

Anyone can Animate

(Even if they can't draw)

Andy Dent's
strongly-held opinions
weakly-held pencils

Animation as Life?

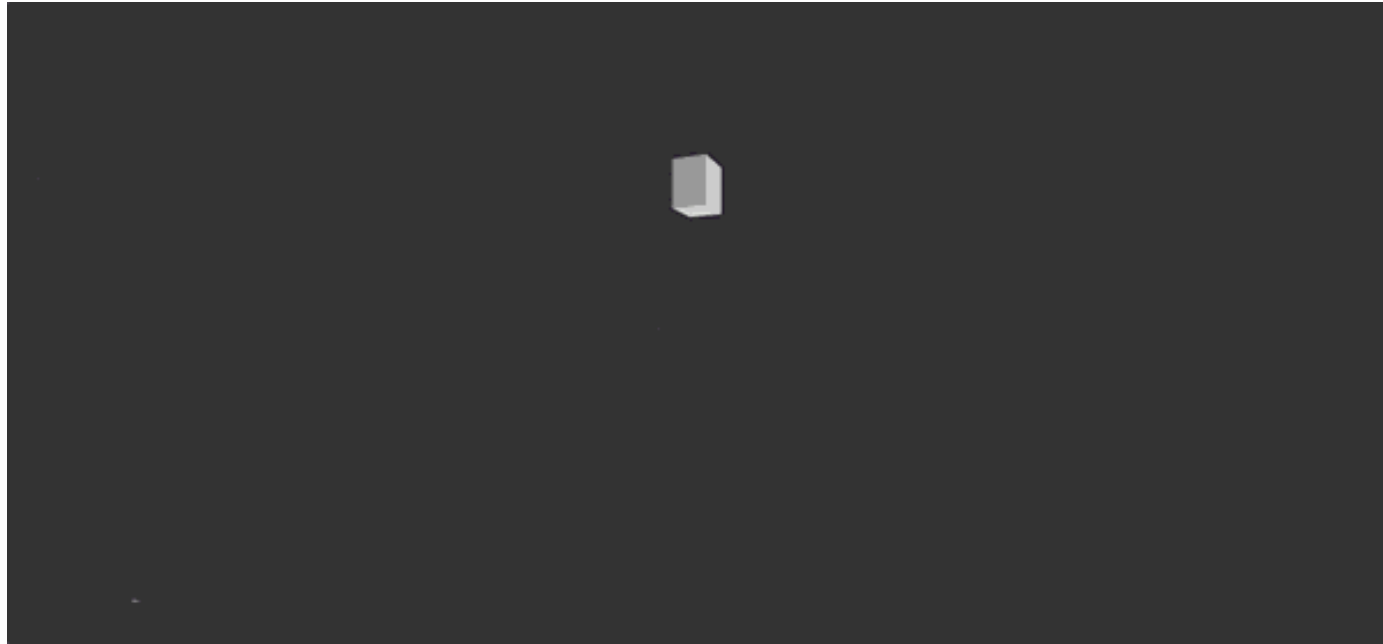
(but not as we know it)

See lots of links on my github site (also be on the last slide)

github.com/AndyDentFree/DDDI9

- Val Head's book *Interface Animation*, videos etc
- Disney's book [The Illusion of Life](#)
- [As video](#) and [gifs](#) 8 of which follow
- [Really good article](#) illustrating them with longer clips from Disney and Pixar movies, and explanations

Principle 1 - Squash and Stretch



Principle 2 - Anticipation



Principle 5 - Follow Through & Overlapping



Principle 6 - Slow In and Slow Out



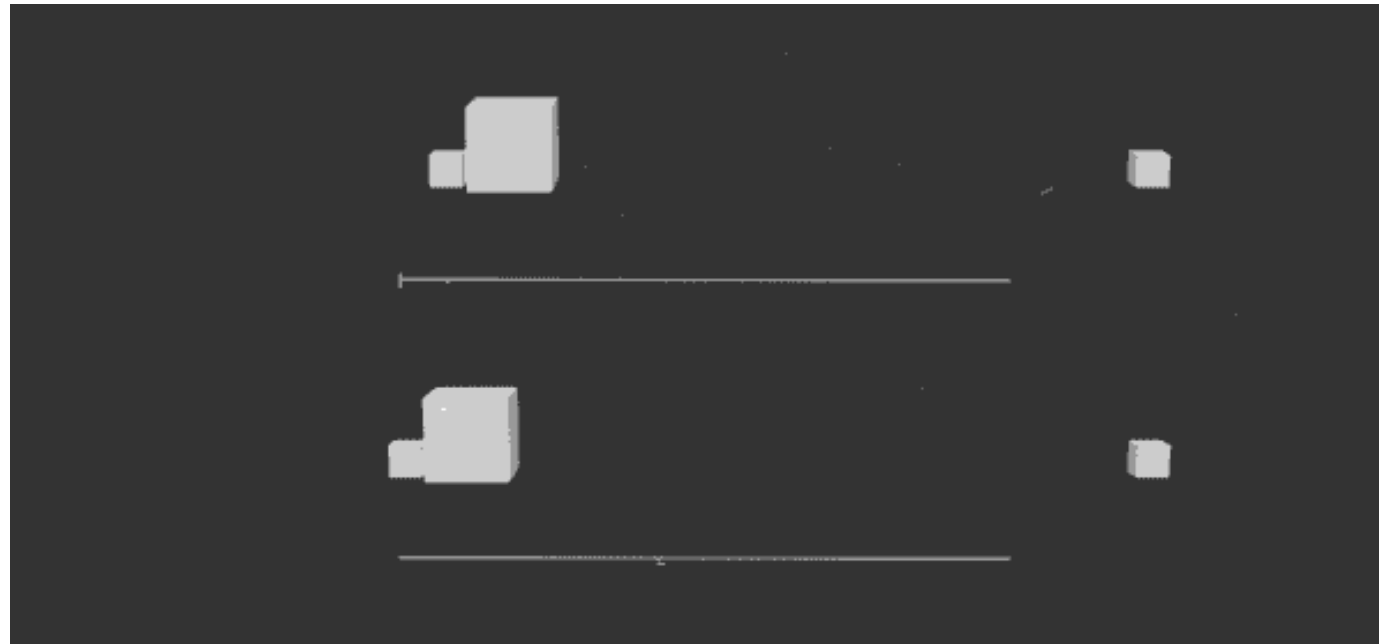
Principle 7 - Arcs



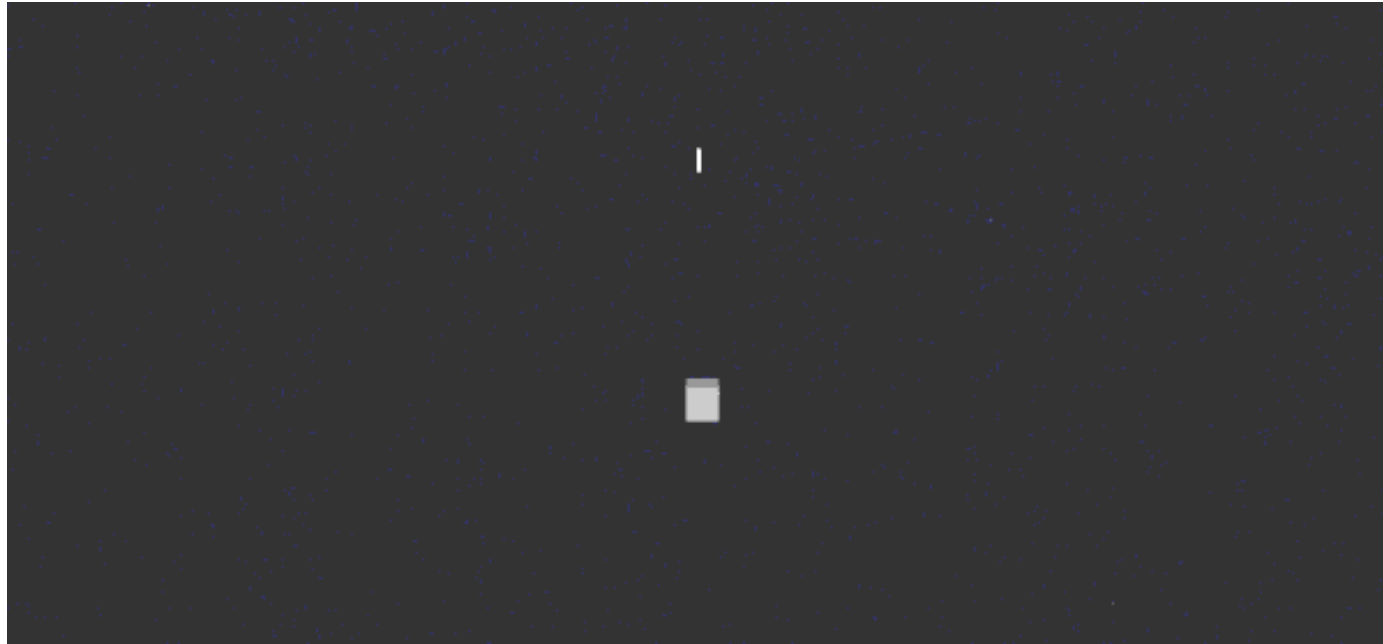
Principle 8 - Secondary Action



Principle 9 - Timing



Principle 10 - Exaggeration

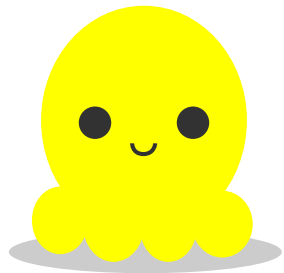


Why Animate? - Designers

Amuse

Distract

Inform



Why Animate? - Designers

Amuse

Distract

Inform

Why Animate? - Designers

Amuse

Distract

Inform



Why Animate? - Developer's View

Distract

Inform

Entertain

Animation as a new Developer Hell

Animation's aren't the only thing that go in circles.

The designer-developer loop finds new things to design that
have to be manually translated to code.

and then the loop closes

Coming to **save us** (last time)



New Breed of Tools

Start with Sketch or other vector prototype

Add animation

Generate working code

Web world

SVG-based vectors

Pure CSS

GSAP Greensock JS-driven canvas

and...

Lottie

Simple animated vectors from Adobe After Effects via *bodysmovin* plugin to JSON, effectively a [standard](#)

Fast native & web players by [AirBnb](#)

Other editors such as [Haiku](#) and [Keyshape](#)

As seen on the Why Animate? - Designers slide, content from [lottiefles.com](#)

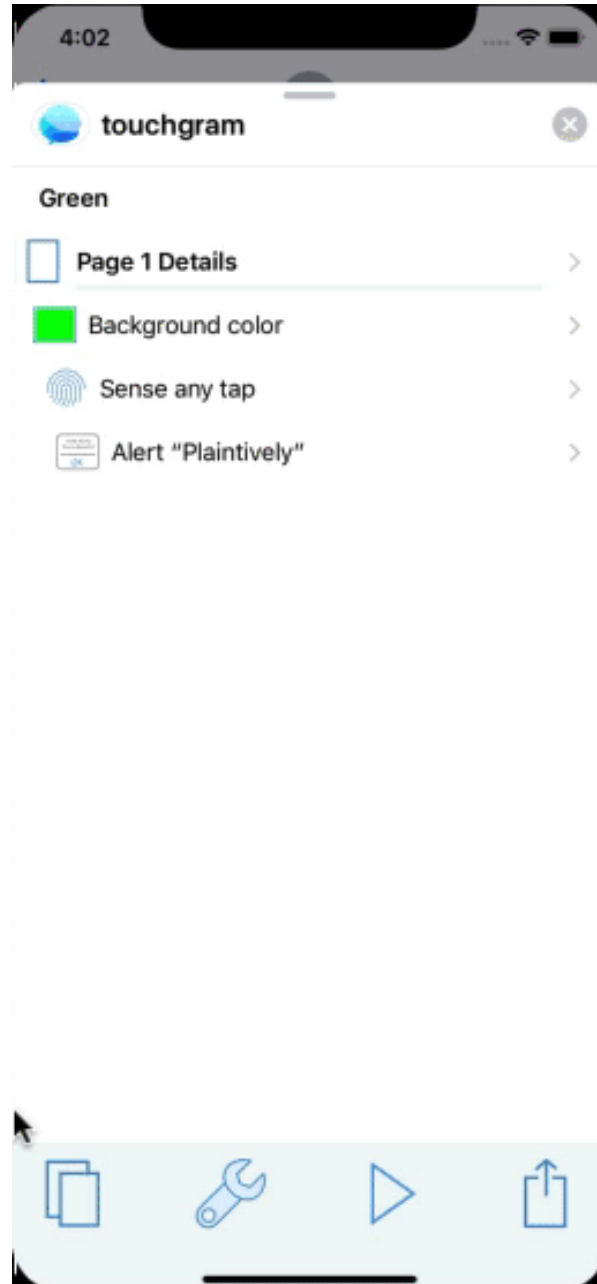
Mostly Mobile Tools

Supernova Studio

PaintCode

Flow

Simple Animation Example

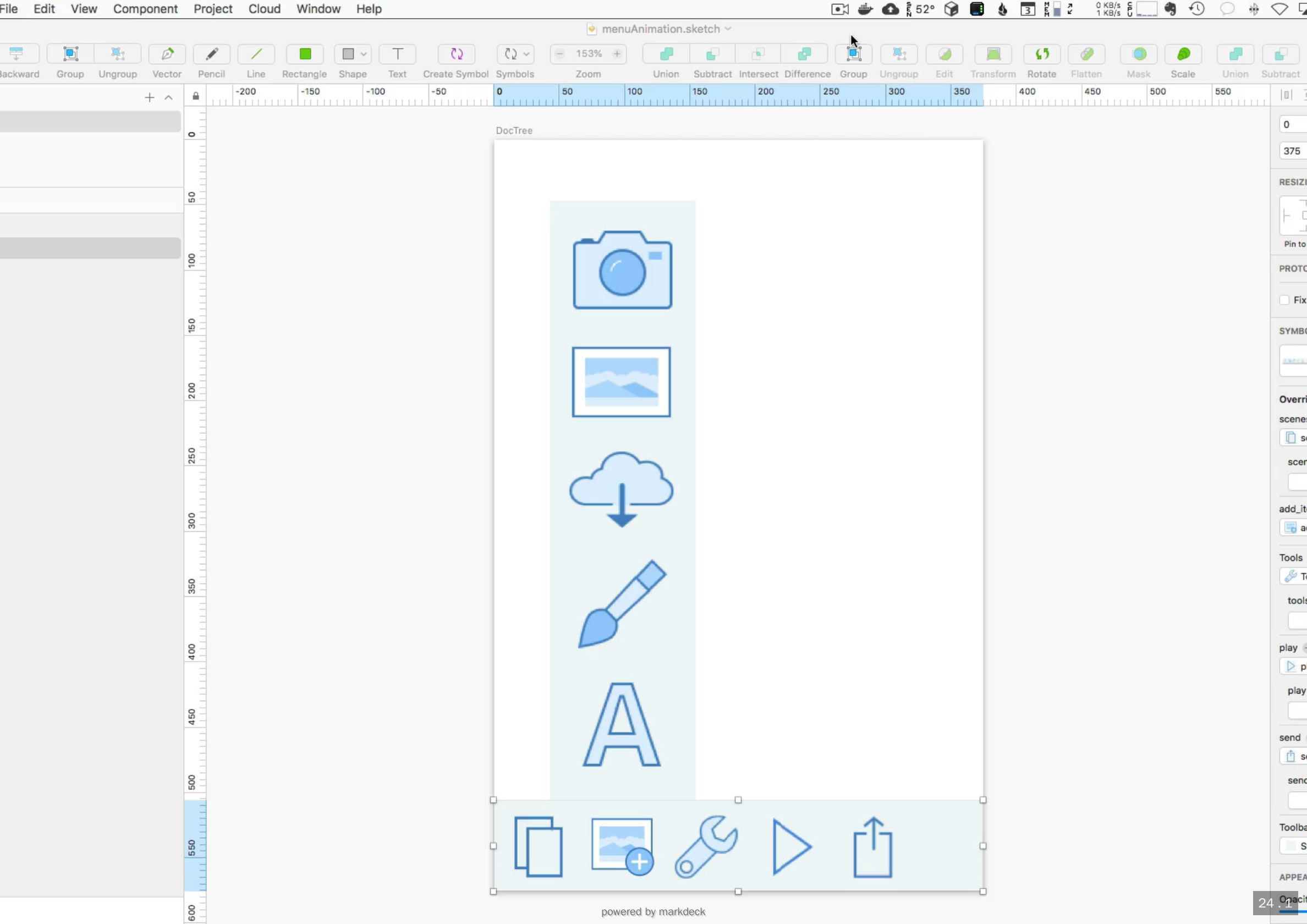


Supernova Editor

The Supernova Editor interface is divided into several sections:

- Top Bar:** Contains icons for Refresh, Project, Localization, and platform targets (iOS, Android, React, Flutter). It also includes zoom controls (100%), a Screen/Navigation toggle, Simulator/Publish buttons, Export/CodeX options, and Settings/Left/Right panels.
- Left Panel:** A list of animation groups. The selected group is "Animate on Tap on replay", which includes a "play-editing-toolbar" and a "Translate X" action.
- Center Canvas:** A preview of the mobile app design. It shows a dark screen with a toolbar at the bottom. The toolbar has a "B" button and a play button. A red dashed line indicates the animation path.
- Right Panel:** The "Animation group" settings panel. It shows the name "Animate on Tap on replay", the trigger "replay", and the trigger action "Tap".
- Code Editor:** A Swift code editor showing the implementation of the animation. The code defines two animation functions: `animationTwo()` and `animationThree()`. Both functions create a `CAAnimationGroup` and add a `CABasicAnimation` to it. The `animationTwo()` function sets the animation to move the toolbar from `0` to `-375` over a duration of `0.5` seconds.

```
152     self.playEditingToolbarView.layer.add(gr
153 }
154
155 private func animationTwo() {
156     let groupOne = CAAAnimationGroup()
157     groupOne.beginTime = CACurrentMediaTime(
158     groupOne.repeatCount = Float(1)
159     groupOne.duration = 0.5
160     groupOne.autoreverses = false
161     groupOne.isRemovedOnCompletion = false
162     groupOne.fillMode = .removed
163     groupOne.animations = []
164
165     let animationOne = CABasicAnimation()
166     animationOne.beginTime = 0
167     animationOne.repeatCount = Float(1)
168     animationOne.duration = 0.5
169     animationOne.autoreverses = false
170     animationOne.isRemovedOnCompletion = fal
171     animationOne.timingFunction = CAMediaTim
172     animationOne.keyPath = "transform.transl
173     animationOne.fromValue = 0
174     animationOne.toValue = -375
175     groupOne.animations?.append(animationOne
176
177     self.playEditingToolbarView.layer.add(gr
178 }
179
180 private func animationThree() {
181     let groupOne = CAAAnimationGroup()
182     groupOne.beginTime = CACurrentMediaTime(
183     groupOne.repeatCount = Float(1)
184     groupOne.duration = 0.5
185     groupOne.autoreverses = false
186     groupOne.isRemovedOnCompletion = false
187     groupOne.fillMode = .removed
188     groupOne.animations = []
189
190     let animationOne = CABasicAnimation()
191     animationOne.beginTime = 0
192     animationOne.repeatCount = Float(1)
193     animationOne.duration = 0.5
```



Supernova Generated - Swift

```
let animationElement1 = CAAAnimationGroup()
animationElement1.isRemovedOnCompletion = false
animationElement1.fillMode = .removed
animationElement1.animations = []

let animationProperty1 = CAKeyframeAnimation()
animationProperty1.beginTime = 0
animationProperty1.repeatCount = Float(1)
animationProperty1.duration = 1
animationProperty1.autoreverses = false
animationProperty1.isRemovedOnCompletion = false
animationProperty1.timingFunction = CAMediaTimingFunction(controlPoints: 0.22,
animationProperty1.keyPath = "transform.translation.y"
animationProperty1.keyTimes = [ 0, 0.6, 0.75, 0.9, 1 ]
animationProperty1.values = [ 3000, -25, 10, -5, 0 ]
animationProperty1.animations?.append(animationProperty1)
```

Supernova Generated - Kotlin

```
val animator1 = ObjectAnimator.ofPropertyValuesHolder(addItemMenuPopupConstrai
animator1.duration = 1000
animator1.interpolator = PathInterpolatorCompat.create(0.22f, 0.61f, 0.61f, 1f

val animator2 = ObjectAnimator.ofPropertyValuesHolder(addItemMenuPopupConstrai
animator2.duration = 1000
animator2.interpolator = PathInterpolatorCompat.create(0.22f, 0.61f, 0.61f, 1f

val animatorSet1 = AnimatorSet()
animatorSet1.playTogether(animator1, animator2)
animatorSet1.setTarget(addItemMenuPopupConstraintLayout)

val animatorSet2 = AnimatorSet()
animatorSet2.playTogether(animatorSet1)
animatorSet2.start()
```

Supernova Generated - React Native

```
this.state.addItemMenuPopupViewTranslateY.setValue(0)
this.state.addItemMenuPopupViewOpacity.setValue(0)

// Configure animation and trigger
Animated.parallel([Animated.parallel([Animated.timing(this.state.addItemMenuPo
  duration: 1000,
  easing: Easing.bezier(0.22, 0.61, 0.61, 1),
  toValue: 1,
]), Animated.timing(this.state.addItemMenuPopupViewOpacity, {
  duration: 1000,
  easing: Easing.bezier(0.22, 0.61, 0.61, 1),
  toValue: 1,
})])]).start()
```

Further Details

github.com/AndyDentFree/DDDI9

Contains this presentation and links to all tools mentioned, full samples and media to build anything shown in the movies.

Twitter: @andydentperth