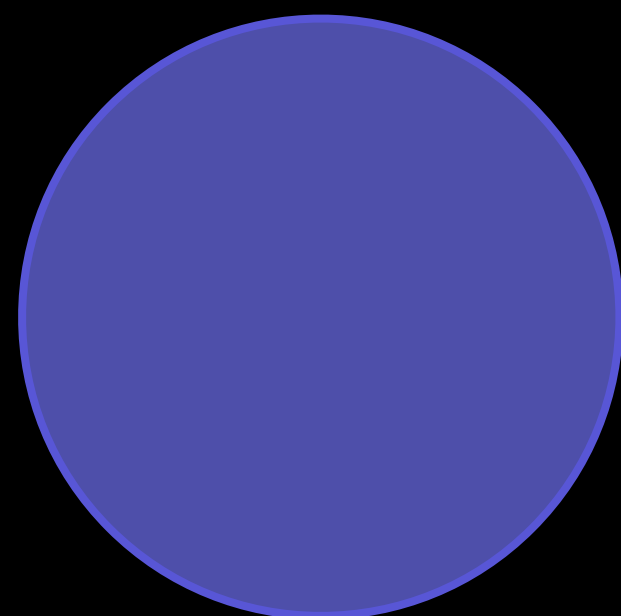




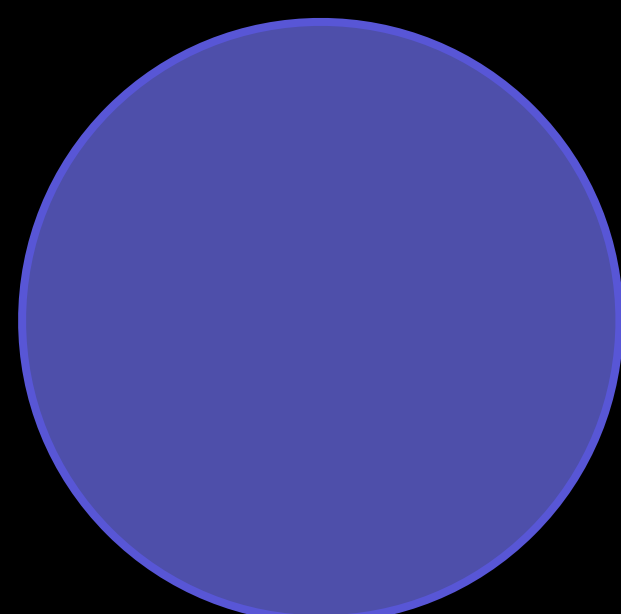




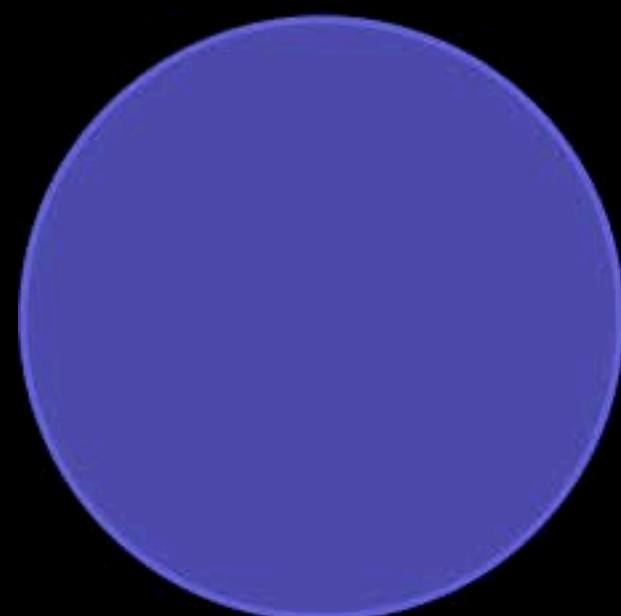
Absolute  
(iOS 7 default)



BeginFrom  
CurrentState



Absolute  
(iOS 7 default)



BeginFrom  
CurrentState



Absolute  
(iOS 7 default)



BeginFrom  
CurrentState





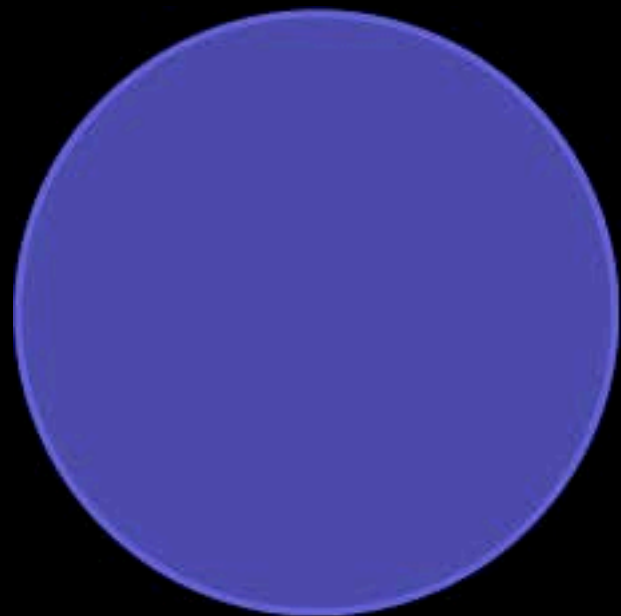
Absolute  
(iOS 7 default)



BeginFrom  
CurrentState



Additive



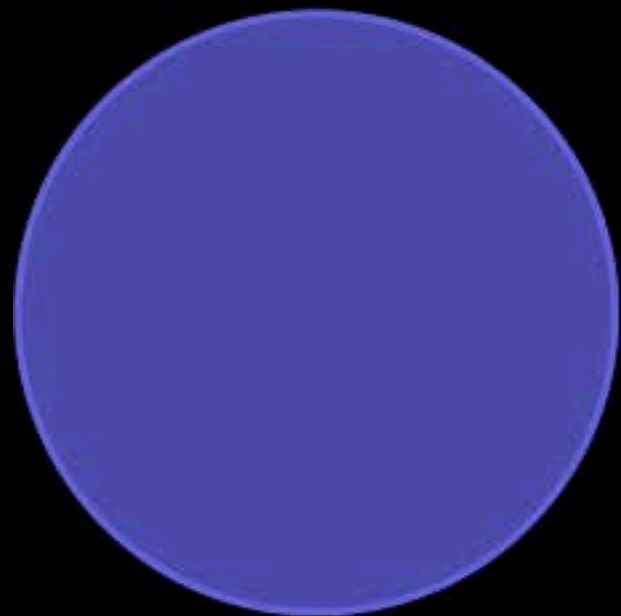
Absolute  
(iOS 7 default)



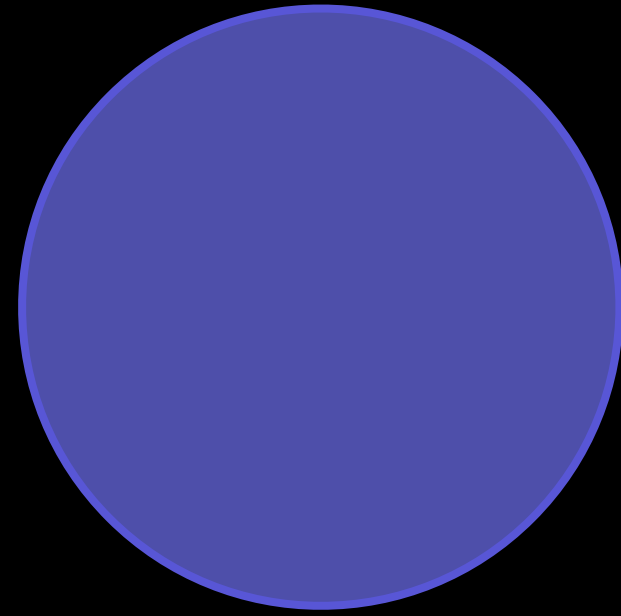
BeginFrom  
CurrentState



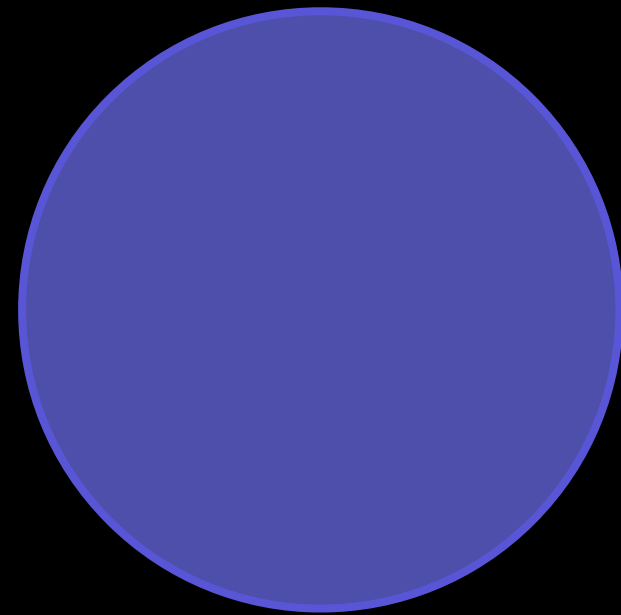
Additive



Absolute  
(iOS 7 default)



BeginFrom  
CurrentState



Additive  
(iOS 8 default)

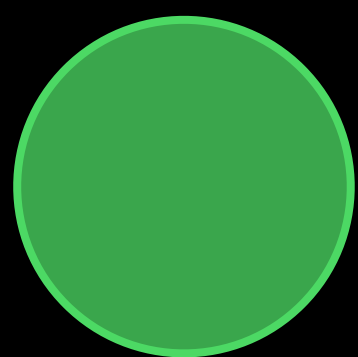


```
UIView.animateWithDuration(1) {  
    circle.center.x = 300  
}
```

# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0
Model	(0, 0)
Animation	n/a
Presentation	(0, 0)

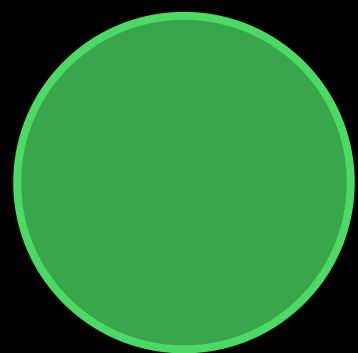


# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0
Model	(0, 0)
Animation	n/a
Presentation	(0, 0)

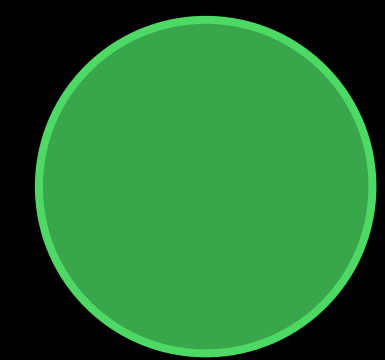
```
UIView.animateWithDuration(1) {  
    circle.center.x = 500  
}
```



# How Cocoa Touch Manages Animations

## Absolute animations

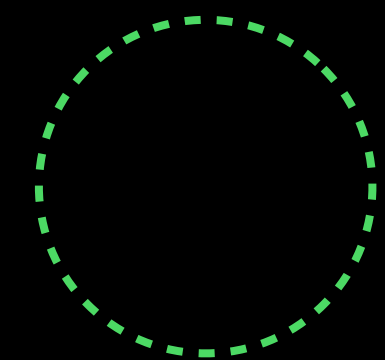
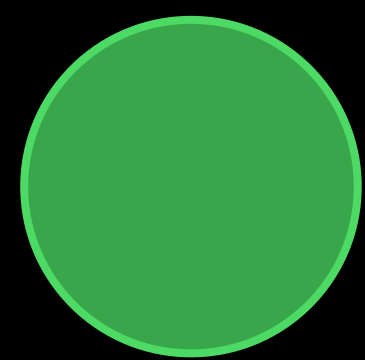
Time	1000.0	1000.1
Model	(0, 0)	(500, 0)
Animation	n/a	
Presentation	(0, 0)	



# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1
Model	(0, 0)	(500, 0)
Animation	n/a	
Presentation	(0, 0)	(0, 0)



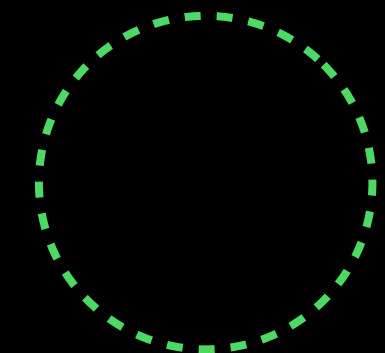
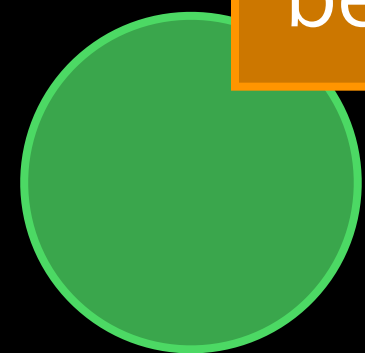


# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1
Model	(0, 0)	(500, 0)
Animation	n/a	
Presentation	(0, 0)	(0, 0)

CAAnimation	
fromValue	(0, 0)
duration	1.0
beginTime	1000.1

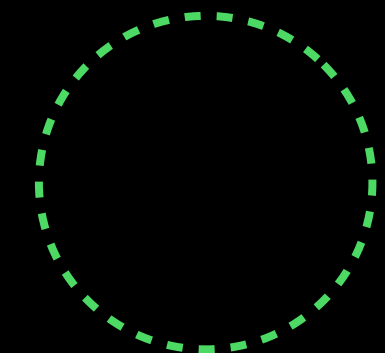
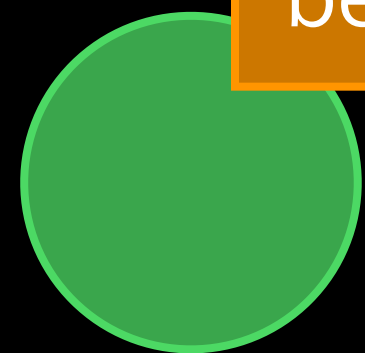


# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1
Model	(0, 0)	(500, 0)
Animation	n/a	(0, 0)
Presentation	(0, 0)	(0, 0)

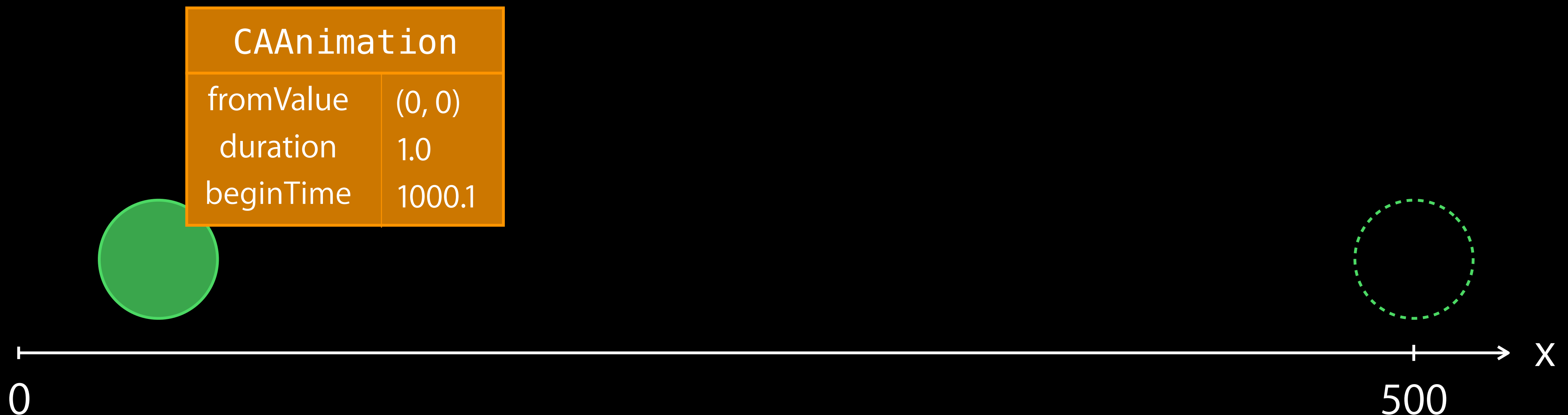
CAAnimation	
fromValue	(0, 0)
duration	1.0
beginTime	1000.1



# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1	1000.2
Model	(0, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)

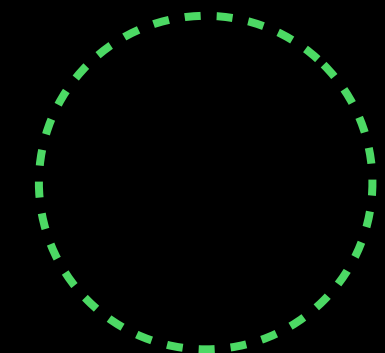
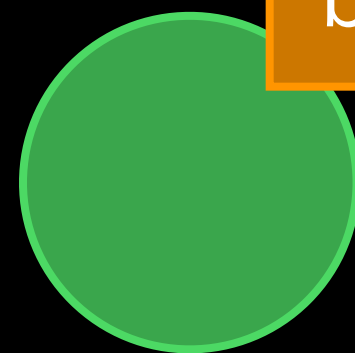


# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1	1000.2	1000.3
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)

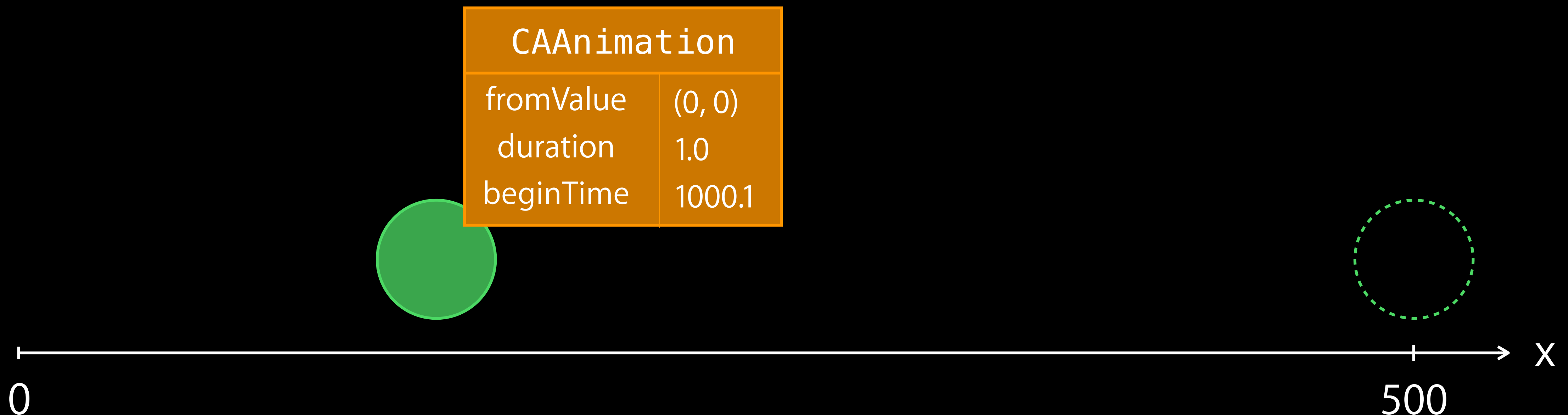
CAAnimation	
fromValue	(0, 0)
duration	1.0
beginTime	1000.1



# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)



# How Cocoa Touch Manages Animations

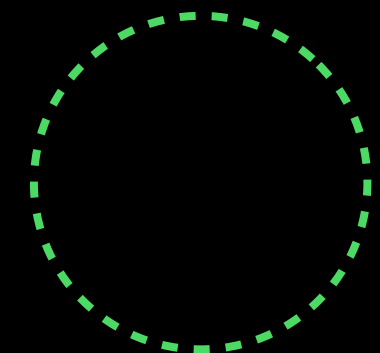
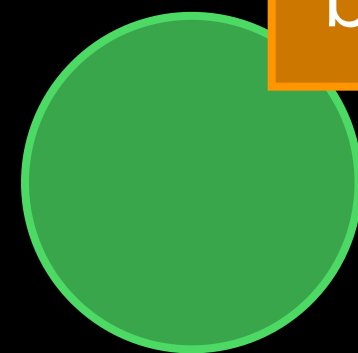
## Absolute animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)

```
UIView.animateWithDuration(1) {  
    circle.center.x = 0  
}
```

```
CAA }
```

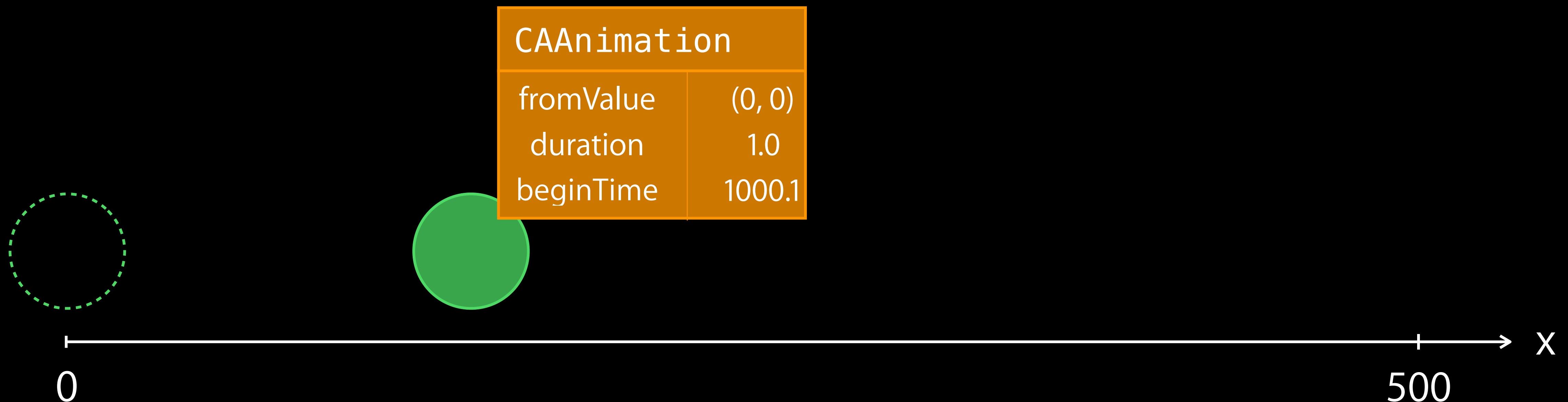
```
fromValue (0, 0)  
duration 1.0  
beginTime 1000.1
```



# How Cocoa Touch Manages Animations

## Absolute animations

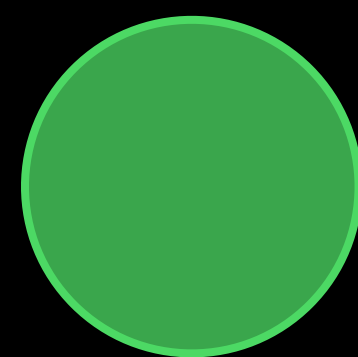
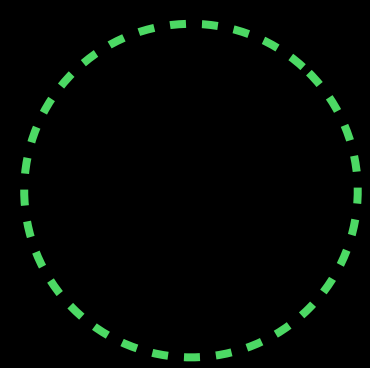
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	



# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	

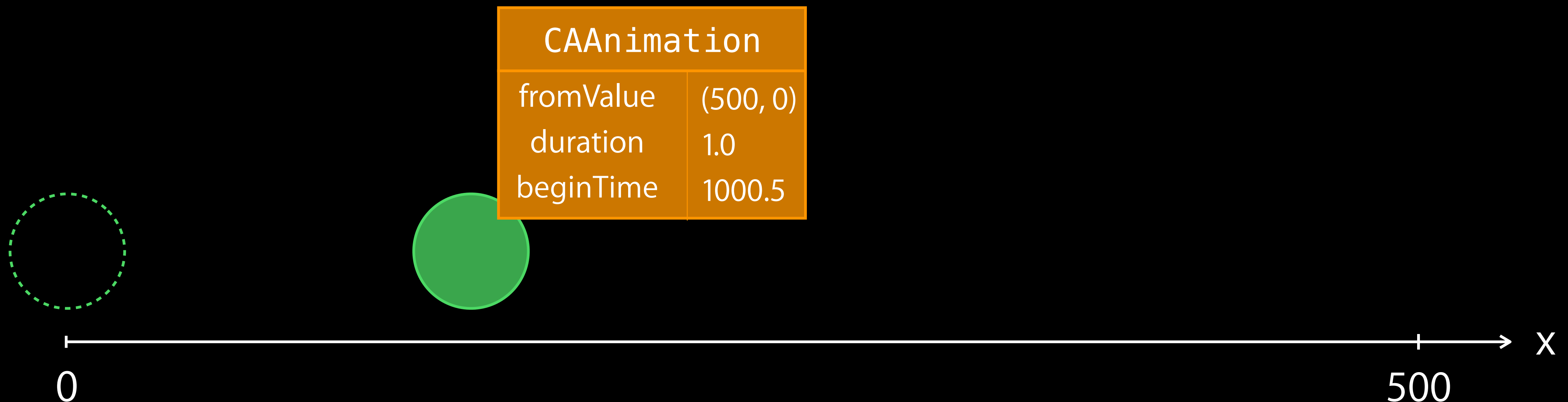




# How Cocoa Touch Manages Animations

## Absolute animations

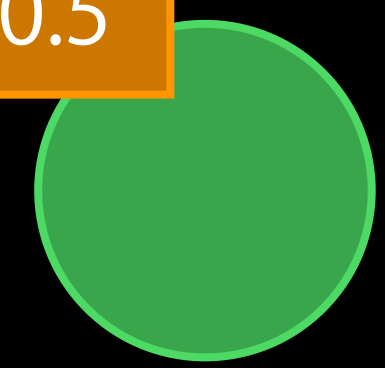
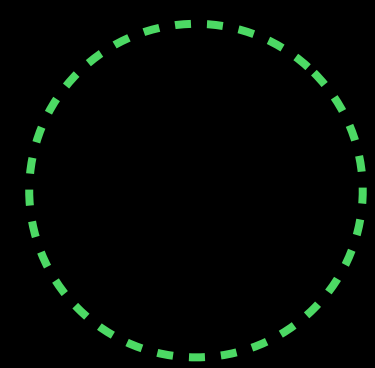
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	



# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)



CAAnimation	
fromValue	(500, 0)
duration	1.0
beginTime	1000.5



# How Cocoa Touch Manages Animations

## Absolute animations

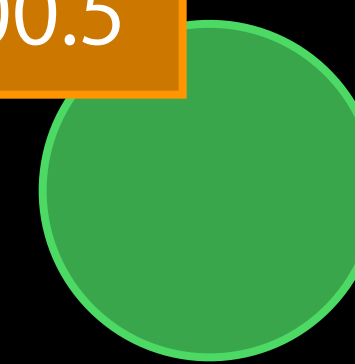
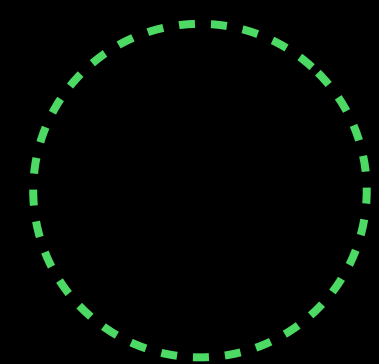
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)	(450, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)	(450, 0)



# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6	1000.7
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)	(450, 0)	(400, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)	(450, 0)	(400, 0)



CAAnimation	
fromValue	(500, 0)
duration	1.0
beginTime	1000.5



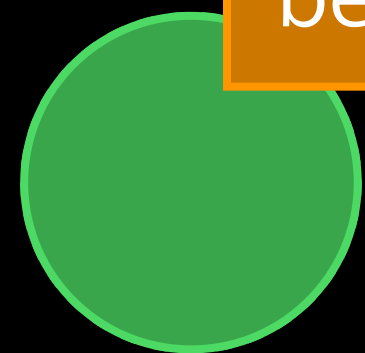
# How Cocoa Touch Manages Animations

## Absolute animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6	1000.7	1001.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)	(450, 0)	(400, 0)	(0, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(500, 0)	(450, 0)	(400, 0)	(0, 0)

### CAAnimation

fromValue	(500, 0)
duration	1.0
beginTime	1000.5



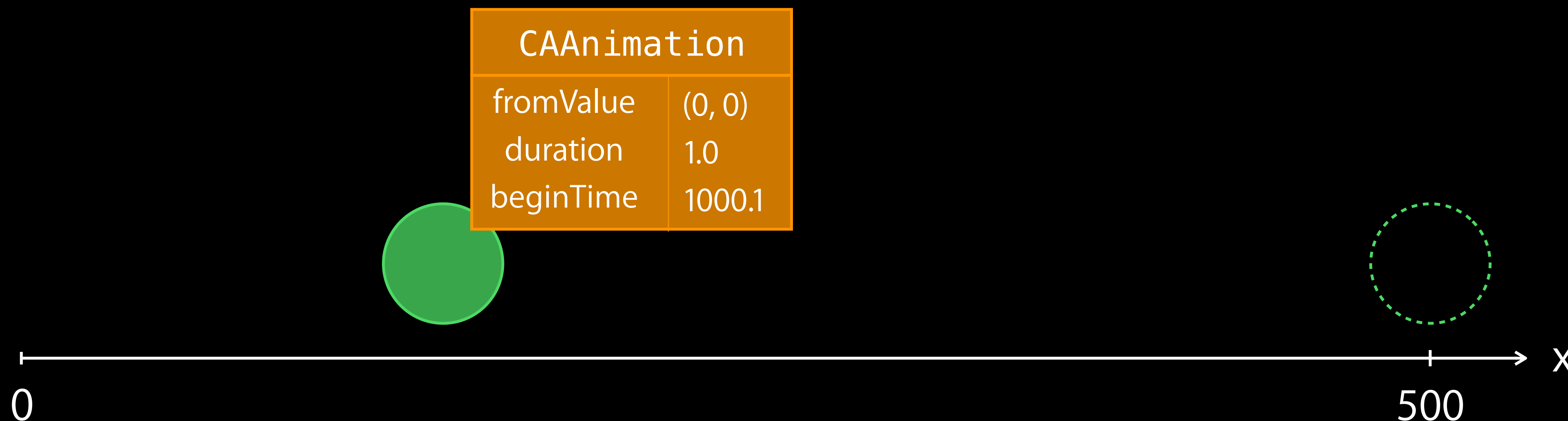
# How Cocoa Touch Manages Animations

Absolute animations and `BeginFromCurrentState`

# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)



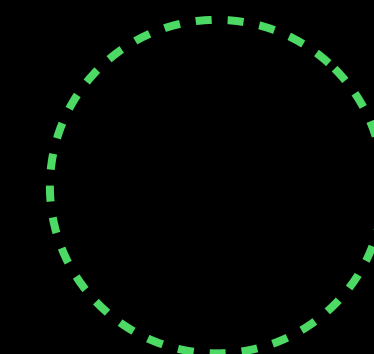
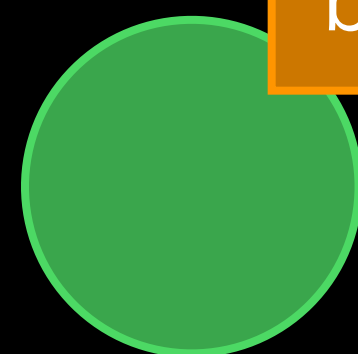
# How Cocoa Touch Manages Animations

## Absolute animations and BeginFromCurrentState

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)

```
UIView.animateWithDuration(1,  
    delay: 0,  
    options: .BeginFromCurrentState,  
    animations: { circle.center.x = 0 }  
    completion: nil)
```

beginTime 1000.1

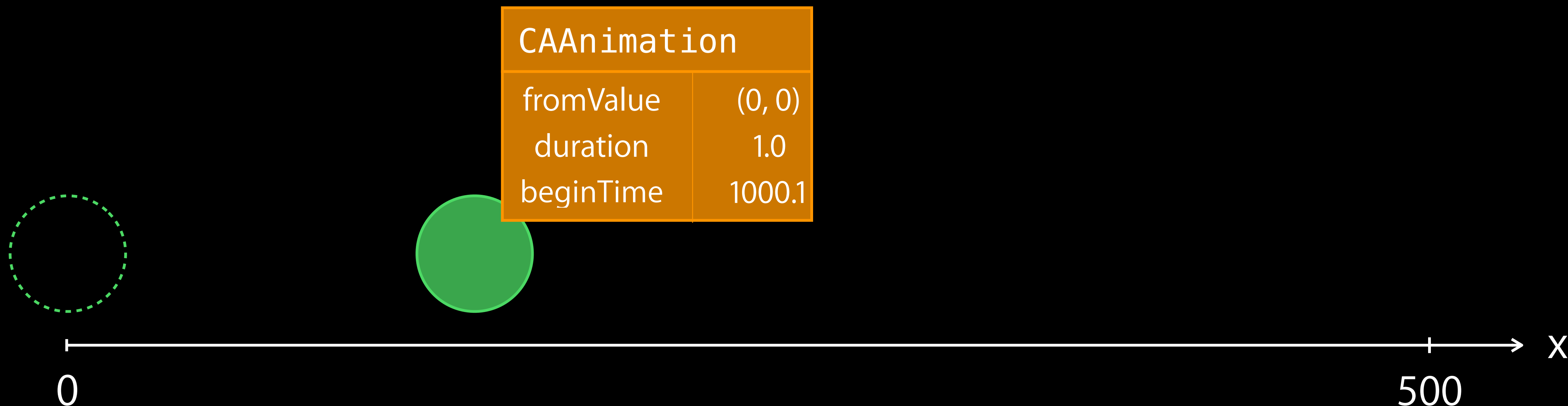




# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

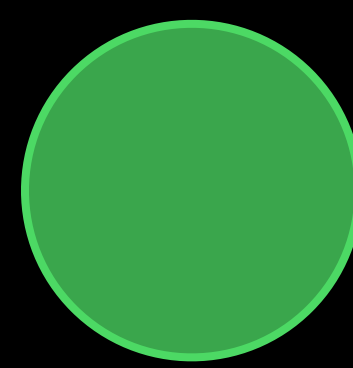
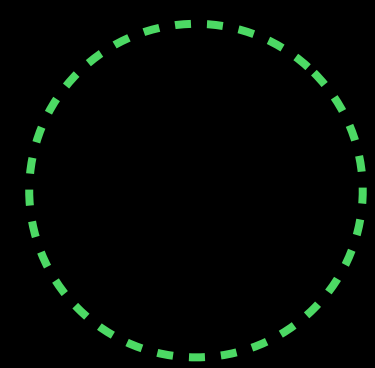
Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)



# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

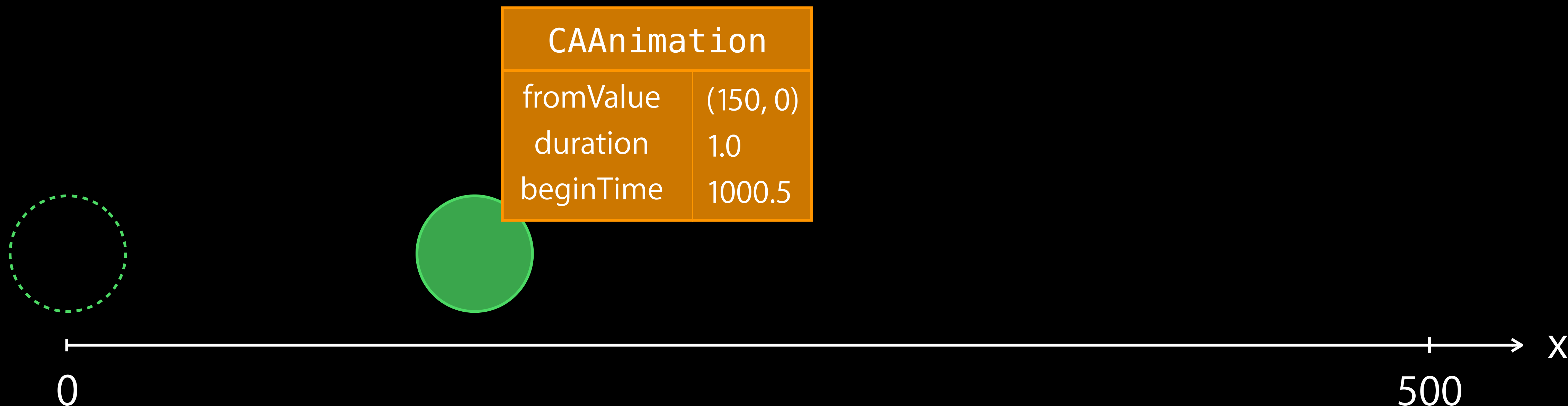
Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)



# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

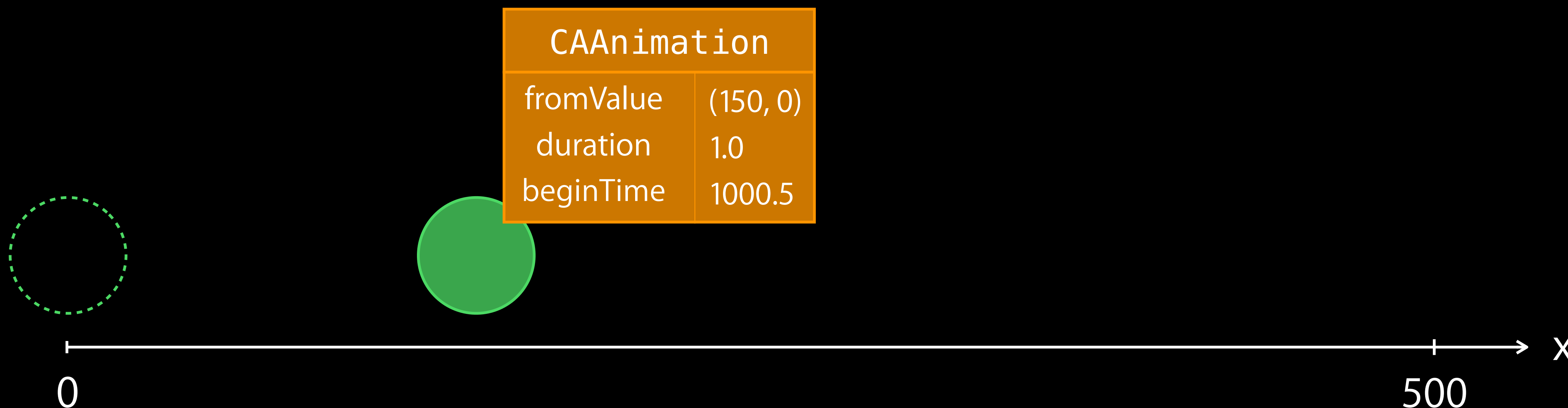
Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)



# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

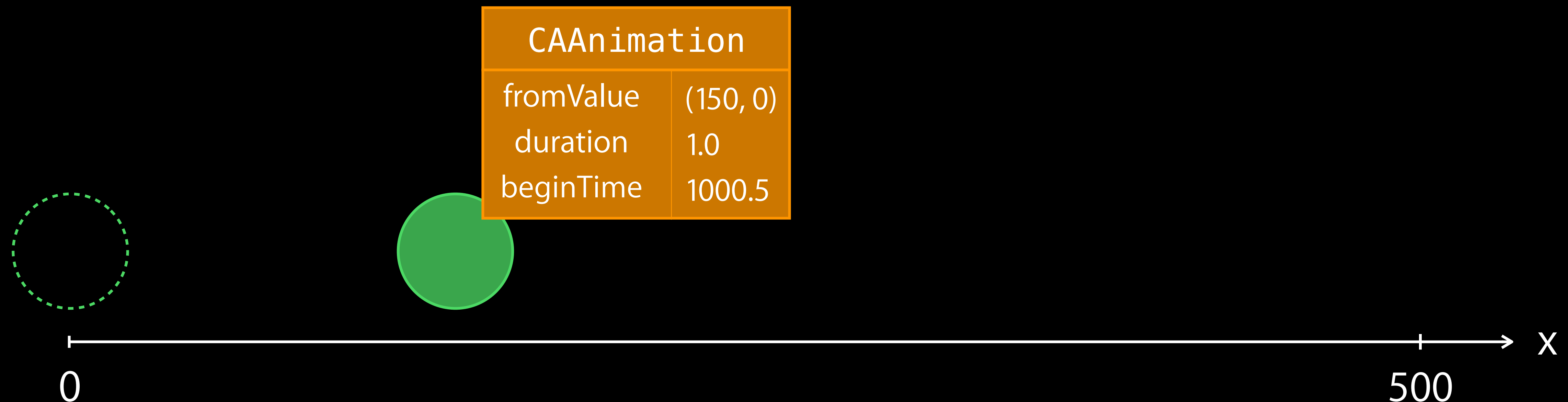
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(150, 0)



# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(150, 0)	(135, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(150, 0)	(135, 0)



# How Cocoa Touch Manages Animations

Absolute animations and `BeginFromCurrentState`



# How Cocoa Touch Manages Animations

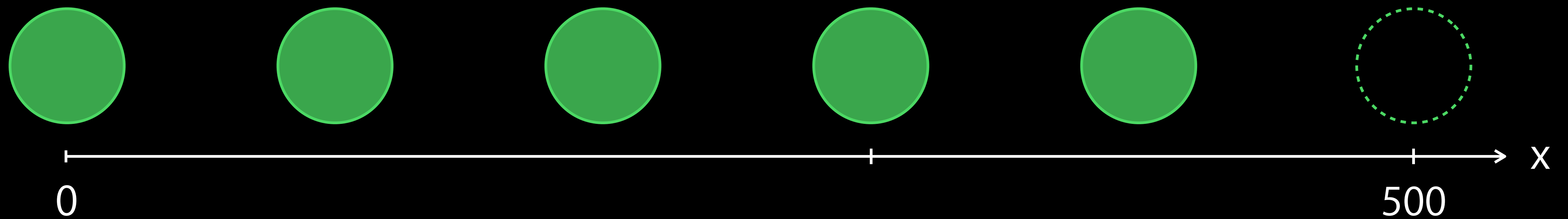
Absolute animations and `BeginFromCurrentState`



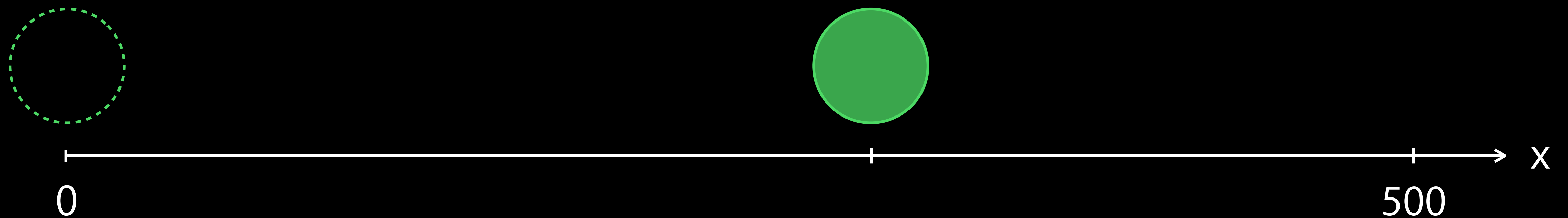
# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

Initial animation



Reverse animation

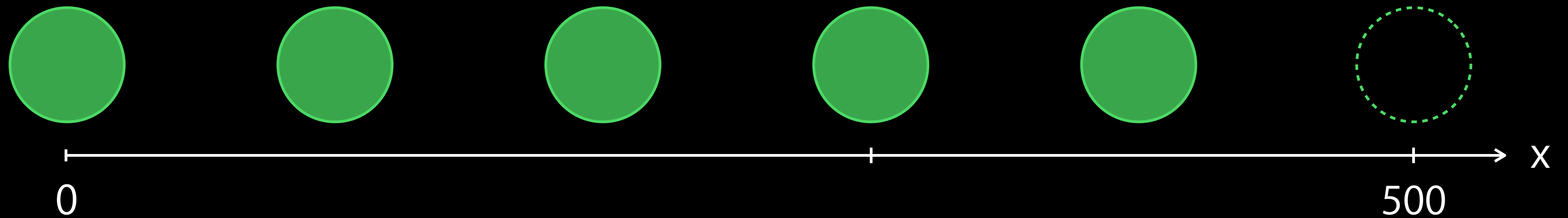




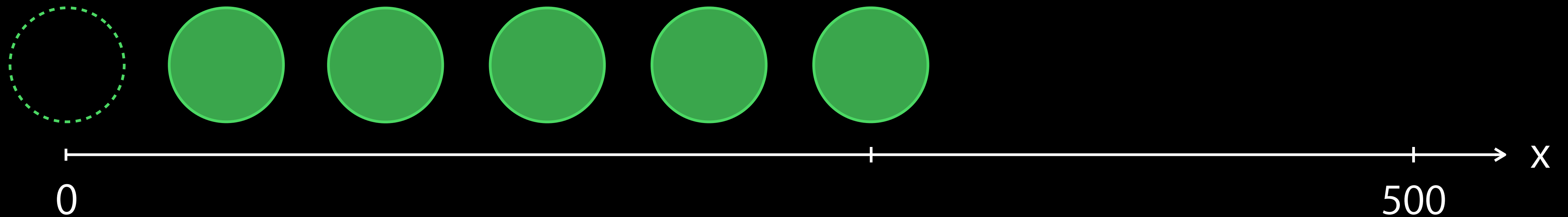
# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

Initial animation



Reverse animation



# How Cocoa Touch Manages Animations

Absolute animations and `BeginFromCurrentState`



# How Cocoa Touch Manages Animations

Absolute animations and `BeginFromCurrentState`

Initial animation



# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

Initial animation



Reverse animation



# How Cocoa Touch Manages Animations

## Absolute animations and `BeginFromCurrentState`

Initial animation

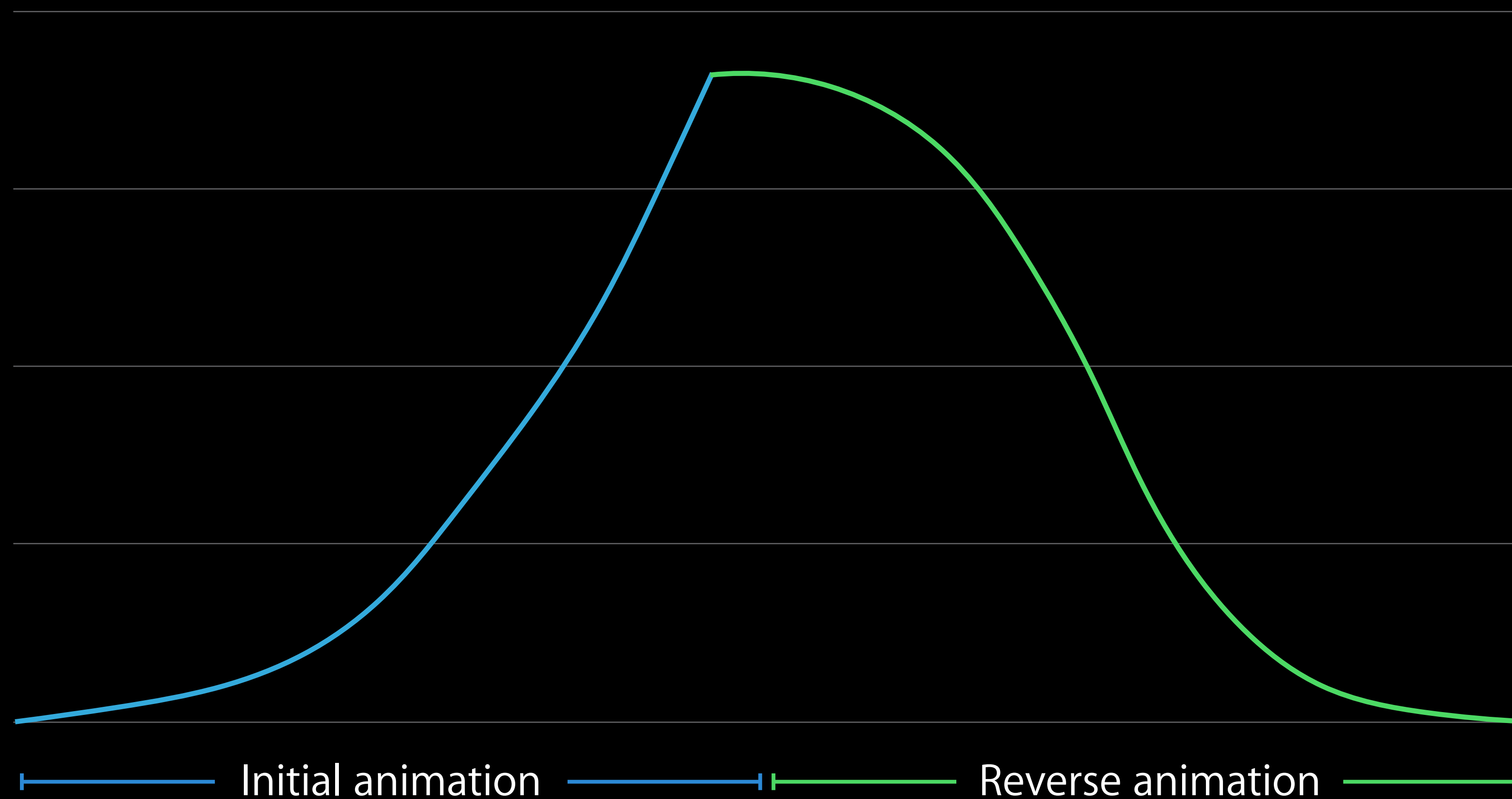


Reverse animation



# How Cocoa Touch Manages Animations

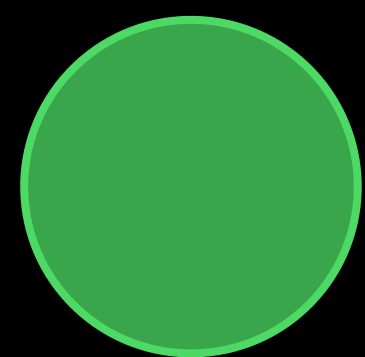
Absolute animations and `BeginFromCurrentState`



# Additive Animations

# Introducing Additive Animations

Time	1000.0
Model	(0, 0)
Animation	n/a
Presentation	(0, 0)

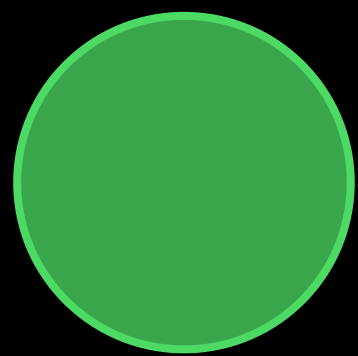




# Introducing Additive Animations

Time	1000.0
Model	(0, 0)
Animation	n/a
Presentation	(0, 0)

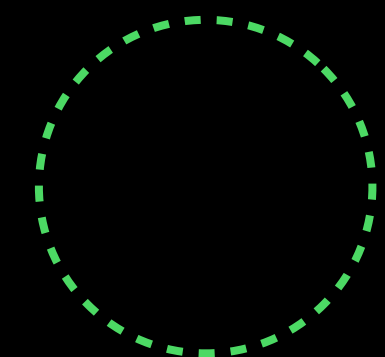
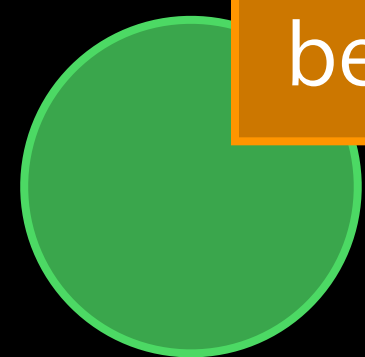
```
UIView.animateWithDuration(1) {  
    circle.center.x = 500  
}
```



# Introducing Additive Animations

Time	1000.0	1000.1
Model	(0, 0)	(500, 0)
Animation	n/a	
Presentation	(0, 0)	

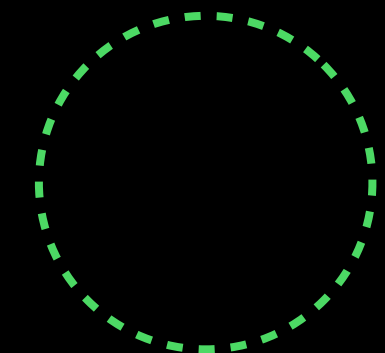
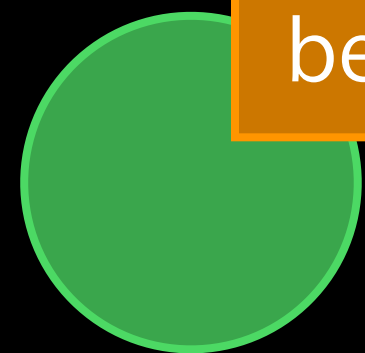
CAAnimation	
additive	YES
fromValue	(-500, 0)
toValue	(0, 0)
duration	1.0
beginTime	1000.1



# Introducing Additive Animations

Time	1000.0	1000.1
Model	(0, 0)	(500, 0)
Animation	n/a	(-500, 0)
Presentation	(0, 0)	(0, 0)

CAAnimation	
additive	YES
fromValue	(-500, 0)
toValue	(0, 0)
duration	1.0
beginTime	1000.1



# Introducing Additive Animations

Time	1000.0	1000.1	1000.2
Model	(0, 0)	(500, 0)	(500, 0)
Animation	n/a	(-500, 0)	(-450, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)

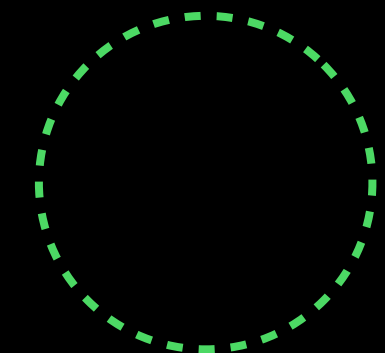
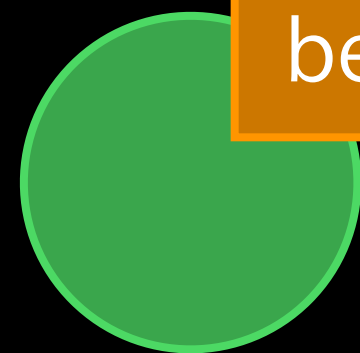
CAAnimation	
additive	YES
fromValue	(-500, 0)
toValue	(0, 0)
duration	1.0
beginTime	1000.1



# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)

CAAnimation	
additive	YES
fromValue	(-500, 0)
toValue	(0, 0)
duration	1.0
beginTime	1000.1



# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)

## CAAnimation

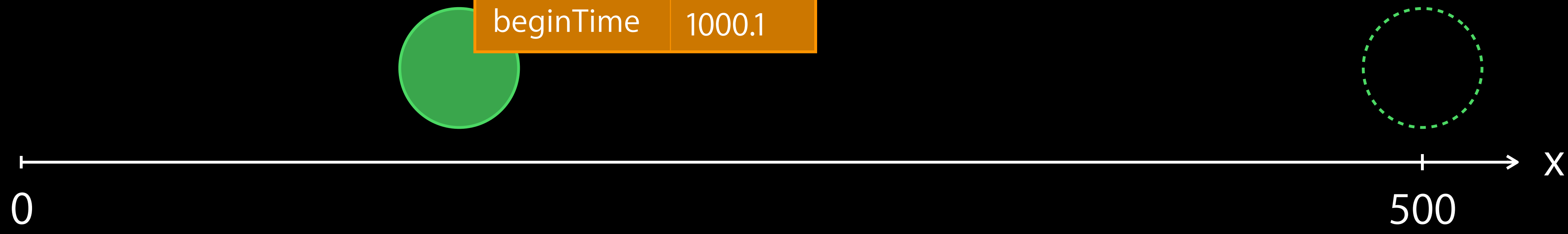
additive	YES
fromValue	(-500, 0)
toValue	(0, 0)
duration	1.0
beginTime	1000.1



# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)

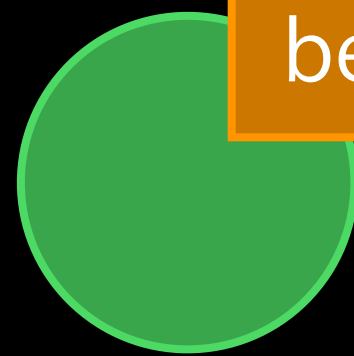
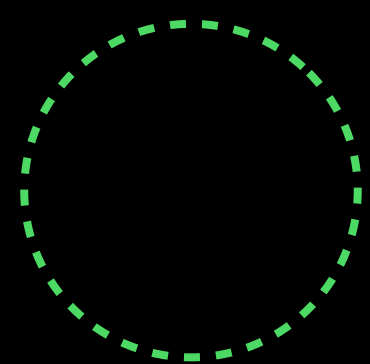
```
CAAnimation *anim = [CAAnimation animation];  
[anim addFromValue:(500, 0) toValue:(0, 0) duration:1.0 beginTime:1000.1];  
[UIView animateWithDuration(0.4) {  
    circle.center.x = 0;  
}];
```



# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	

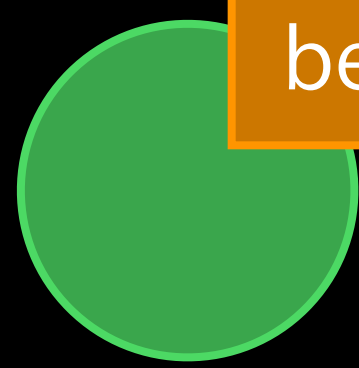
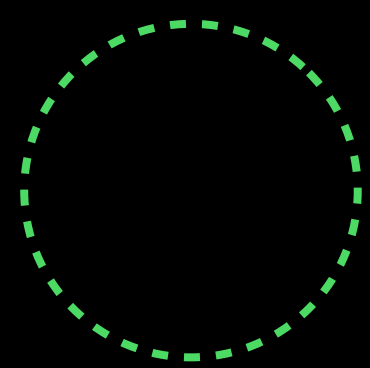
CAAnimation	
additive	YES
fromValue	(-500, 0)
toValue	(0, 0)
duration	1.0
beginTime	1000.1





# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	

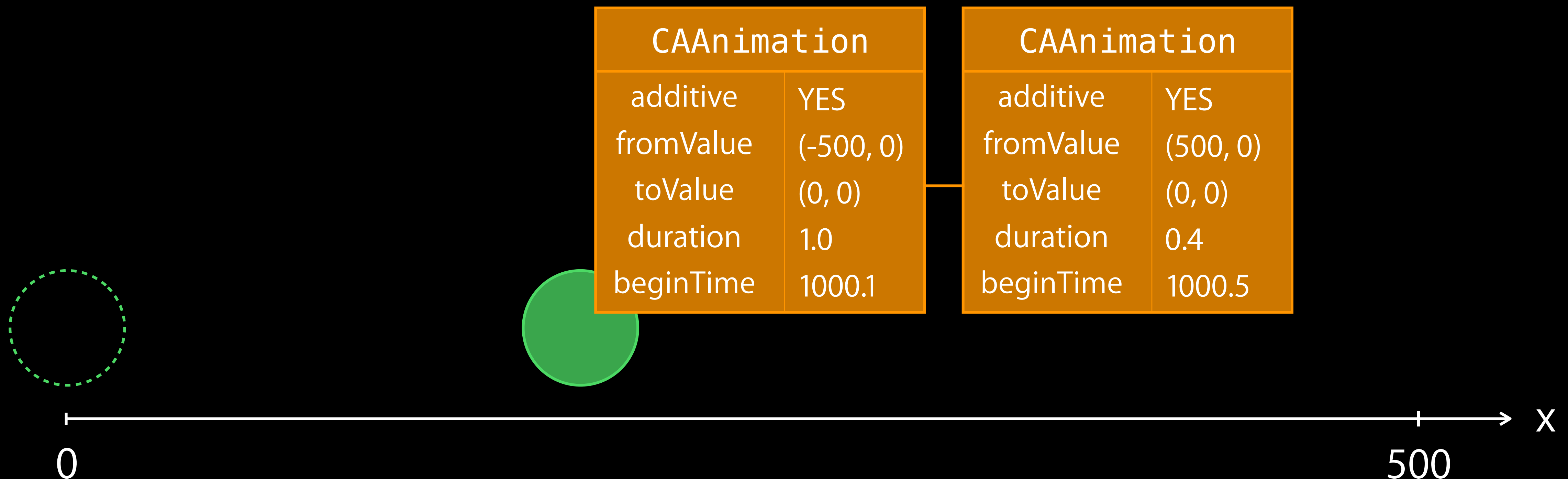


CAAnimation		CAAnimation	
additive	YES	additive	YES
fromValue	(-500, 0)	fromValue	(500, 0)
toValue	(0, 0)	toValue	(0, 0)
duration	1.0	duration	0.4
beginTime	1000.1	beginTime	1000.5



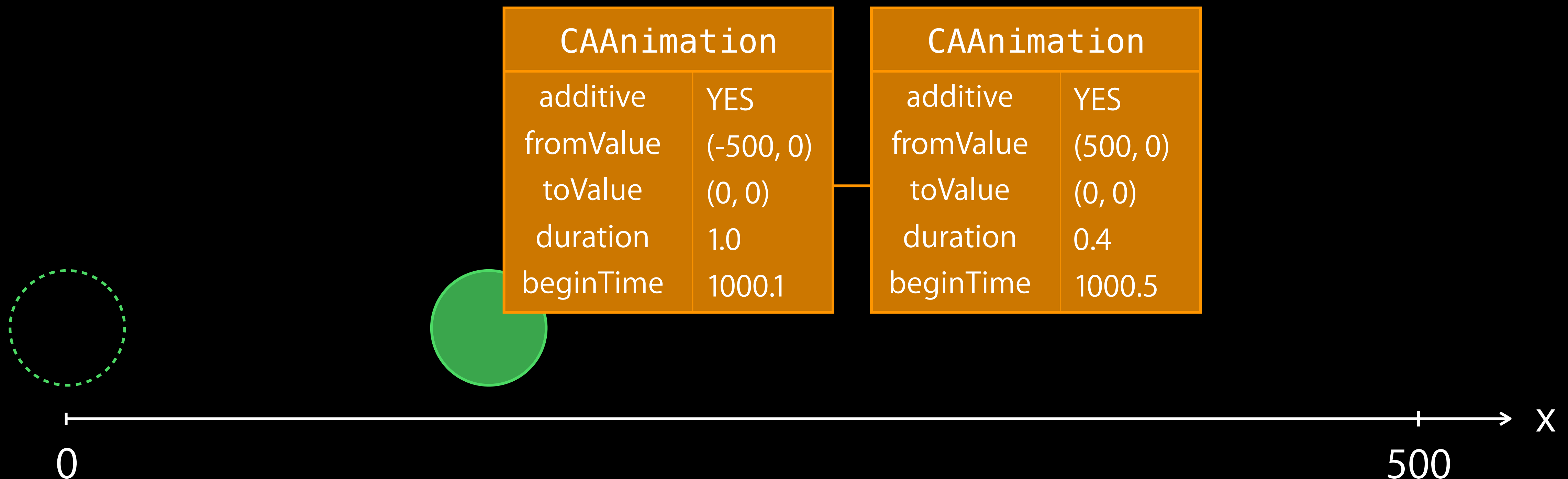
# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation 1	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	(-300, 0)
Animation 2	n/a	n/a	n/a	n/a	n/a	(500, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(200, 0)



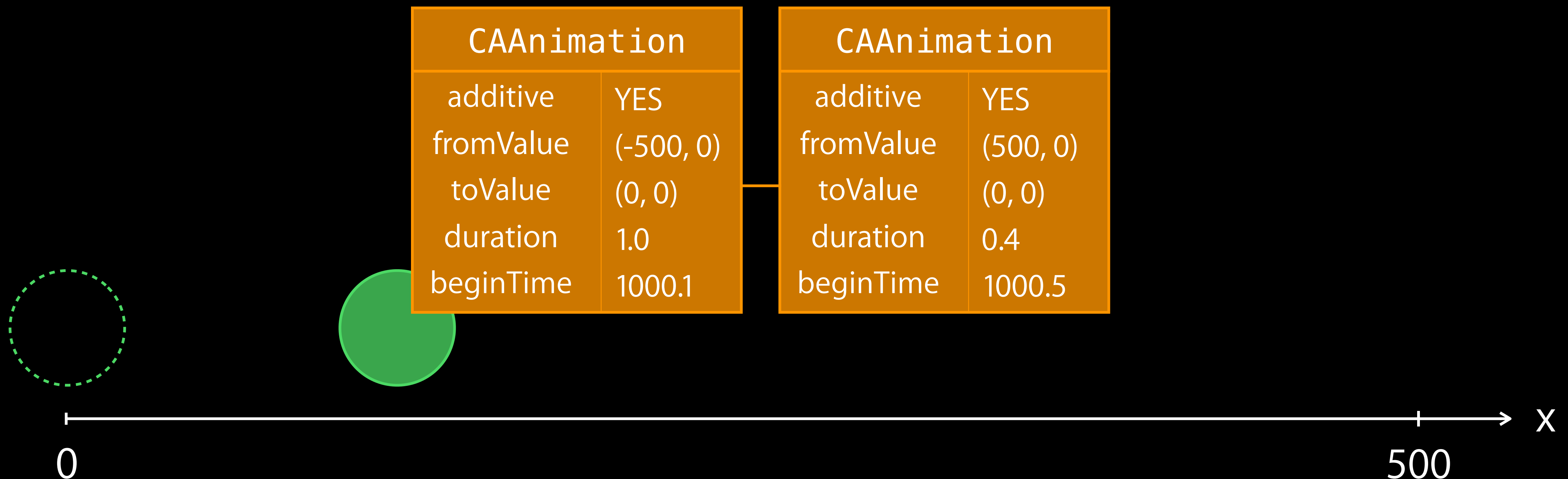
# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)
Animation 1	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	(-300, 0)	(-250, 0)
Animation 2	n/a	n/a	n/a	n/a	n/a	(500, 0)	(400, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(200, 0)	(150, 0)



# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6	1000.7
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)	(0, 0)
Animation 1	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	(-300, 0)	(-250, 0)	(-200, 0)
Animation 2	n/a	n/a	n/a	n/a	n/a	(500, 0)	(400, 0)	(300, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(200, 0)	(150, 0)	(100, 0)

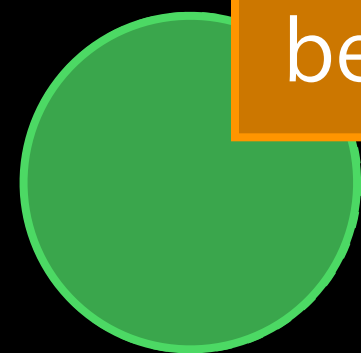


# Introducing Additive Animations

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6	1000.7	1001.1
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Animation 1	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	(-300, 0)	(-250, 0)	(-200, 0)	(0, 0)
Animation 2	n/a	n/a	n/a	n/a	n/a	(500, 0)	(400, 0)	(300, 0)	(0, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(200, 0)	(150, 0)	(100, 0)	(0, 0)

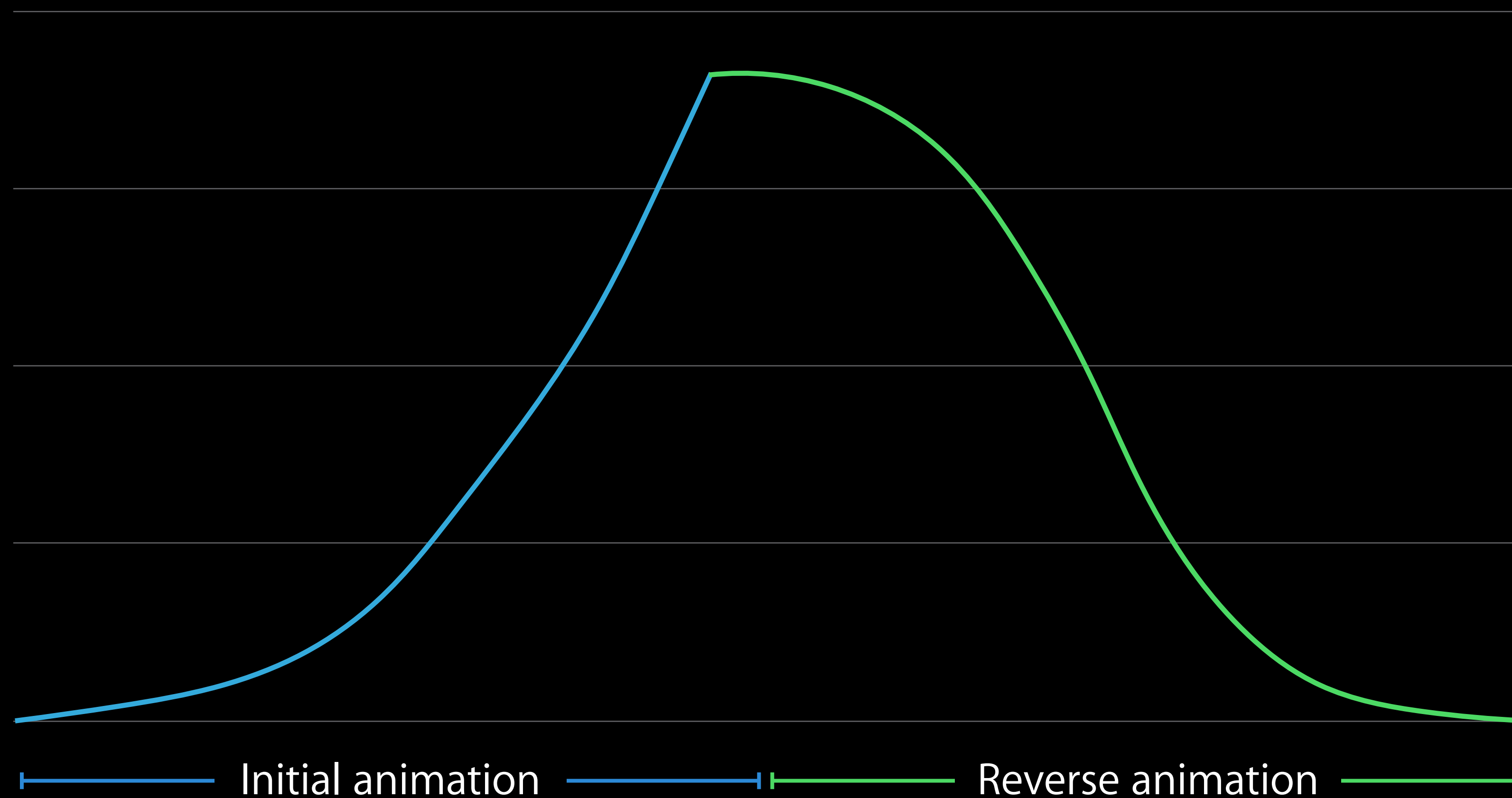
CAAnimation	
additive	YES
fromValue	(-500, 0)
toValue	(0, 0)
duration	1.0
beginTime	1000.1

CAAnimation	
additive	YES
fromValue	(500, 0)
toValue	(0, 0)
duration	0.4
beginTime	1000.5



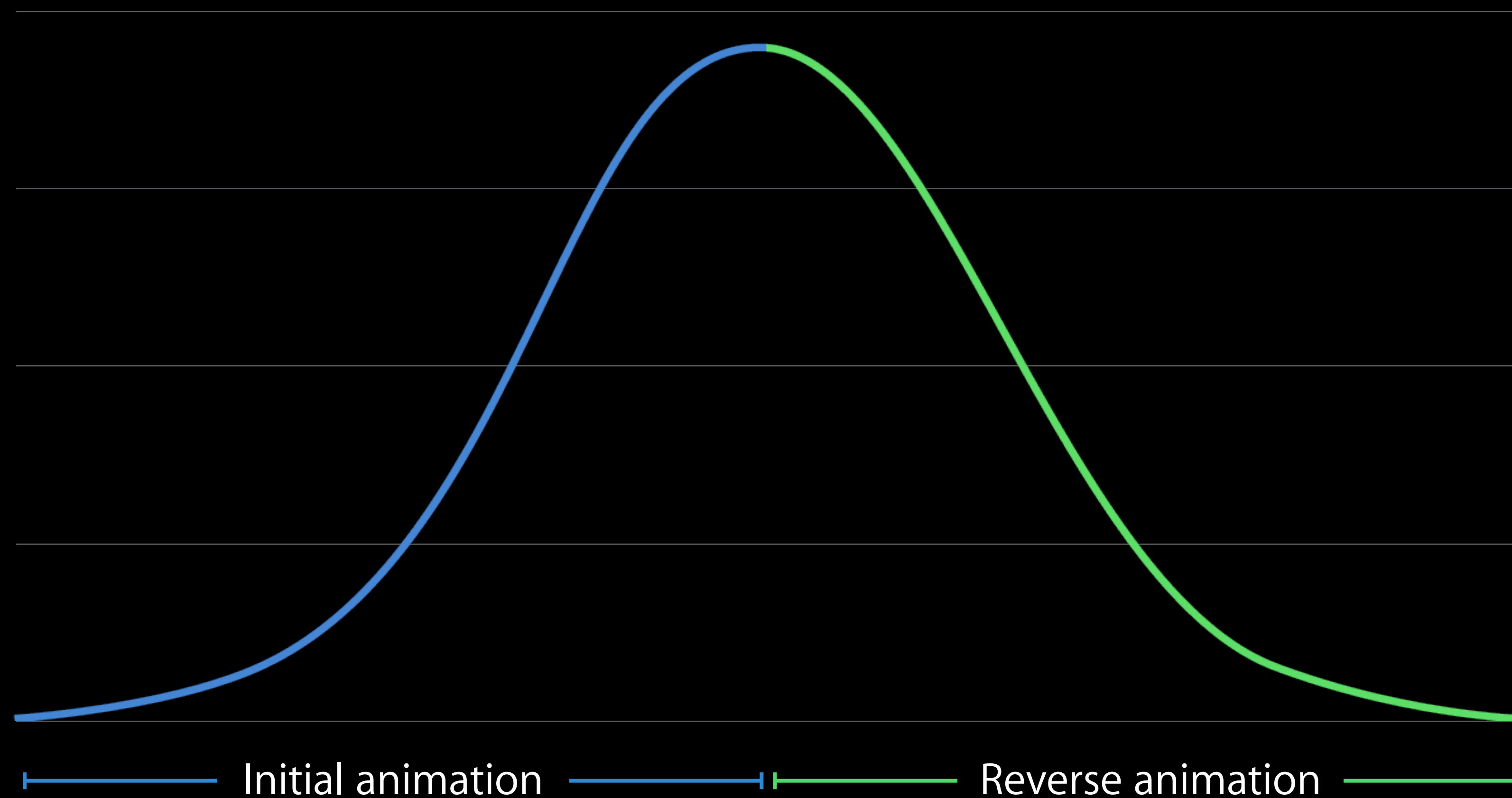
# Introducing Additive Animations

Absolute animations and `BeginFromCurrentState`



# Introducing Additive Animations

Smooth transitions with easing



# Introducing Additive Animations

Supported keys



# Introducing Additive Animations

## Supported keys

center

frame

bounds

transform

# Introducing Additive Animations

## Supported keys

center

frame

bounds

transform

layer.transform

# Introducing Additive Animations

Affine transforms



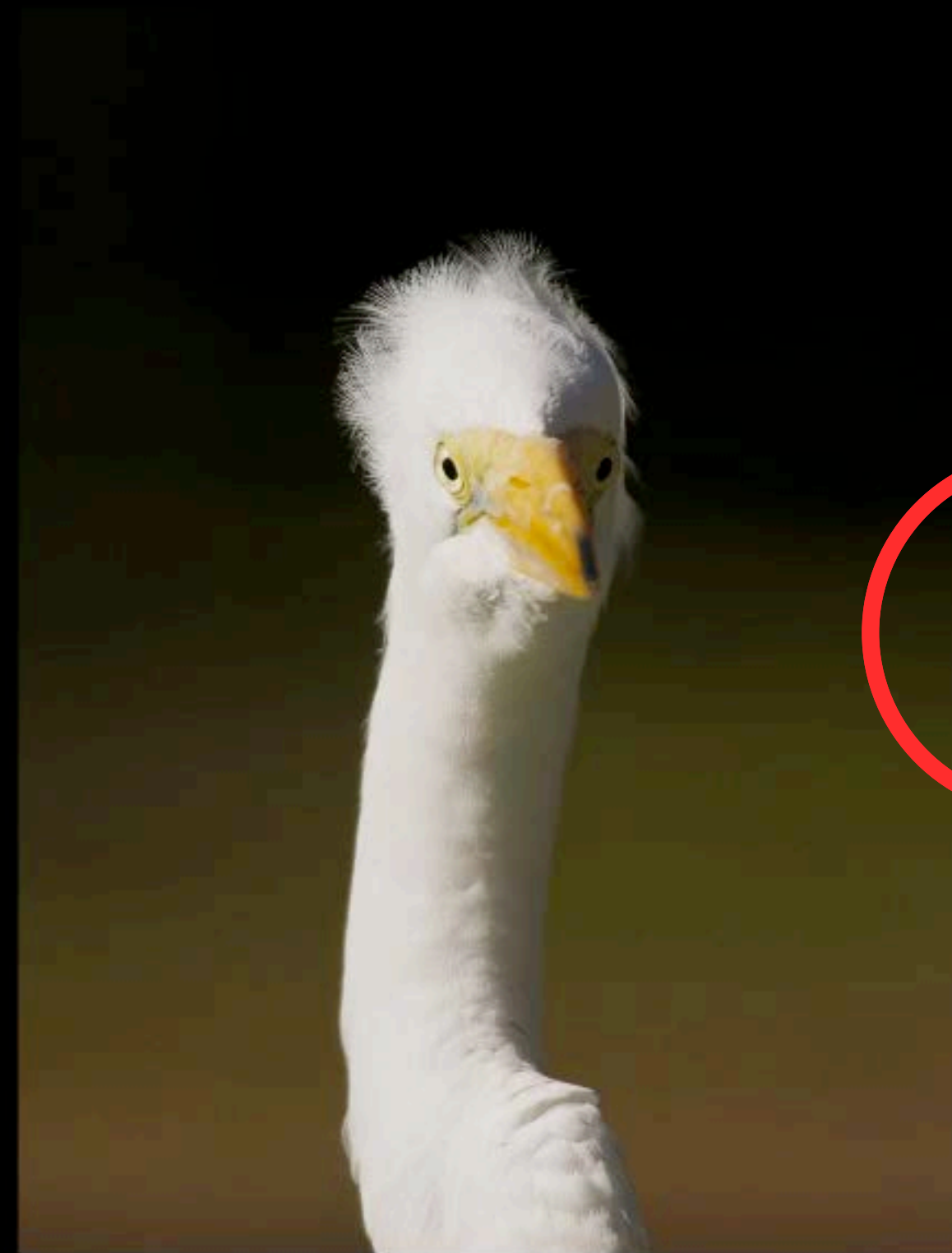
# Introducing Additive Animations

Affine transforms



# Introducing Additive Animations

Affine transforms



# Introducing Additive Animations

Compatibility requirements

# Introducing Additive Animations

Compatibility requirements

No keyframe animations

# Introducing Additive Animations

Compatibility requirements

No keyframe animations

No pre-existing repeating animations



# Introducing Additive Animations

## Compatibility requirements

No keyframe animations

No pre-existing repeating animations

No pre-existing absolute animations

# Introducing Additive Animations

Use `BeginFromCurrentState` defensively

```
UIView.animateWithDuration(1) {  
    circle.center.x = finalValue  
    circle.alpha = 0  
    circle.tintColor = UIColor.redColor  
}
```

# Introducing Additive Animations

Use `BeginFromCurrentState` defensively

```
UIView.animateWithDuration(1,  
                           delay: 0,  
                           options: .BeginFromCurrentState,  
                           animations: {  
circle.center.x = finalValue  
circle.alpha = 0  
circle.tintColor = UIColor.redColor  
                           },  
                           completion: nil)
```

# Introducing Additive Animations

Canceling animations

# Introducing Additive Animations

## Canceling animations

```
UIView.animateWithDuration(0) {  
    circle.center.x = finalValue  
}
```



# Introducing Additive Animations

## Canceling animations

```
UIView.animateWithDuration(0) {  
    circle.center.x = finalValue  
}
```



```
circle.layer.removeAnimation(animation)
```



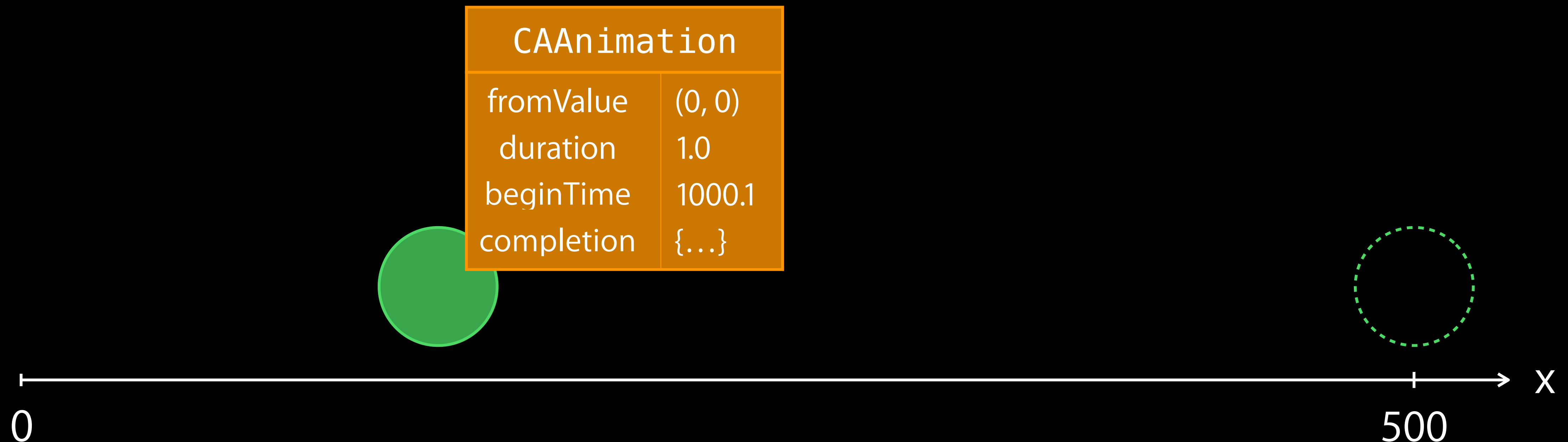
# Introducing Additive Animations

Completion handlers

# Introducing Additive Animations

## Completion handlers (then)

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)





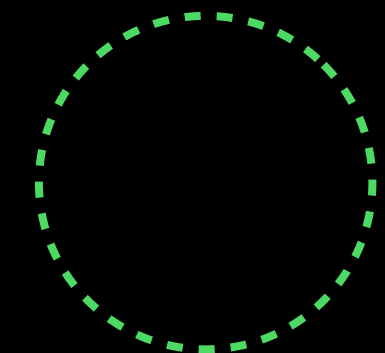
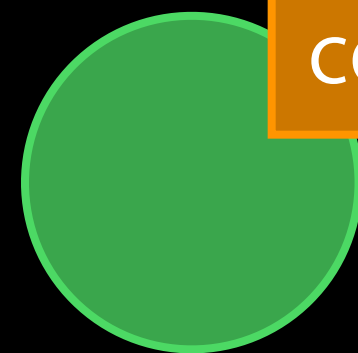
# Introducing Additive Animations

Completion handlers (then)

Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)

```
UIView.animateWithDuration(1) {  
    circle.center.x = 0  
}
```

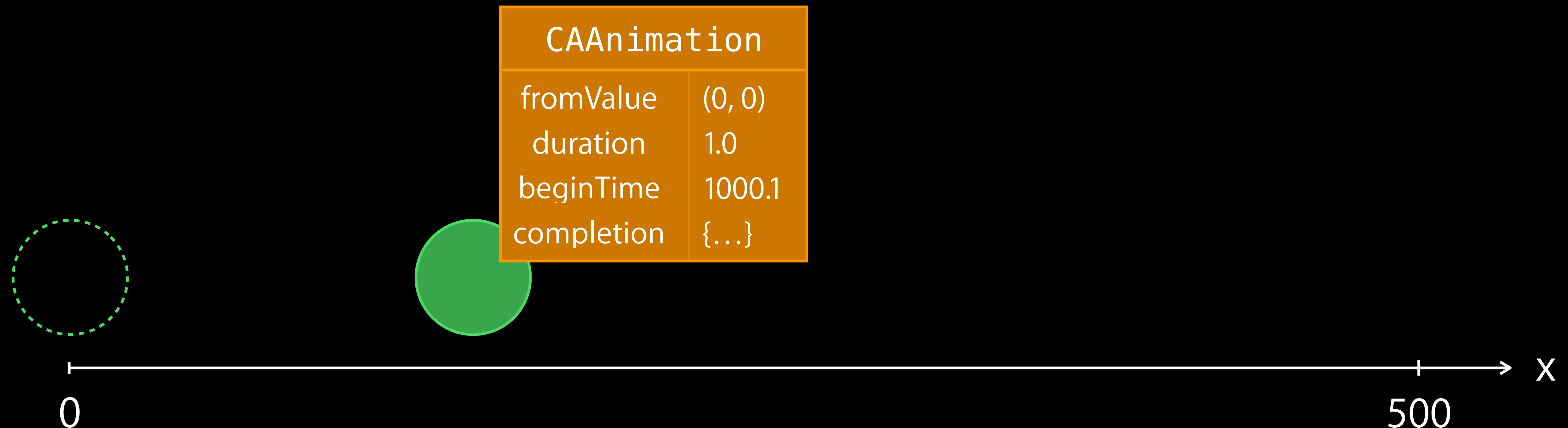
```
CAA  
fromValue (0, 0)  
duration 1.0  
beginTime 1000.1  
completion {...}
```



# Introducing Additive Animations

## Completion handlers (then)

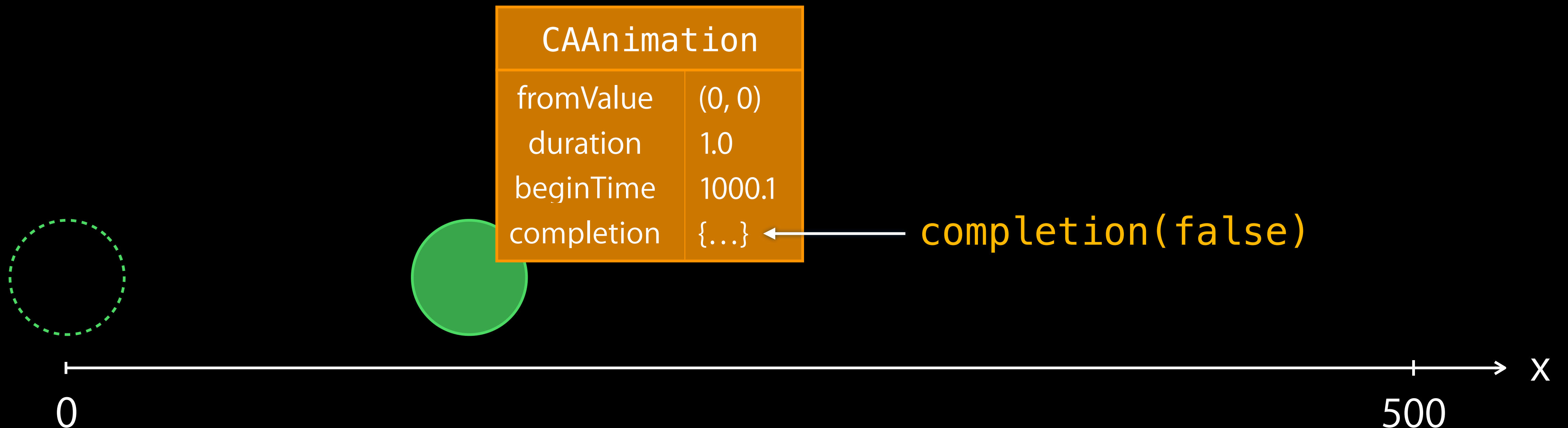
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	



# Introducing Additive Animations

## Completion handlers (then)

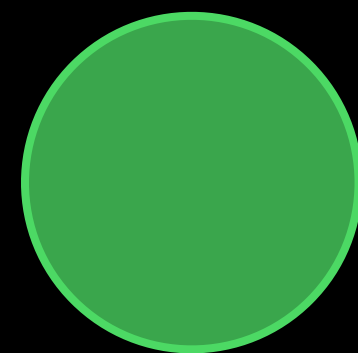
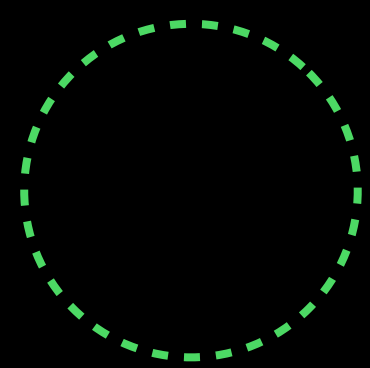
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	



# Introducing Additive Animations

Completion handlers (then)

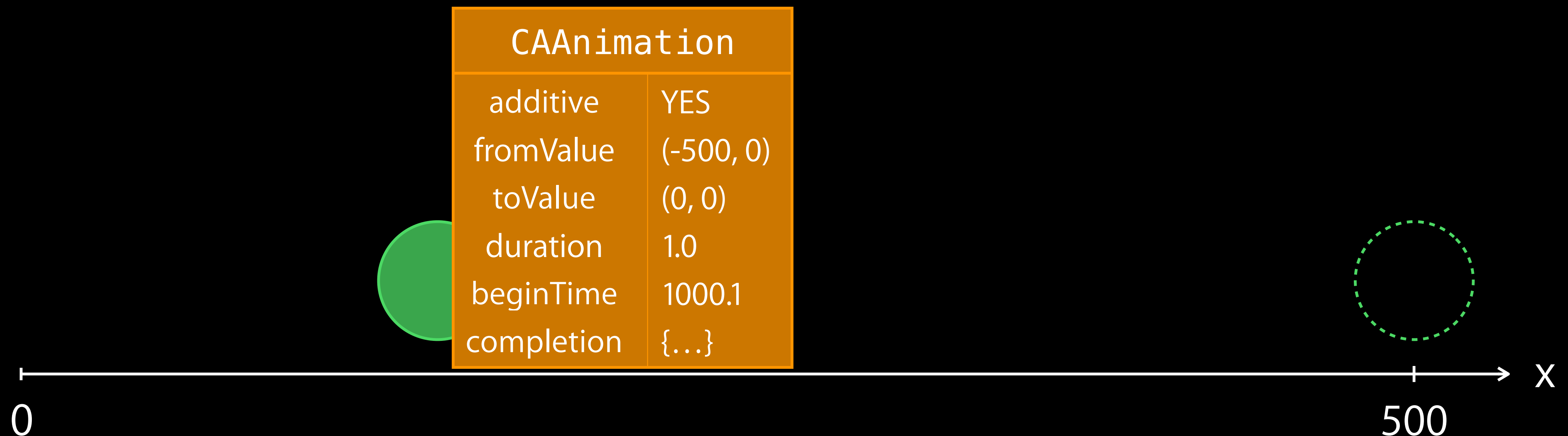
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(0, 0)	(50, 0)	(100, 0)	(150, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	



# Introducing Additive Animations

Completion handlers (now)

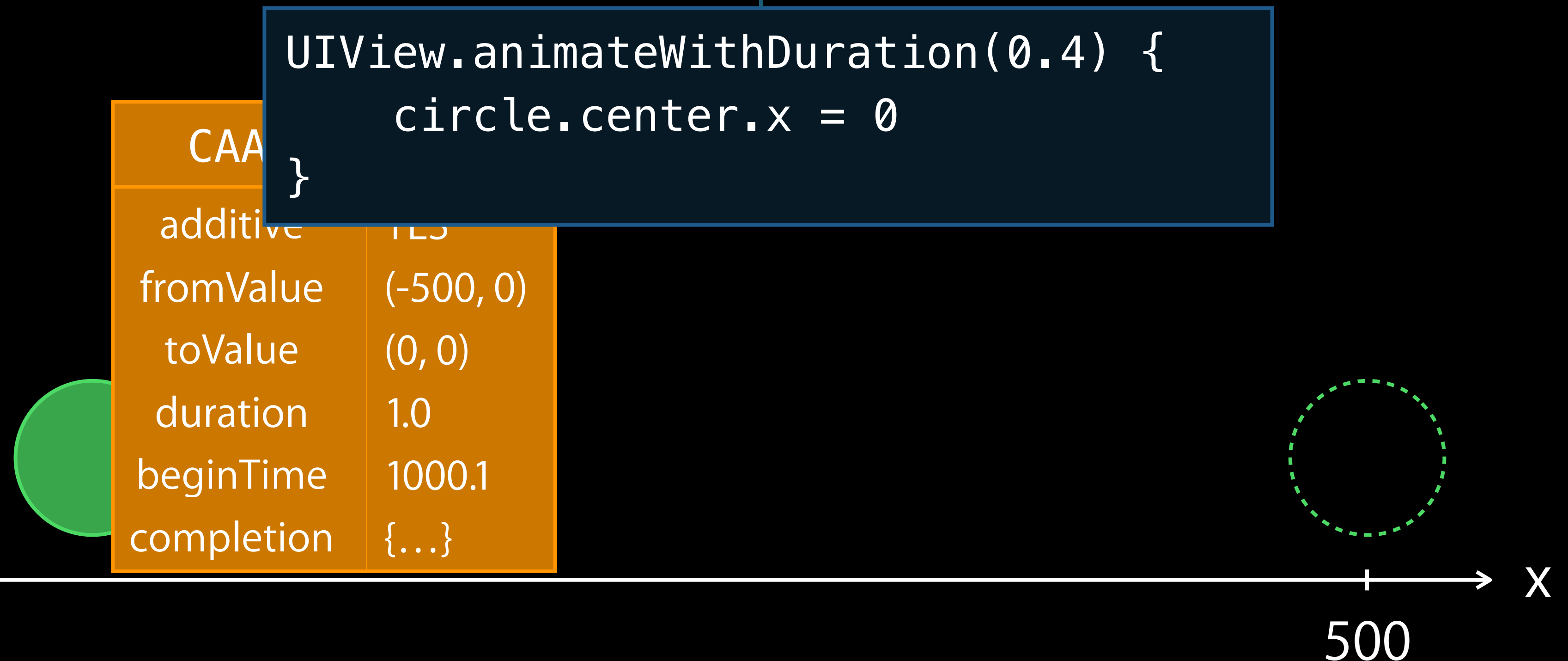
Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)



# Introducing Additive Animations

## Completion handlers (now)

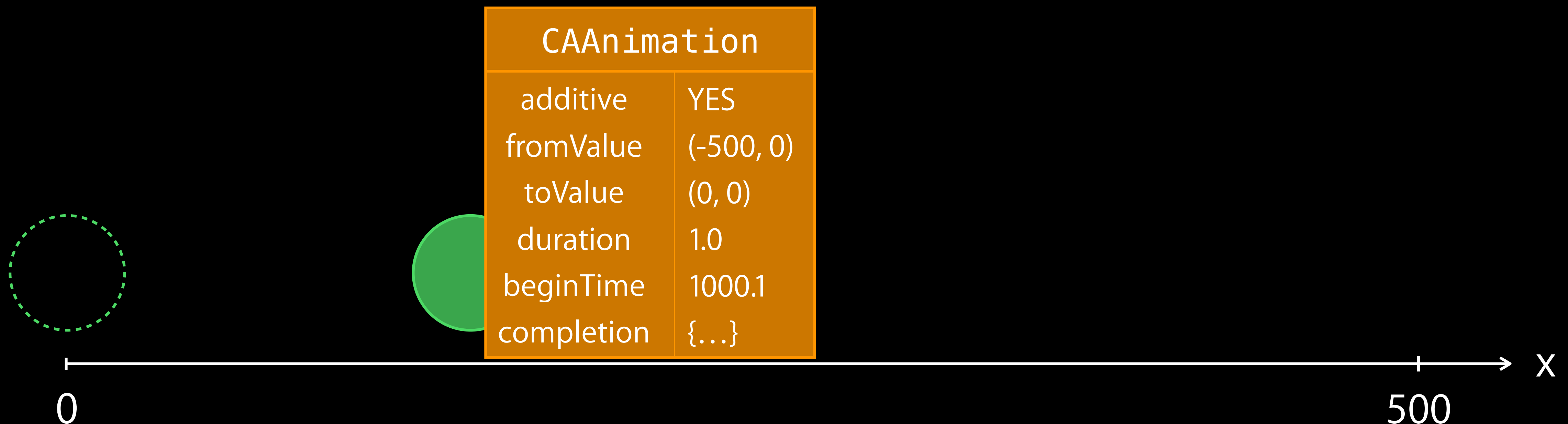
Time	1000.0	1000.1	1000.2	1000.3	1000.4
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)



# Introducing Additive Animations

Completion handlers (now)

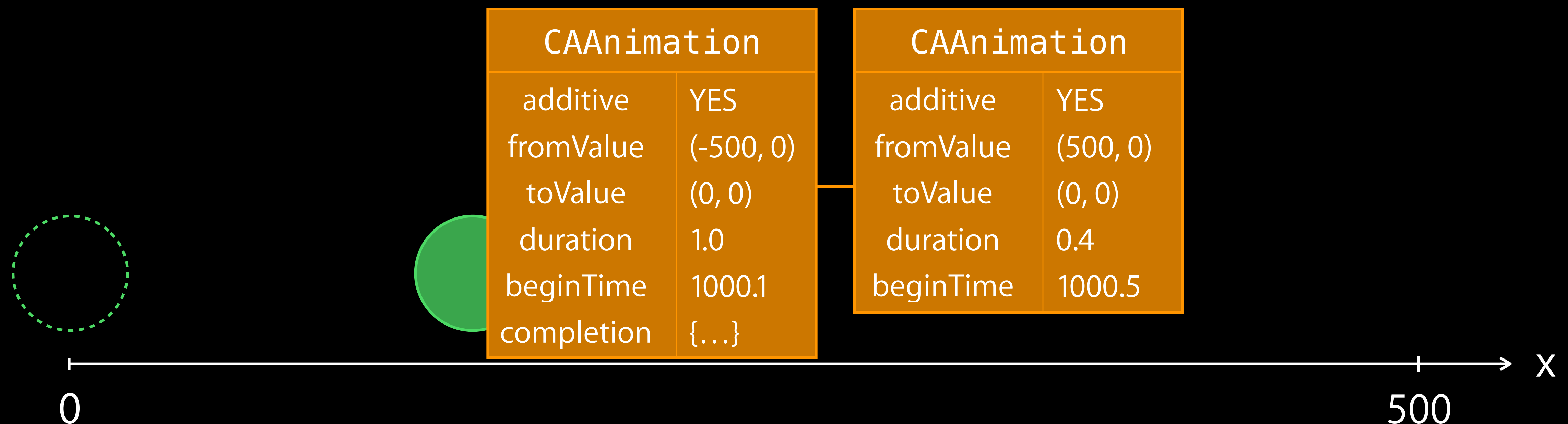
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	



# Introducing Additive Animations

## Completion handlers (now)

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)
Animation	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	

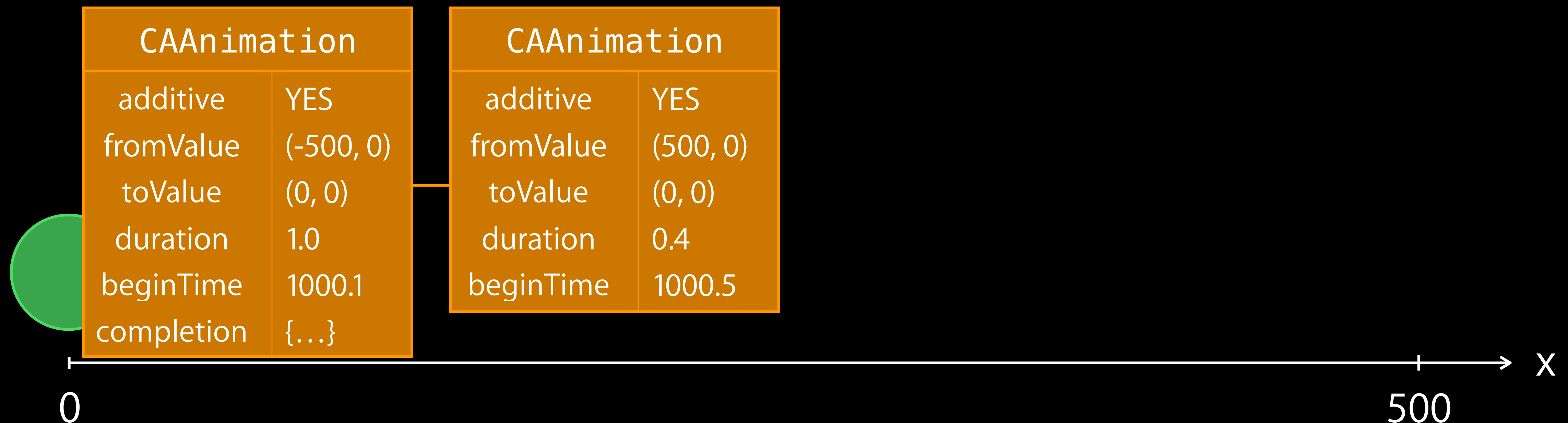




# Introducing Additive Animations

## Completion handlers (now)

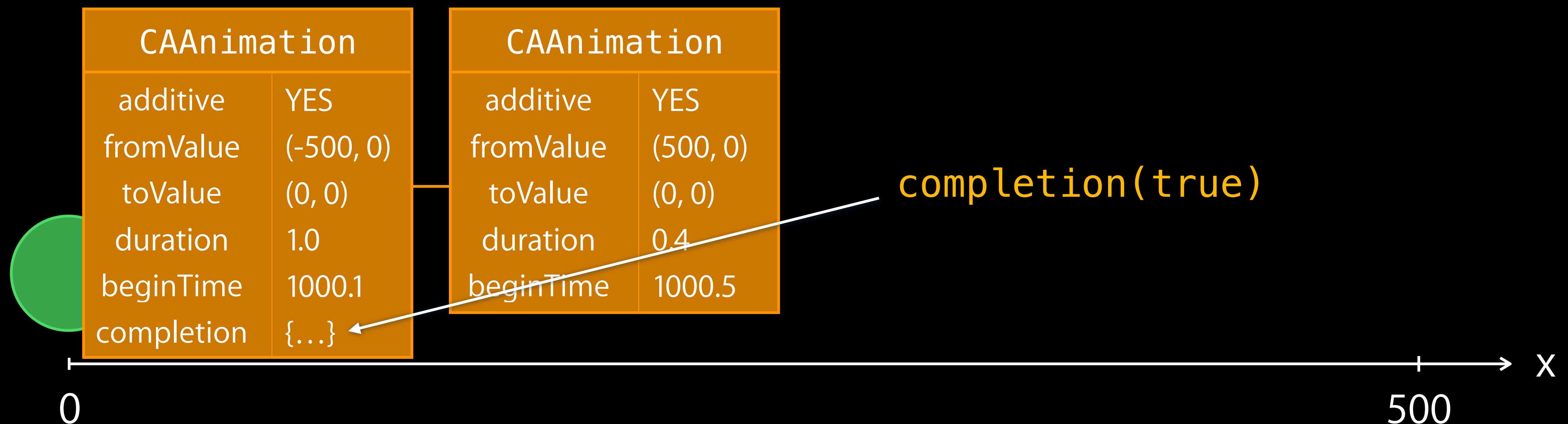
Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6	1000.7	1001.1
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Animation 1	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	(-300, 0)	(-250, 0)	(-200, 0)	(0, 0)
Animation 2	n/a	n/a	n/a	n/a	n/a	(500, 0)	(400, 0)	(300, 0)	(0, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(200, 0)	(150, 0)	(100, 0)	(0, 0)



# Introducing Additive Animations

## Completion handlers (now)

Time	1000.0	1000.1	1000.2	1000.3	1000.4	1000.5	1000.6	1000.7	1001.1
Model	(0, 0)	(500, 0)	(500, 0)	(500, 0)	(500, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Animation 1	n/a	(-500, 0)	(-450, 0)	(-400, 0)	(-350, 0)	(-300, 0)	(-250, 0)	(-200, 0)	(0, 0)
Animation 2	n/a	n/a	n/a	n/a	n/a	(500, 0)	(400, 0)	(300, 0)	(0, 0)
Presentation	(0, 0)	(0, 0)	(50, 0)	(100, 0)	(150, 0)	(200, 0)	(150, 0)	(100, 0)	(0, 0)



# Introducing Additive Animations

Summary

# Introducing Additive Animations

## Summary

Smoother transitions by default

# Introducing Additive Animations

## Summary

Smoother transitions by default

Still use `BeginFromCurrentState` if unsure

# Introducing Additive Animations

## Summary

Smoother transitions by default

Still use `BeginFromCurrentState` if unsure

Completion handlers may stack

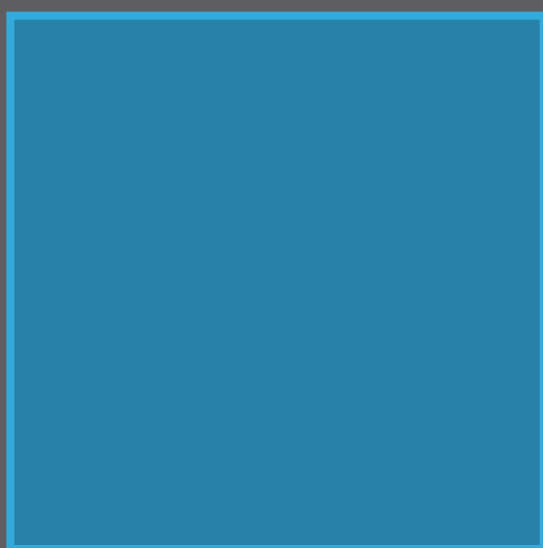
# From Animation to Gesture

Josh Shaffer

iOS Apps and Frameworks Engineer

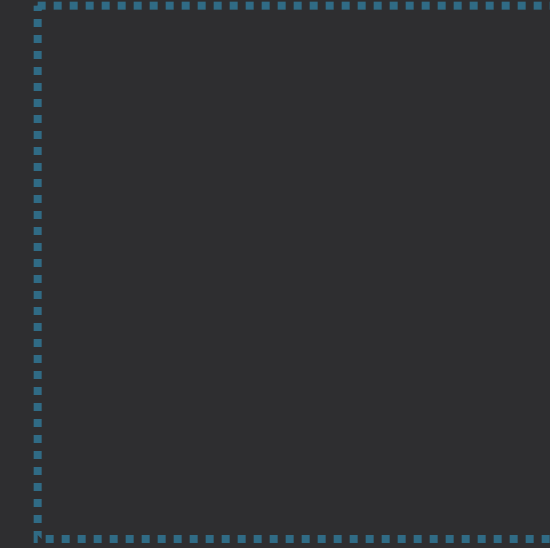
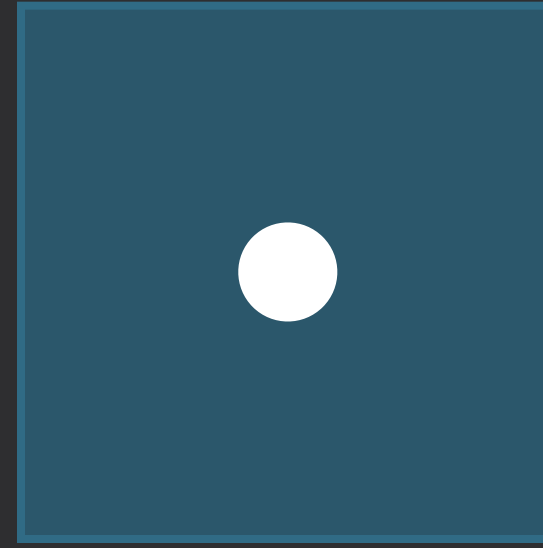


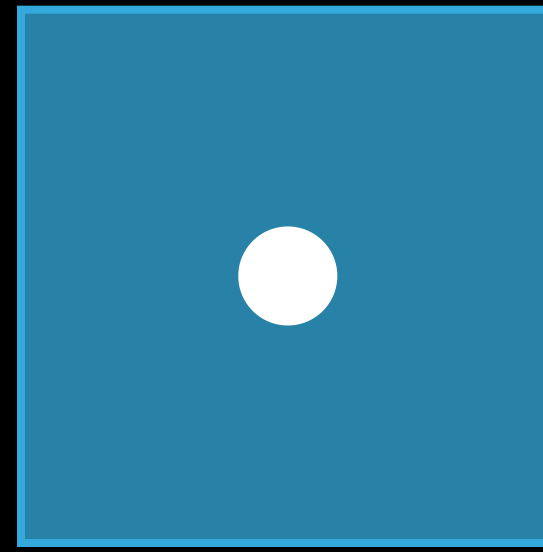


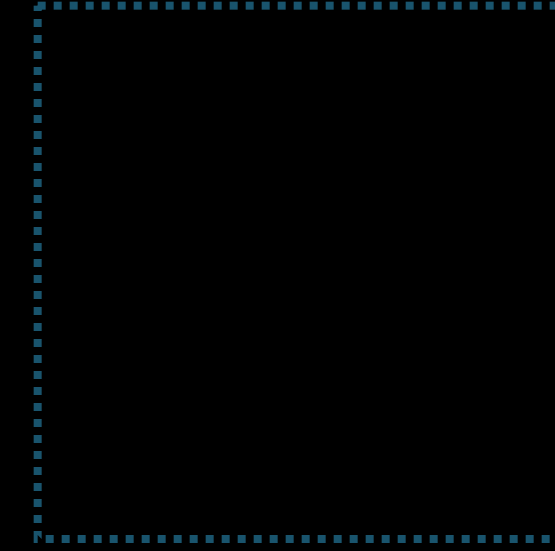




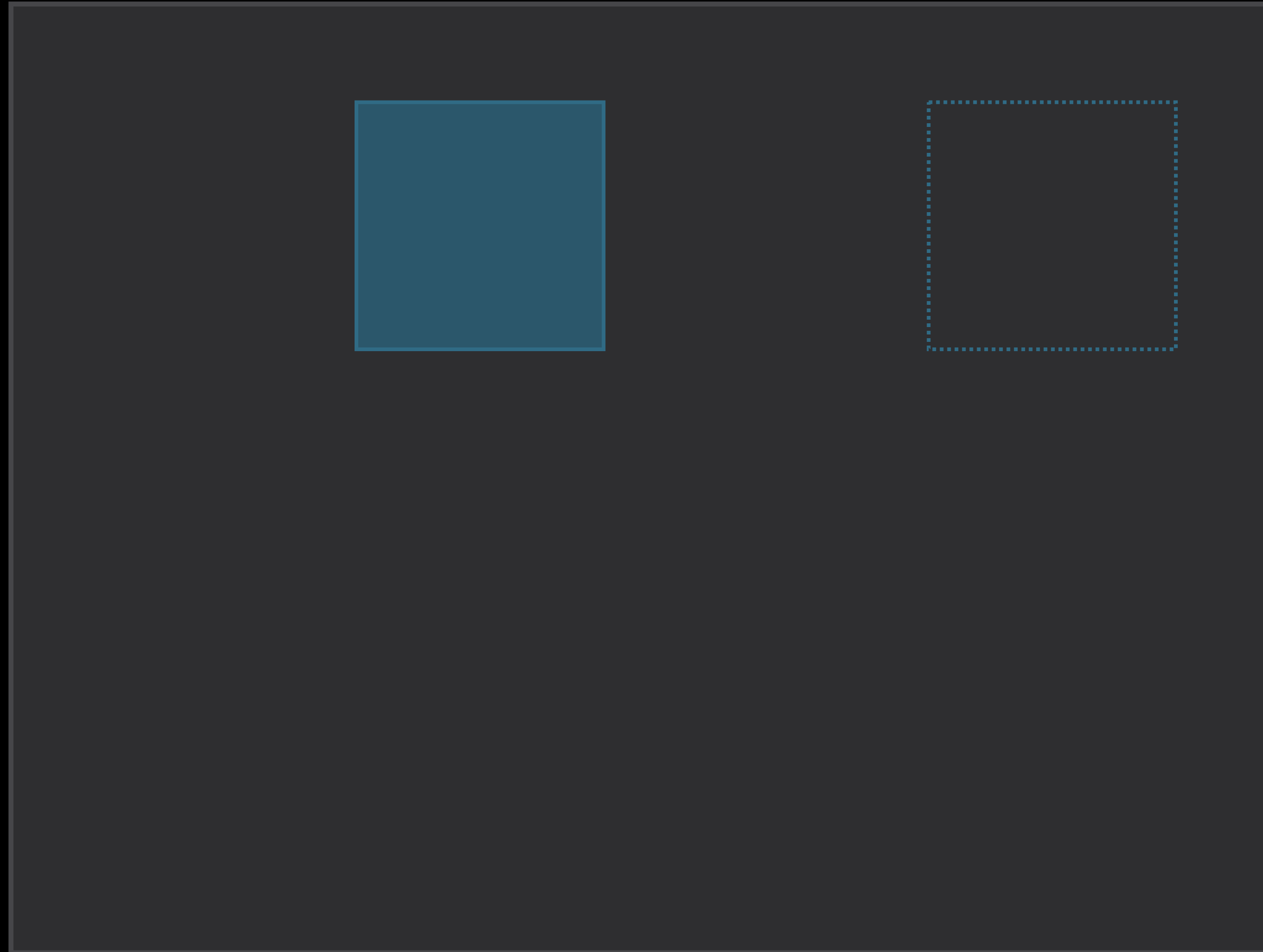




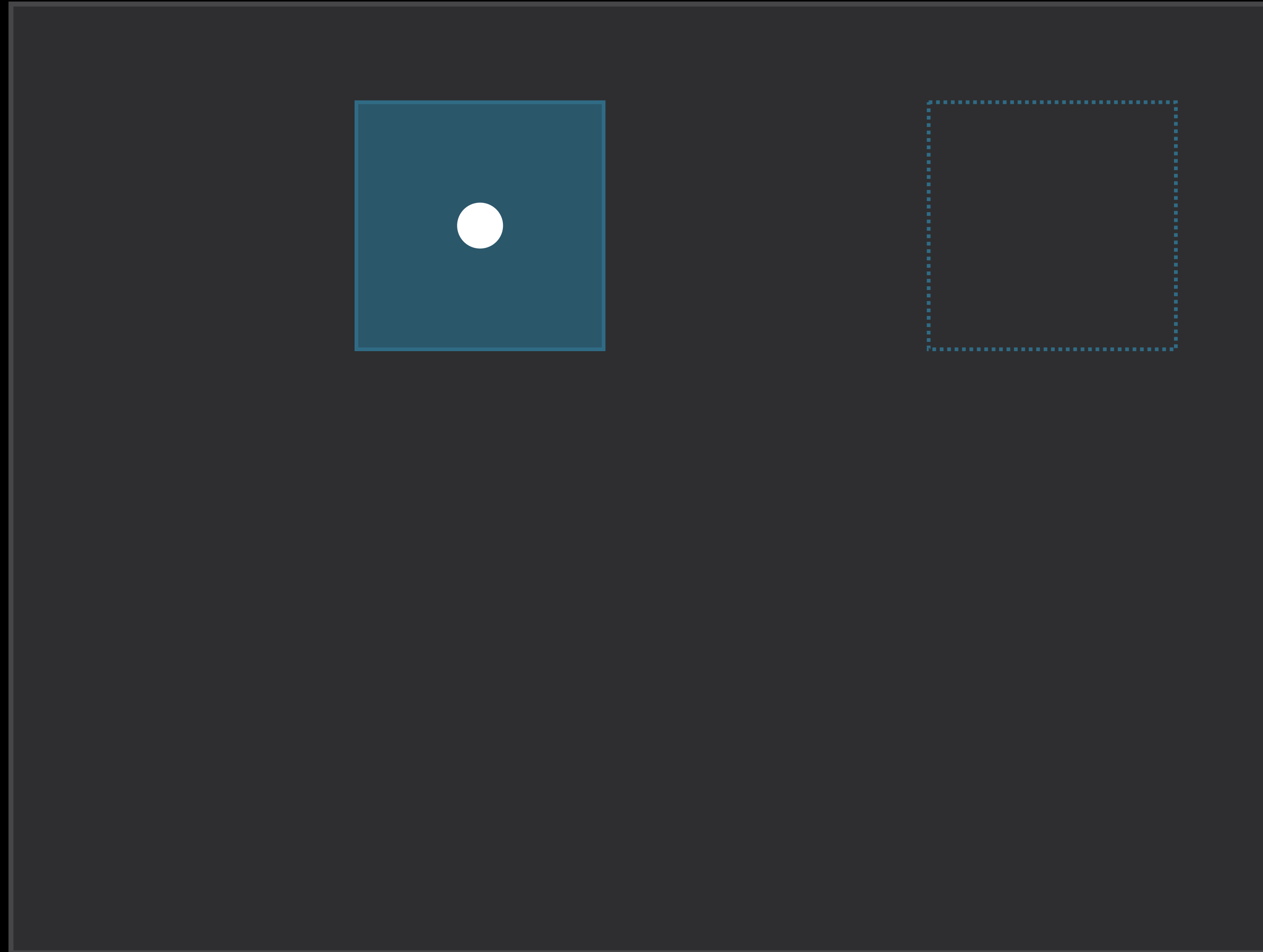




# .AllowsUserInteraction

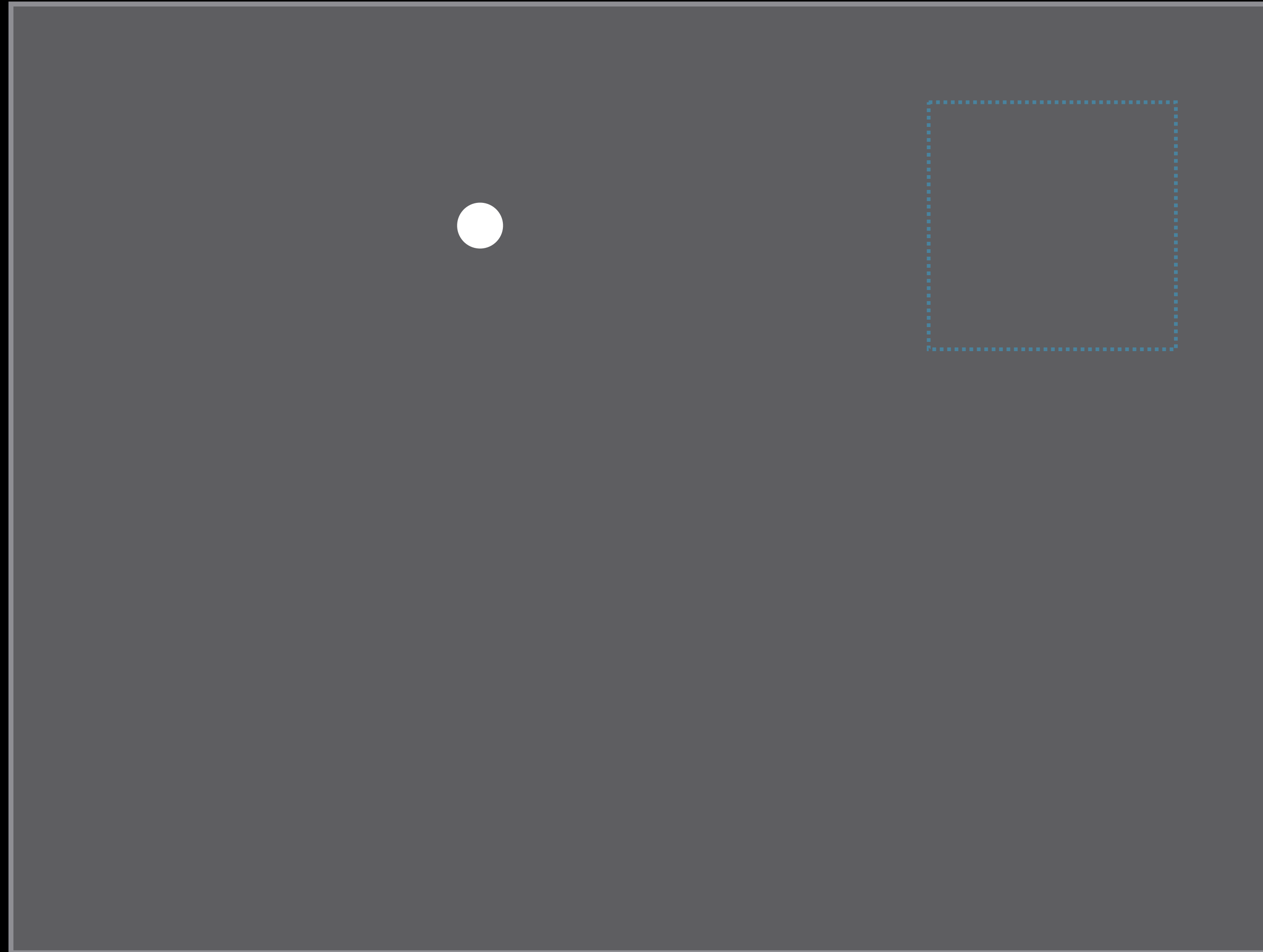


# .AllowsUserInteraction





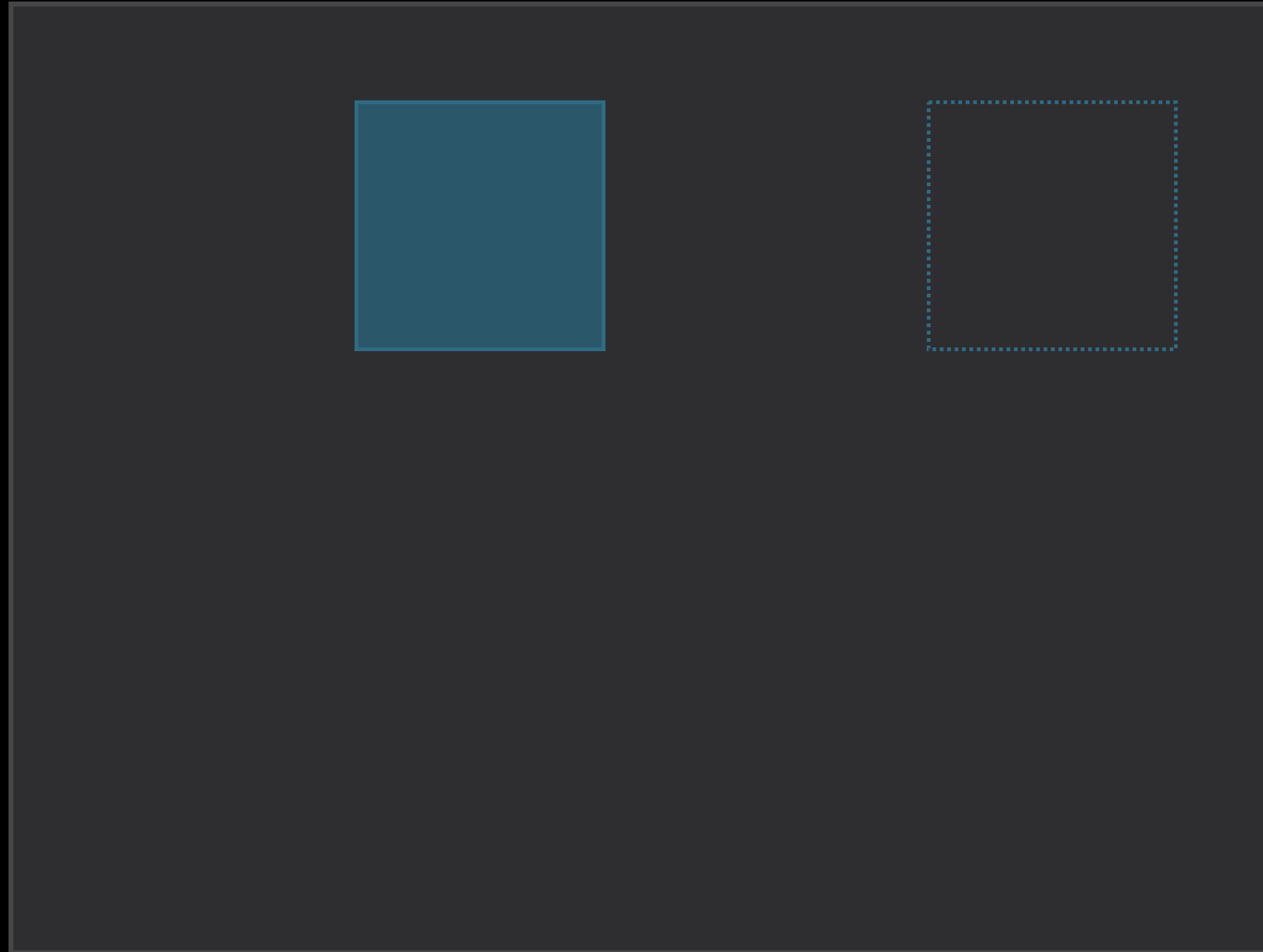
# .AllowsUserInteraction



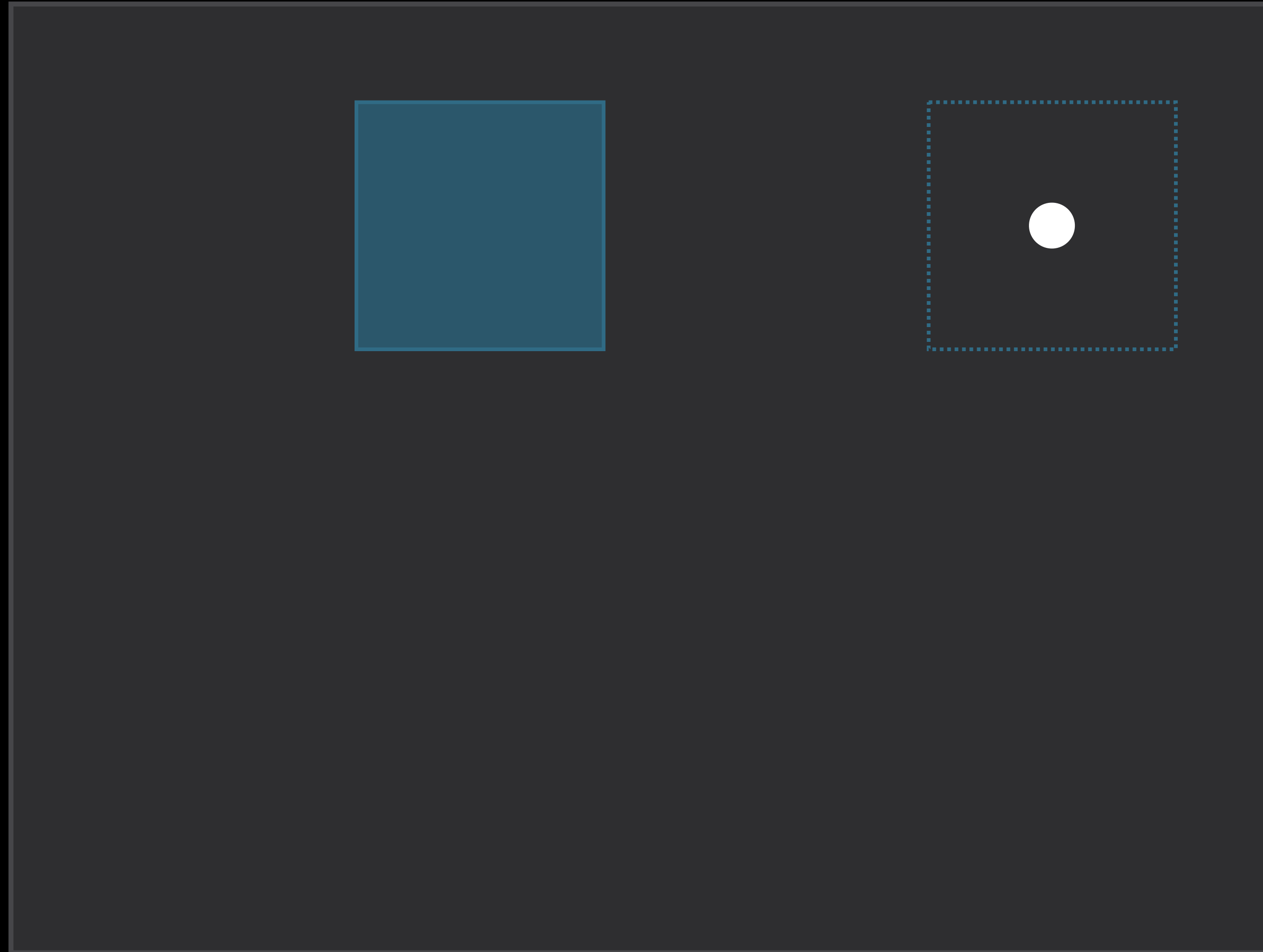
# .AllowsUserInteraction



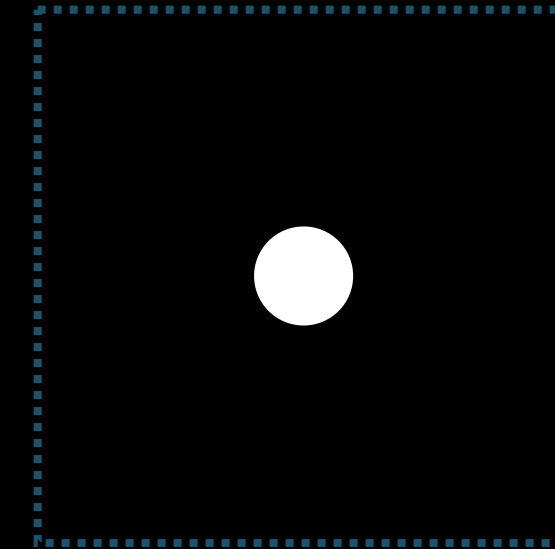
# .AllowsUserInteraction



# .AllowsUserInteraction



# .AllowsUserInteraction



# .AllowsUserInteraction



# Presentation Layer Hit Test





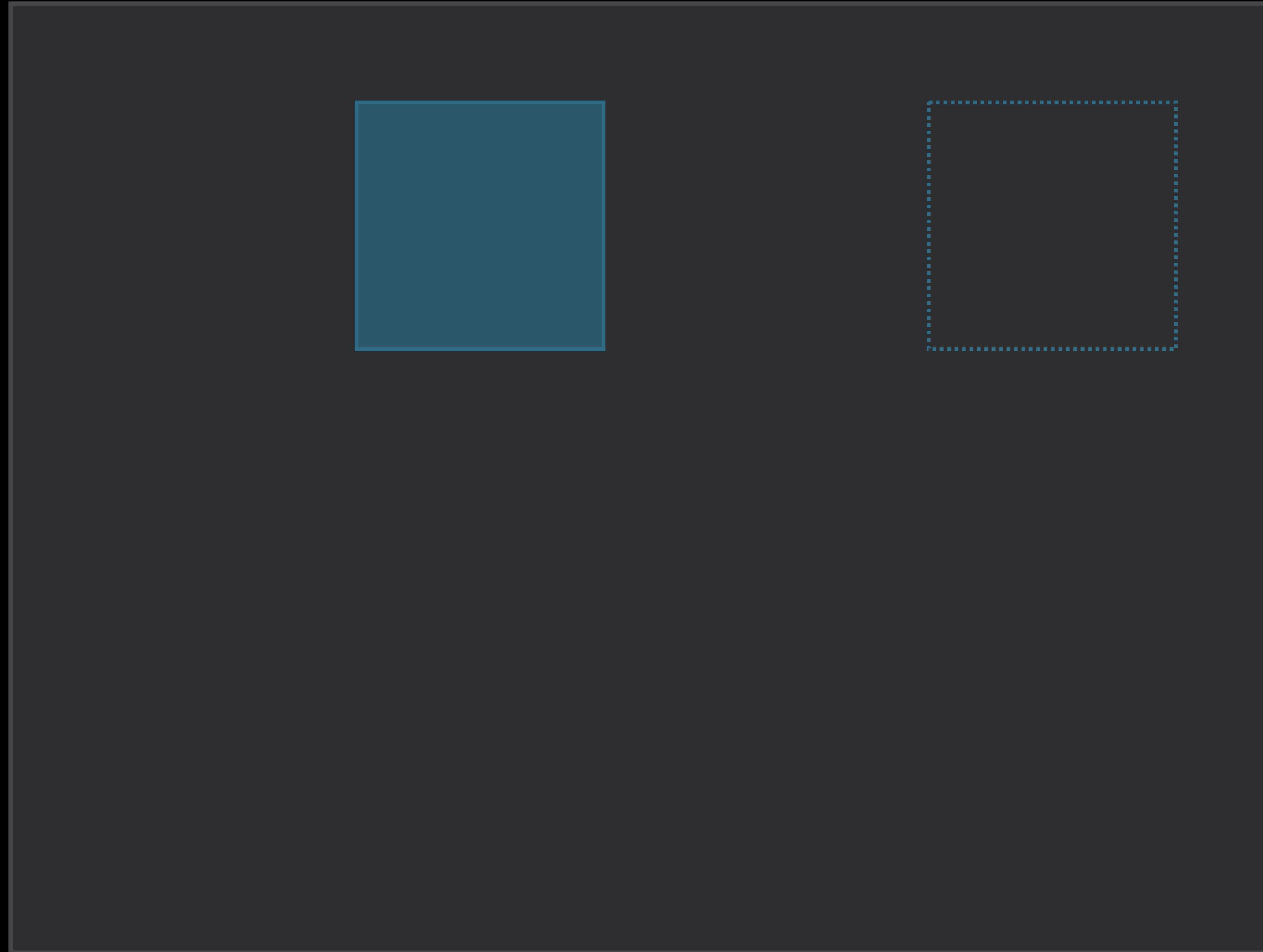




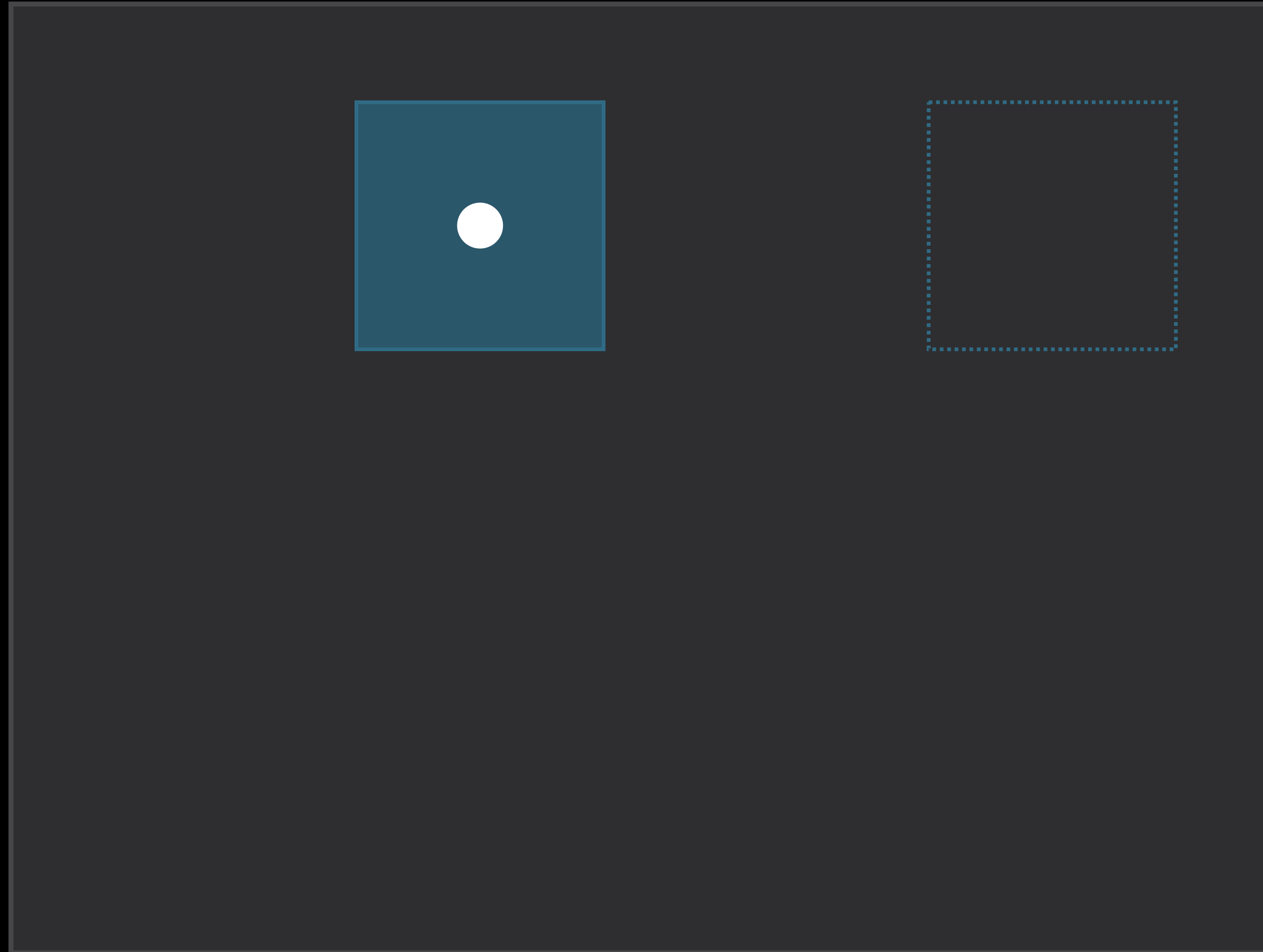
# Presentation Layer Hit Test

```
override func hitTest(point: CGPoint, withEvent event: UIEvent!) -> UIView! {  
    let superviewPoint = convertPoint(point, toView: superview)  
    let point = layer.presentationLayer.convertPoint(superviewPoint,  
                                                    fromLayer: superview.layer)  
  
    return super.hitTest(point, withEvent: event)  
}
```

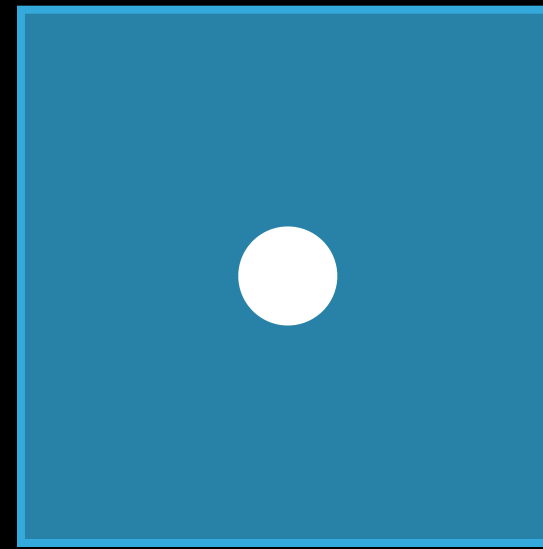
# .AllowsUserInteraction



# .AllowsUserInteraction



# .AllowsUserInteraction



# .AllowsUserInteraction



touch.locationInView



Stop Animating

# Stop Animating

```
let presentationPosition = view.layer.presentationLayer().position
```

# Stop Animating

```
let presentationPosition = view.layer.presentationLayer().position  
view.center = presentationPosition
```

# Stop Animating

```
let presentationPosition = view.layer.presentationLayer().position
view.center = presentationPosition
view.layer.removeAllAnimations()
```

UIDynamicAnimator

# UIDynamicAnimator

```
switch (panGestureRecognizer.state) {  
    case .Ended:  
        let v = panGestureRecognizer.velocityInView(targetView.superview)  
  
        dynamicItemBehavior.addLinearVelocity(v, forItem: targetView)  
    }  
}
```

# UIDynamicAnimator

```
switch (panGestureRecognizer.state) {  
    case .Began:  
  
    case .Ended:  
        let v = panGestureRecognizer.velocityInView(targetView.superview)  
        dynamicItemBehavior.addLinearVelocity(v, forItem: targetView)  
}
```

# UIDynamicAnimator

```
switch (panGestureRecognizer.state) {  
    case .Began:  
        dynamicItemBehavior.removeItem(targetView)  
  
    case .Ended:  
        let v = panGestureRecognizer.velocityInView(targetView.superview)  
        dynamicItemBehavior.addLinearVelocity(v, forItem: targetView)  
}
```



# UIDynamicAnimator

```
switch (panGestureRecognizer.state) {  
    case .Began:  
        dynamicItemBehavior.removeItem(targetView)  
  
    case .Ended:  
        let v = panGestureRecognizer.velocityInView(targetView.superview)  
  
        dynamicItemBehavior.addItem(targetView)  
        dynamicItemBehavior.addLinearVelocity(v, forItem: targetView)  
}
```

Gesture to Animation

---

Animation to Animation

---

Animation to Gesture

# Transient Animation State

*Demo*

Andy Matuschak

# Summary

# Summary

Use gesture velocity in animations

# Summary

Use gesture velocity in animations

Smoothly transition between animations

# Summary

Use gesture velocity in animations

Smoothly transition between animations

Make animations interruptible



# Summary

Use gesture velocity in animations

Smoothly transition between animations

Make animations interruptible

Mind your state

# More Information

Jake Behrens

App Frameworks Evangelist

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

Documentation

Core Animation Programming Guide

<http://developer.apple.com>

Apple Developer Forums

<http://devforums.apple.com>

# Labs

- 
- Open Hours

Frameworks Lab A and B

Friday 2:00PM

---

 WWDC14