Custom View Controller Transitions

Korhan Bircan

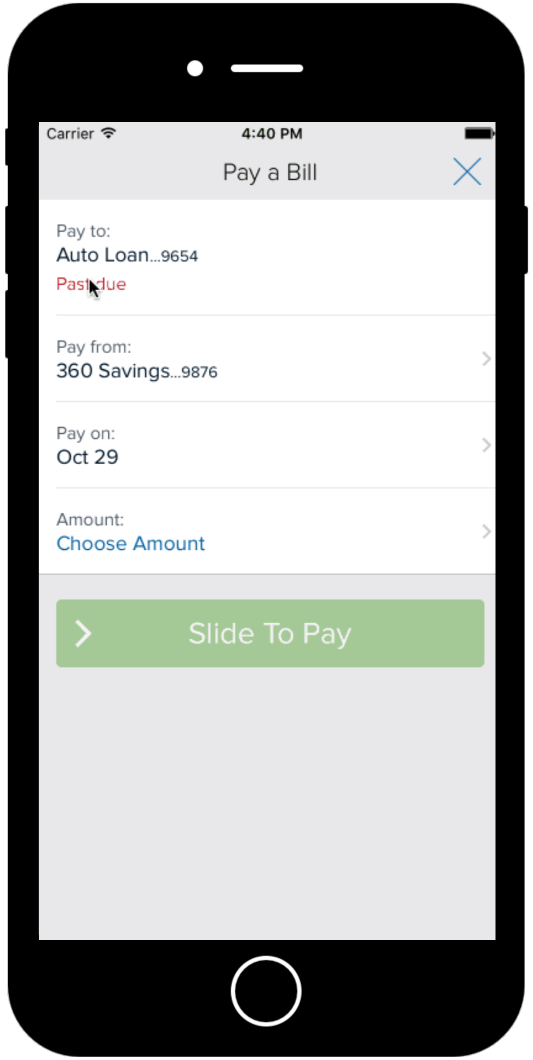
iOS Summit 2016

Overview

- What?
- Why?
- How?

Demo – Shrink Transition

```
let animationController = ShrinkAnimationController()
navigationController.transitioningDelegate = animationController
```



Carrier 4:40 PM

Pay a Bill

Pay to:
Auto Loan...9654
Past due

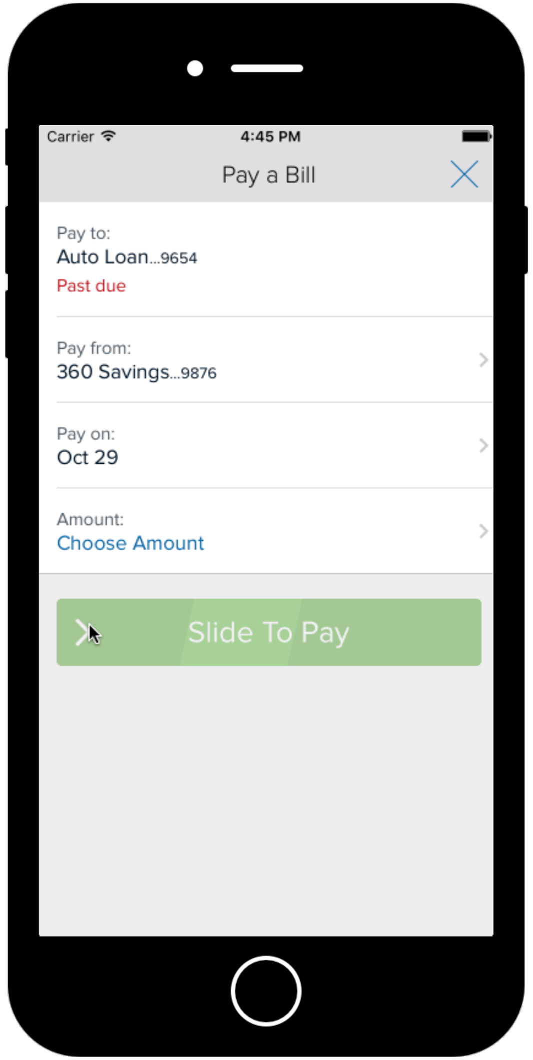
Pay from:
360 Savings...9876

Pay on:
Oct 29

Amount:
Choose Amount

Demo – Circular Transition

```
let animationController = CircularAnimationController()
navigationController.transitioningDelegate = animationController
```



Carrier 4:45 PM

Pay a Bill

Pay to:
Auto Loan...9654
Past due

Pay from:
360 Savings...9876

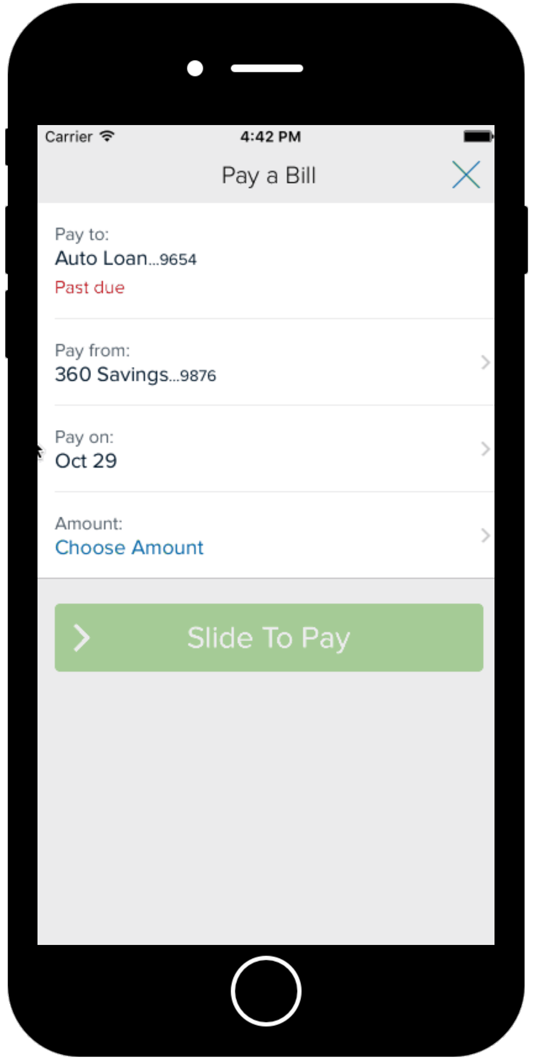
Pay on:
Oct 29

Amount:
Choose Amount

Slide To Pay

Demo – Fade Transition

```
let animationController = FadeAnimationController()
navigationController.transitioningDelegate = animationController
```



Carrier 4:42 PM

Pay a Bill

Pay to:
Auto Loan...9654
Past due

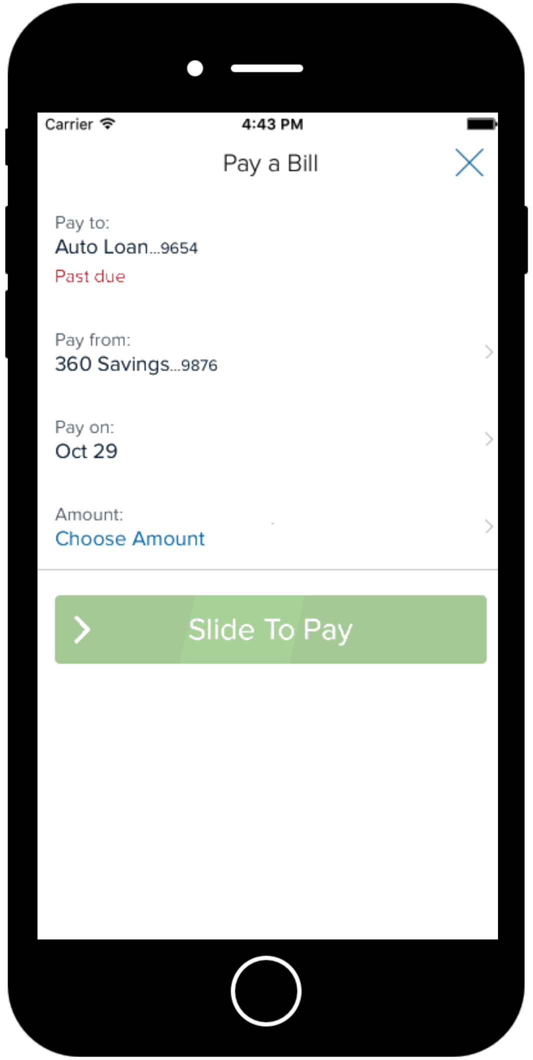
Pay from:
360 Savings...9876

Pay on:
Oct 29

Amount:
Choose Amount

Demo – Pop Transition

```
let animationController = PopAnimationController()
navigationController.transitioningDelegate = animationController
```



Carrier 4:43 PM

Pay a Bill

Pay to:
Auto Loan...9654
Past due

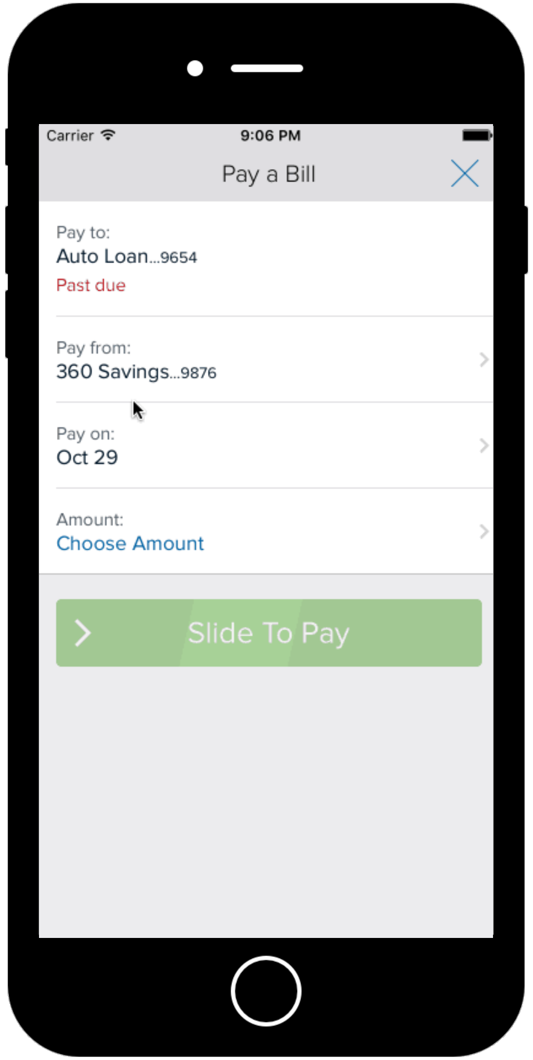
Pay from:
360 Savings...9876

Pay on:
Oct 29

Amount:
Choose Amount

Demo – Sideways Transition

```
let animationController = SidewaysAnimationController()
navigationController.transitioningDelegate = animationController
```



Pay to:
Auto Loan...9654
Past due

Pay from:
360 Savings...9876

Pay on:
Oct 29

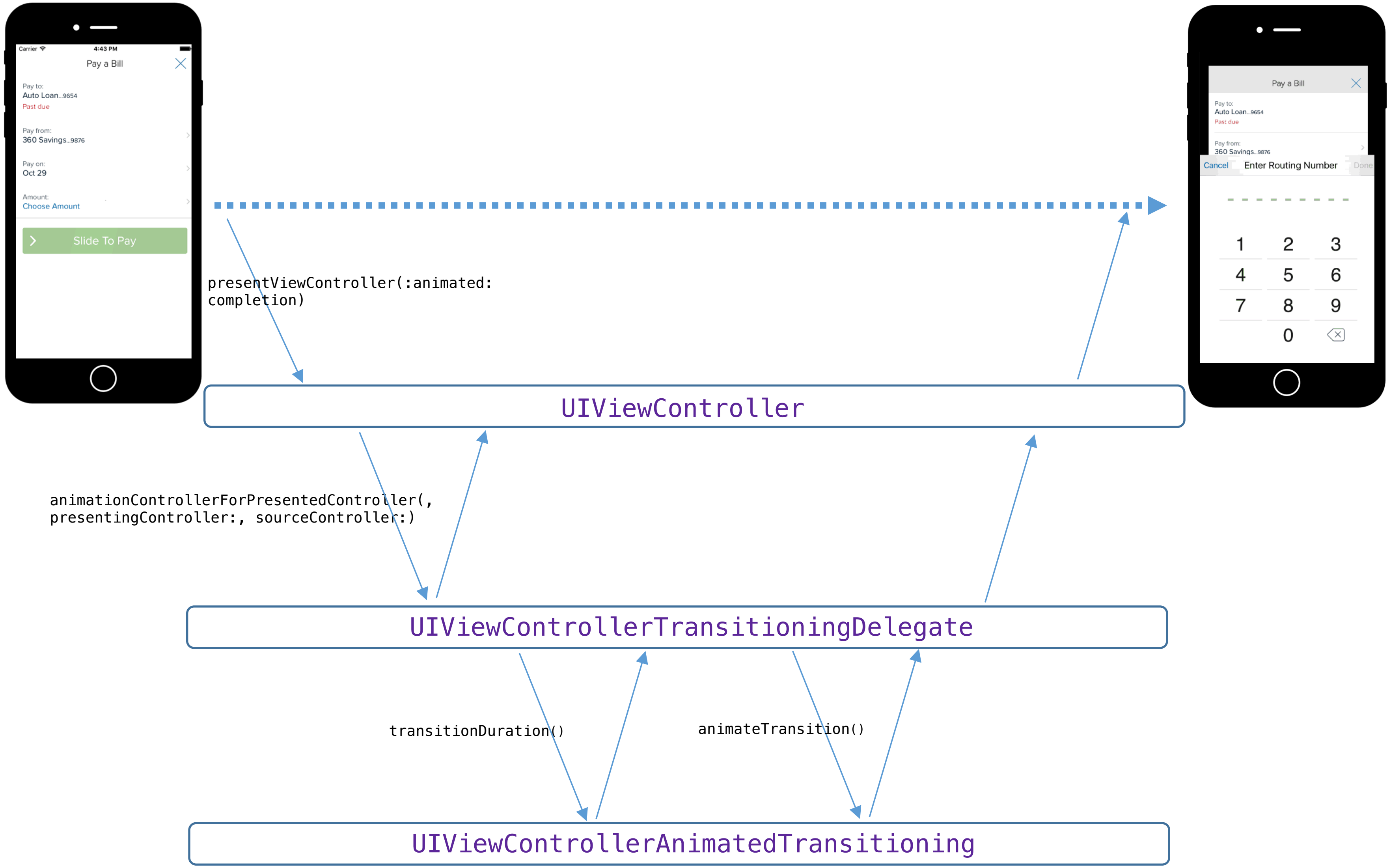
Amount:
Choose Amount

Custom view controller transitions

Which transitions can be customized?

- Presentations and dismissals
- UITabBarController
- UINavigationController
- UICollectionViewController layout-to-layout transitions

Transition Animation Lifecycle



presentViewController(:animated: completion)

UIViewController

animationControllerForPresentedController(, presentingController:, sourceController:)

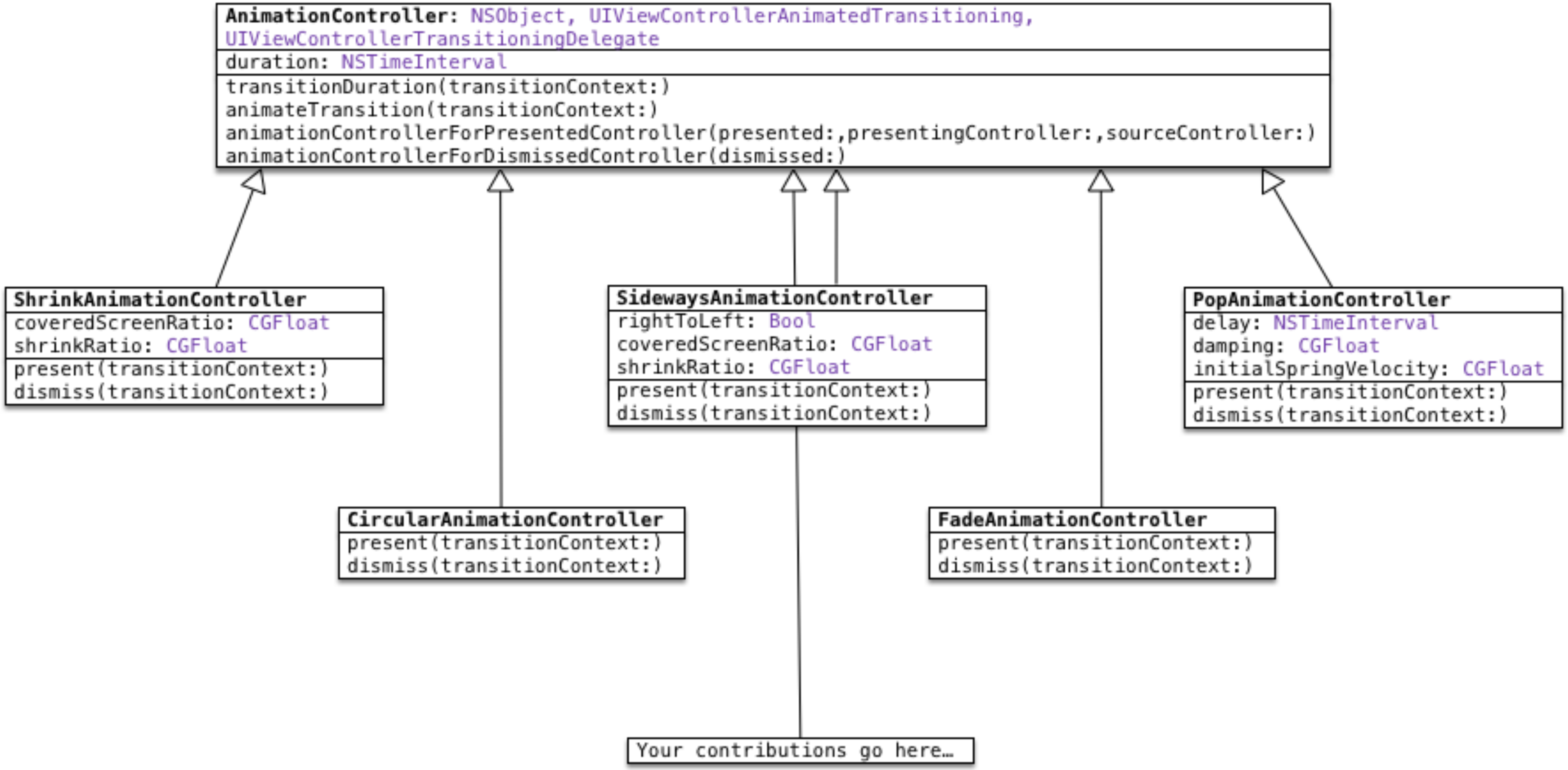
UIViewControllerTransitioningDelegate

transitionDuration()

animateTransition()

UIViewControllerAnimatedTransitioning

C1TransitionAnimations



AnimationController: NSObject, UIViewControllerAnimatedTransitioning, UIViewControllerTransitioningDelegate

duration: NSTimeInterval

transitionDuration(transitionContext:)
animateTransition(transitionContext:)
animationControllerForPresentedController(presented:,presentingController:,sourceController:)
animationControllerForDismissedController(dismissed:)

ShrinkAnimationController

coveredScreenRatio: CGFloat
shrinkRatio: CGFloat

present(transitionContext:)
dismiss(transitionContext:)

SidewaysAnimationController

rightToLeft: Bool
coveredScreenRatio: CGFloat
shrinkRatio: CGFloat

present(transitionContext:)
dismiss(transitionContext:)

PopAnimationController

delay: NSTimeInterval
damping: CGFloat
initialSpringVelocity: CGFloat

present(transitionContext:)
dismiss(transitionContext:)

CircularAnimationController

present(transitionContext:)
dismiss(transitionContext:)

FadeAnimationController

present(transitionContext:)
dismiss(transitionContext:)

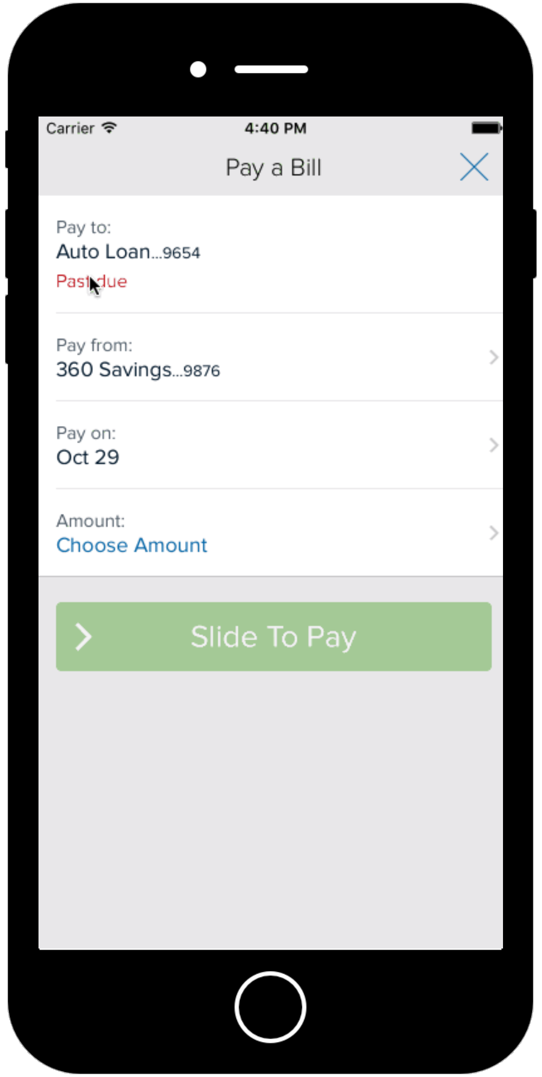
Your contributions go here…

Animation Toolkit

- Block based UIView animation API
- Spring animations
- Key-frame animations
- UIKit Dynamics

Block based UIView animation API

```
UIView.animateWithDuration(duration,
           animations: {
               // Shrink the background view.
               if let screenshotView = self.screenshotView {
                   screenshotView.transform =
                   CGAffineTransformMakeScale(self.shrinkRatio, self.shrinkRatio)
               }
               // Animate numPadView upwards.
               numPadView.frame.origin.y -=
               self.coveredScreenRatio * (UIScreen.mainScreen().bounds.height)
           }, completion: { _ in
               transitionContext.completeTransition(true)
       })
```



Pay a Bill

Pay to:
Auto Loan...9654
Past due

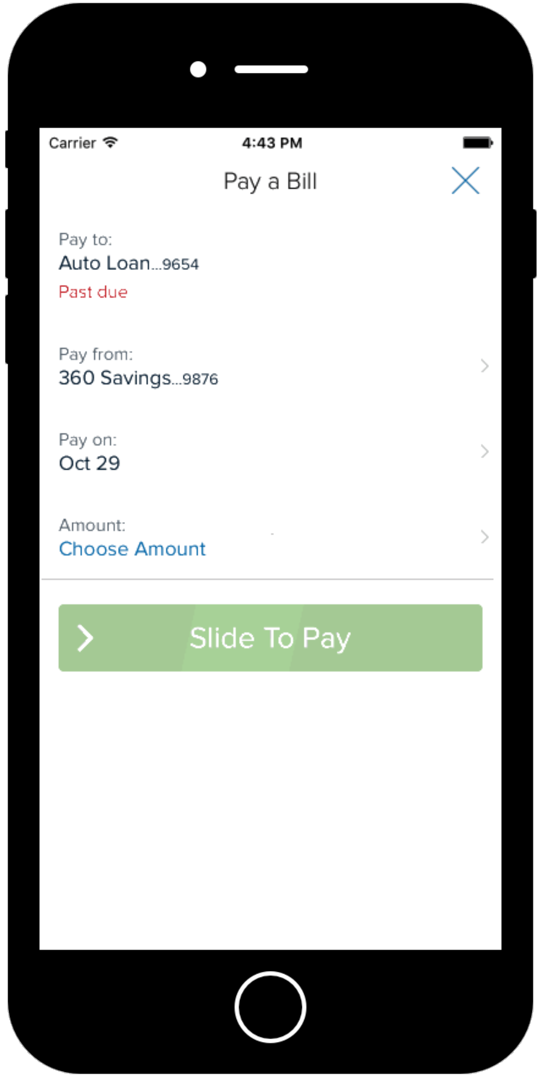
Pay from:
360 Savings...9876

Pay on:
Oct 29

Amount:
Choose Amount

Spring animation API

```
UIView.animateWithDuration(duration,
          delay:delay,
          usingSpringWithDamping: damping,
          initialSpringVelocity: initialSpringVelocity,
          options: [],
          animations: {
          numPadView.transform = CGAffineTransformIdentity
          }, completion: { _ in
              transitionContext.completeTransition(true)
      })
```



Pay to:
Auto Loan...9654
Past due

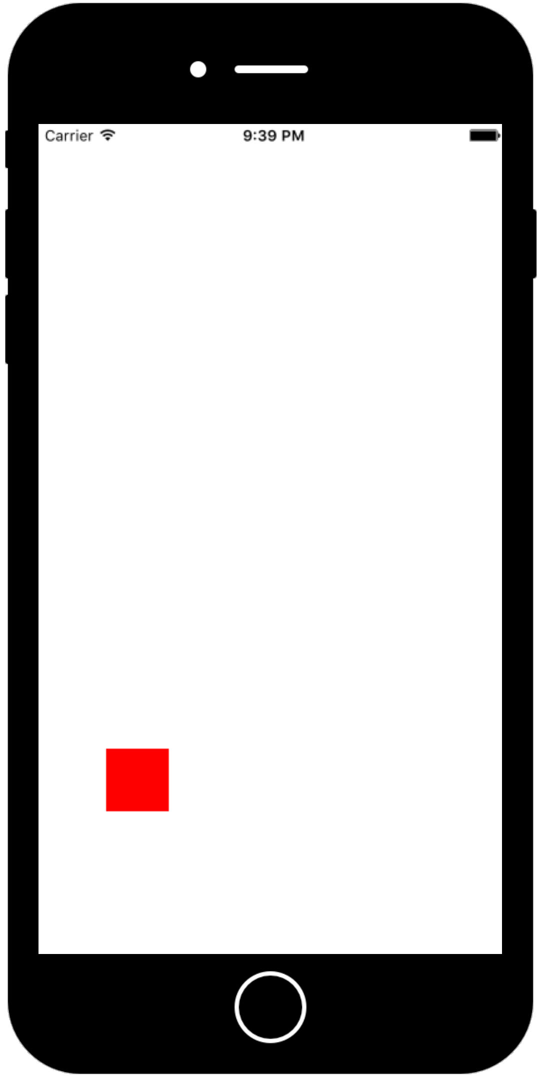
Pay from:
360 Savings...9876

Pay on:
Oct 29

Amount:
Choose Amount

Keyframe animation API

```
UIView.animateKeyframesWithDuration(4, delay: 0.0, options: [.Repeat], animations: {
            UIView.addKeyframeWithRelativeStartTime(0.0, relativeDuration: 0.25, animations: {
                square.center.x += 250.0
            })
            UIView.addKeyframeWithRelativeStartTime(0.25, relativeDuration: 0.25) {
                square.center.y -= 400.0
                var transfrom = CGAffineTransformIdentity
                transfrom = CGAffineTransformRotate(transfrom, CGFloat(-2*M_PI_4))
                transfrom = CGAffineTransformScale(transfrom, 1.5, 1.5)
                square.transform = transfrom
            }
            UIView.addKeyframeWithRelativeStartTime(0.5, relativeDuration: 0.25, animations: {
                square.center.x -= 250.0
                square.transform = CGAffineTransformMakeRotation(CGFloat(-3*M_PI_4))
                square.backgroundColor = UIColor.yellowColor()
            })
            UIView.addKeyframeWithRelativeStartTime(0.75, relativeDuration: 0.25) {
                square.center.y += 400.0
                square.transform = CGAffineTransformMakeRotation(CGFloat(-4*M_PI_4))
                square.backgroundColor = UIColor.redColor()
            }
        }, completion: nil)
```



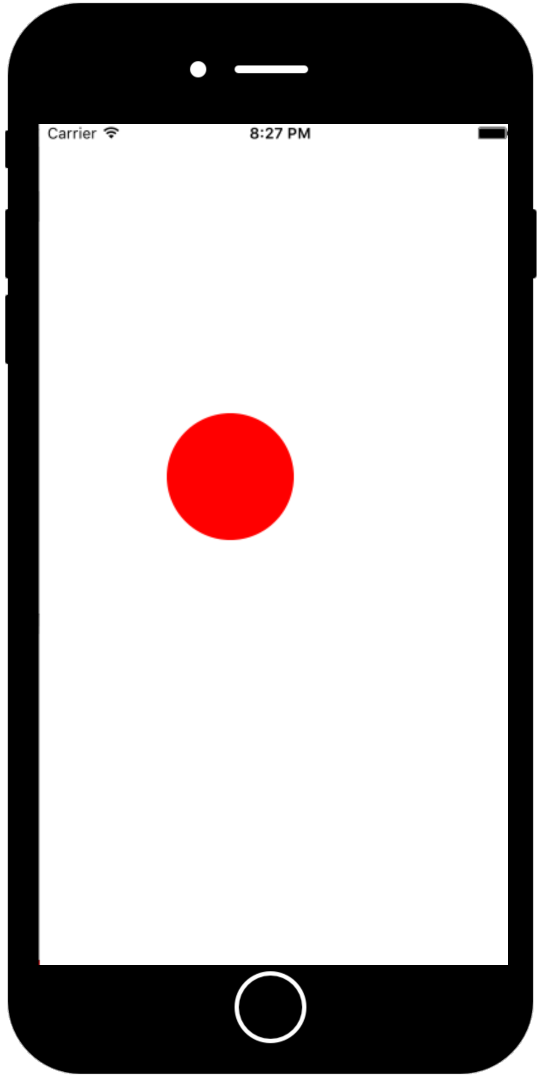
UIKit Dynamics

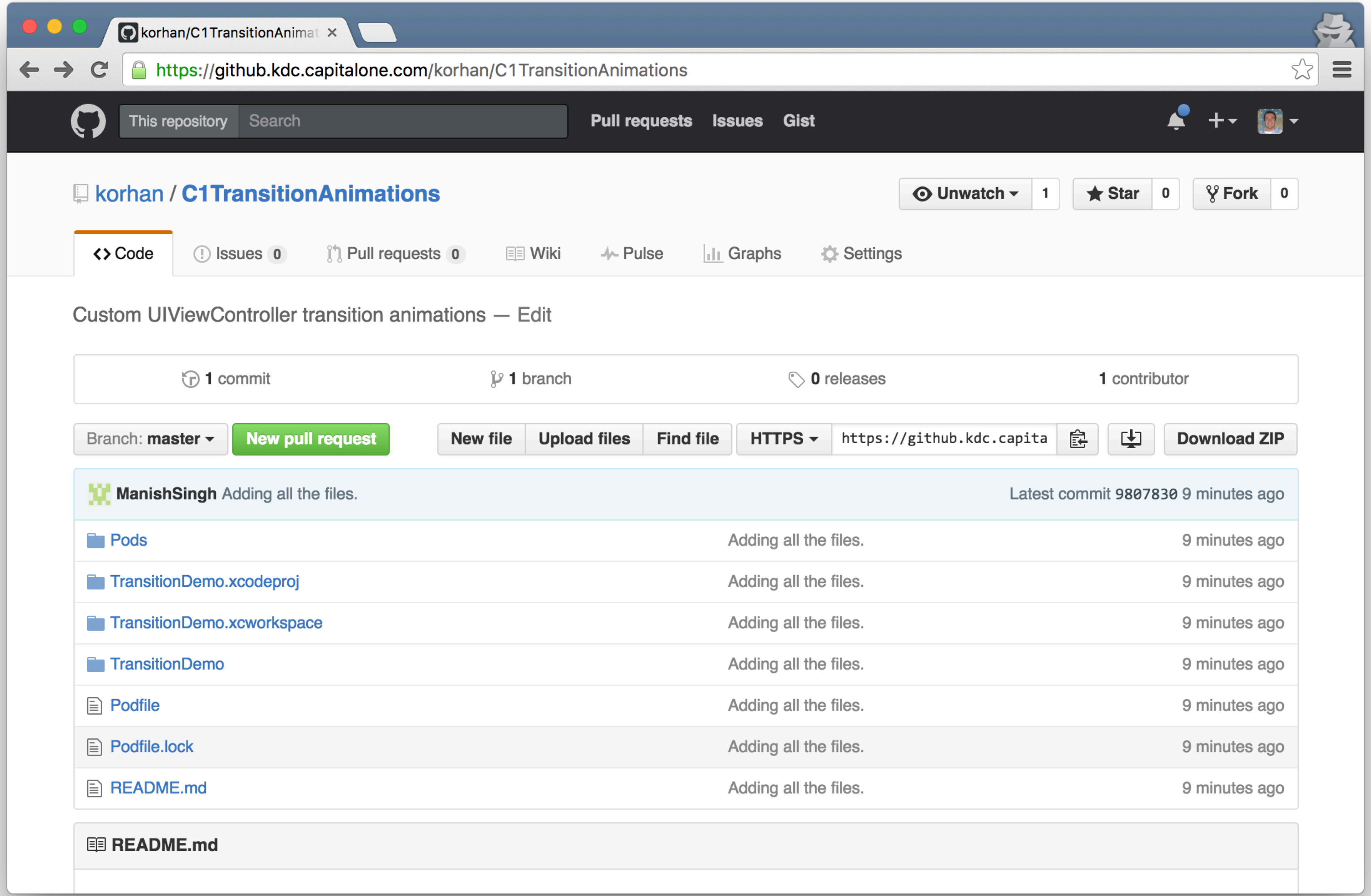
```
let circle = UIView(frame: CGRect(x: 100, y: 100, width: 100, height: 100))
circle.backgroundColor = UIColor.redColor()
circle.layer.cornerRadius = 50
view.addSubview(circle)

animator = UIDynamicAnimator(referenceView: view)
gravity = UIGravityBehavior(items: [circle])
animator.addBehavior(gravity)

collision = UICollisionBehavior(items: [circle])
collision.translatesReferenceBoundsIntoBoundary = true
animator.addBehavior(collision)

let itemBehaviour = UIDynamicItemBehavior(items: [circle])
itemBehaviour.elasticity = 0.9
animator.addBehavior(itemBehaviour)
```





korhan / C1TransitionAnimations

Custom UIViewController transition animations — Edit

1 commit | 1 branch | 0 releases | 1 contributor

ManishSingh Adding all the files. Latest commit 9807830 9 minutes ago

Pods	Adding all the files.	9 minutes ago
TransitionDemo.xcodeproj	Adding all the files.	9 minutes ago
TransitionDemo.xcworkspace	Adding all the files.	9 minutes ago
TransitionDemo	Adding all the files.	9 minutes ago
Podfile	Adding all the files.	9 minutes ago
Podfile.lock	Adding all the files.	9 minutes ago
README.md	Adding all the files.	9 minutes ago

README.md