



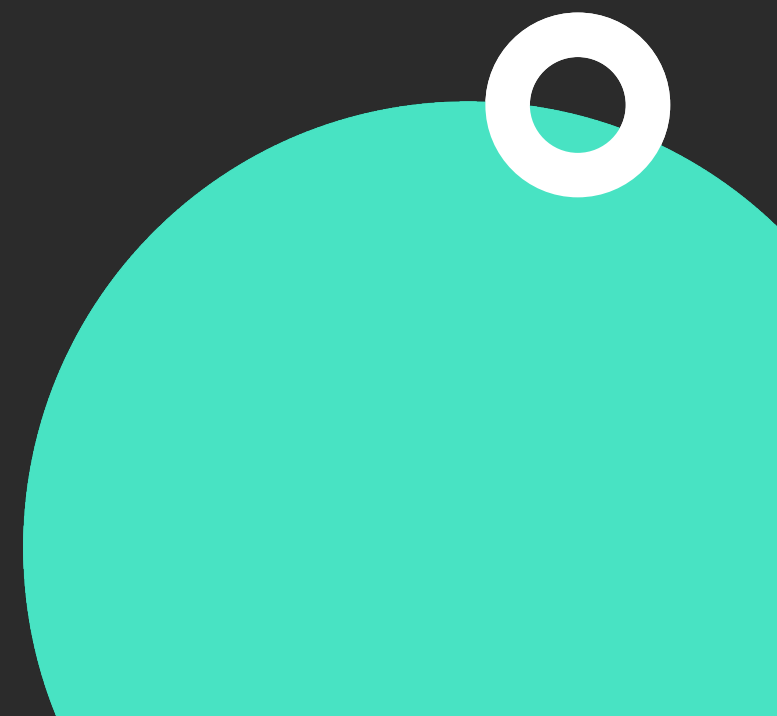
Android Animations and Transitions with Jetpack Compose



Mehdi Haghgoo

 @IAmMehdiHaghgoo

 in/mehdihaghgoo





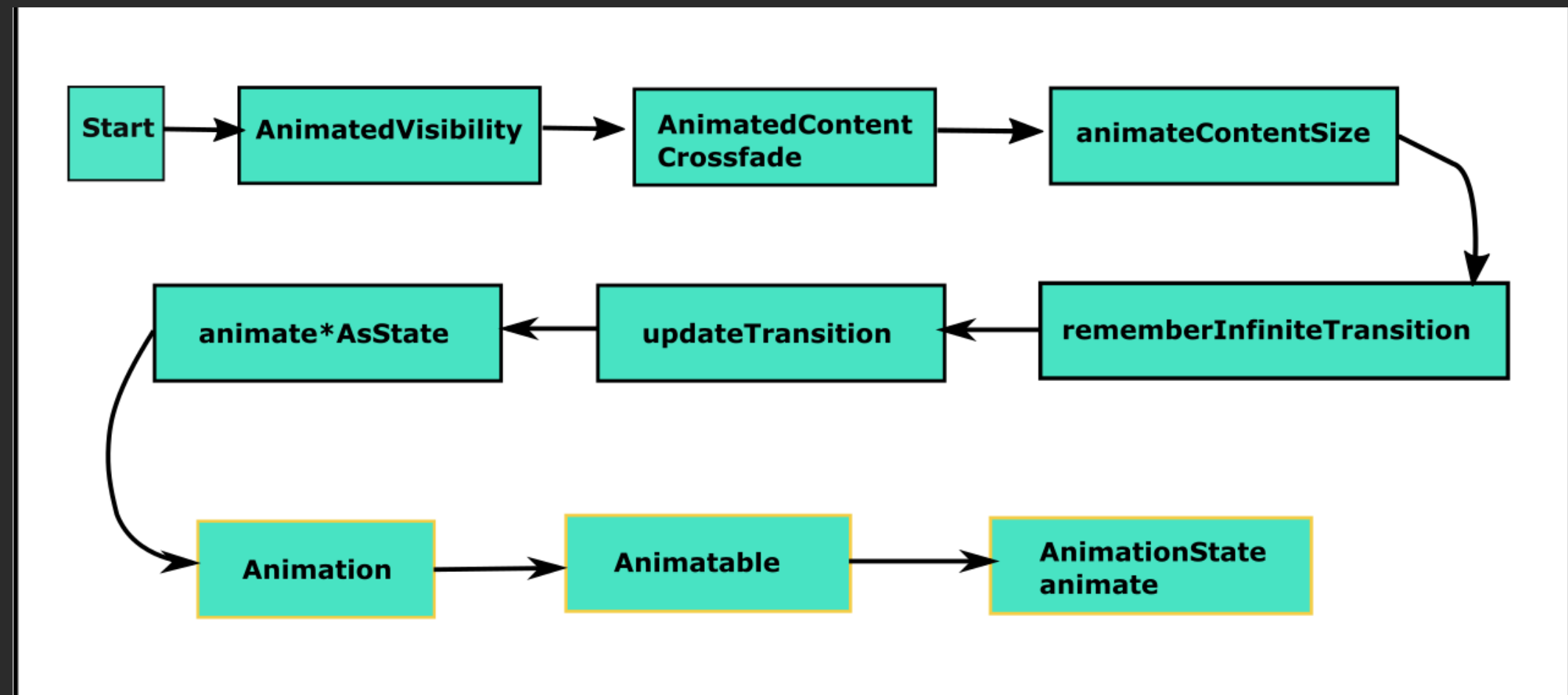
Section 4

Exploring Low-Level Animations



Section Overview

- Low-Level Animations
 - Animation
 - Animatable
 - animate
 - AnimationState
- Code
- Recap



Animation

- Lowest animation API
- A core animation engine
- Has two subtypes:
 - TargetBasedAnimation
 - DecayAnimation
- They usually run in a coroutine

```
TargetBasedAnimation(  
    animationSpec = tween(ANIM_DURATION),  
    typeConverter = Int.VectorConverter,  
    initialValue = 1,  
    initialVelocity = 1,  
    targetValue = 1000  
)
```

```
DecayAnimation(  
    animationSpec = exponentialDecay(),  
    typeConverter =  
Float.VectorConverter,  
    initialValue = 100f,  
    initialVelocity = 4000f  
)
```



Animatable

- Value holder
- API backing up animate*AsState
- Useful for implementing touch-based animation
- Animation via:
 - animateTo
 - snapTo
 - animateDecay

```
val offset = remember {
    Animatable(
        initialValue = Offset(0f, 0f),
        typeConverter =
            Offset.VectorConverter
    )
}

coroutineScope {
    val position = //Get position
    launch {
        offset.animateTo(position)
    }
}
```



animate

- A target-based animation
- Animate from initialValue to targetValue with optional velocity
- Customize with animationSpec

```
//Animate translate
animate(
    initialValue = 50f,
    targetValue = 200f,
    animationSpec = infiniteRepeatable(
        animation = tween(durationMillis =
ANIMATION_DURATION),
        repeatMode = RepeatMode.Reverse
    )
) { value, velocity ->
    translate = value
}
```



AnimationState

- Holds the state of an animation
- Can be queried for initial value, initial velocity, and isRunning
- Must be launched in a coroutine scope

```
//Make animation
val animState = remember {
    AnimationState(
        typeConverter =
            Color.VectorConverter(ColorSpaces.AdobeRgb),
        initialValue = Color.Cyan
    )
}
// Run animation
animationState.animateTo(
    targetValue = Color.Green
)
```



Code Time

- Folder : s4-exploring-lowlevel-animations/starter-project
- Files:
 - AnimationSheet.kt
 - TargetBasedAnimation
 - DecayAnimation
 - AnimatableSheet.kt
 - Animatable
 - AnimationStateOrAnimateSheet.kt
 - AnimationState
 - animate



Section Summary

- Low-Level Animations
 - Animation
 - Animatable
 - animate
 - AnimationState
- Now, you are familiar with all Compose animation APIs and how to use them
- What's next?
 - Apply what we have learned so far on a real-world app

