Advanced Animations with UIKit

devliubo.com

- 基础动画与时间曲线
- 可交互与可中断的动画
- iOS11新增属性

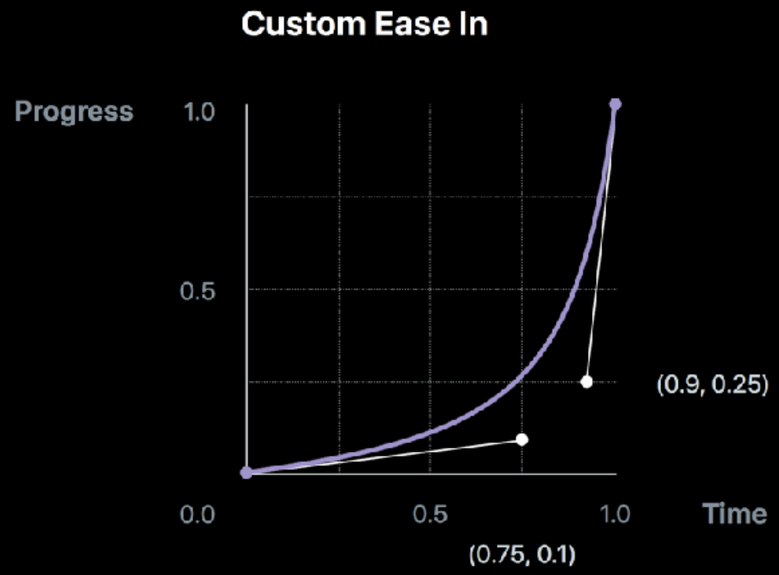
基础动画

- ```
  [UIView animateWithDuration:4.f animations:^{
      CGRect oriFrame = self.viewToMove.frame;
      oriFrame.origin.x += 200;

      [self.viewToMove setFrame:oriFrame];
  }];
  ```

- ```
  self.propertyAnimator = [[UIViewPropertyAnimator alloc] initWithDuration:
  4.f curve:UIViewAnimationCurveEaseOut animations:^{
      CGRect oriFrame = self.viewToMove.frame;
      oriFrame.origin.x += 200;

      [self.viewToMove setFrame:oriFrame];
  }];

  [self.propertyAnimator startAnimation];
  ```

时间曲线

- iOS提供四种曲线：Linear、Easy-In、Easy-Out、Easy-In-Out
- UICubicTimingParameters，通过修改控制点自定义时间曲线
- http://cubic-bezier.com/

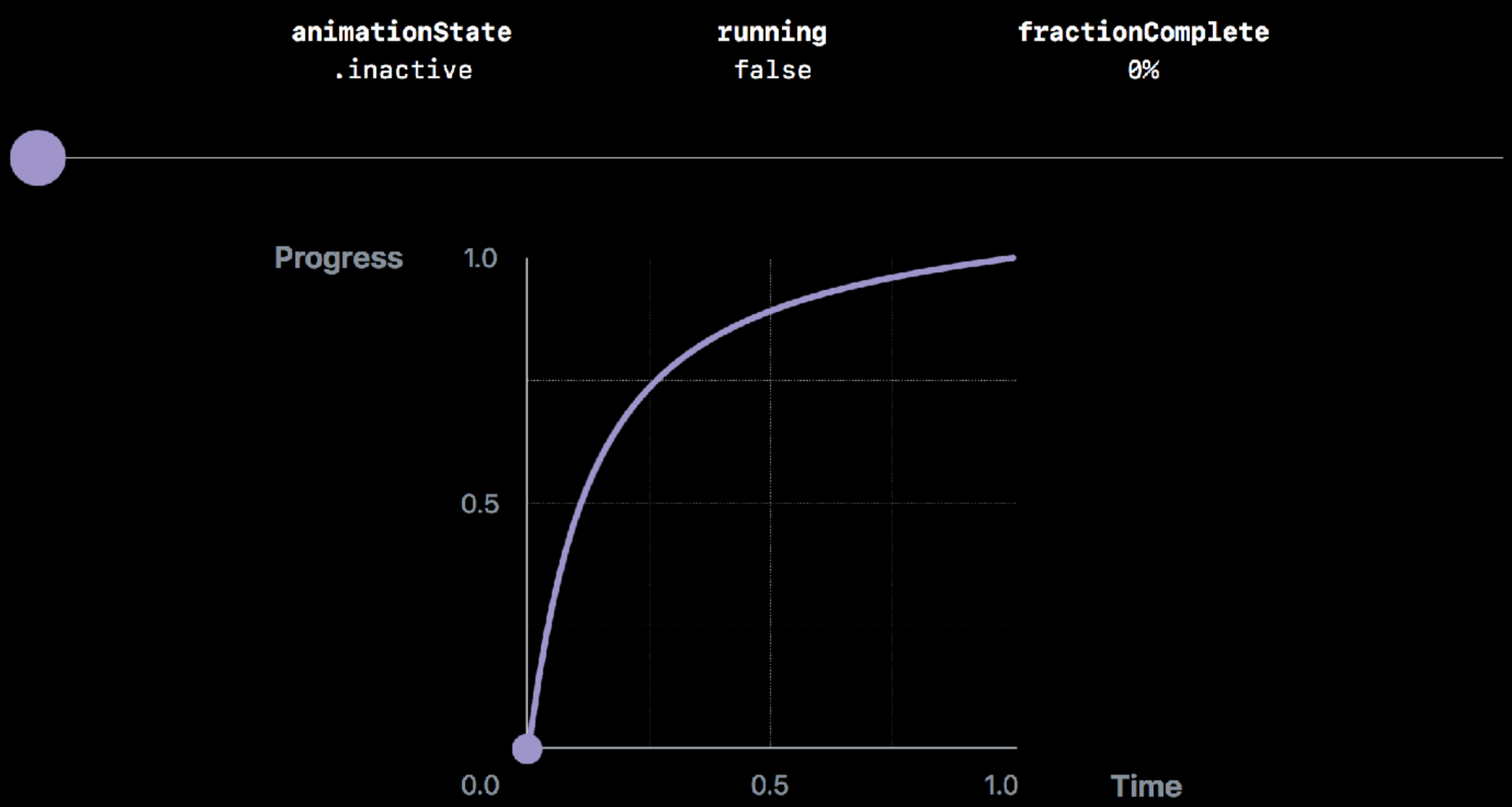



```
[[UICubicTimingParameters alloc] initWithControlPoint1:CGPointMake(0.75, 0.1)
                                         controlPoint2:CGPointMake(0.9, 0.25)];
```

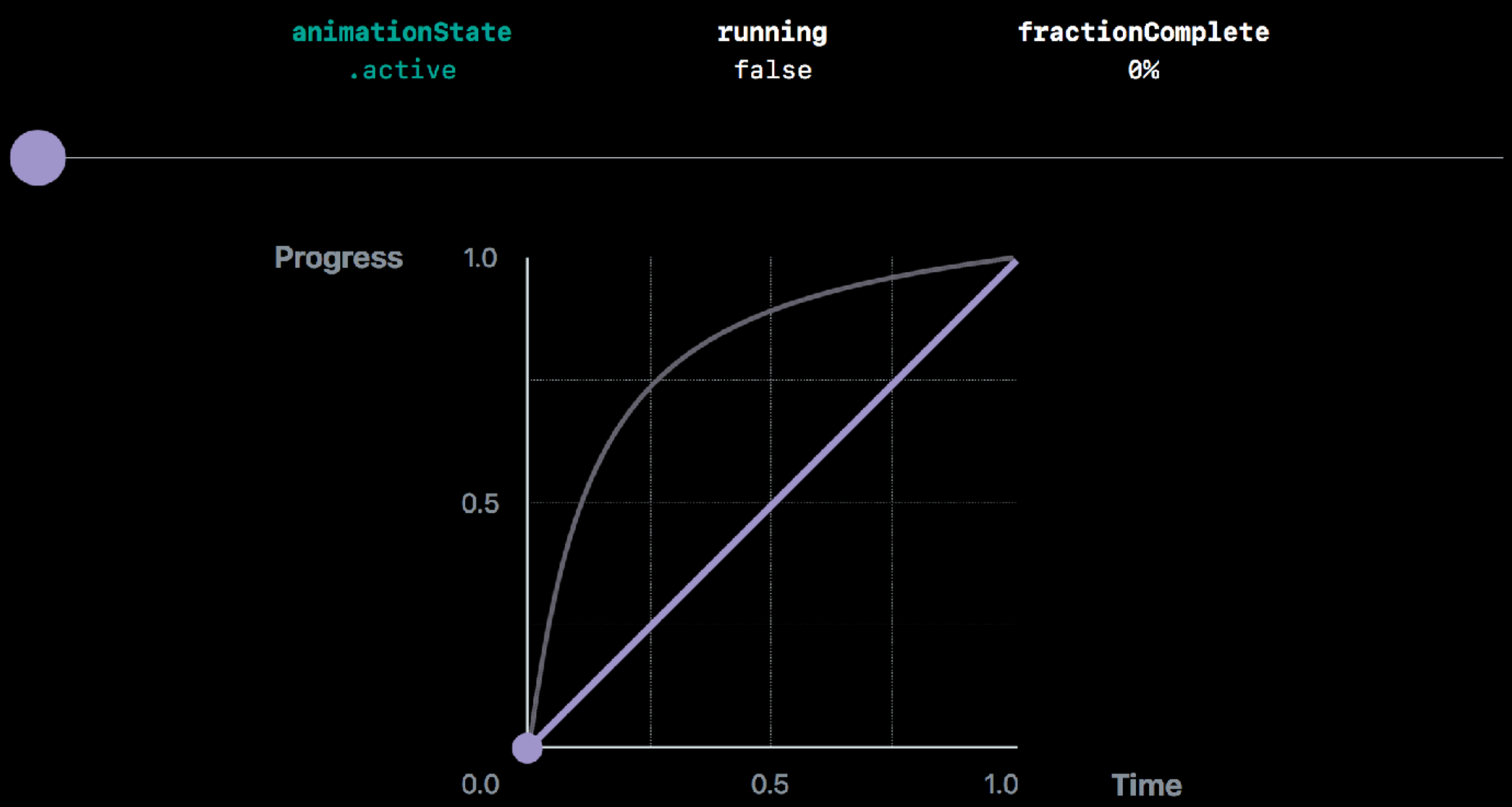
可交互动画

```
- (void)panGestureRecognizerAction:(UIPanGestureRecognizer *)panGestureRecognizer {
    if (panGestureRecognizer.state == UIGestureRecognizerStateBegan) {

        self.propertyAnimator = [[UIViewPropertyAnimator alloc] initWithDuration:4.f
curve:UIViewAnimationCurveEaseOut animations:^{
            CGRect oriFrame = self.viewToMove.frame;
            oriFrame.origin.x += 200;

            [self.viewToMove setFrame:oriFrame];
        }];

        [self.propertyAnimator pauseAnimation];
    }
    else if (panGestureRecognizer.state == UIGestureRecognizerStateChanged) {
        CGPoint translation = [panGestureRecognizer translationInView:self.viewToMove];
        self.propertyAnimator.fractionComplete = translation.x / 200;
    }
    else if (panGestureRecognizer.state == UIGestureRecognizerStateEnded) {
        [self.propertyAnimator continueAnimationWithTimingParameters:nil durationFactor:0];
    }
}
```

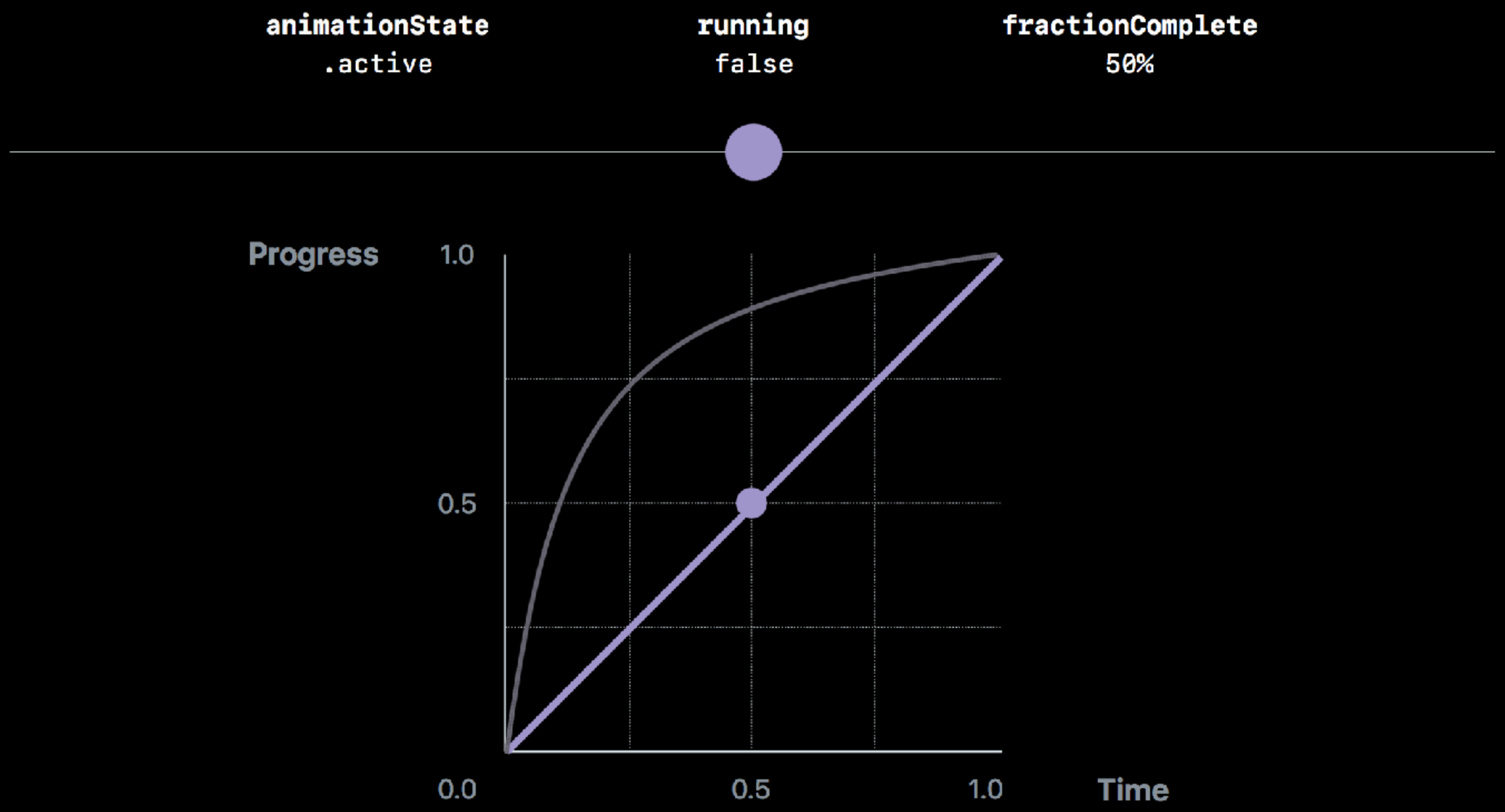
```
self.propertyAnimator = [[UIViewPropertyAnimator alloc] initWithDuration:4.f
curve:UIViewAnimationCurveEaseOut animations:^{ ..... }];
```



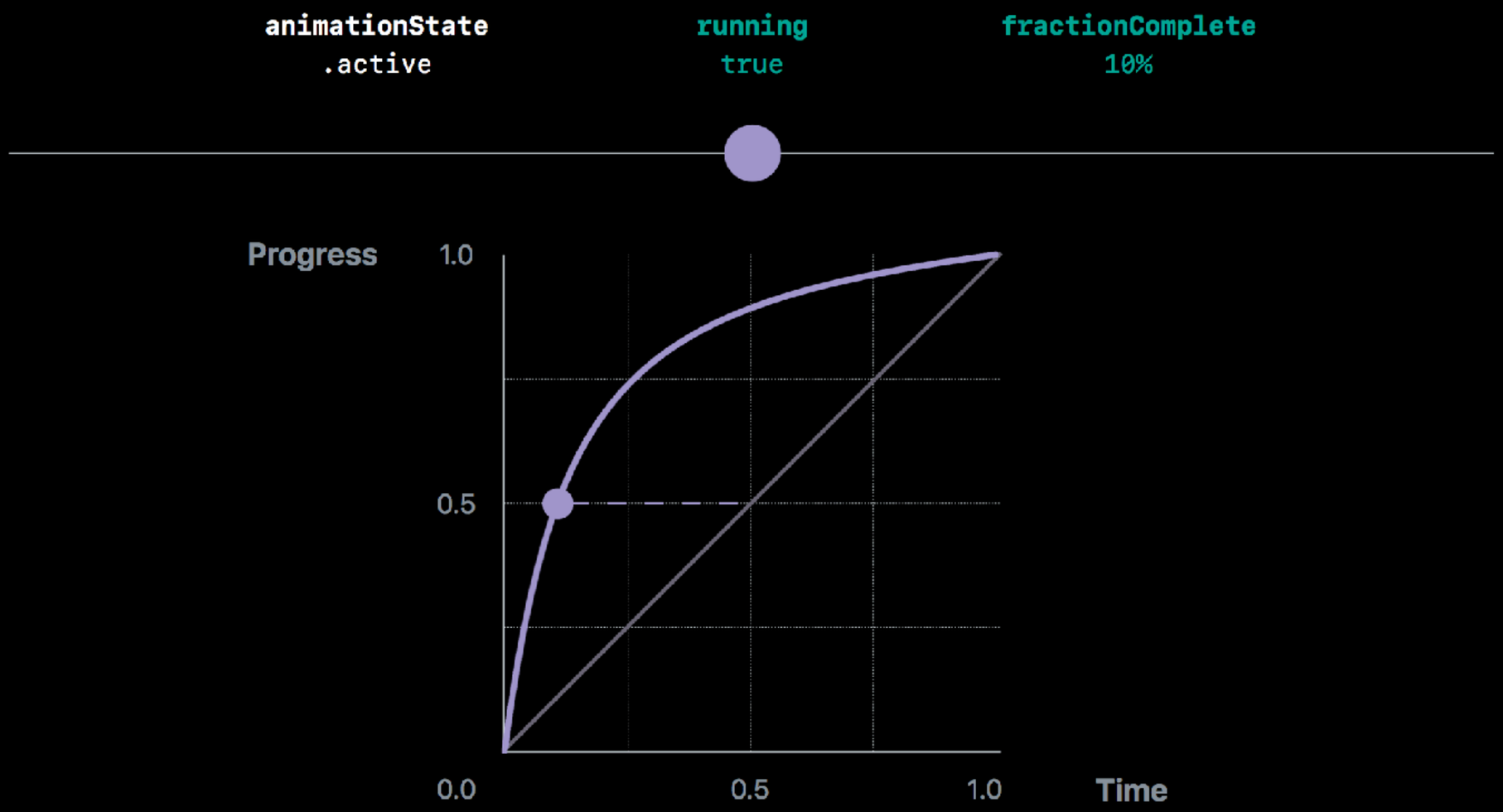
```
[self.propertyAnimator pauseAnimation];
```



```
self.propertyAnimator.fractionComplete = translation.x / 200;
```



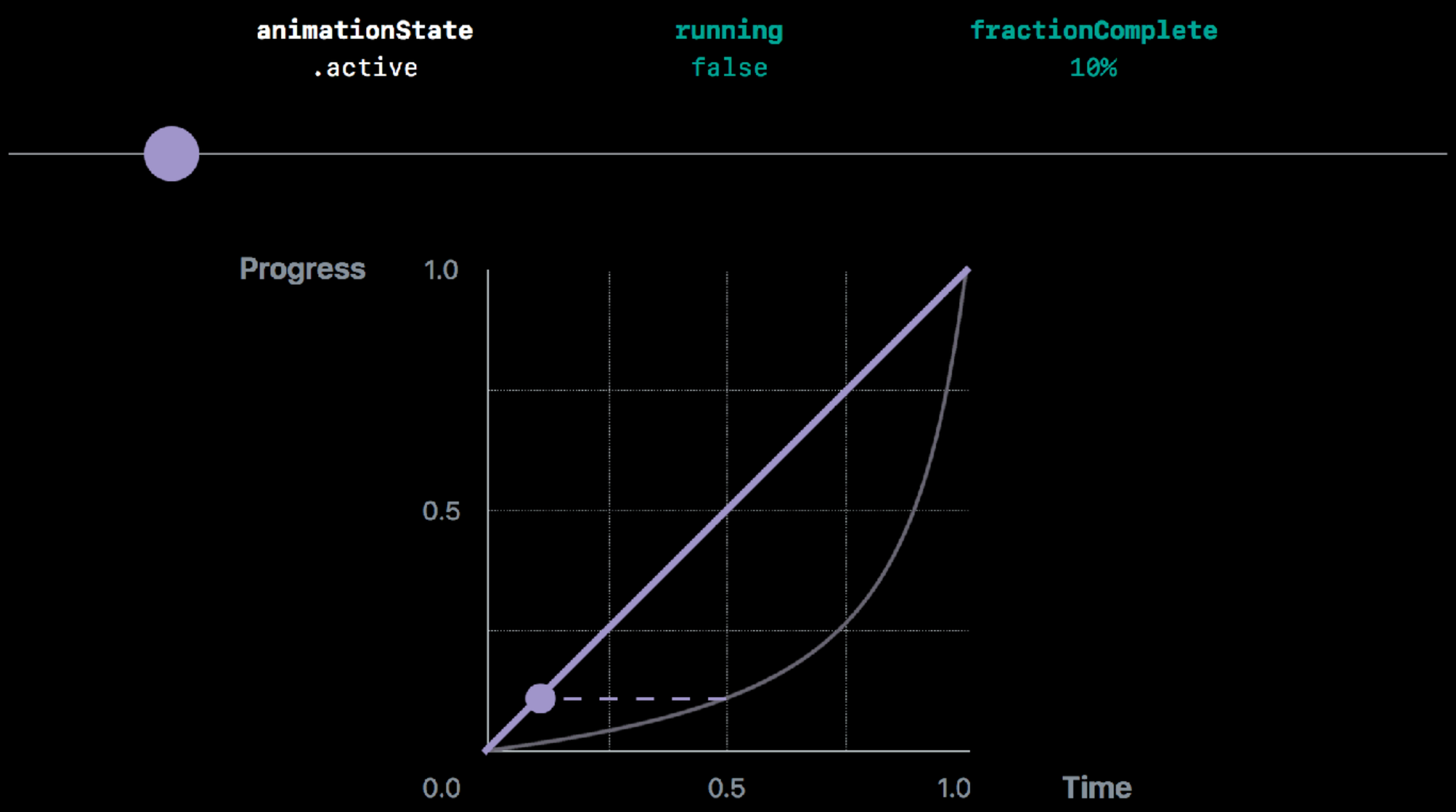
```
[self.propertyAnimator continueAnimationWithTimingParameters:nil durationFactor:0];
```



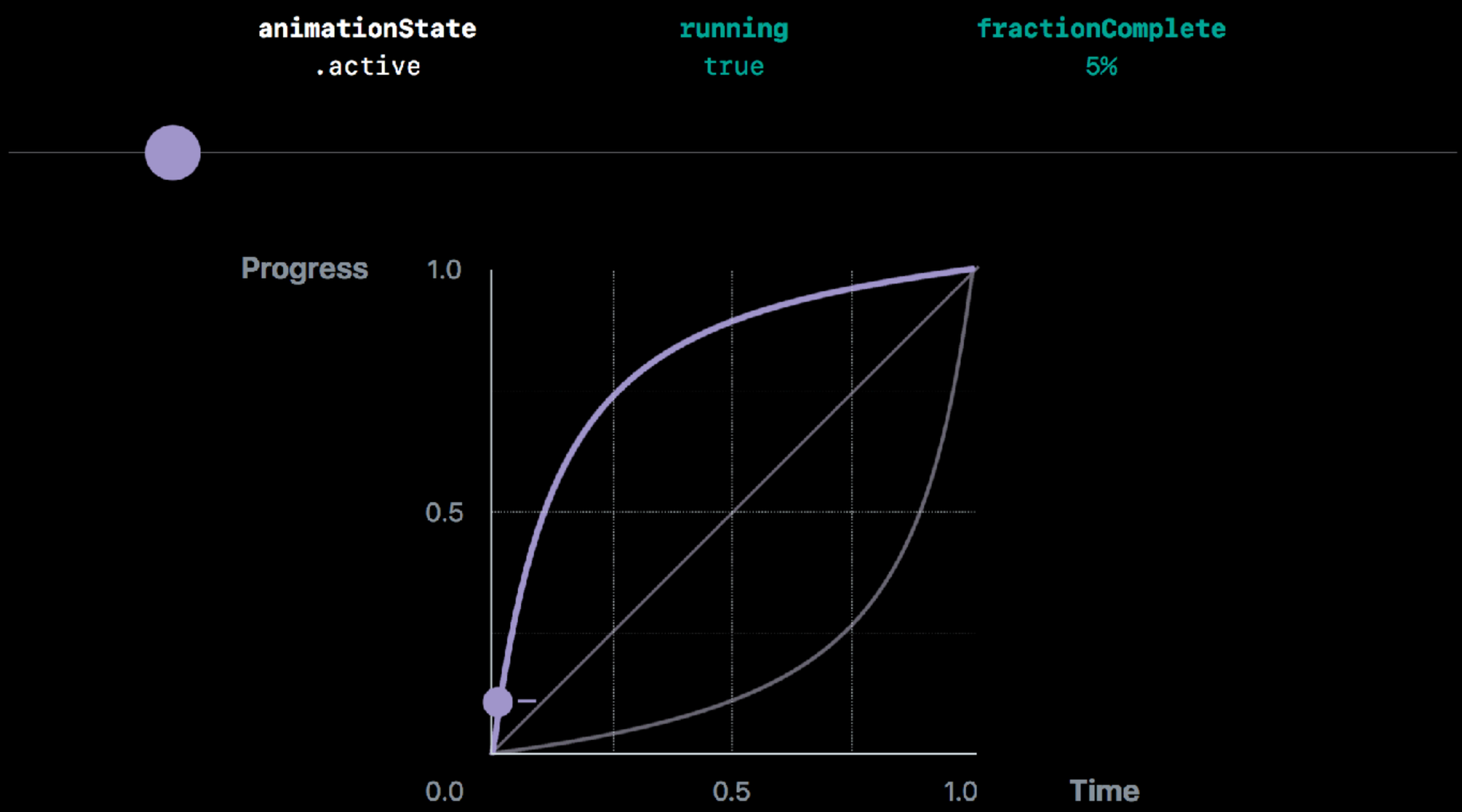
可中断动画

```
- (void)panGestureRecognizerAction:(UIPanGestureRecognizer *)panGestureRecognizer
{
    if (panGestureRecognizer.state == UIGestureRecognizerStateBegan) {
        [self createPropertyAnimatorIfNeed];
        [self.propertyAnimator pauseAnimation];

        self.progressWhenInterrupted = self.propertyAnimator.fractionComplete;
    }
    else if (panGestureRecognizer.state == UIGestureRecognizerStateChanged) {
        CGPoint translation = [panGestureRecognizer translationInView:self.viewToMove];

        self.propertyAnimator.fractionComplete = translation.x / 200 +
self.progressWhenInterrupted;
    }
    else if (panGestureRecognizer.state == UIGestureRecognizerStateEnded) {
        UICubicTimingParameters *timingParameter = [[UICubicTimingParameters alloc]
initWithAnimationCurve:UIViewAnimationCurveEaseOut];

        [self.propertyAnimator continueAnimationWithTimingParameters:timingParameter
durationFactor:0];
    }
}
```

```
[self.propertyAnimator pauseAnimation];
```

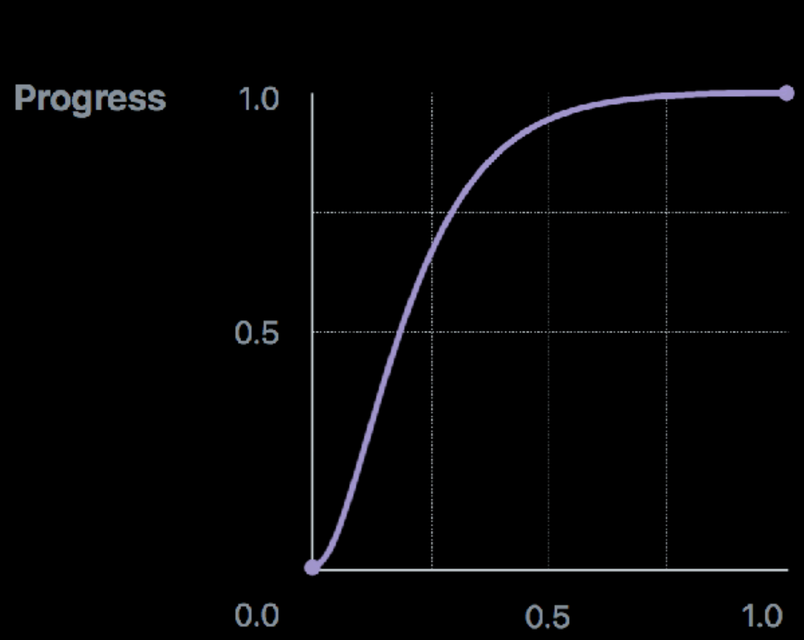


```
[self.propertyAnimator continueAnimationWithTimingParameters:timingParameter
durationFactor:0];
```



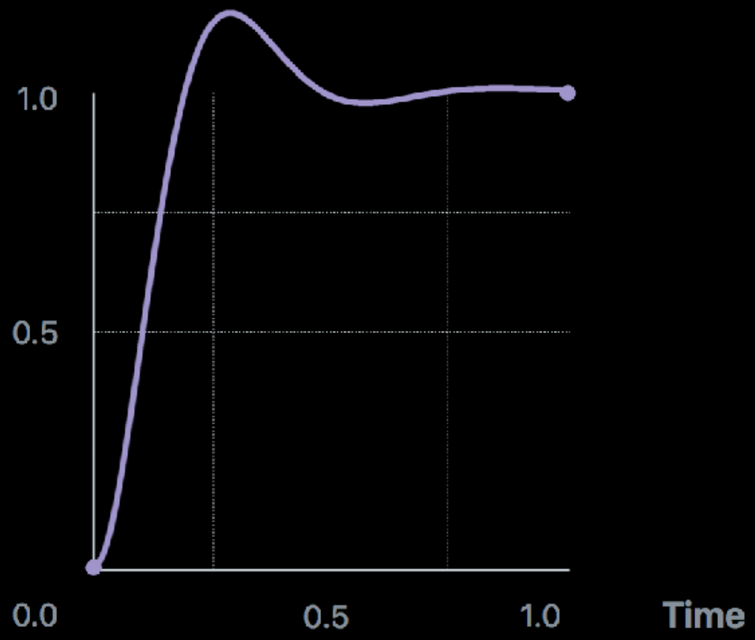
Spring Animations

`UISpringTimingParameters`



Critically damped spring

Damping ratio = 1.0

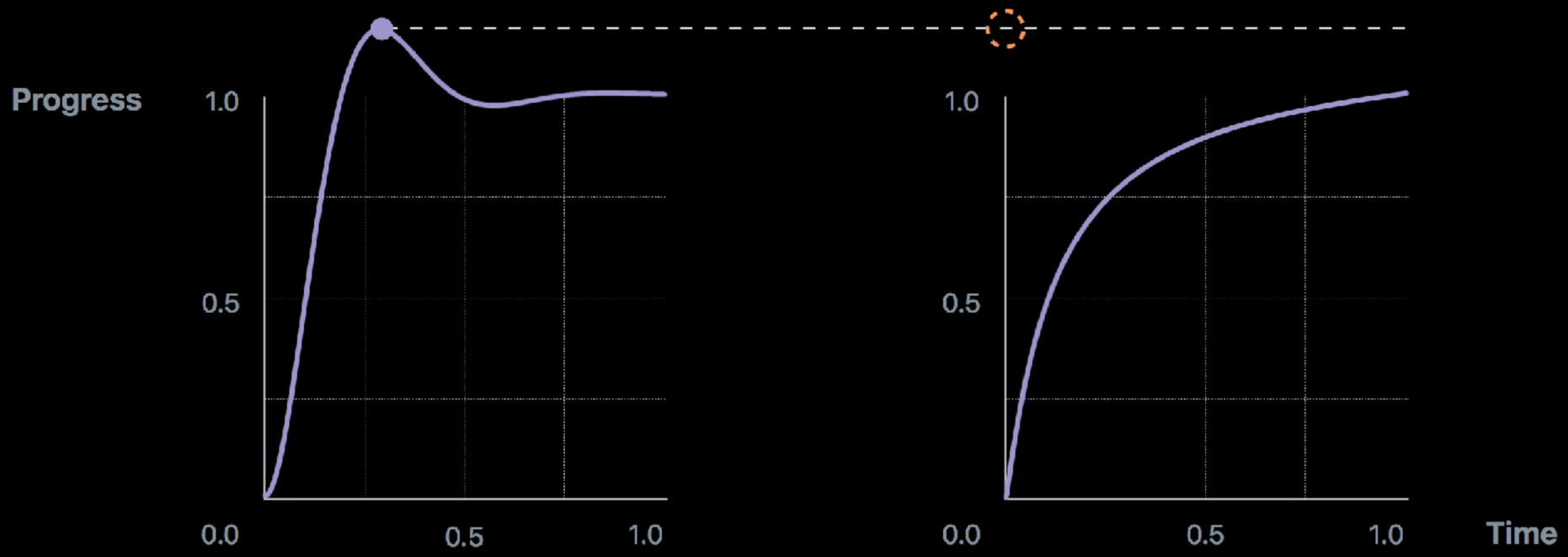


Under damped spring

Damping ratio < 1.0

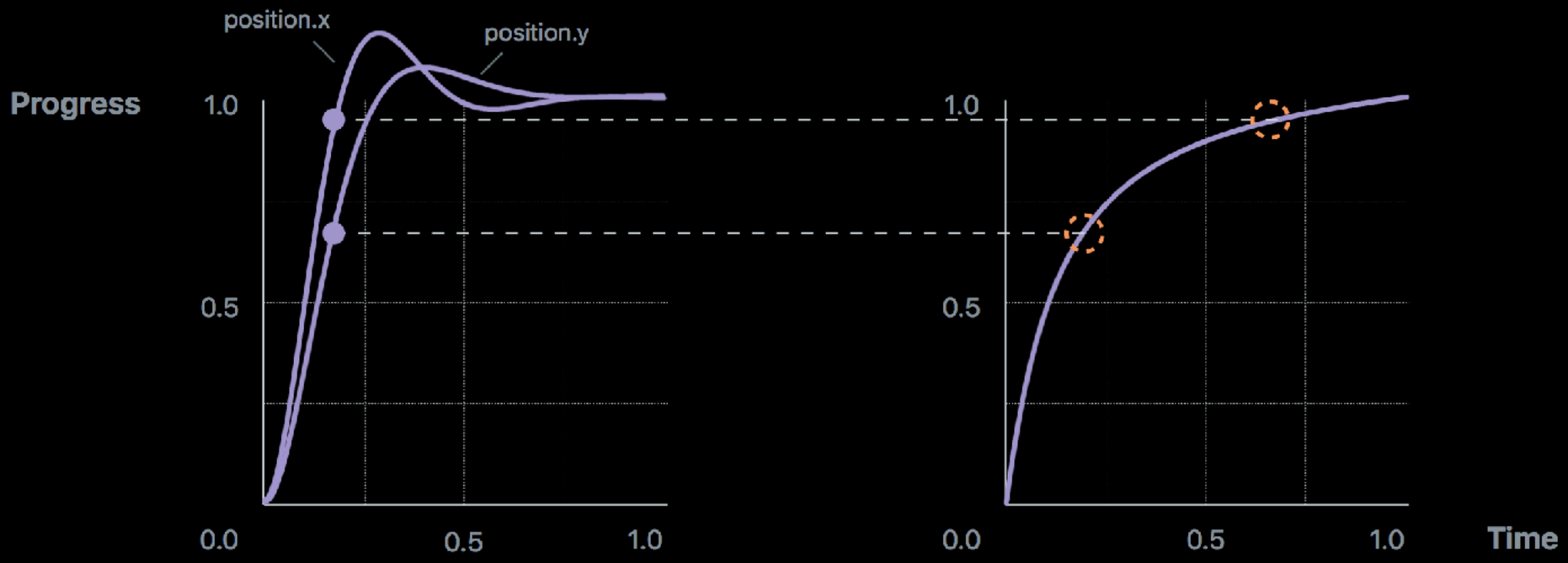
Spring Animations

- SpringTiming映射到UICubicTimingParameters的时候，有可能不存在对应点的。



Spring Animations

- 带初速度的SpringTiming映射到UICubicTimingParameters的时候可能不同步。



Spring Animations

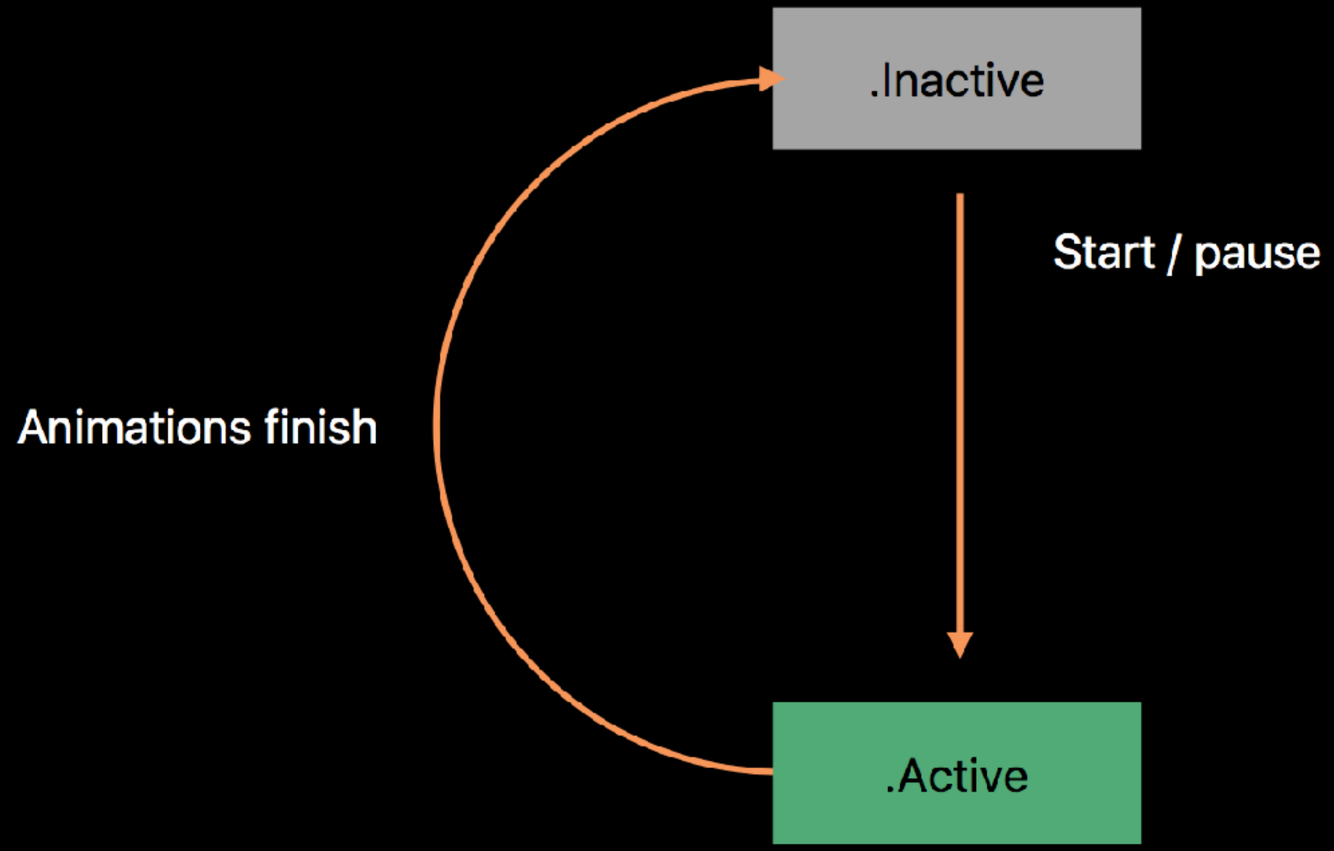
- 中断后不再继续Spring，将presentationLayer的状态赋值到modelLayer上，后续创建一个全新的动画。
- 如果想使用弹簧动画，则只能使用critically damped springs，同时不能指定初始速度。
- 如果想使用带初速度的弹簧动画，则建议手动去将这一个动画分解成这两个不同速度上的动画，对这两个动画逐个进行控制。

iOS 11 新增属性

```
/// Defaults to YES. Provides the ability for an animator to pause and scrub either linearly or using the animator’s current timing.
@property(nonatomic) BOOL scrubsLinearly NS_AVAILABLE_IOS(11_0);


/// Defaults to NO. Provides the ability for an animator to pause on completion instead of transitioning to the .inactive state.
@property(nonatomic) BOOL pausesOnCompletion NS_AVAILABLE_IOS(11_0);
```

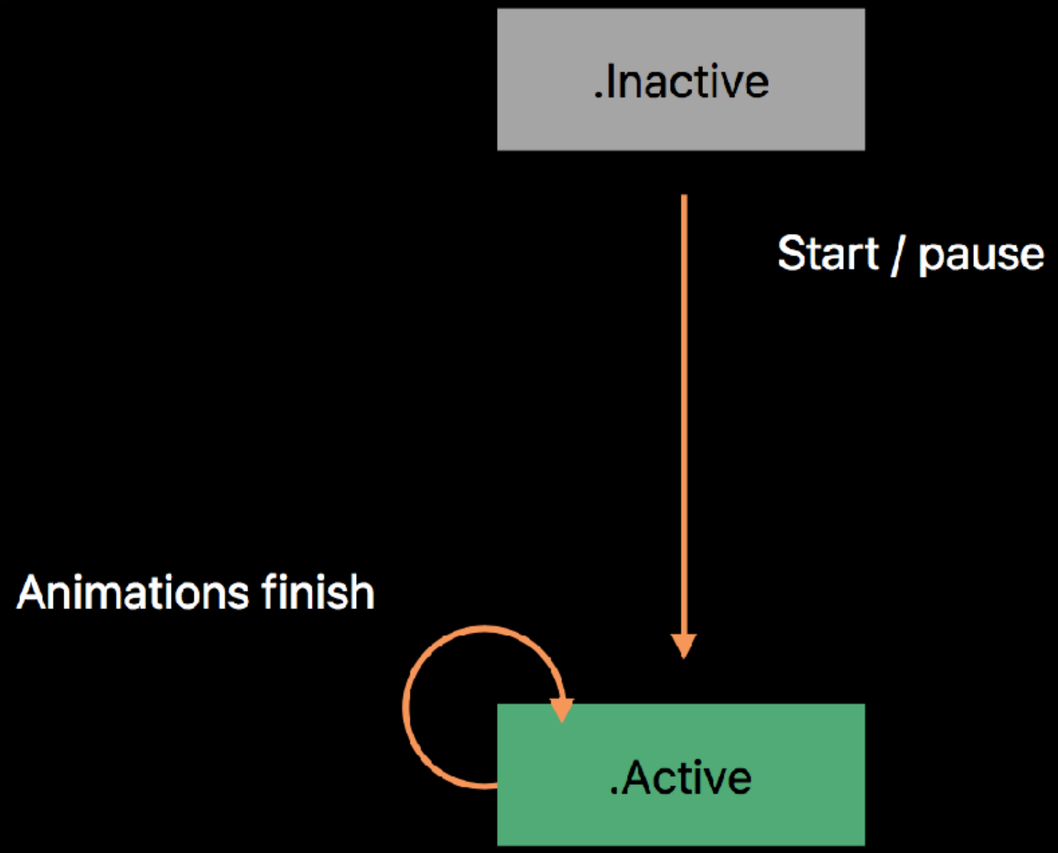
iOS 11 新增属性

```
pausesOnCompletion = NO;
```



iOS 11 新增属性

```
pausesOnCompletion = YES;
```



https://developer.apple.com/wwdc17/230

Thanks ~