

Анимации и жесты

Артур Сардарян
Руководитель команды
разработки Юлы,
проект VK

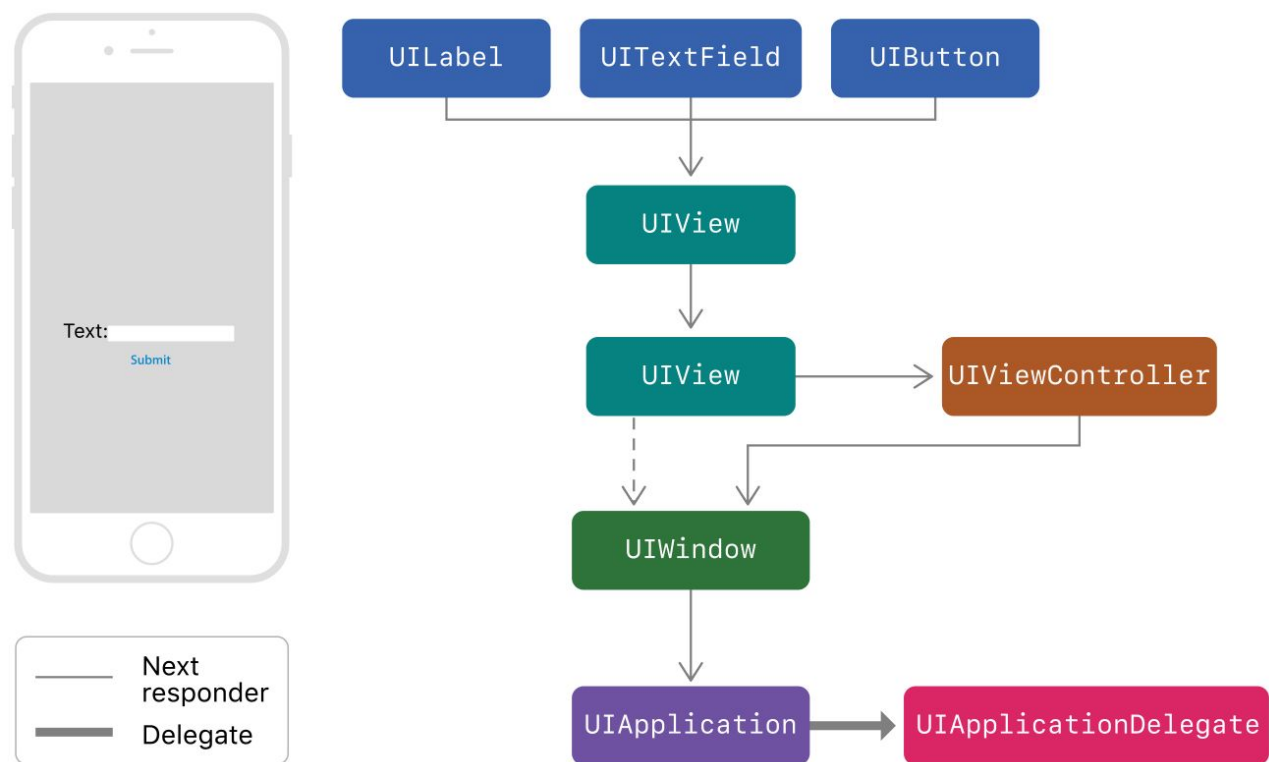


О чем будем говорить

- Responder Chain
- hitTest
- Распознавание жестов
- Анимации
 - UIView.animate
 - draw
 - CAAAnimation

Responder chain

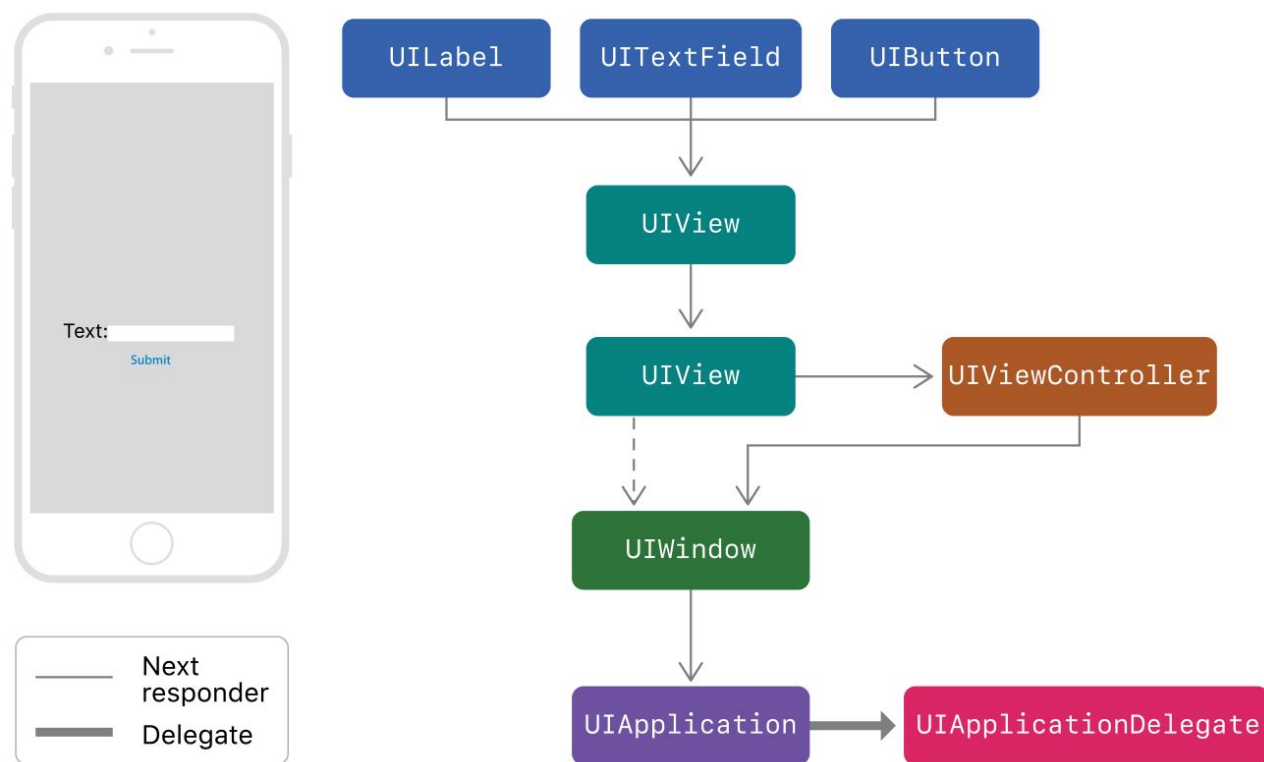
Цепочка пробрасывания события



Responder chain

- Система создает событие
- Оно поступает к UIApplication
- Через sendEvent пробрасывается к window
- UIWindow запускает проход по дереву через hitTest

Документация [IUI](#)



Реализация hitTest

Стандартная реализация использует `point(inside:with:)`- проверяет содержит ли получатель эту точку

Документация [тут](#)

Declaration

```
func hitTest(_ point: CGPoint,  
             with event: UIEvent?) -> UIView?
```

Реализация point

Возвращает `true`, если получатель содержит указанную точку, иначе - `false`

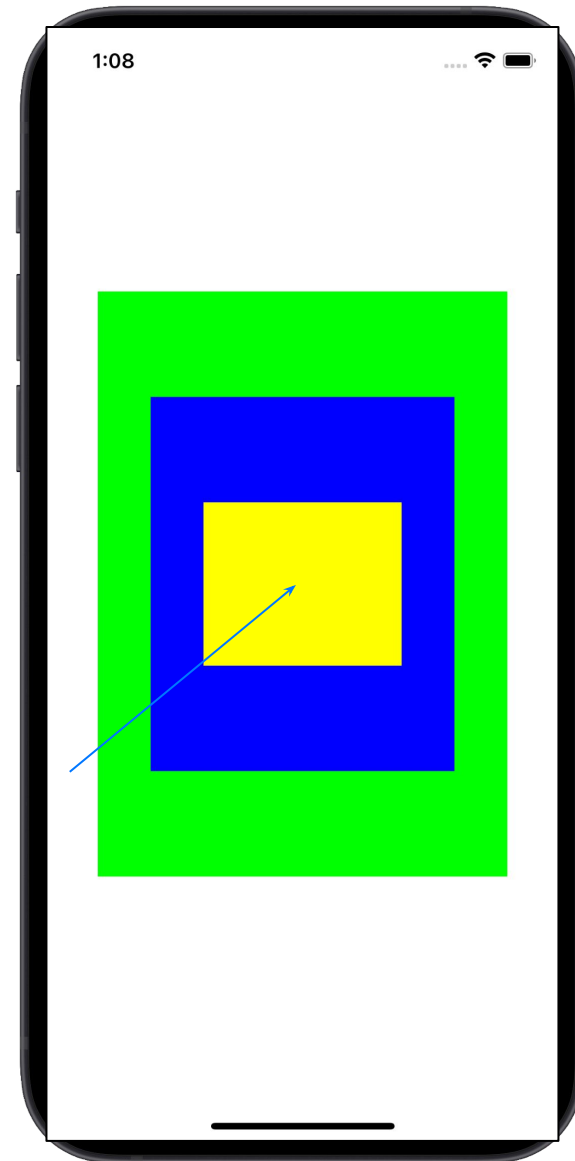
Документация [ТУТ](#)

Declaration

```
func point(inside point: CGPoint,  
           with event: UIEvent?) -> Bool
```

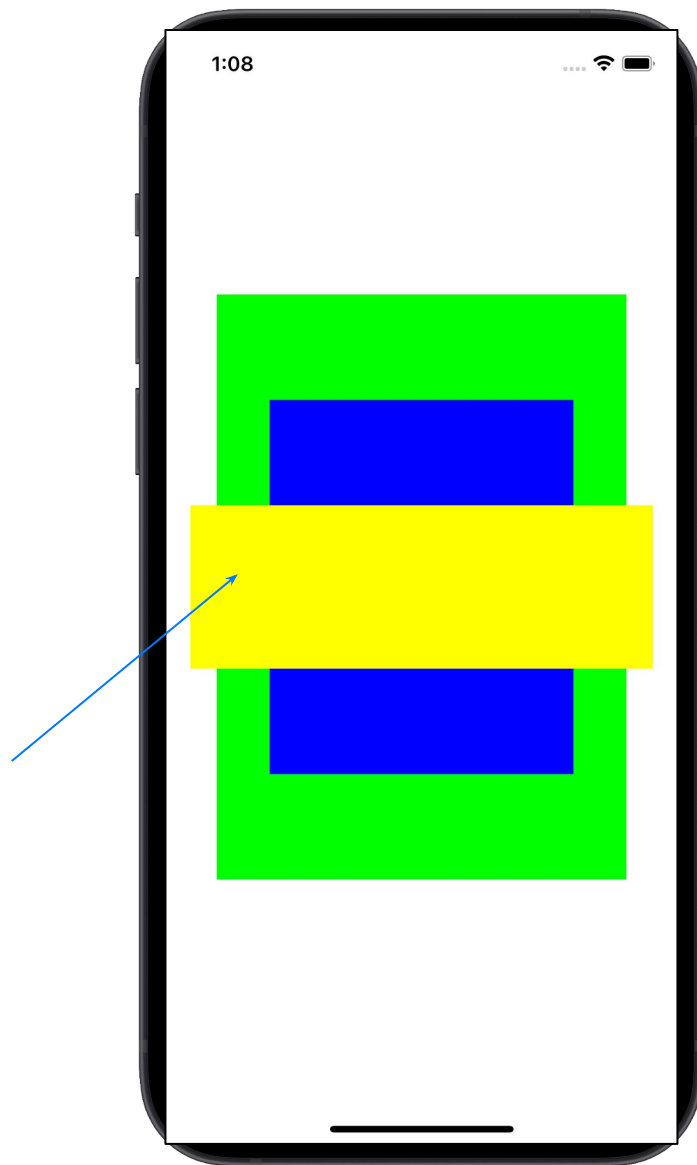
Пример

Кто responder?



Пример

Кто responder?



Coding

Рассмотрим проект HitTest

На заметку

Если свойство `isUserInteractionEnabled` будет `false`, дочерние вью не будут опрашиваться и `hitTest` вернет `nil`

Лайфхаки

- Увеличение области нажатия вьюхи
- Кастомный прогон по subviews в методе `point (inside:with:)`

Распознавание жестов

Какие варианты есть?

Распознавание жестов вручную

- `func touchesMoved(Set<UITouch>, with: UIEvent?)`
- `func touchesEnded(Set<UITouch>, with: UIEvent?)`
- `func touchesCancelled(Set<UITouch>, with: UIEvent?)`
- `func touchesBegan(Set<UITouch>, with: UIEvent?)`

Gesture Recognizers

Какие бывают?

Gesture Recognizers

- `UITapGestureRecognizer`
- `UIPinchGestureRecognizer`
- `UIRotationGestureRecognizer`
- `UISwipeGestureRecognizer`
- `UIPanGestureRecognizer`
- `UIScreenEdgePanGestureRecognizer`
- `UILongPressGestureRecognizer`

Добавление GestureRecognizer

```
let tapGestureRecognizer = UITapGestureRecognizer(target: self, action:  
#selector(handleTap))  
view?.addGestureRecognizer(tapGestureRecognizer)
```


Дискретные GestureRecognizers

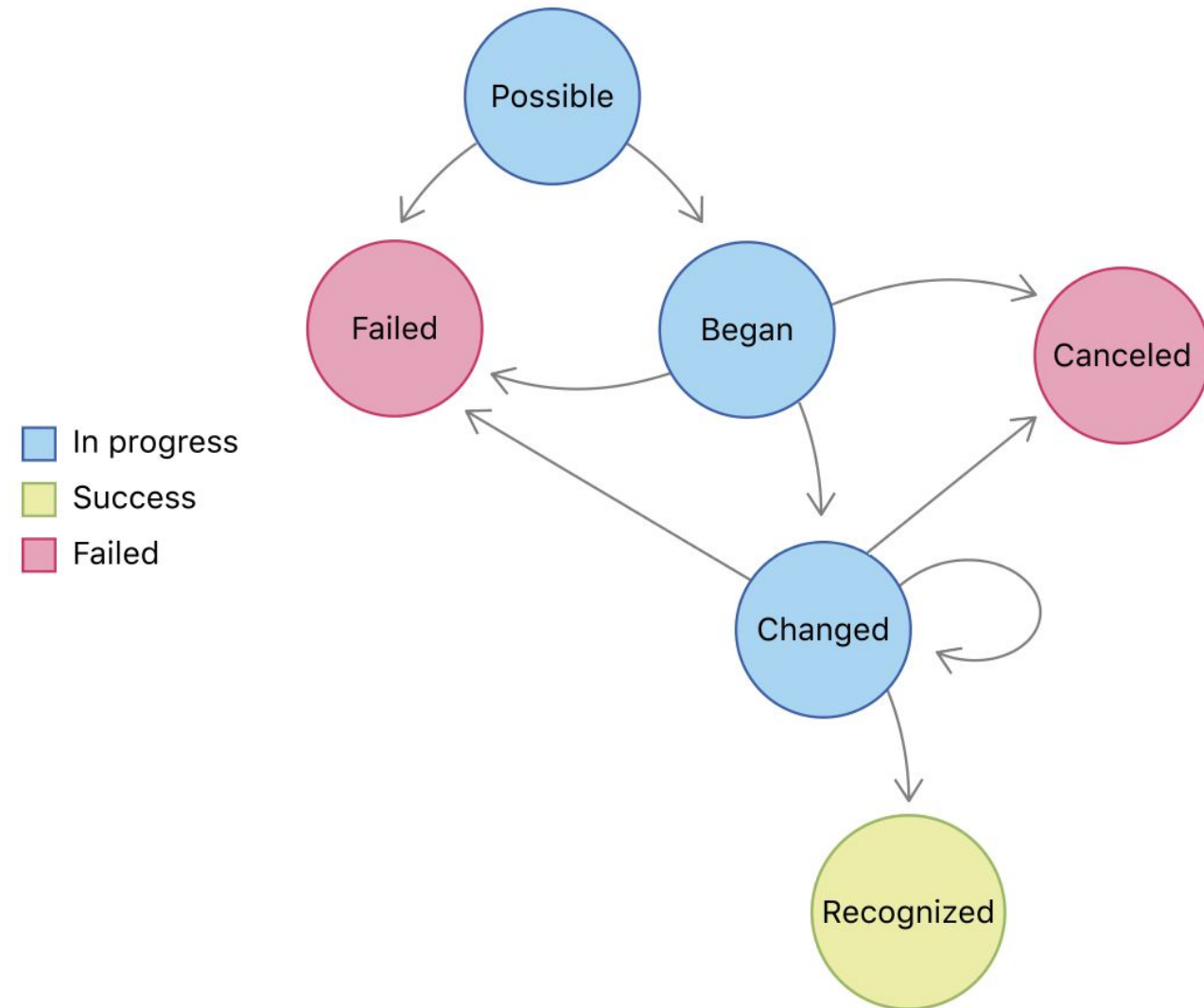
- UITapGestureRecognizer
- UISwipeGestureRecognizer
- UILongPressGestureRecognizer

Непрерывные GestureRecognizers

- `UIPinchGestureRecognizer`
- `UIRotationGestureRecognizer`
- `UIPanGestureRecognizer`
- `UIScreenEdgePanGestureRecognizer`

UIGestureRecognizer.State

- possible — рекогнайзер готов к работе
- began — начался распознаваемый жест
- changed — изменение состояния, например, движение пальца при скролле
- recognized (он же ended) — жест закончился
- cancelled — аналог touchesCancelled
- failed — жест не был распознан (например ожидали скролл двумя пальцами, но экрана коснулся лишь один)



Coding

Рассмотрим проект GestureRecognizers

Перерыв

 образование

Анимация – это?

Анимация — это изменение свойств
графического объекта с течением
времени

Анимации

Как их делать?

Анимации

- `UIView.animate`
- `CAAnimation`
- `draw(_ rect)`

UIView.animate

Declaration

```
class func animate(withDuration duration: TimeInterval,  
                    delay: TimeInterval,  
                    options: UIView.AnimationOptions = [],  
                    animations: @escaping () -> Void,  
                    completion: ((Bool) -> Void)? = nil)
```

AnimationOptions

- repeat - анимация будет повторяться
- autoreverse - после окончания анимации она будет воспроизведена в обратном направлении
- linear - анимация будет проходить равномерно
- curveEaseIn - сначала анимация будет проходить медленно, потом ускорится
- curveEaseOut - сначала анимация будет проходить ускоренно, а в конце замедлится
- curveEaseInOut - и сначала и в конце анимация будет проходить медленнее чем в середине

UIView & CALayer

UIView:

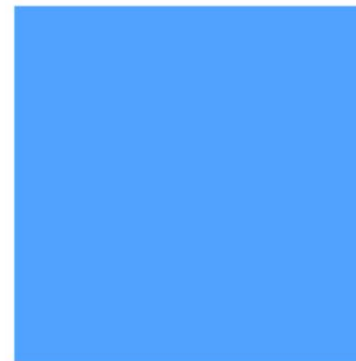
- Содержит CALayer
- Обрабатывает жесты пользователя

CALayer:

- Содержит информацию о геометрии своего контента
- Отображает графический контент
- Позволяет анимировать графический контент

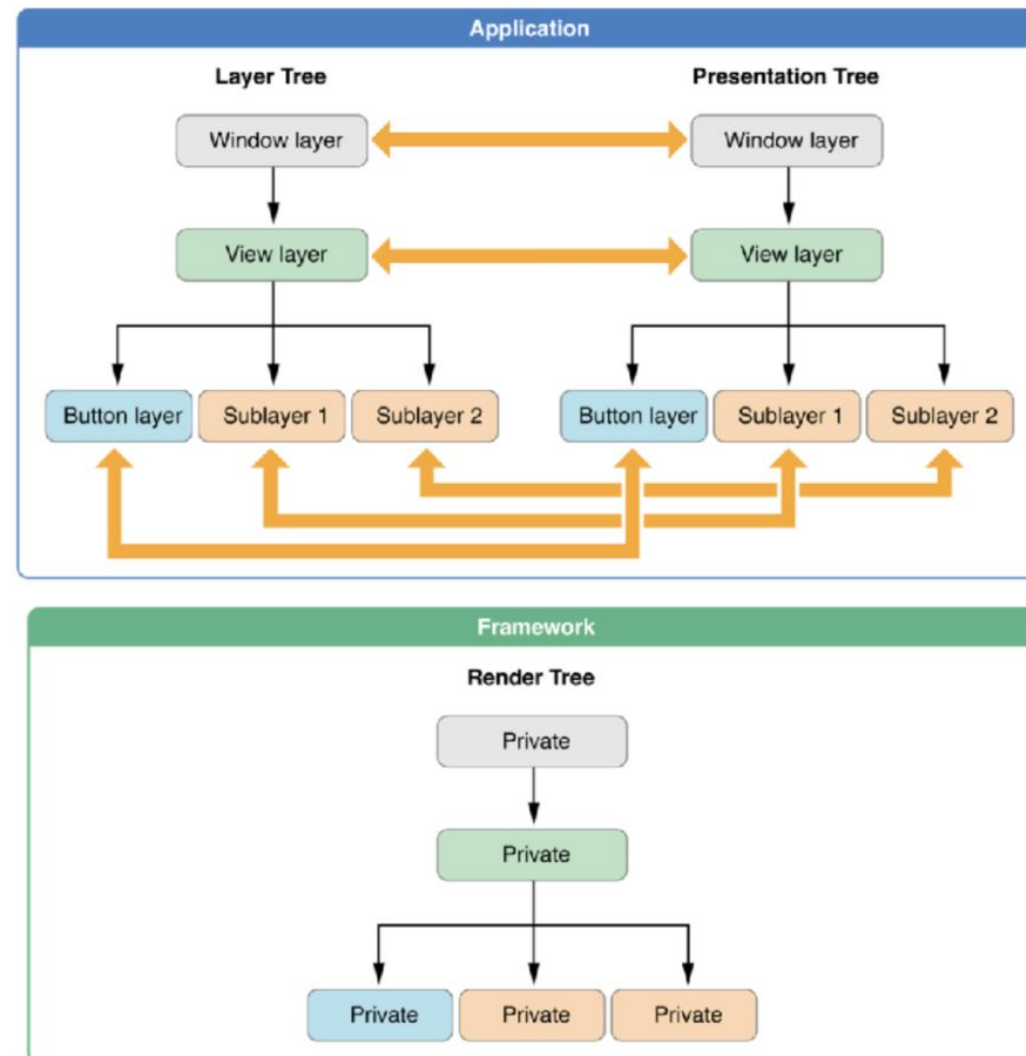
Анимирование CALayer

```
let animation = CABasicAnimation(keyPath: "opacity")
animation.beginTime = CACurrentMediaTime() + 0.5
animation.fromValue = 0.5
animation.toValue = 0
animation.duration = 0.5
animation.fillMode = kCAFillModeRemoved
layer.add(animation, forKey: nil)
```



Анимирование CALayer

- Перед запуском анимации слой скрывается
- Появляется копия слоя
- Происходит анимация копии слоя
- Копия слоя скрывается
- Появляется исходный слой



Анимирование CALayer

- **Слой переходит в исходное состояние после анимации**
- Исходный слой не меняет состояние. Оно изменяется только в presentation слое.

●

Если нужно спрятать исходный слой

Изменить состояние слоя
после добавления анимации:

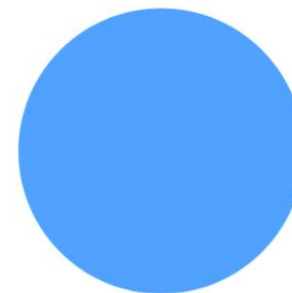
```
let animation = CABasicAnimation(keyPath: "opacity")
animation.beginTime = CACurrentMediaTime() + 0.5
animation.fromValue = 0.5
animation.toValue = 0
animation.duration = 0.5
animation.fillMode = kCAFillModeRemoved
layer.add(animation, forKey: nil)
layer.opacity = 0
```

Установить свойство
isRemovedOnCompletion:

```
let animation = CABasicAnimation(keyPath: "opacity")
animation.beginTime = CACurrentMediaTime() + 0.5
animation.fromValue = 0.5
animation.toValue = 0
animation.duration = 0.5
animation.fillMode = kCAFillModeRemoved
animation.isRemovedOnCompletion = false
layer.add(animation, forKey: nil)
```


Групповые анимации

```
let animationsGroup = CAAAnimationGroup()  
animationsGroup.duration = 0.5  
animationsGroup.animations = [anim1, anim2]  
layer.add(animationsGroup, forKey: nil)
```



Что можно анимировать?

UIView

- backgroundColor
- bounds
- frame
- center
- transform
- zPosition
- alpha

CALayer

- backgroundColor
- bounds
- frame
- opacity
- center
- transform
- zPosition
- anchorPoint
- backgroundFilters
- compositingFilter
- filters
- borderColor
- borderWidth
- contents
- contentsRect
- cornerRadius
- doubleSided
- mask
- masksToBounds
- opacity
- shadowColor
- shadowRadius
- shadowPath
- shadowOffset
- shadowOpacity
- sublayers
- sublayerTransform

draw(_ rect)

Declaration

```
func draw(_ rect: CGRect)
```

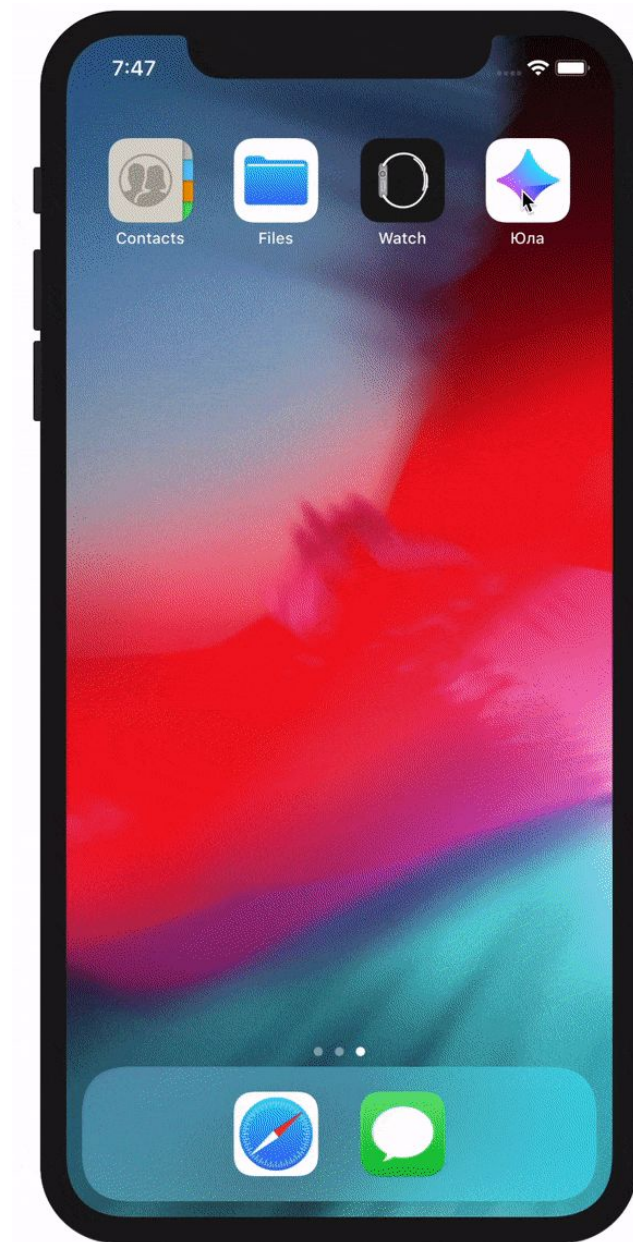
Документация [ТУТ](#)

Coding



Splash Screen

<https://habr.com/en/company/youla/blog/456394/>



QA

 образование