



Animations

It can be very simple



Implicitly Animated Widgets





AnimatedAlign

AnimatedContainer

AnimatedDefaultTextStyle

AnimatedOpacity

AnimatedPadding

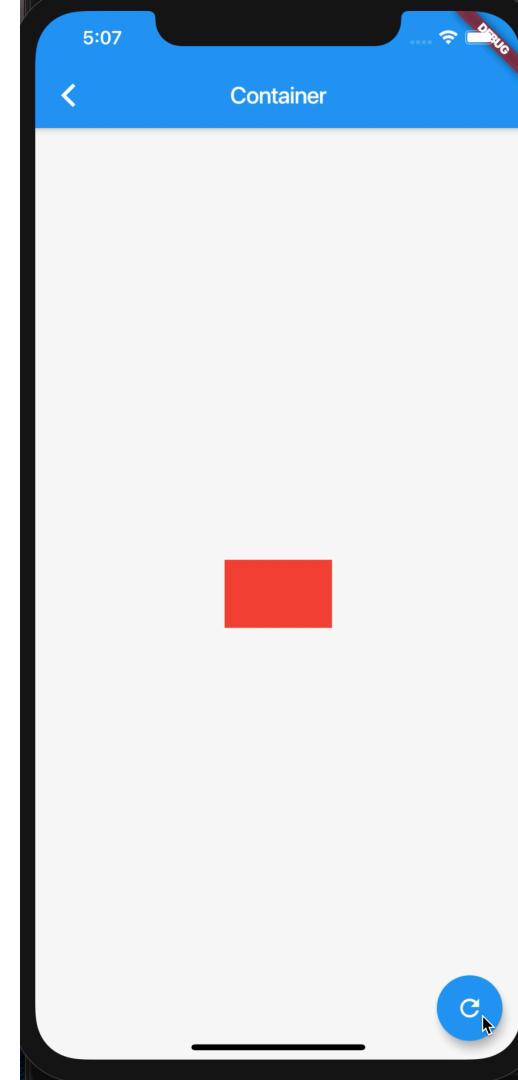
AnimatedPhysicalModel

AnimatedPositioned

AnimatedPositionedDirectional

AnimatedContainer

```
AnimatedContainer(  
    color: Colors.red,  
    width: width,  
    height: height,  
    duration: const Duration(seconds: 1),  
)
```



AnimatedPositioned

```
AnimatedPositioned(  
    top: shouldShowText ? 200 : 300,  
    left: 100,  
    child: Container(  
        width: 200,  
        height: 100,  
        color: Colors.blue,  
    ),  
    duration: const Duration(seconds: 2),  
    curve: Curves.easeInOut,  
)
```

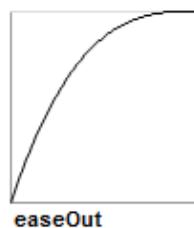
AnimatedOpacity

```
AnimatedOpacity(  
    opacity: shouldShowText ? 1 : 0,  
    child: const Text('Hello new world'),  
    duration: const Duration(seconds: 2),  
    curve: Curves.easeIn,  
)
```

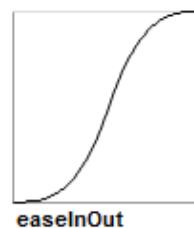
Curves



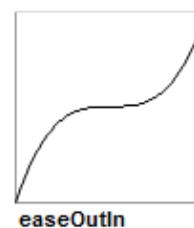
easeIn



easeOut



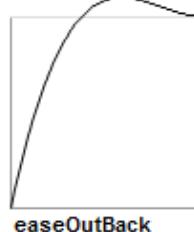
easeInOut



easeOutIn



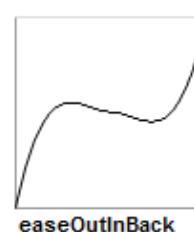
easeInBack



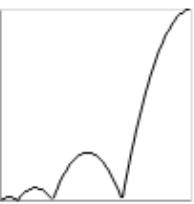
easeOutBack



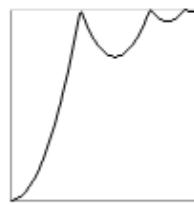
easeInOutBack



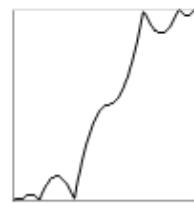
easeOutInBack



easeInBounce



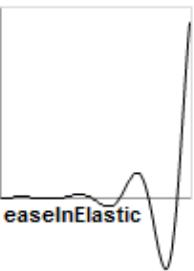
easeOutBounce



easeInOutBounce



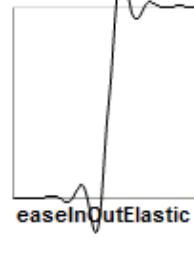
easeOutInBounce



easeInElastic



easeOutElastic



easeInOutElastic



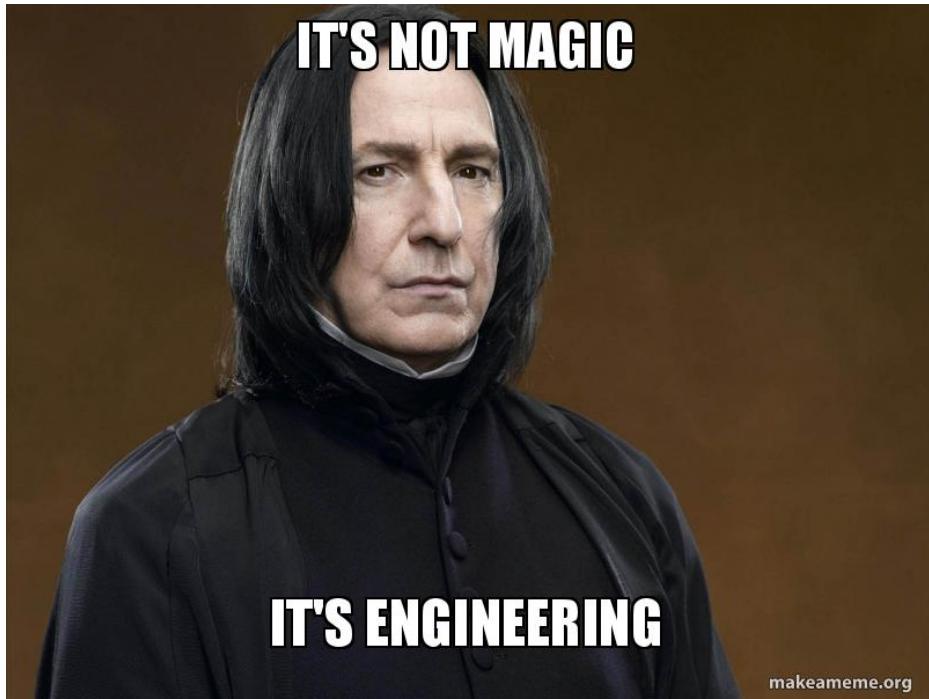
easeOutInElastic

4:52

Animated position

DEBUG

Curves Interval



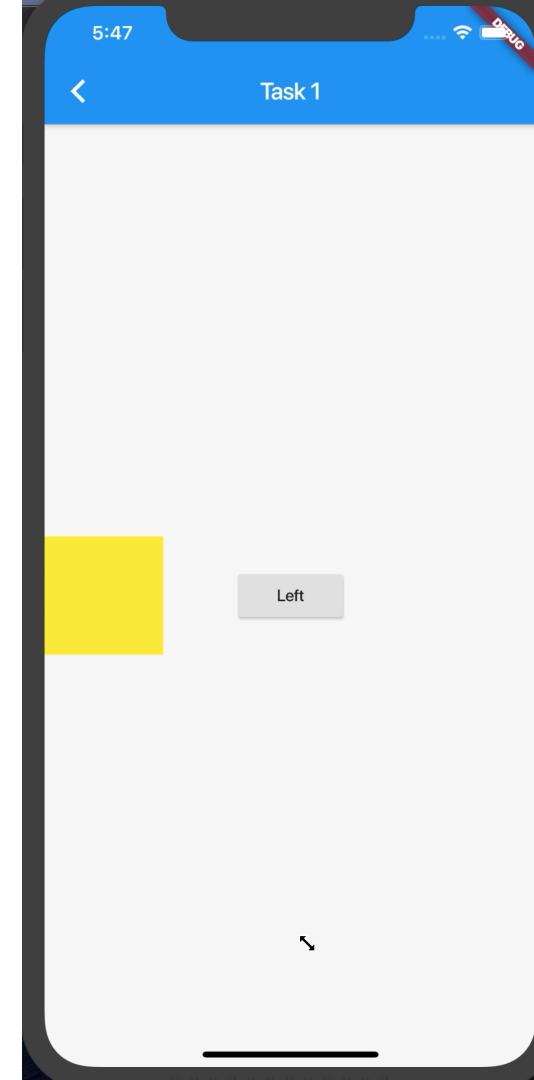
Hello world





It's coffee time

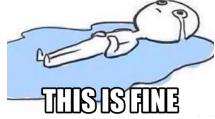
15 minutes

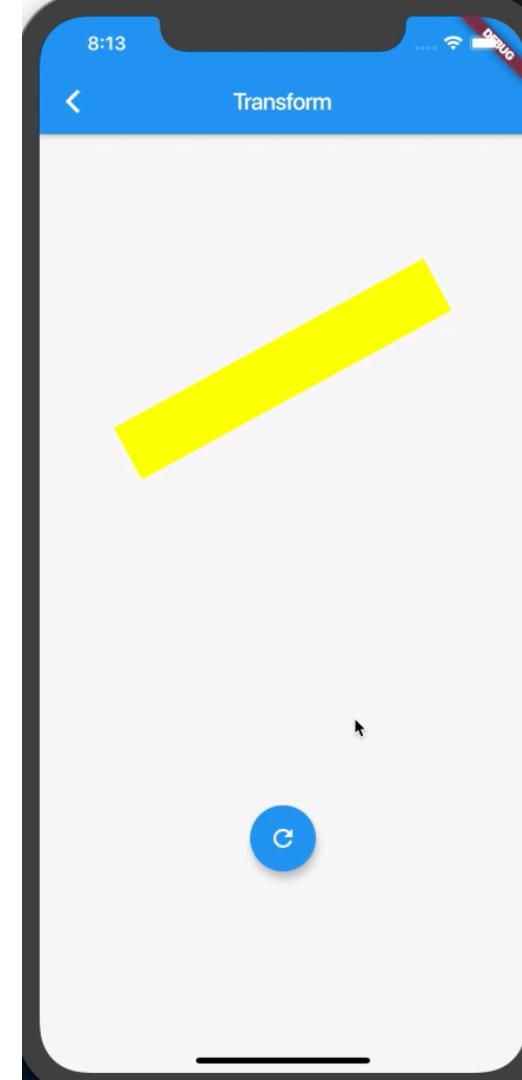


Animation and AnimationController



Transform

```
_controller =  
    AnimationController(vsync: this, duration: const Duration(seconds: 2));  
_animation = Tween(begin: 0.0, end: 200.0).animate(_controller)  
.addListener(() {  
  
    setState(() {});  
  
});  
  
    transform.rotate(  
        angle: -2 * pi * _animation.value / 200,  
        child: Container(  
            width: 300,  
            height: 50,  
            color: Colors.yellowAccent,  
        ),  
),
```





C

AnimatedBuilder

```
AnimatedBuilder(  
    child: Container(  
        width: 300,  
        height: 50,  
        color: Colors.yellowAccent,  
    ),  
    animation: _controller,  
    builder: (BuildContext context, Widget child) {  
        return Transform.rotate(  
            angle: -2 * pi * _animation.value / 200,  
            child: child,  
        );  
    },
```



FadeTransition

```
animation = CurvedAnimation(parent: controller, curve: Curves.easeIn);
```

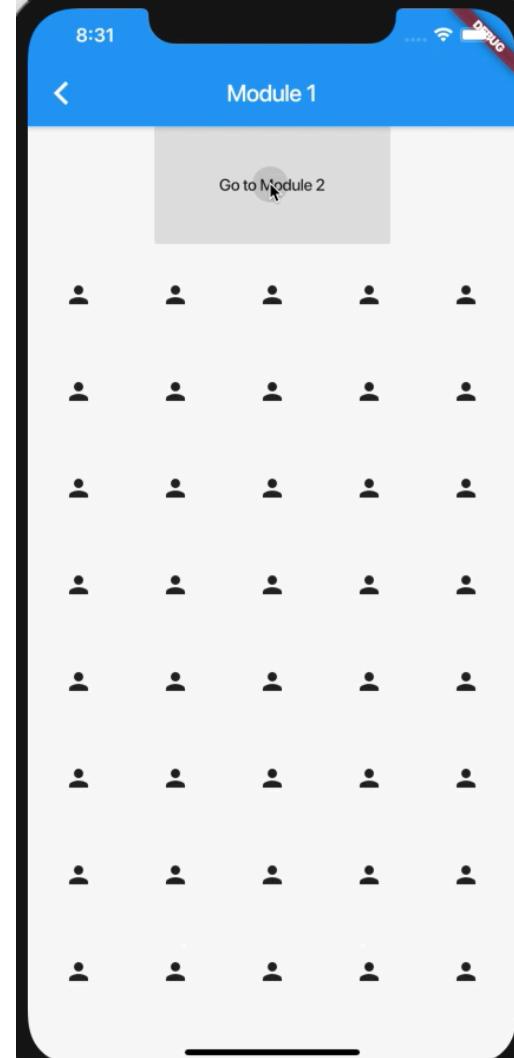
```
FadeTransition(  
    opacity: animation,  
    child:  
        Row(mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
        Icon(  
            Icons.check,  
            size: 100.0,  
            color: Colors.green,  
        ),  
    ]))
```



Hero

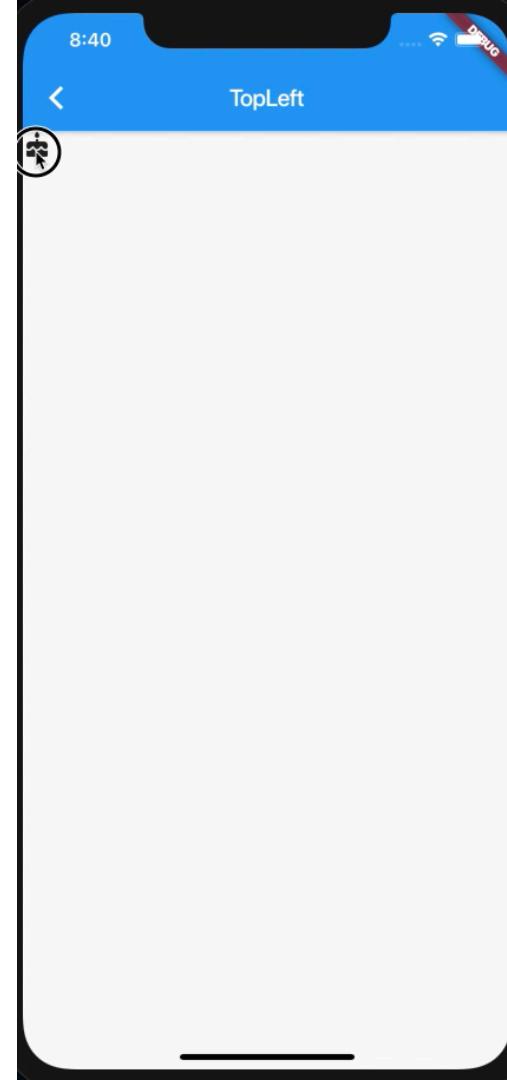


Hero Transition



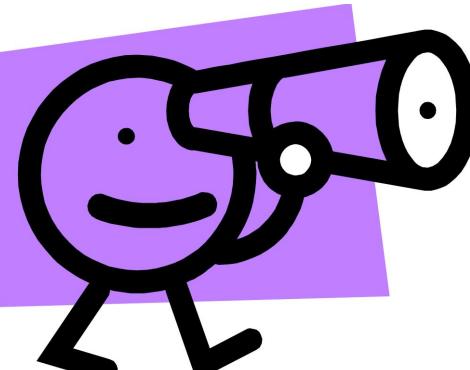
Hero Transition

Hero(tag: 'My hero icon', child: Icon(Icons.cake),)



Have fun and explore more

https://github.com/rxlabz/flutter_animation_explorer



12 Principles of animation

Homework

1. In the final app from module 1 add animation for image transition between screens.
2. Add animation for changing image when tap “refresh”.
3. Elements on the details page should appear with some delay.