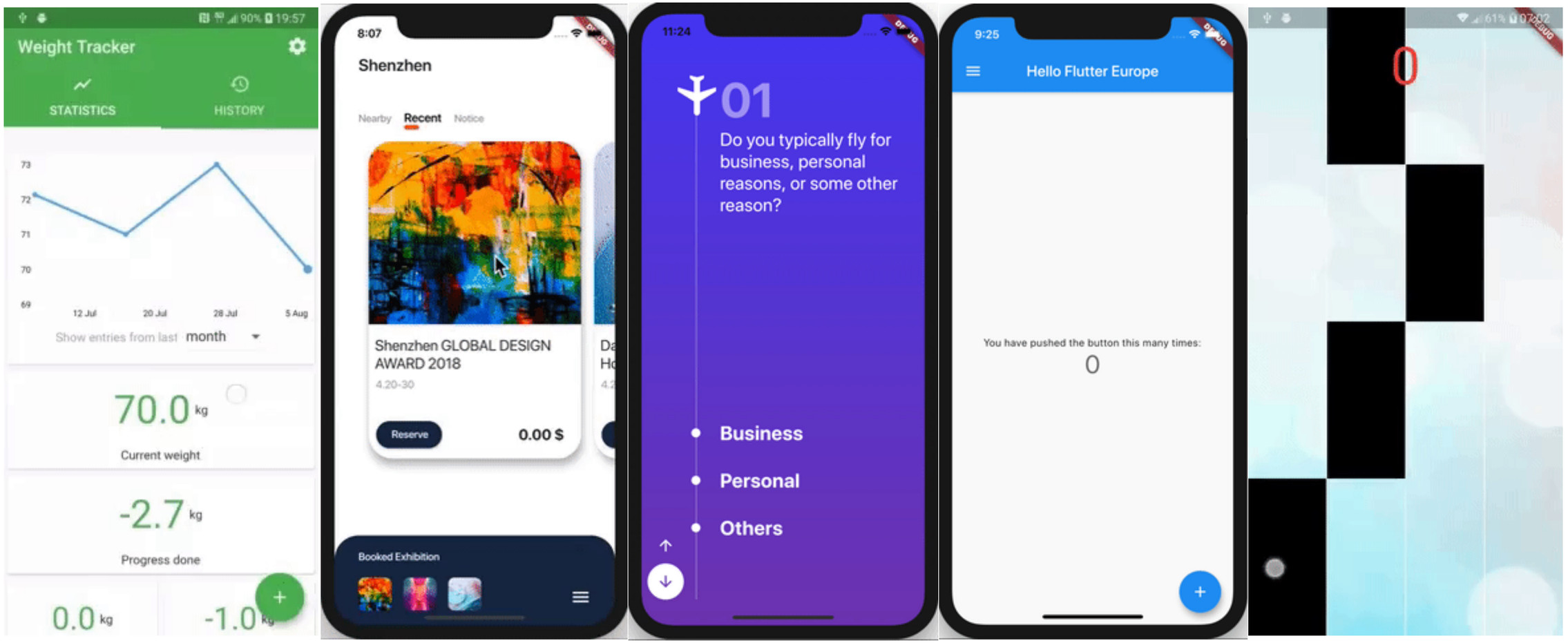
Animations in Flutter

Marcin Szałek
@marcin_szalek

Warsaw
January 23, 2023

Who am I?

- mgr inż. Marcin Szałek
- Flutter developer since Alpha (May 2017)
- Co-organizer and speaker at Flutter Europe
- Currently Mobile Lead at Cheddar
- Recently developer at FIFA World Cup Qatar 2022 ⚽

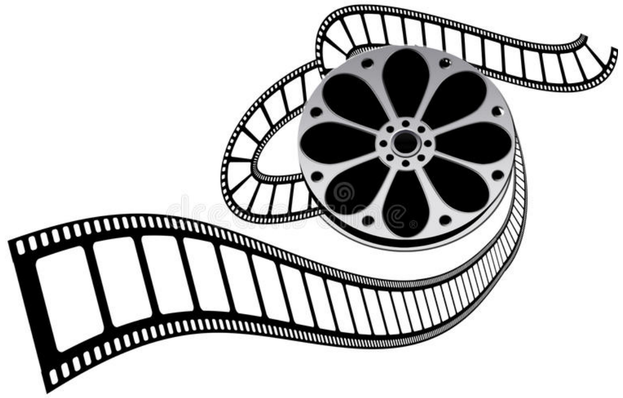

Weight Tracker
STATISTICS
HISTORY
73
72
71
70
69
12 Jul
20 Jul
28 Jul
5 Aug
Show entries from last month
70.0 kg
Current weight
-2.7 kg
Progress done
0.0 kg
-1.0 kg
8:07
Shenzhen
Nearby
Recent
Notice
Shenzhen GLOBAL DESIGN AWARD 2018
4.20-30
Reserve
0.00 $
Booked Exhibition
11:24
01
Do you typically fly for business, personal reasons, or some other reason?
Business
Personal
Others
9:25
Hello Flutter Europe
You have pushed the button this many times:
0
0

What are animations?

It’s the motion on the screen

And what is motion?

Many still images shown in quick successions



Why animations?

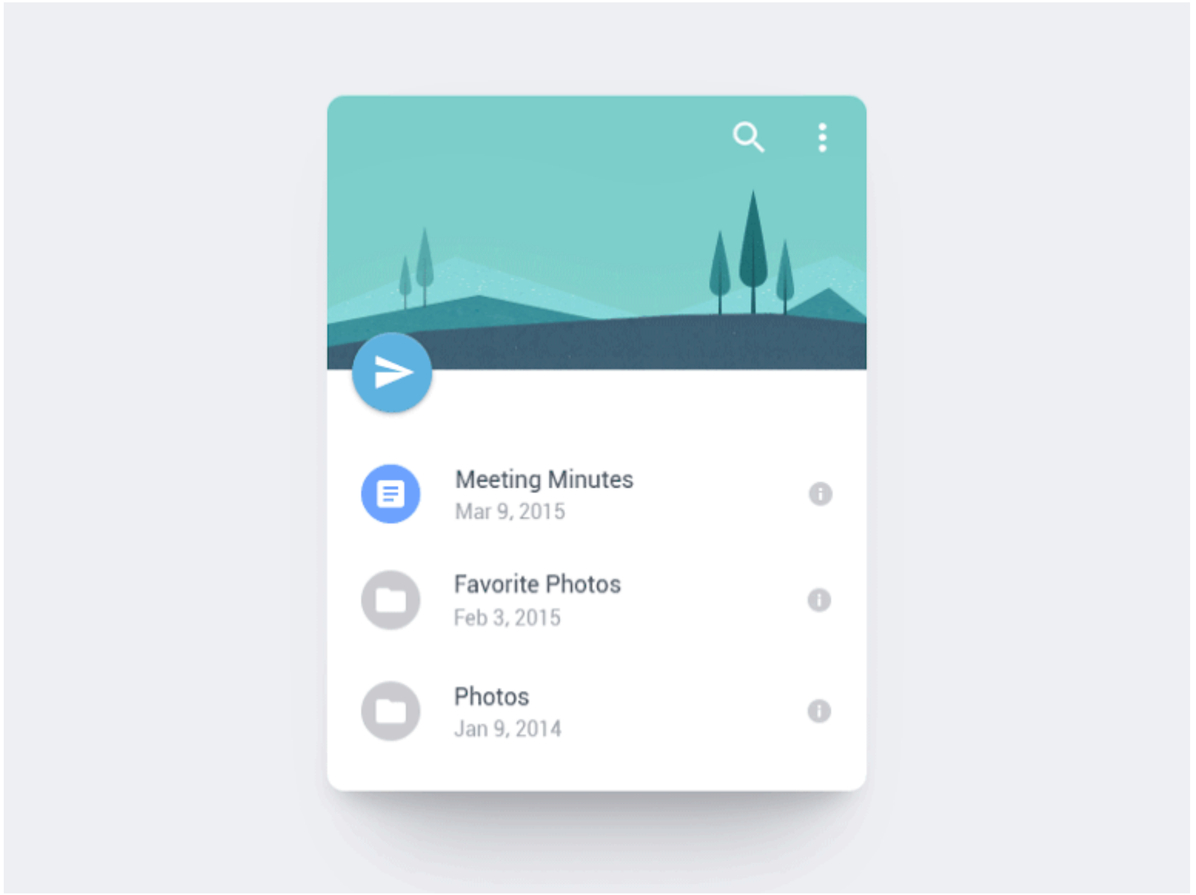
Functional

- Meaningful transitions
- Provide visual feedback
- Show system status
- Help user get started

Why animations?

Delightful

- To demonstrate polish
- To bring personality and humanity to the app
- To entertain users



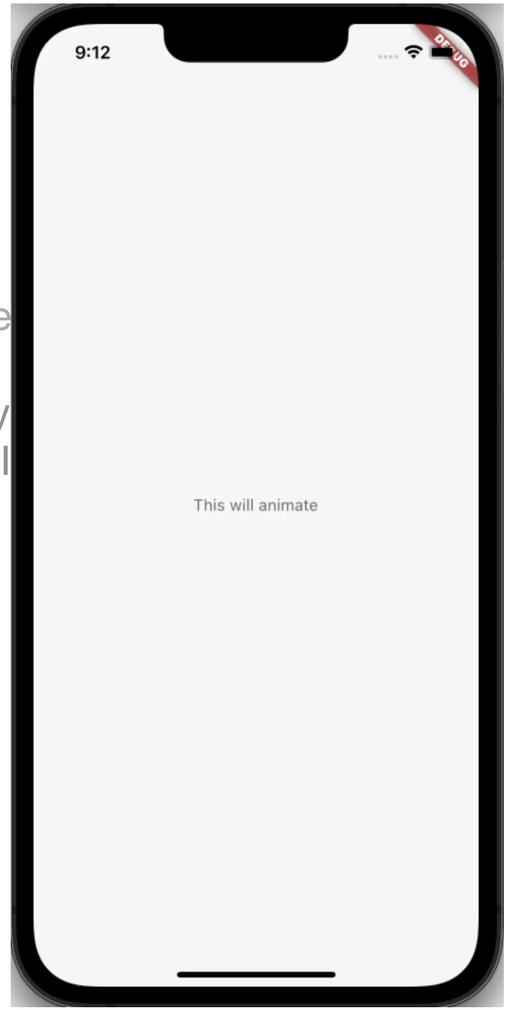
Implicit vs Explicit Animations

Implicit animations

AnimatedContainer
AnimatedAlign
AnimatedPositioned
AnimatedPositionedDirectional
AnimatedOpacity
AnimatedDefaultTextStyle
AnimatedPhysicalModel
AnimatedPadding
AnimatedTheme
AnimatedCrossFade
AnimatedSwitcher

Implicit animations

AnimatedContainer
AnimatedAlign
AnimatedPositioned
AnimatedPositionedDire
AnimatedOpacity
AnimatedDefaultTextSty
AnimatedPhysicalModel
AnimatedPadding
AnimatedTheme
AnimatedCrossFade
AnimatedSwitcher




```dart
class _PageState extends State<Page> {
  double _opacity = 1.0;

  @override
  Widget build(BuildContext context) {
    return AnimatedOpacity(
      opacity: _opacity,
      duration: const Duration(milliseconds: 300),
      child: const Text('This will animate'),
    );
  }
}
```

Implicit animations

AnimatedContainer
AnimatedAlign
AnimatedPositioned
AnimatedPositionedDirectional
AnimatedOpacity
AnimatedDefaultTextStyle
AnimatedPhysicalModel
AnimatedPadding
AnimatedTheme
AnimatedCrossFade
AnimatedSwitcher

```dart
class _PageState extends State<Page> {
  EdgeInsetsGeometry _padding = const EdgeInsets.all(8);

  @override
  Widget build(BuildContext context) {
    return AnimatedPadding(
      padding: _padding,
      duration: const Duration(milliseconds: 300),
      child: const Text('This will animate'),
    );
  }
}
```

Implicit animations

AnimatedContainer
AnimatedAlign
AnimatedPositioned
AnimatedPositionedDirectional
AnimatedOpacity
AnimatedDefaultTextStyle
AnimatedPhysicalModel
AnimatedPadding
AnimatedTheme
AnimatedCrossFade
AnimatedSwitcher

```dart
class _PageState extends State<Page> {
  bool _isActive = true;

  @override
  Widget build(BuildContext context) {
    return AnimatedContainer(
      height: _isActive ? 300 : 200,
      color: _isActive ? Colors.green : Colors.yellow,
      duration: Duration(milliseconds: 300),
      child: const Text('This will animate'),
    );
  }
}
```

Implicit animations

AnimatedContainer
AnimatedAlign
AnimatedPositioned
AnimatedPositionedDirectional
AnimatedOpacity
AnimatedDefaultTextStyle
AnimatedPhysicalModel
AnimatedPadding
AnimatedTheme
AnimatedCrossFade
AnimatedSwitcher

```dart
class _PageState extends State<Page> {
  bool _isActive = true;

  @override
  Widget build(BuildContext context) {
    return AnimatedSwitcher(
      duration: Duration(milliseconds: 300),
      child: _isActive
          ? const Text('Active')
          : const CircularProgressIndicator(),
    );
  }
}
```

Explicit animations

AnimationController

AnimationController

```dart
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(vsync: this);
  }

  @override
  void dispose() {
    _animationController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return ...;
  }
}
```

AnimationController

```
_animationController = AnimationController(
  vsync: this,
  duration: const Duration(seconds: 1),
  lowerBound: 0.0,
  upperBound: 1.0,
);
```

```
_animationController.value;
_animationController.status;
_animationController.isCompleted;
_animationController.duration;
_animationController.velocity;
```

```
_animationController.forward(from: 0.0);
_animationController.reverse(from: 1.0);
_animationController.animateTo(0.45);
_animationController.repeat(reverse: false);
```

```
_animationController.addListener(() {
  //do sth whenever value changes
});
_animationController.addStatusListener((status) {
  //do sth whenever status changes
});
```

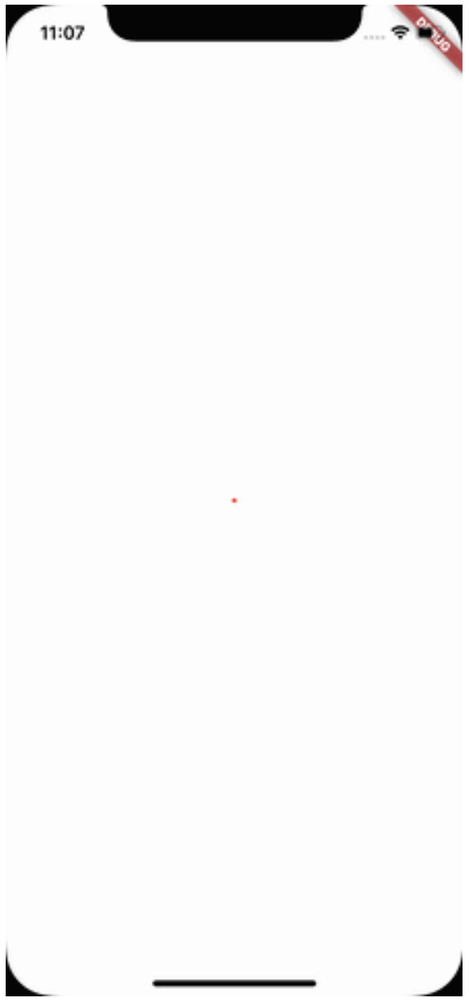
AnimationController

Bad, ugly and good

AnimationController

Bad, ugly and good

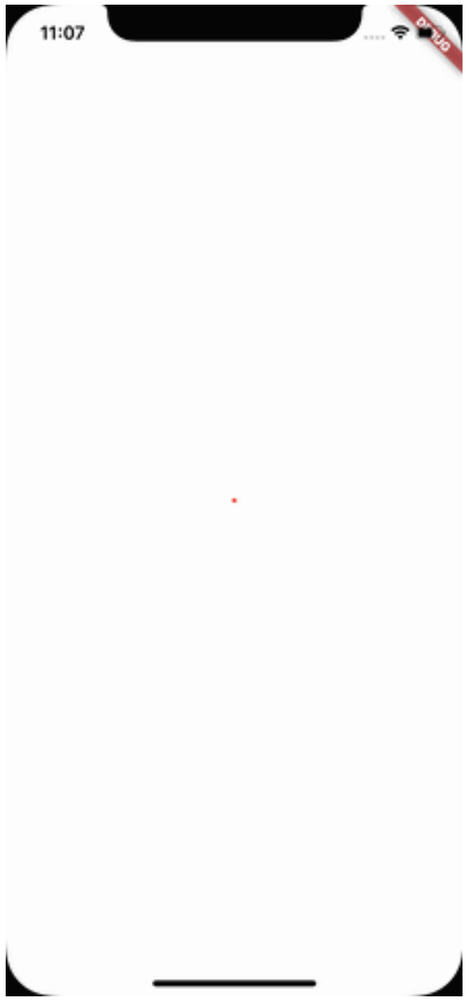
```dart
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animationController.repeat();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Transform.scale(
          scale: _animationController.value,
          child: Container(
            color: Colors.red,
            height: 300,
            width: 300,
          ),
        ),
      ),
    );
  }
}
```



AnimationController

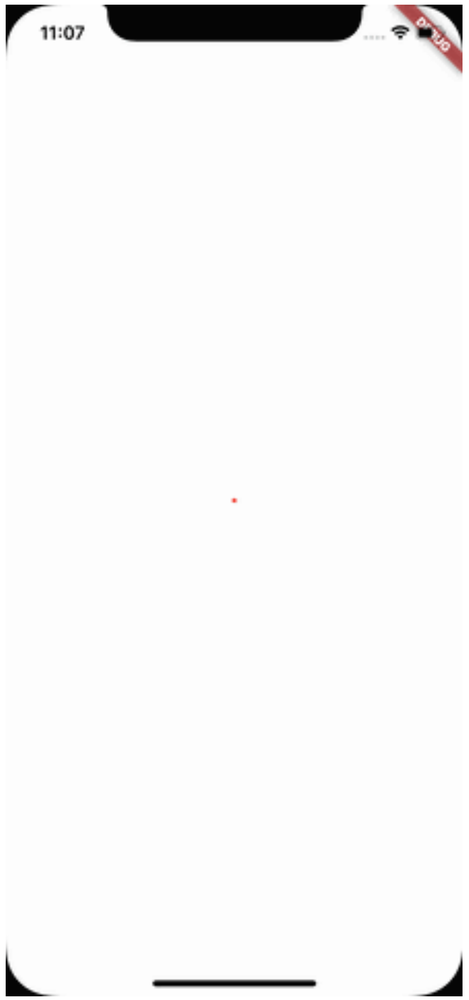
Bad, ugly and good

```dart
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animationController.addListener(() => setState(() {}));
    _animationController.repeat();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Transform.scale(
          scale: _animationController.value,
          child: Container(
            color: Colors.red,
            height: 300,
            width: 300,
          ),
        ),
      ),
    );
  }
}
```



AnimationController

```
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animationController.repeat();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: AnimatedBuilder(
          animation: _animationController,
          builder: (context, child) => Transform.scale(
            scale: _animationController.value,
            child: child,
          ),
          child: Container(
            color: Colors.red,
            height: 300,
            width: 300,
          ),
        ),
      ),
    );
  }
}
```



AnimationController

```dart
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animationController.repeat();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: AnimatedBuilder(
          animation: _animationController,
          builder: (context, child) => Transform.scale(
            scale: _animationController.value,
            child: child,
          ),
          child: Container(
            color: Colors.red,
            height: 300,
            width: 300,
          ),
        ),
      ),
    );
  }
}
```

AnimationController

```
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;
  late final Animation<double> _animation;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animationController.repeat();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: AnimatedBuilder(
          animation: _animationController,
          builder: (context, child) => Transform.scale(
            scale: _animationController.value,
            child: child,
          ),
          child: Container(
            color: Colors.red,
            height: 300,
            width: 300,
          ),
        ),
      ),
    );
  }
}
```

```dart
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;
  late final Animation<double> _animation;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animation = CurvedAnimation(
      parent: _animationController,
      curve: Curves.elasticOut,
    );
    _animationController.repeat();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: AnimatedBuilder(
          animation: _animationController,
          builder: (context, child) => Transform.scale(
            scale: _animationController.value,
            child: child,
          ),
          child: Container(
            color: Colors.red,
            height: 300,
            width: 300,
          ),
        ),
      ),
    );
  }
}
```

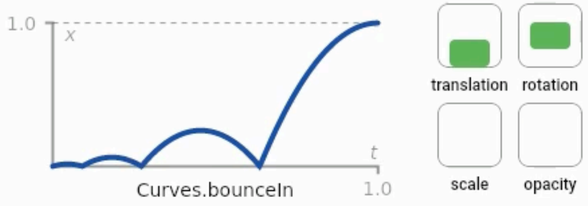
```dart
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;
  late final Animation<double> _animation;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 2),
    );
    _animation = CurvedAnimation(
      parent: _animationController,
      curve: Curves.elasticOut,
    );
    _animationController.repeat();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: AnimatedBuilder(
          animation: _animationController,
          builder: (context, child) => Transform.scale(
            scale: _animation.value,
            child: child,
          ),
          child: Container(
            color: Colors.red,
            height: 300,
            width: 300,
          ),
        ),
      ),
    );
  }
}
```

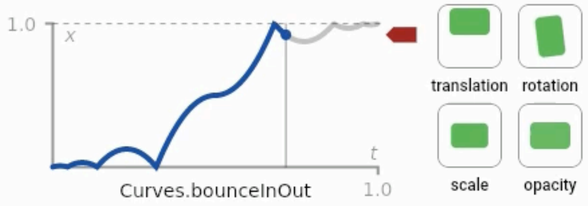
Animation curves

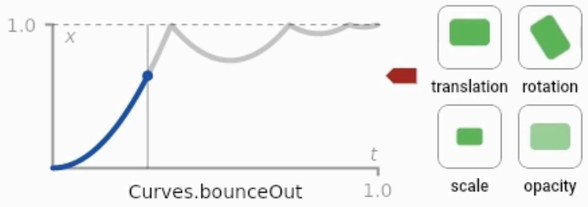
Curves class

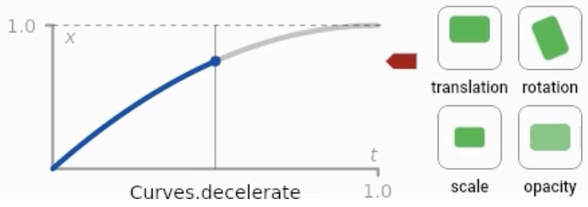
Null safety

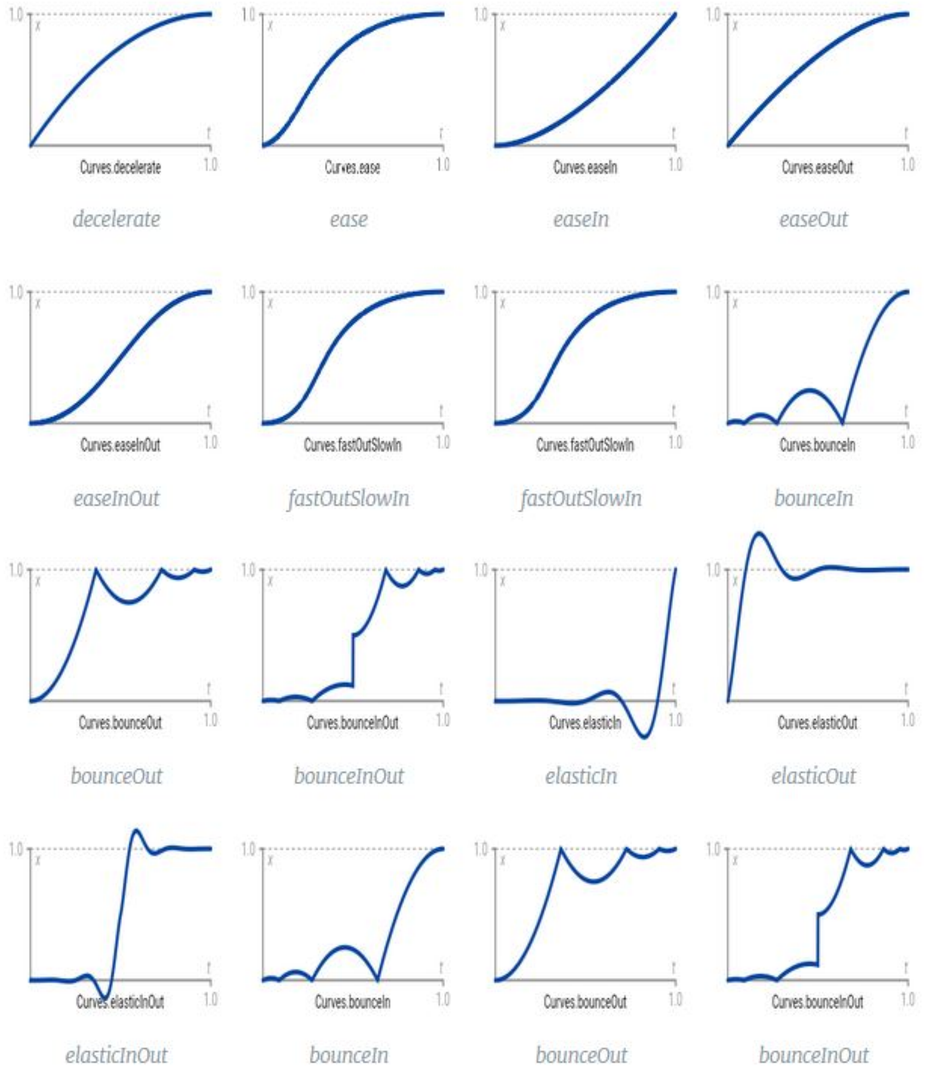
A collection of common animation curves.







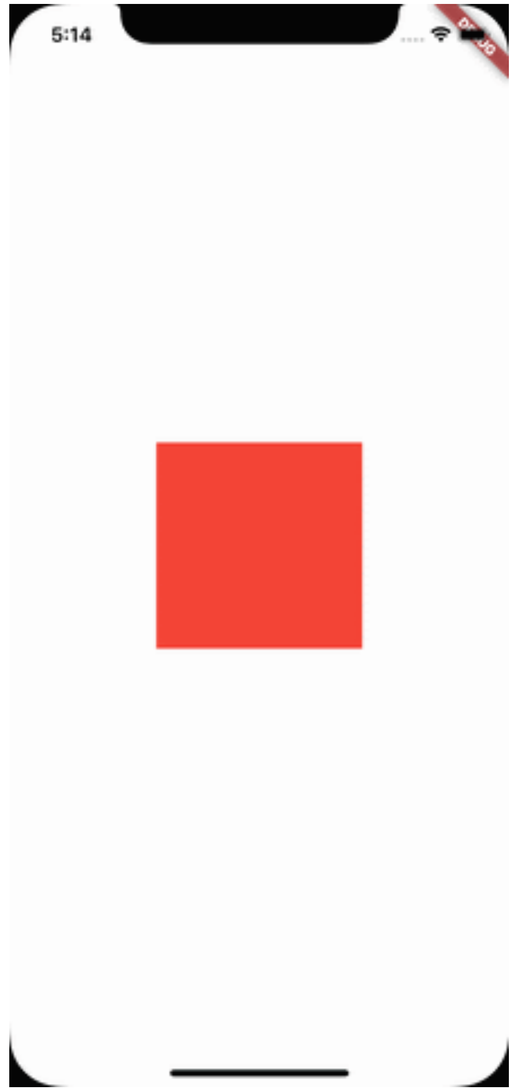




What can be animated?

Tweens

```
@override
void initState() {
  super.initState();
  _animationController = AnimationController(
    vsync: this,
    duration: const Duration(seconds: 2),
  );
  _animation = Tween(begin: 0.5, end: 2.0).animate(_animationController);
  _animationController.repeat();
}
```



Tweens

```dart
@override
void initState() {
  super.initState();
  _animationController = AnimationController(
    vsync: this,
    duration: const Duration(seconds: 5),
  );
  _animation = IntTween(begin: 0, end: 100).animate(CurvedAnimation(
    parent: _animationController,
    curve: Curves.easeOutExpo,
  ));
  _animationController.repeat();
}
```

Tweens

```dart
@override
void initState() {
  super.initState();
  _animationController = AnimationController(
    vsync: this,
    duration: const Duration(seconds: 5),
  );
  _animation = IntTween(begin: 0, end: 100).animate(CurvedAnimation(
    parent: _animationController,
    curve: Curves.easeOutExpo,
  ));
  _animationController.repeat();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: AnimatedBuilder(
        animation: _animationController,
        builder: (context, child) => Text(
          _animation.value.toString(),
          style: const TextStyle(fontSize: 50),
        ),
      ),
    ),
  );
}
```

Tweens

```
_animation = ColorTween(
  begin: Colors.yellow,
  end: Colors.orange,
).animate(_animationController);
```

```
_animation = SizeTween(
  begin: const Size(200, 50),
  end: const Size(50, 200),
).animate(_animationController);
```

```
_animation = BorderRadiusTween(
  begin: BorderRadius.circular(0),
  end: BorderRadius.circular(80),
).animate(_animationController);
```

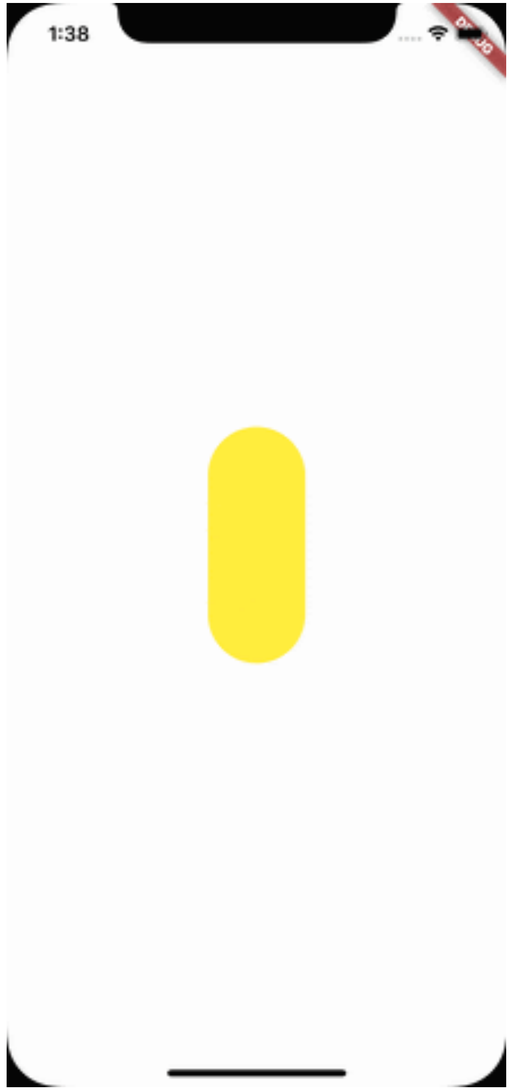
Tweens

```dart
class _MyPageState extends State<MyPage> with SingleTickerProviderStateMixin {
  late final AnimationController _animationController;
  late final Animation _borderAnimation;
  late final Animation _sizeAnimation;
  late final Animation _colorAnimation;

  @override
  void initState() {
    super.initState();
    _animationController = AnimationController(
      vsync: this,
      duration: const Duration(seconds: 1),
    );
    _borderAnimation = BorderRadiusTween(
      begin: BorderRadius.circular(0),
      end: BorderRadius.circular(80),
    ).animate(_animationController);
    _sizeAnimation = SizeTween(
      begin: Size(200, 50),
      end: Size(50, 200),
    ).animate(_animationController);
    _colorAnimation = ColorTween(
      begin: Colors.yellow,
      end: Colors.red,
    ).animate(_animationController);
    _animationController.repeat(reverse: true);
  }
```

```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: AnimatedBuilder(
        animation: _animationController,
        builder: (context, child) => Container(
          decoration: BoxDecoration(
            borderRadius: _borderAnimation.value,
            color: _colorAnimation.value,
          ),
          width: _sizeAnimation.value.width,
          height: _sizeAnimation.value.height,
        ),
      ),
    ),
  );
}
```

Staggered Animations

```
_borderAnimation = BorderRadiusTween(
  begin: BorderRadius.circular(0),
  end: BorderRadius.circular(80),
).animate(CurvedAnimation(
  parent: _animationController,
  curve: const Interval(0, 0.4),
));
_sizeAnimation = SizeTween(
  begin: Size(200, 50),
  end: Size(50, 200),
).animate(CurvedAnimation(
  parent: _animationController,
  curve: const Interval(0.3, 0.7),
));
_colorAnimation = ColorTween(
  begin: Colors.yellow,
  end: Colors.red,
).animate(CurvedAnimation(
  parent: _animationController,
  curve: const Interval(0.7, 1.0),
));
```



TweenAnimationBuilder

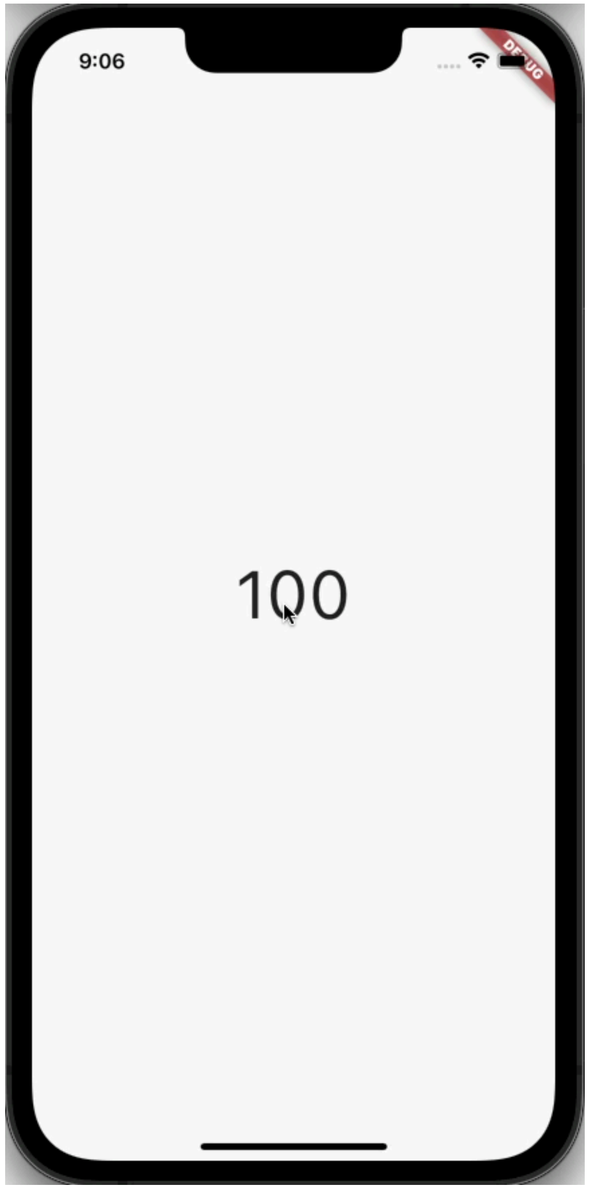
```
class _PageState extends State<Page> {
  @override
  Widget build(BuildContext context) {
    return TweenAnimationBuilder(
      tween: IntTween(begin: 0, end: 100),
      duration: const Duration(seconds: 1),
      builder: (context, value, child) => Text(value.toString()),
    );
  }
}
```

TweenAnimationBuilder

```
class _PageState extends State<Page> {
  int _counter = 100;

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
     onTap: () => setState(() => _counter += 100),
      child: TweenAnimationBuilder(
        tween: IntTween(begin: 0, end: _counter),
        duration: const Duration(seconds: 1),
        builder: (context, value, child) => Text(value.toString()),
      ),
    );
  }
}
```

TweenAnimationBuilder

```dart
class _PageState extends State<Page> {
  int _counter = 100;

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
     onTap: () => setState(() => _counter += 100),
      child: TweenAnimationBuilder(
        tween: IntTween(begin: 0, end: _counter),
        duration: const Duration(seconds: 1),
        builder: (context, value, child) => Text(value.toString()),
      ),
    );
  }
}
```



Can we make it easier?

SlideTransition
ScaleTransition
RotationTransition
SizeTransition
FadeTransition
RelativeRectTween
PositionedTransition
RelativePositionedTransition
DecoratedBoxTransition
AlignTransition
DefaultTextStyleTransition

Can we make it easier?

SlideTransition
ScaleTransition
RotationTransition
SizeTransition
FadeTransition
RelativeRectTween
PositionedTransition
RelativePositionedTransiti
DecoratedBoxTransition
AlignTransition
DefaultTextStyleTransition

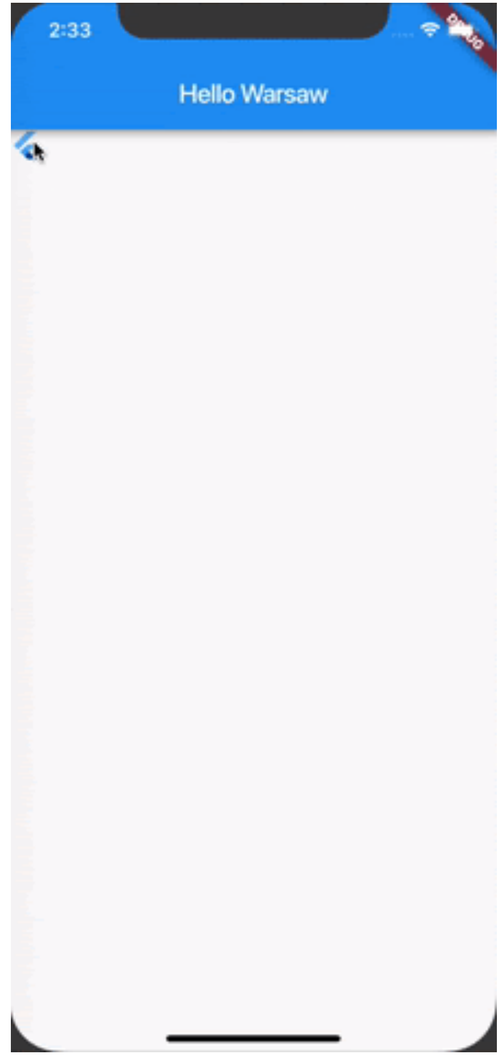
```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: RotationTransition(
        turns: _animationController,
        child: Container(
          color: Colors.red,
          width: 300,
          height: 300,
        ),
      ),
    ),
  );
}
```

Some other tricks…

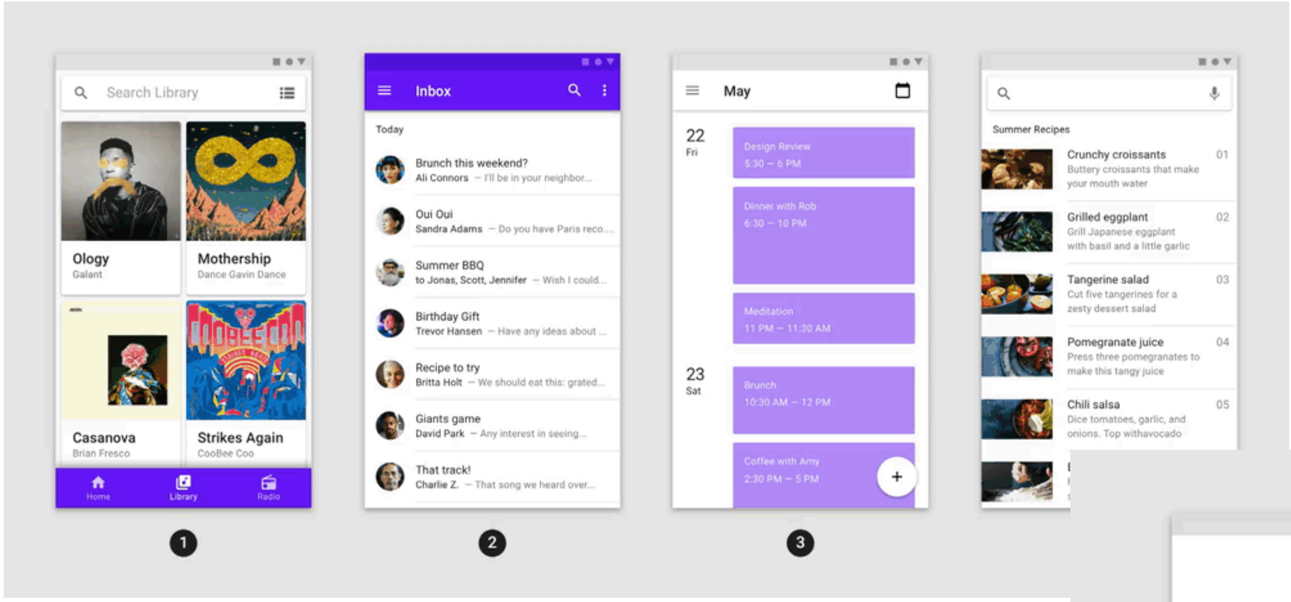
Heroes

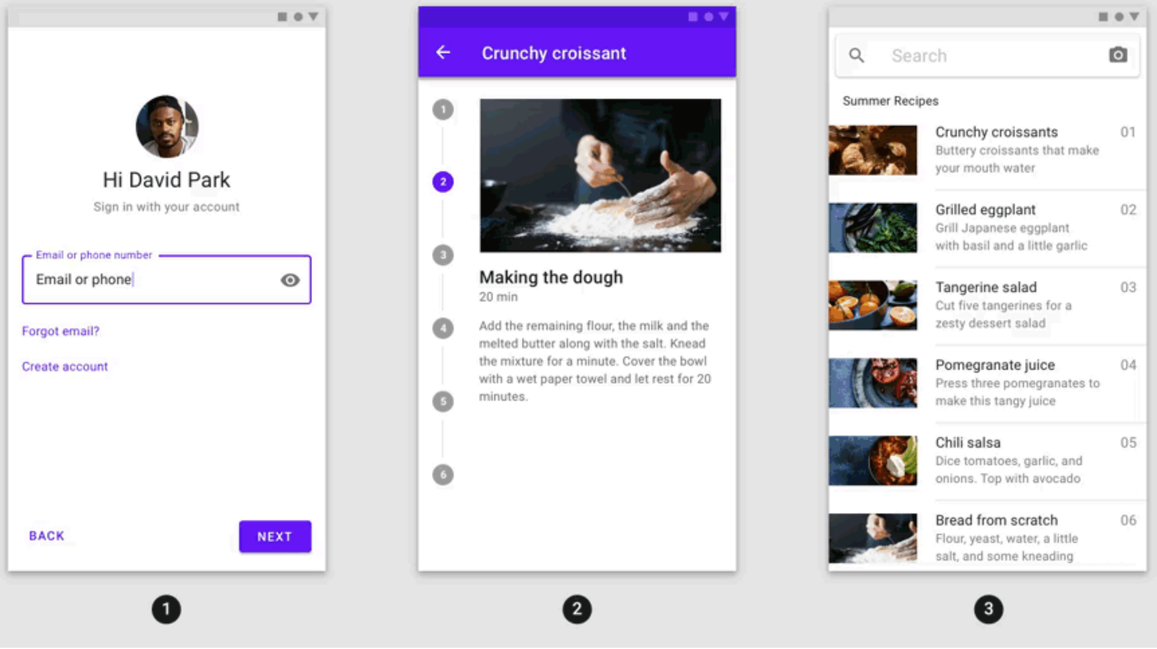
```dart
class _PageState extends State<Page> {
  void _goToPage2() => Navigator.of(context).push(
        MaterialPageRoute(builder: (_) => Page2()),
      );

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title:const Text('Hello Warsaw')),
      body: Hero(
        tag: 'FlutterLogo',
        child: GestureDetector(
          onTap: _goToPage2,
          child: const FlutterLogo(),
        ),
      ),
    );
  }
}
```

```dart
class Page2 extends StatelessWidget {
  const Page2({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Page 2'),
      ),
      body: const Center(
        child: Hero(
          tag: 'FlutterLogo',
          child: FlutterLogo(size: 200),
        ),
      ),
    );
  }
}
```



Animations package

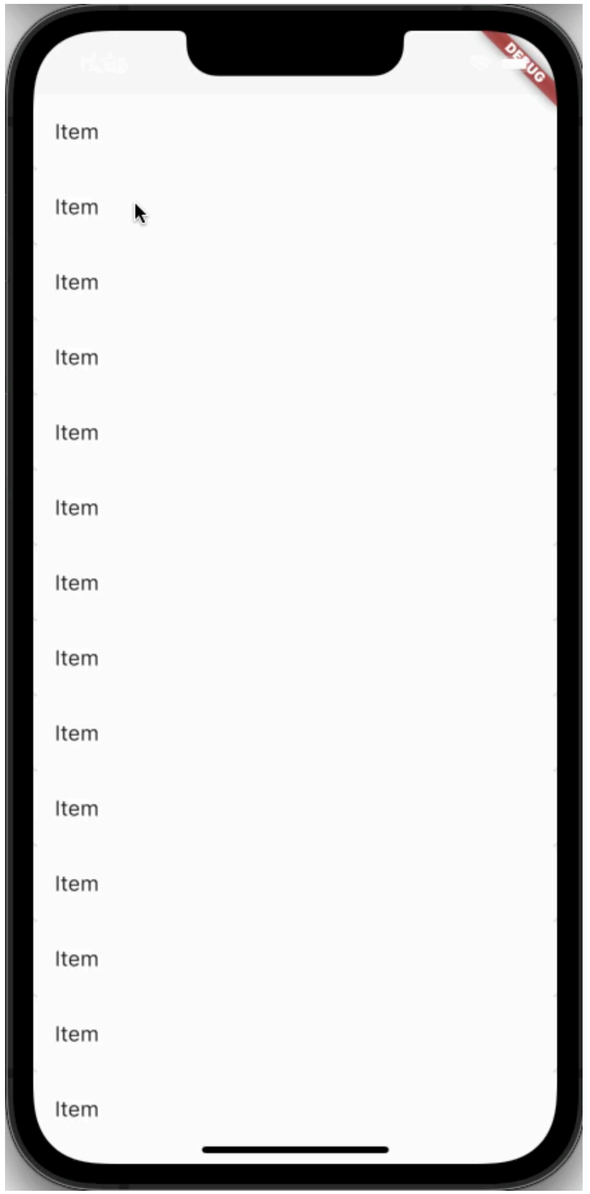




Animations package

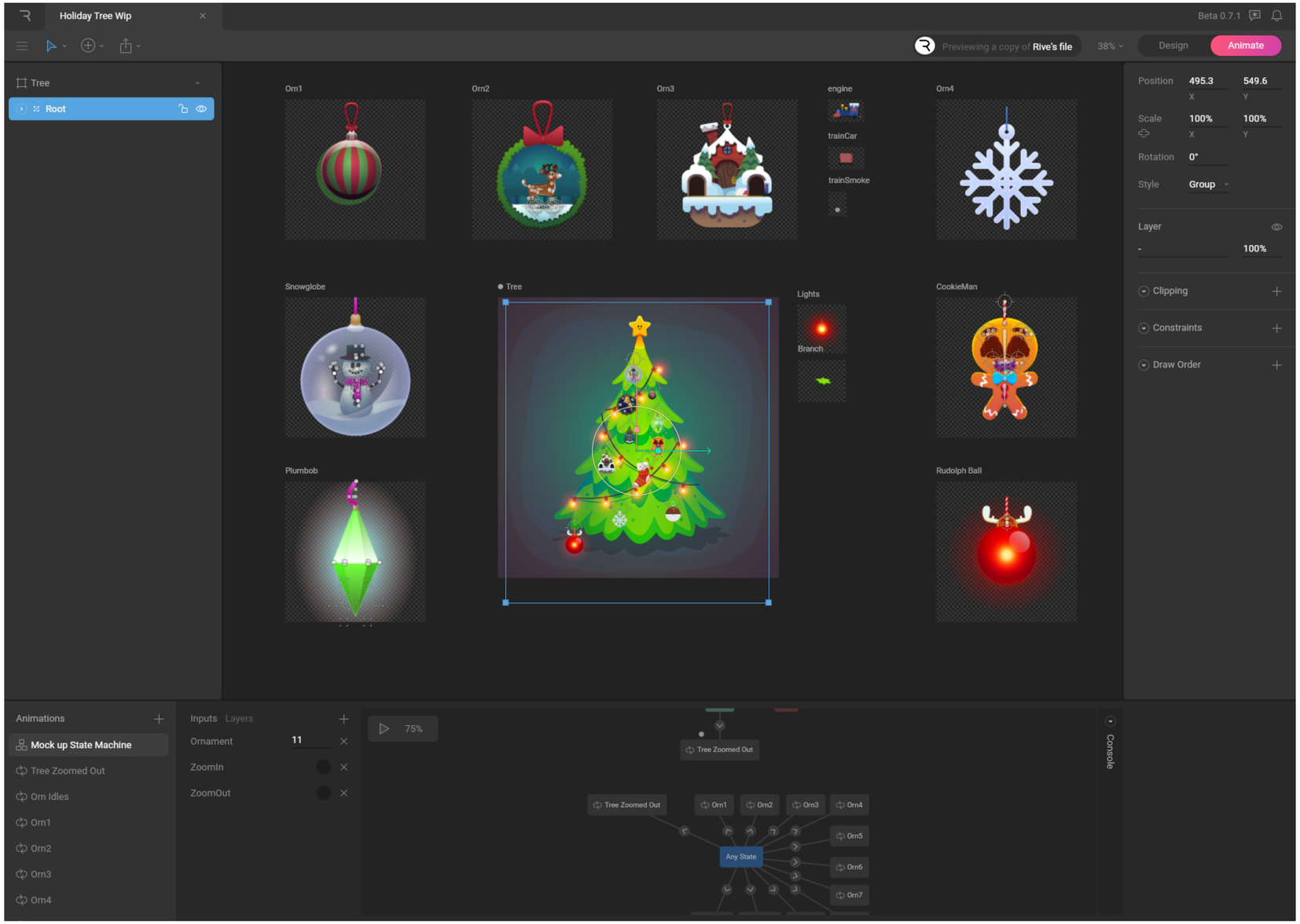
```dart
class _OpenContainer extends StatelessWidget {
  const _OpenContainer({
    Key? key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return OpenContainer(
      transitionDuration: Duration(milliseconds: 1000),
      closedBuilder: (_, __) => const Center(
        child: ListTile(title: Text('Item')),
      ),
      openBuilder: (_, __) => Scaffold(
        appBar: AppBar(title: const Text('Page 2')),
        body: const Center(child: Text('page2')),
      ),
    );
  }
}
```

```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: ListView(
      children: List.generate(
        100,
        (index) => _OpenContainer(),
      ),
    ),
  );
}
```

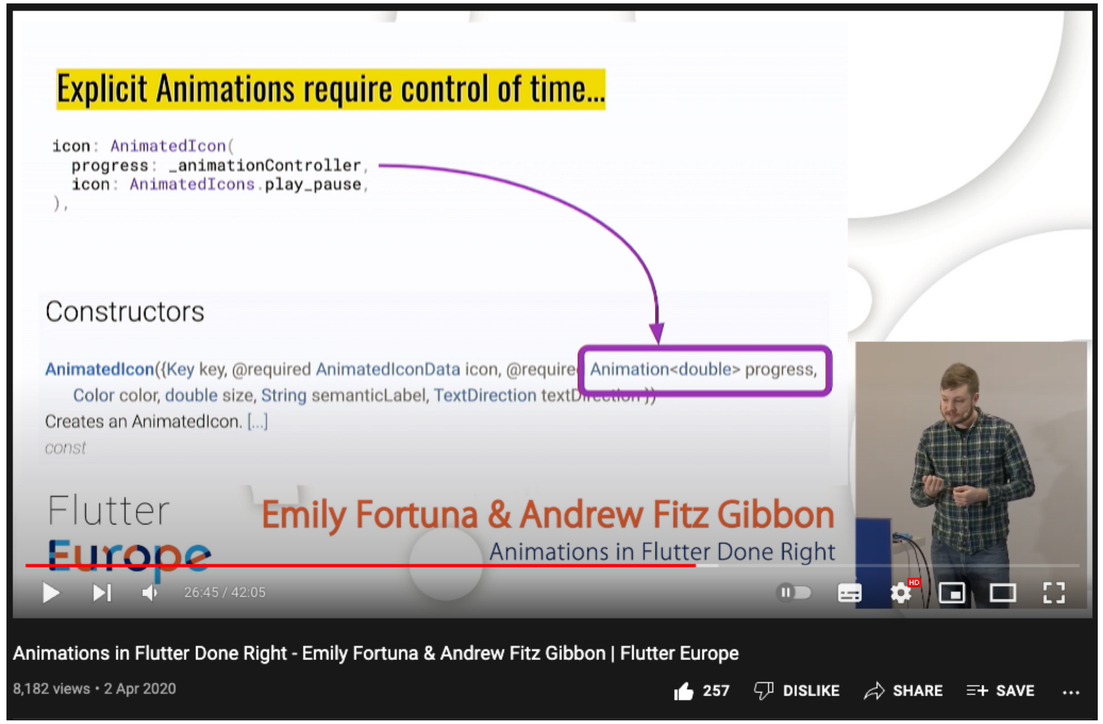


Rive

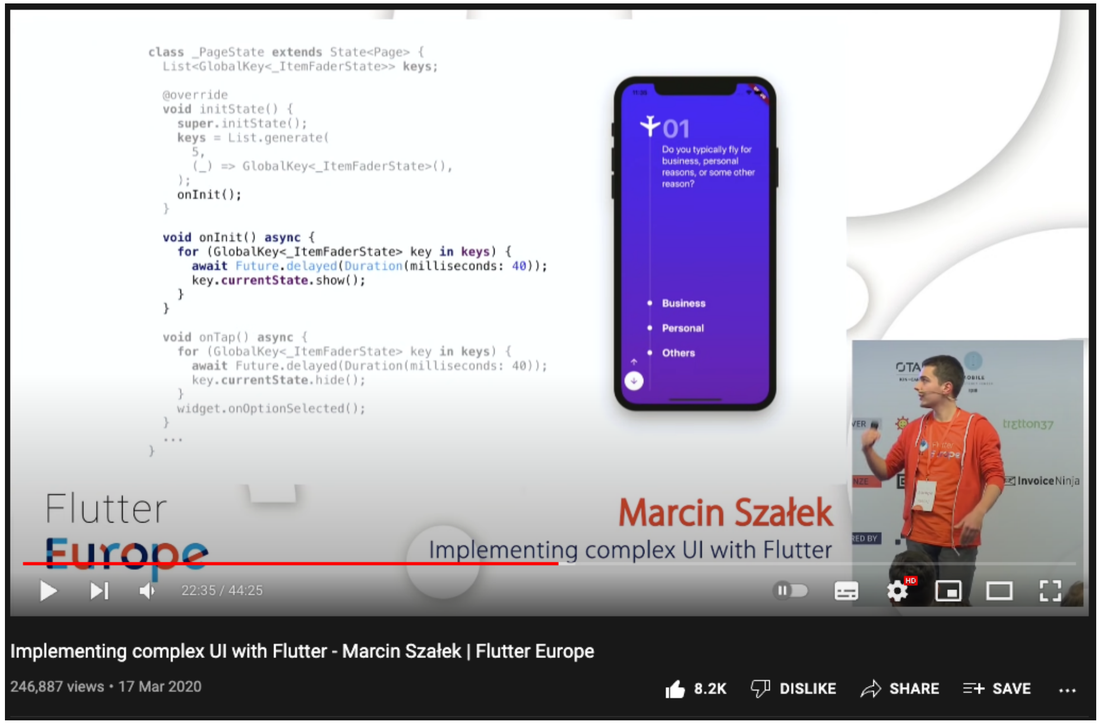
Rive


Holiday Tree Wip
Beta 0.7.1
Previewing a copy of Rive's file
38%
Design
Animate
Tree
Root
Orn1
Orn2
Orn3
engine
trainCar
trainSmoke
Orn4
Snowglobe
Tree
Lights
Branch
CookieMan
Plumbob
Rudolph Ball
Position
495.3
549.6
Scale
100%
100%
Rotation
0°
Style
Group
Layer
100%
Clipping
Constraints
Draw Order
Animations
Mock up State Machine
Tree Zoomed Out
Orn Idles
Orn1
Orn2
Orn3
Orn4
Inputs
Layers
Ornament
11
ZoomIn
ZoomOut
75%
Tree Zoomed Out
Orn1
Orn2
Orn3
Orn4
Orn5
Orn6
Orn7
Any State
Console

Next steps

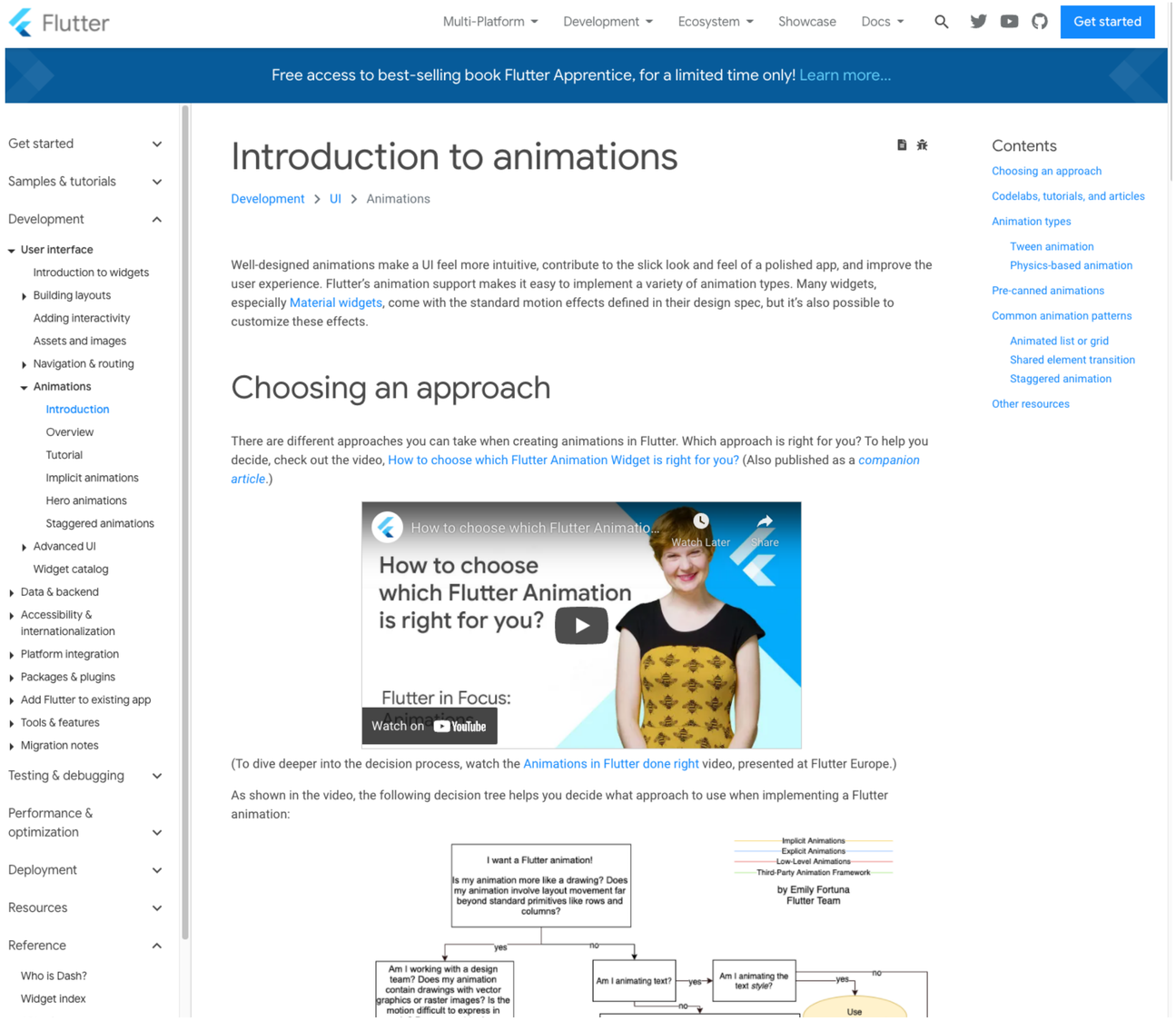


youtube.com/watch?v=wnARLByOtKA



youtube.com/watch?v=FCyoHclCqc8

Next steps



https://docs.flutter.dev/development/ui/animations

Thank you :)

Marcin Szałek

@marcin_szalek

marcin@fidev.io

Questions?

Marcin Szałek

@marcin_szalek

marcin@fidev.io