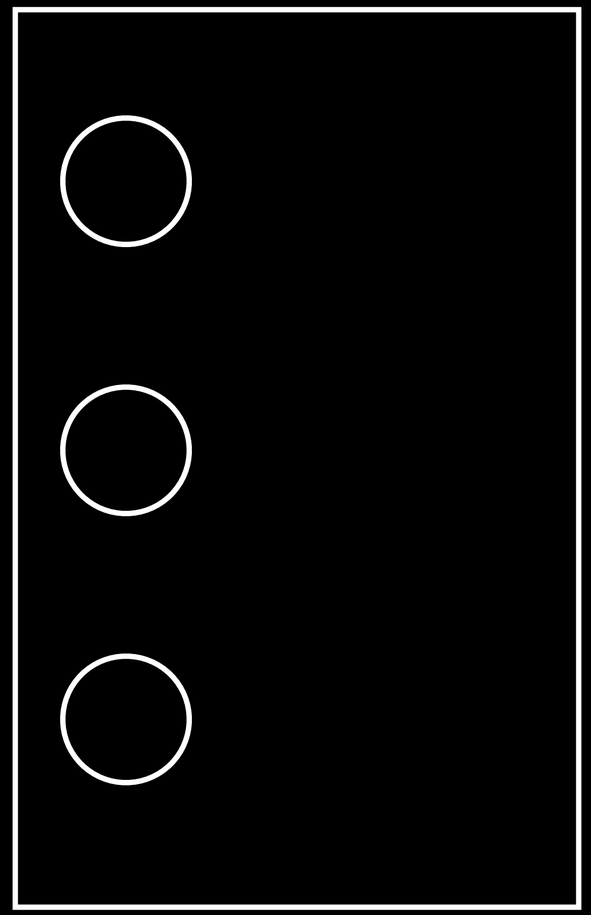
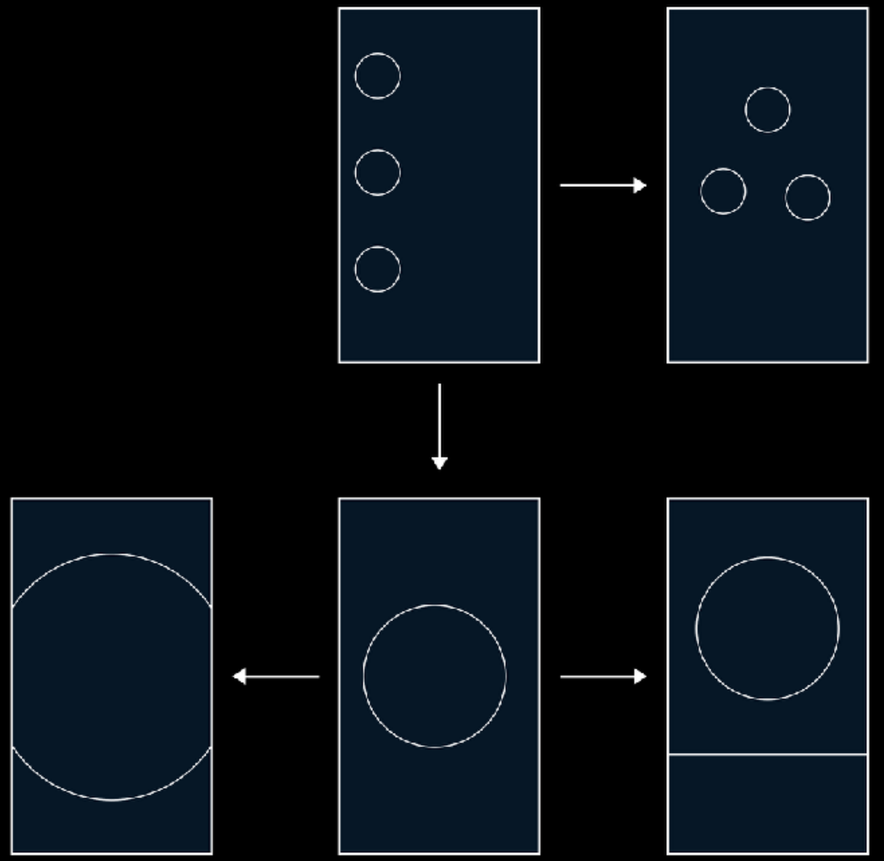
Unit 5—Lesson 3: Practical Animation

Animations



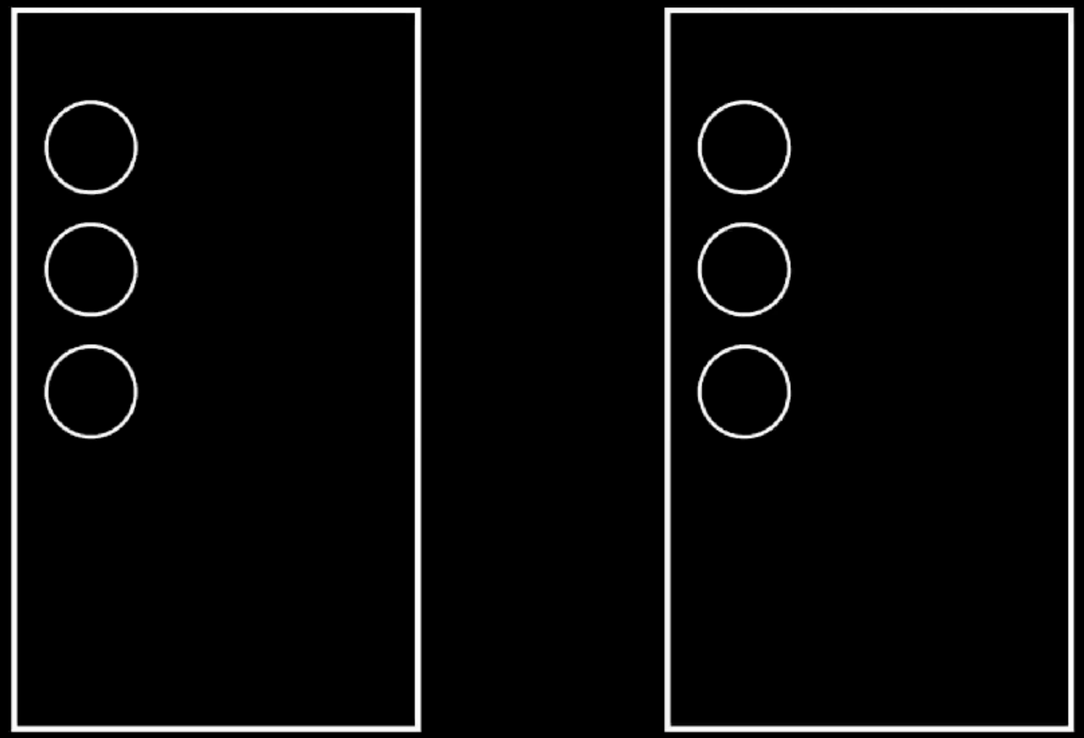
Animations

How do the pieces fit together?



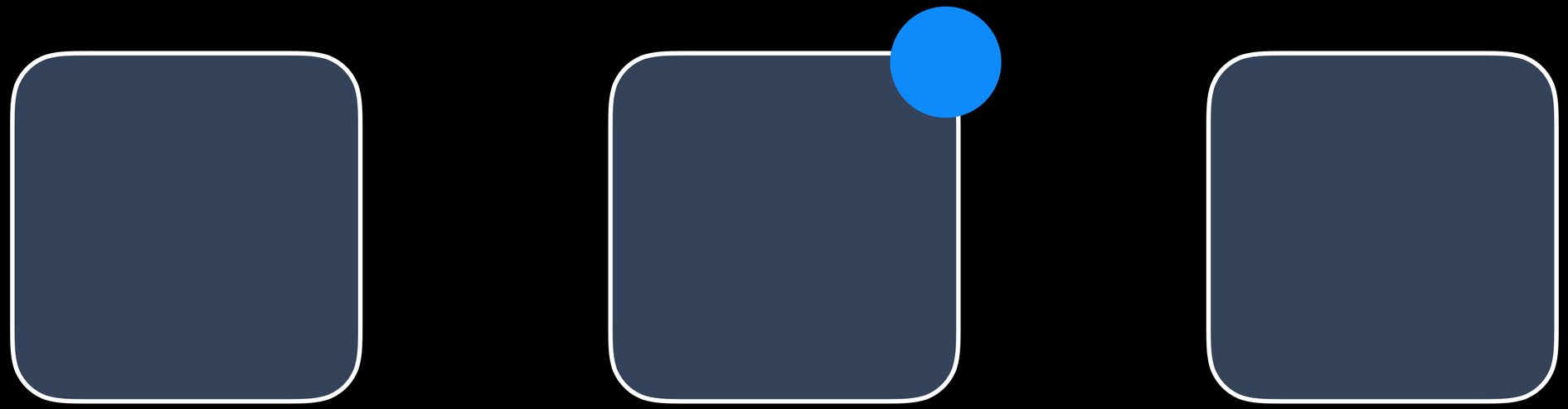
Animations

Conveying an app’s personality



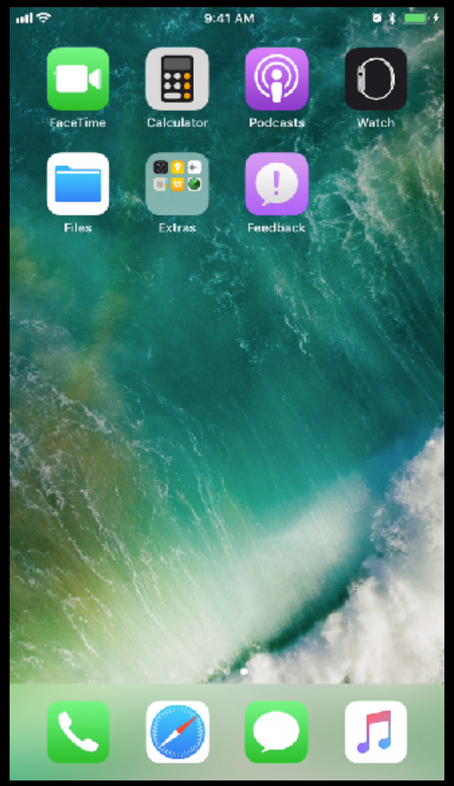
Why animate?

Direct the user’s attention



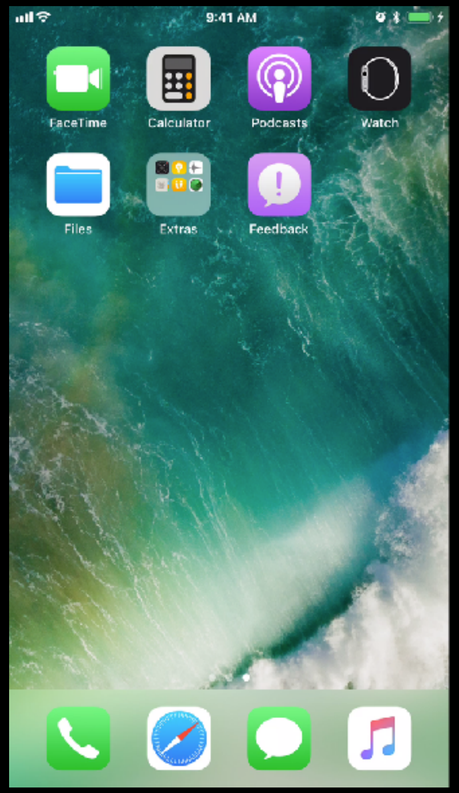
Why animate?

Keep the user oriented



Why animate?

Connect user behaviors



What can be animated?

`UIView`

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

UIView animation methods

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

Animation closures

animate(withDuration:animations:completion:)

```
UIView.animate(withDuration: 2.0, animations: {
    //animation closure
    viewA.alpha = 0.0
}) { (_: Bool) in
    //completion closure
    UIView.animate(withDuration: 2.0, animations: {
        //second animation closure
        viewB.alpha = 1.0
    })
}
```

Animation closures

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

```
UIView.animate(withDuration: 2.0, delay: 5.0, options: [.repeat], animations: {
    aView.center = CGPoint(x: aView.center.x + 10, y: aView.center.y)
}, completion: nil)
```

The transform property

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

The transform property

Identity

```
aView.transform = CGAffineTransform.identity
```

The transform property

Combining transform instances

```
let scaleTransform = CGAffineTransform(scaleX: 2.0, y: 2.0)
let rotateTransform = CGAffineTransform(rotationAngle: .pi)
let combinedTransform = scaleTransform.concatenating(rotateTransform)
```

Animation in practice

Use animation and motion effects judiciously

Strive for realism and credibility

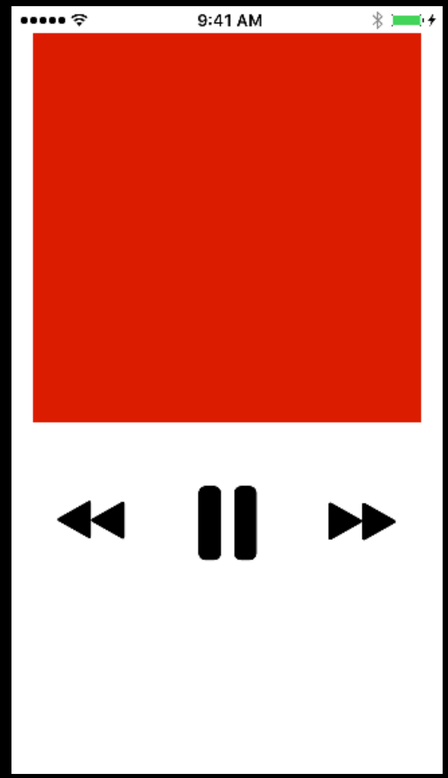
Use consistent animation

Make animations optional

Unit 5—Lesson 3
Practical Animation

Learn how to use the UIView class and closures to add animations that improve the presentation and the functionality of your apps.

Create a wireframe— just the views, without actual functionality—of the Now Playing screen in the Music app



Unit 5—Lesson 3

Lab: Enter to win a Contest

Better understand when is a good time to use an animation by creating an animation that will help your app be more user intuitive