

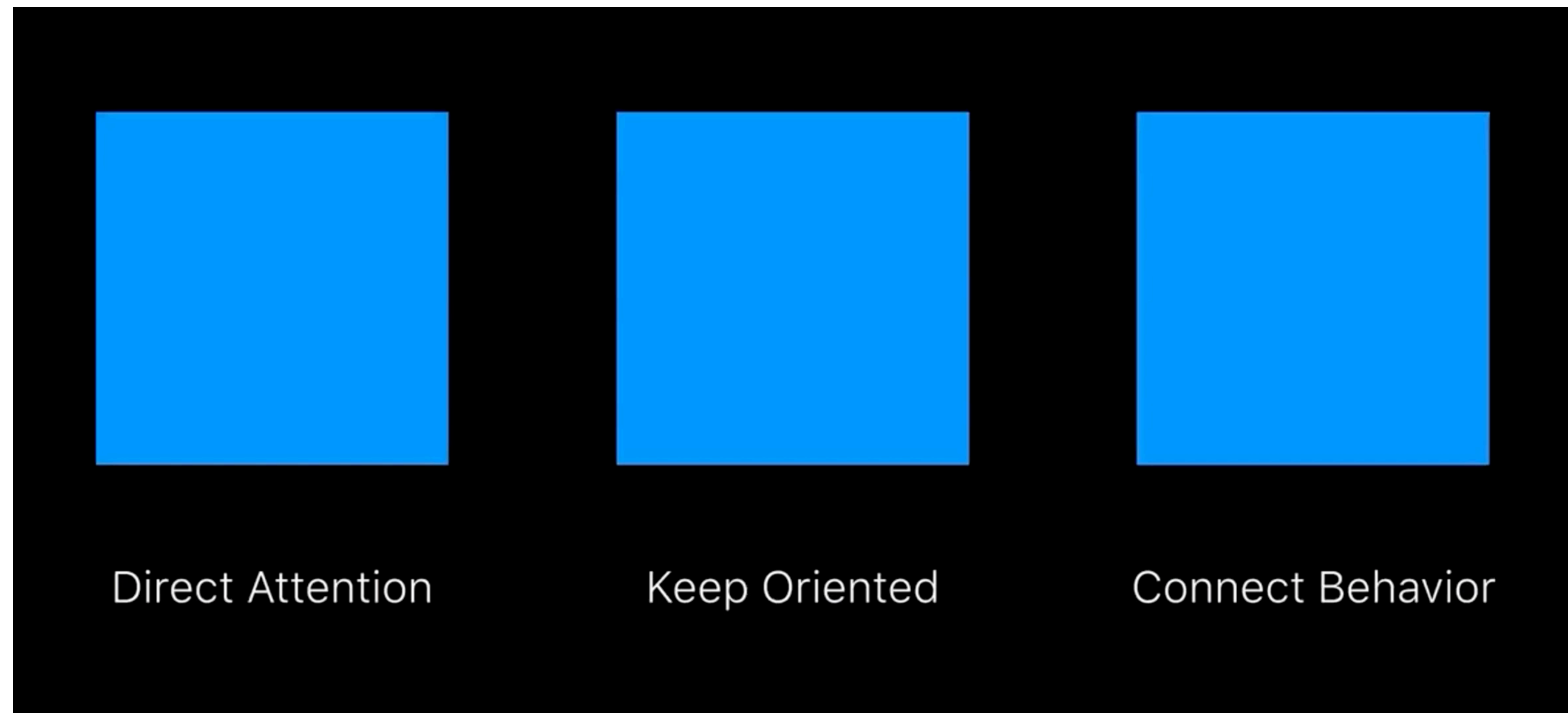
Animations



Why Animate?

Animations should:

- Direct the user's attention
- Keep the user oriented
- Connect user behaviors



What Can Be Animated?

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

Animation Closures

The **UIView** class defines several class methods where you can update the animatable properties

- `animate(withDuration:animations:)`
- `animate(withDuration:animations:completion:)`
- `animate(withDuration:delay:options:animations:completion:)`

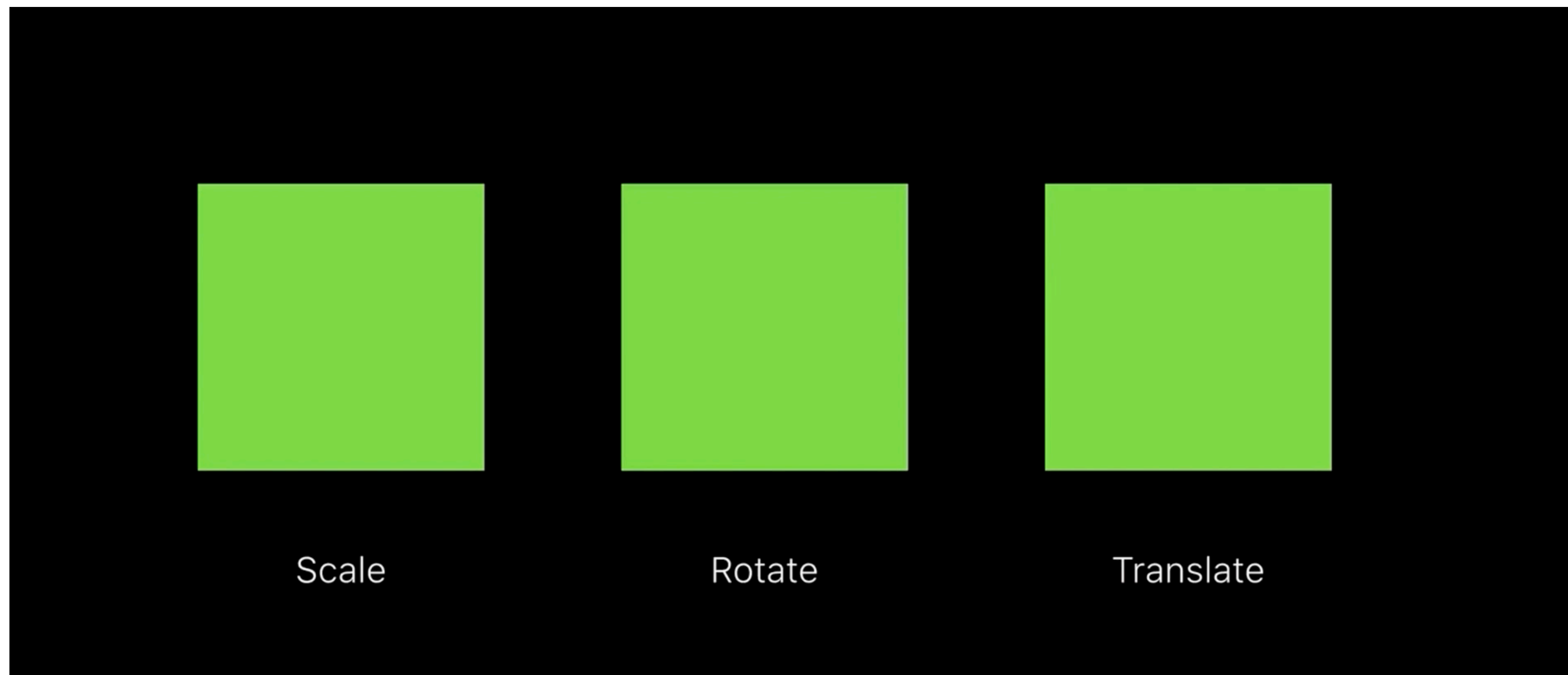
Example of changing view's alpha:

```
UIView.animate(withDuration: 2.0) {  
    aView.alpha = 0.3  
}
```

The Transform Property

Instance of the structure **CGAffineTransform**.

The **transform** property can be used to change the scale, rotate, or move the view without calculating changes to the view's frame.



Transform

Type	Initializer	Parameter Description
Scale	<code>init(scaleX: CGFloat, y: CGFloat)</code>	The factors by which to scale your view.
Rotate	<code>init(rotationAngle : CGFloat)</code>	The angle (in radians) by which to rotate your view. Positive Value = Counterclockwise
Translate	<code>init(translationX: CGFloat, y: CGFloat)</code>	The value by which to move (shift) your view.

```
// Double the height and width of the view  
let scaleTransform = CGAffineTransform(scaleX: 2.0, y: 2.0)
```


Transform in action

```
UIView.animate(withDuration: 2.0) {  
    square.backgroundColor = .orange  
  
    let scaleTransform = CGAffineTransform(scaleX: 2.0, y: 2.0)  
    let rotateTransform = CGAffineTransform(rotationAngle: .pi)  
    let translateTransform = CGAffineTransform(translationX:  
        200, y: 200)  
    let comboTransform =  
        scaleTransform.concatenating(rotateTransform).concatenating  
        (translateTransform)  
  
    square.transform = comboTransform  
}
```



Dynamic Animation

UIDynamicAnimator - an object that provides physics-related animations

UIDynamicBehavior - object that describes animation's behavior

List of all UIDynamicBehavior types:

- **UISnapBehavior**
- **UIPushBehavior**
- **UIFieldBehavior**
- **UICollisionBehavior**
- **UIGravityBehavior**
- **UIAttachmentBehavior**

Dynamic Animation Example



```
import UIKit

class SecondViewController: UIViewController {

    @IBOutlet private weak var appleImageView: UIImageView!

    private lazy var animator = UIDynamicAnimator(referenceView: view)
    private lazy var snapping = UISnapBehavior(item: appleImageView, snapTo: view.center)

    @IBAction func handleApplePan(_ sender: UIPanGestureRecognizer) {
        switch sender.state {
        case .began:
            animator.removeBehavior(snapping)
        case .changed:
            let translation = sender.translation(in: view)
            let center = appleImageView.center

            appleImageView.center = CGPoint(x: center.x + translation.x,
                                             y: center.y + translation.y)

            sender.setTranslation(.zero, in: view)
        case .ended, .failed, .cancelled:
            animator.addBehavior(snapping)
        default:
            break
        }
    }
}
```

Same Using Transform



```
import UIKit

class SecondViewController: UIViewController {

    @IBOutlet private weak var appleImageView: UIImageView!

    @IBAction func handleApplePan(_ sender: UIPanGestureRecognizer) {
        switch sender.state {
        case .changed:
            let translation = sender.translation(in: view)
            appleImageView.transform = CGAffineTransform(translationX: translation.x,
                                                                y: translation.y)

        case .ended, .failed, .cancelled:
            UIView.animate(withDuration: 0.3,
                           delay: 0,
                           usingSpringWithDamping: 0.5,
                           initialSpringVelocity: 0.5,
                           animations: {
                                self.appleImageView.transform = .identity
                            })
        default:
            break
        }
    }
}
```