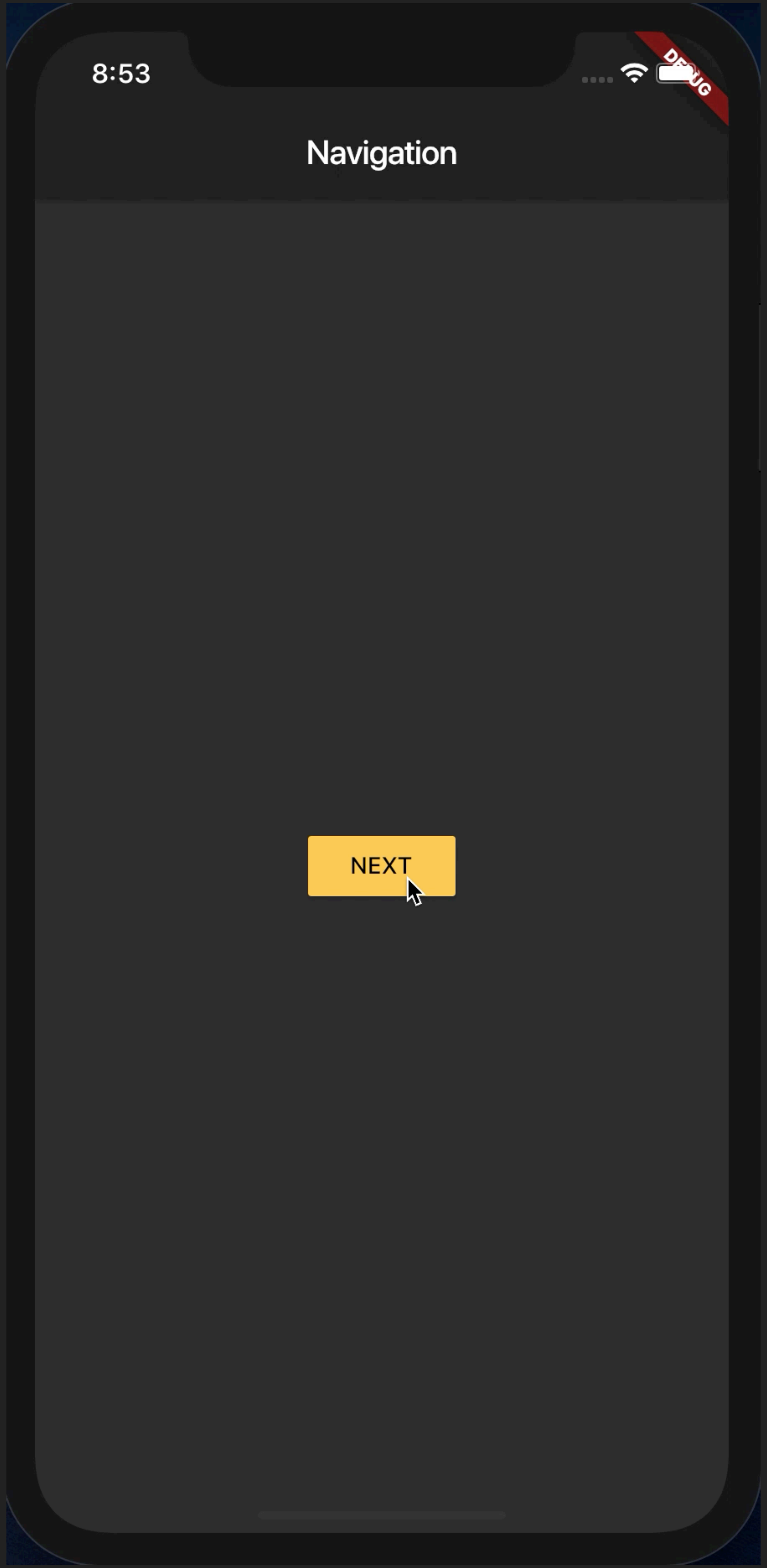
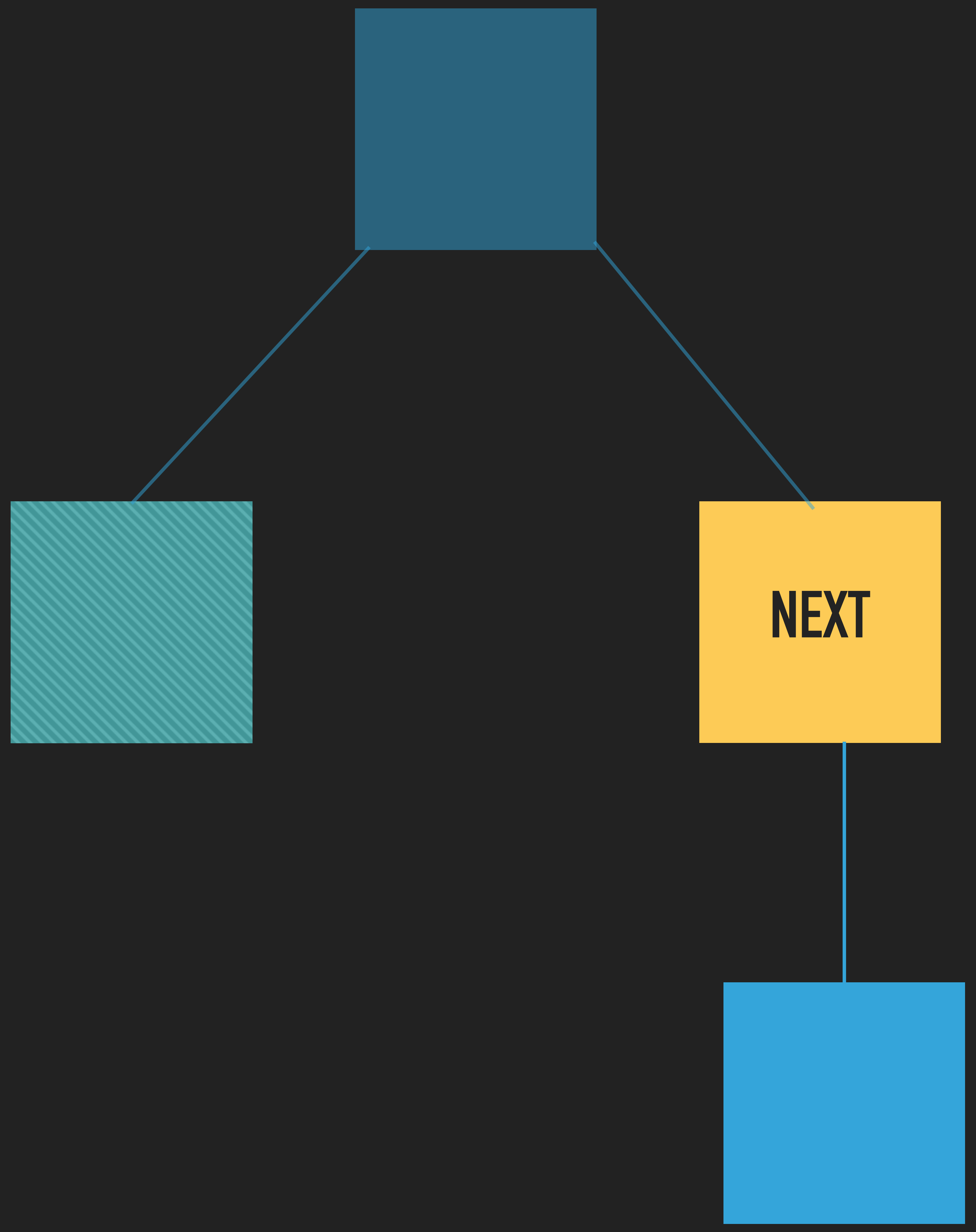
ИЛЬЯ СЕДОВ

НАВИГАЦИЯ И АНИМАЦИЯ

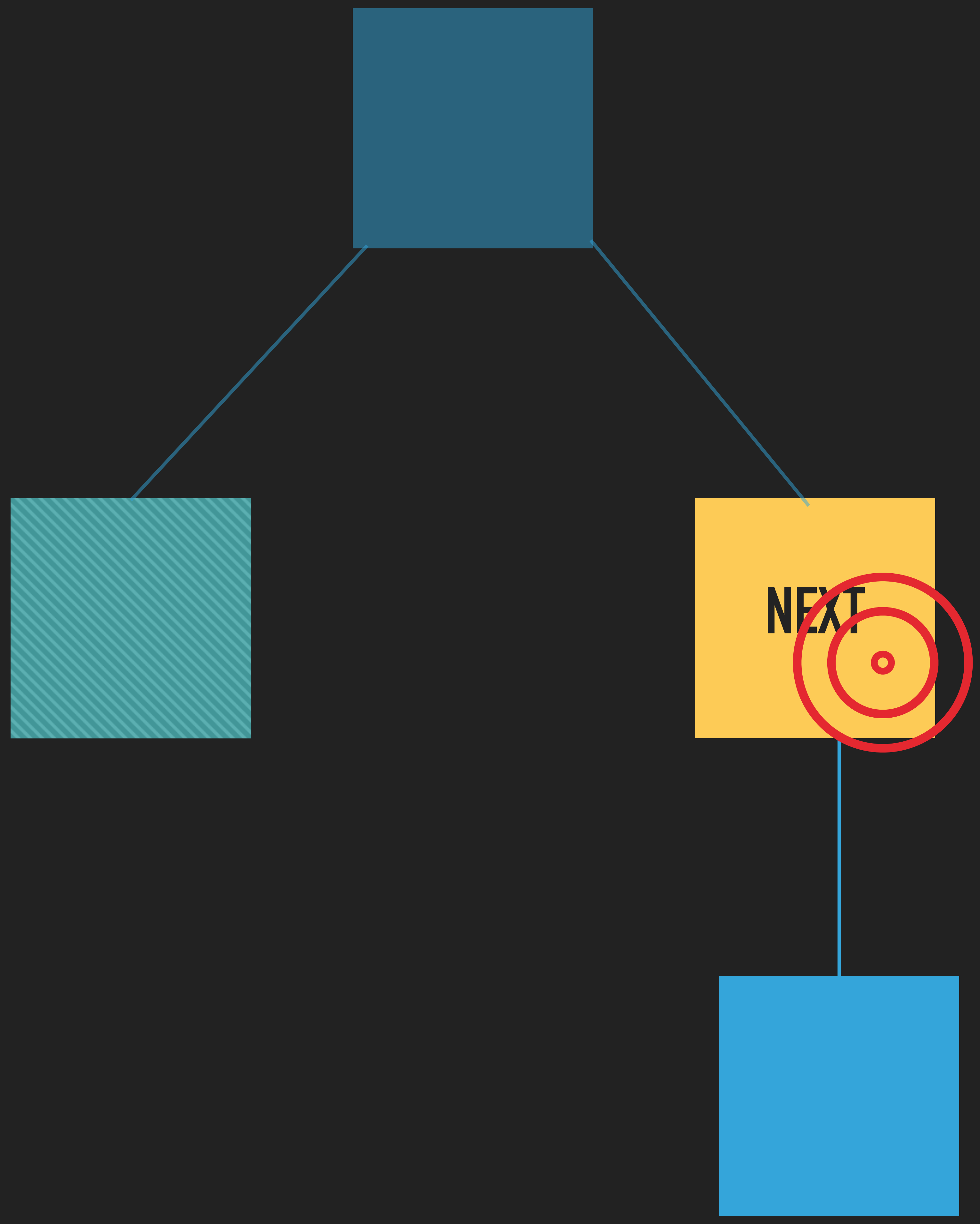

8:53
DEBUG
Navigation
NEXT

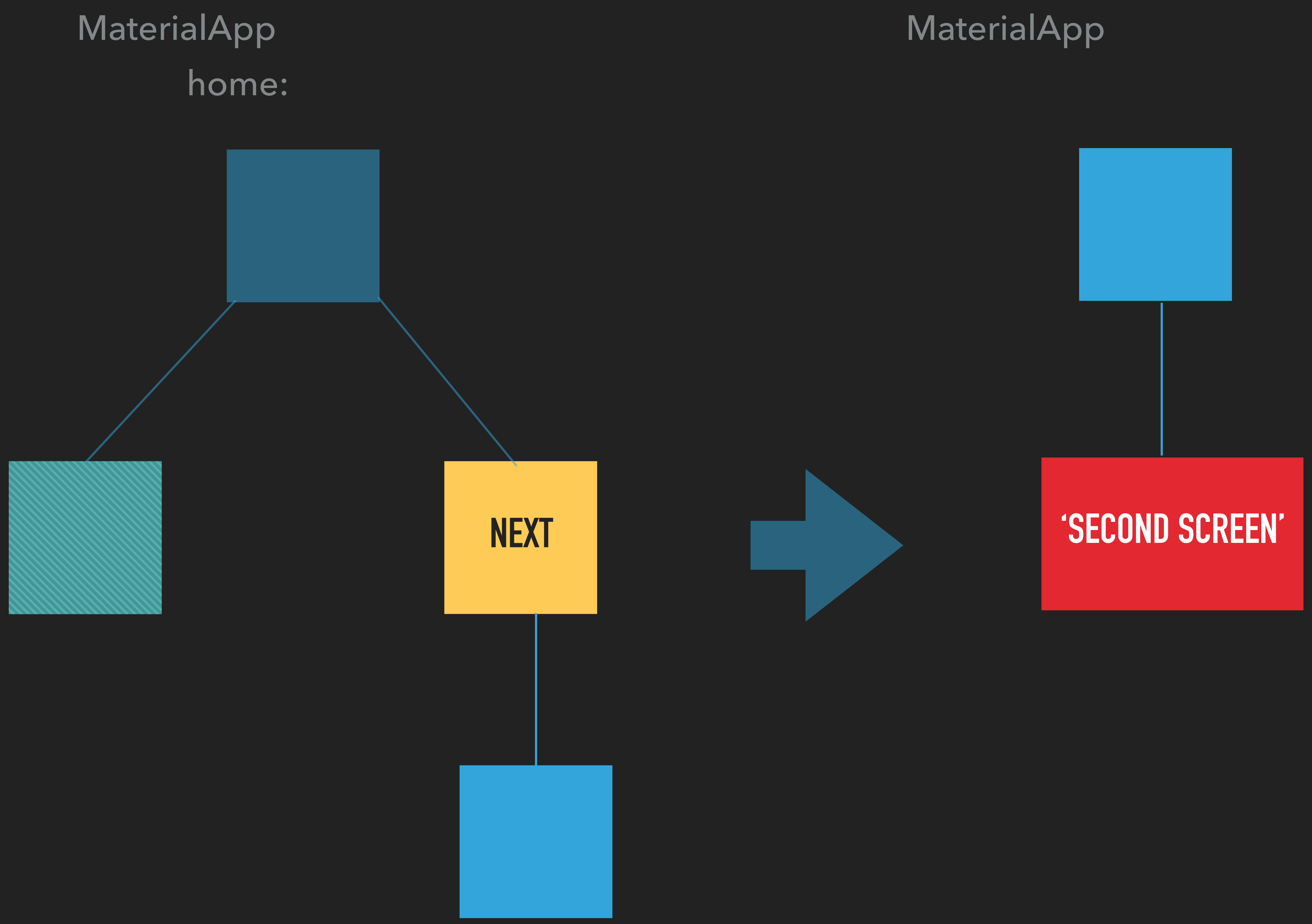
MaterialApp
home:

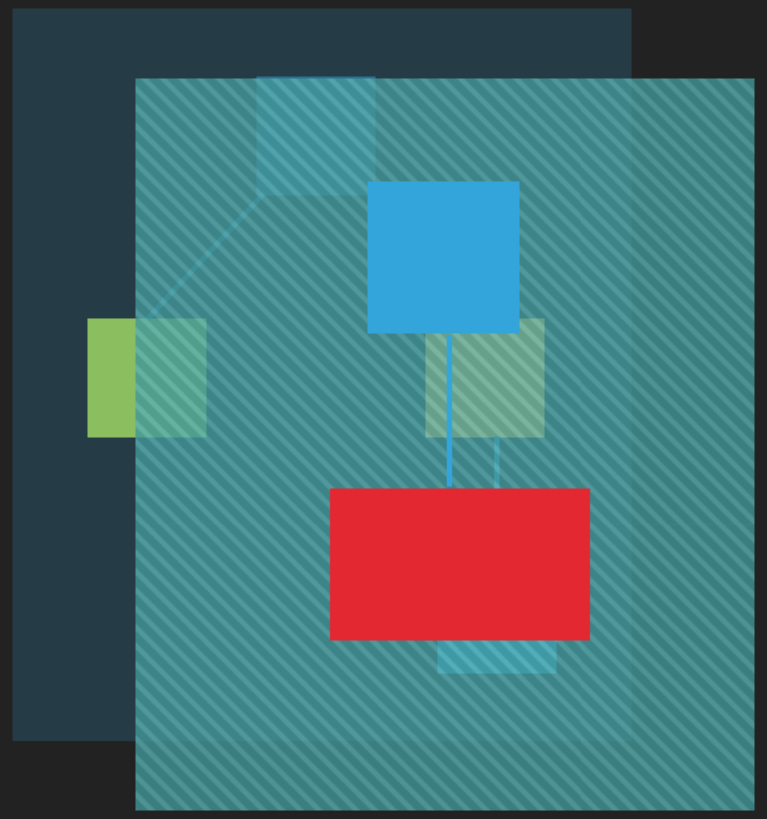


MaterialApp

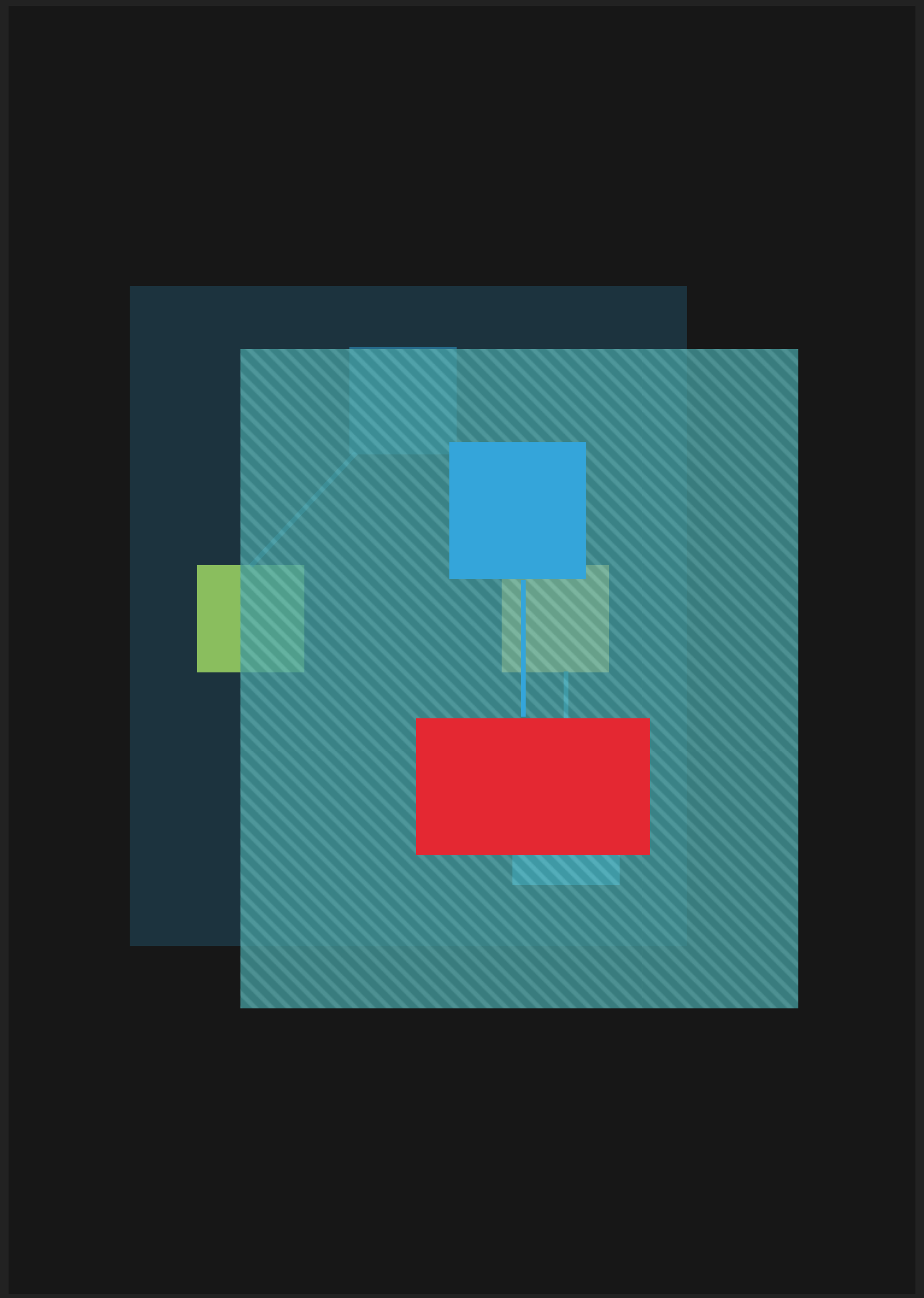
home:



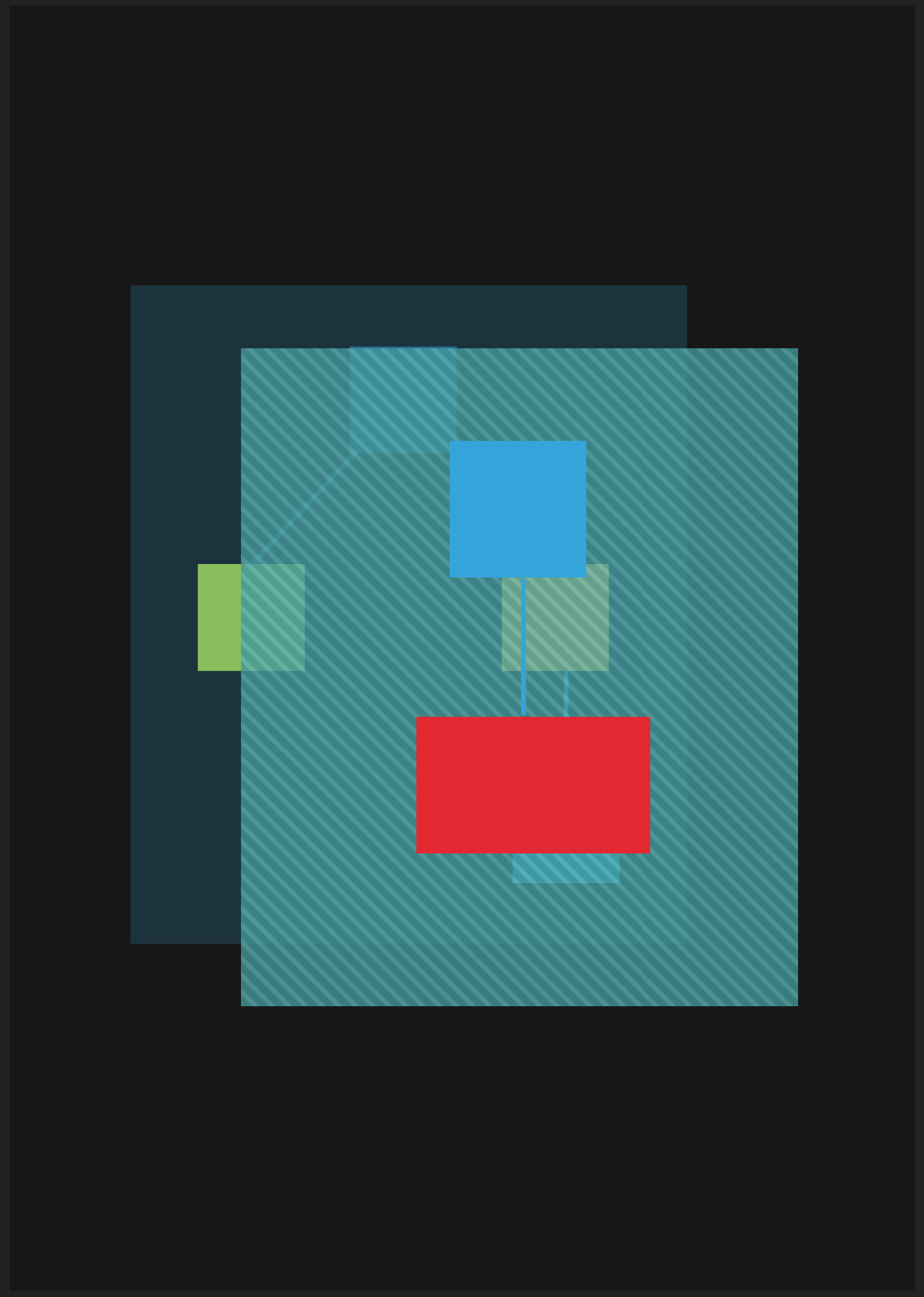

MaterialApp
home:
NEXT
MaterialApp
'SECOND SCREEN'

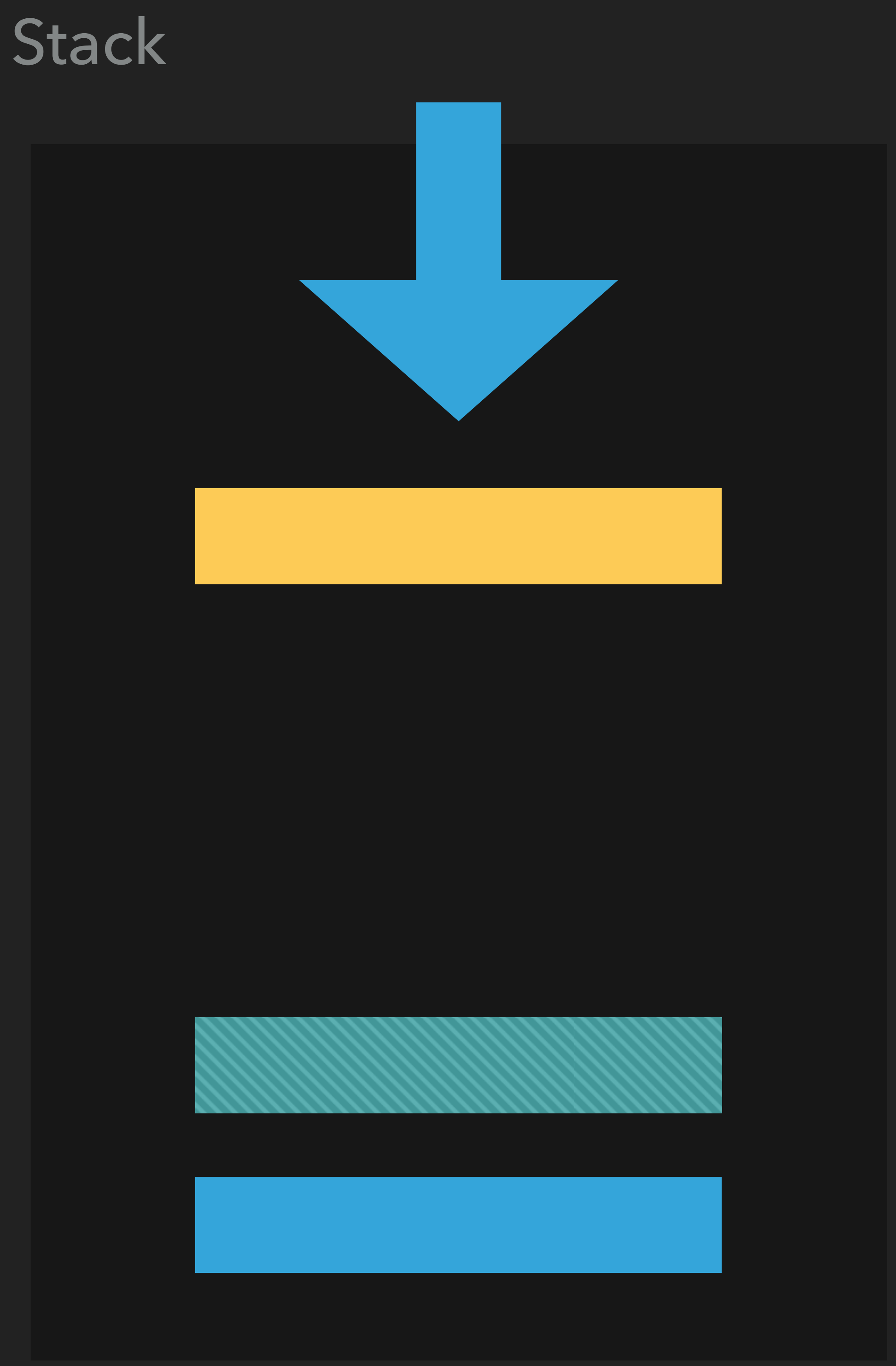


OVERLAY

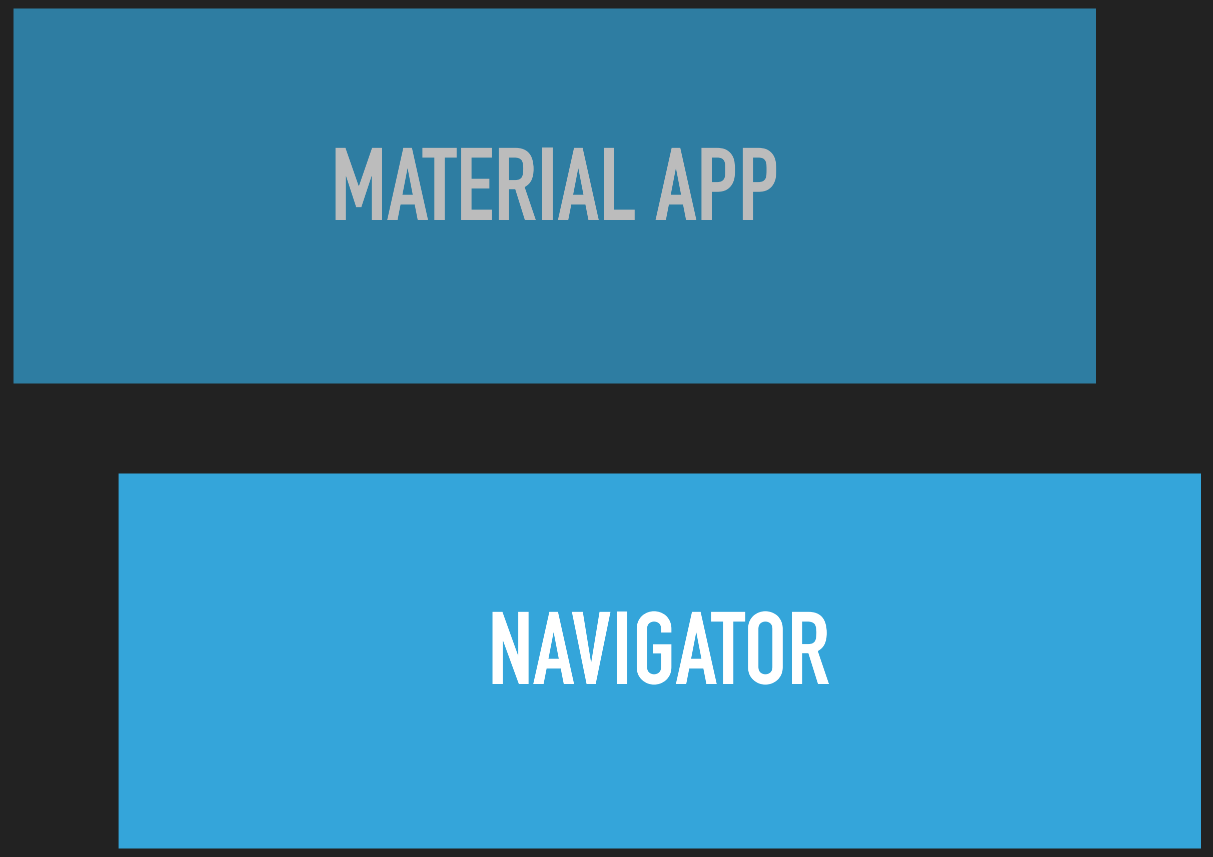


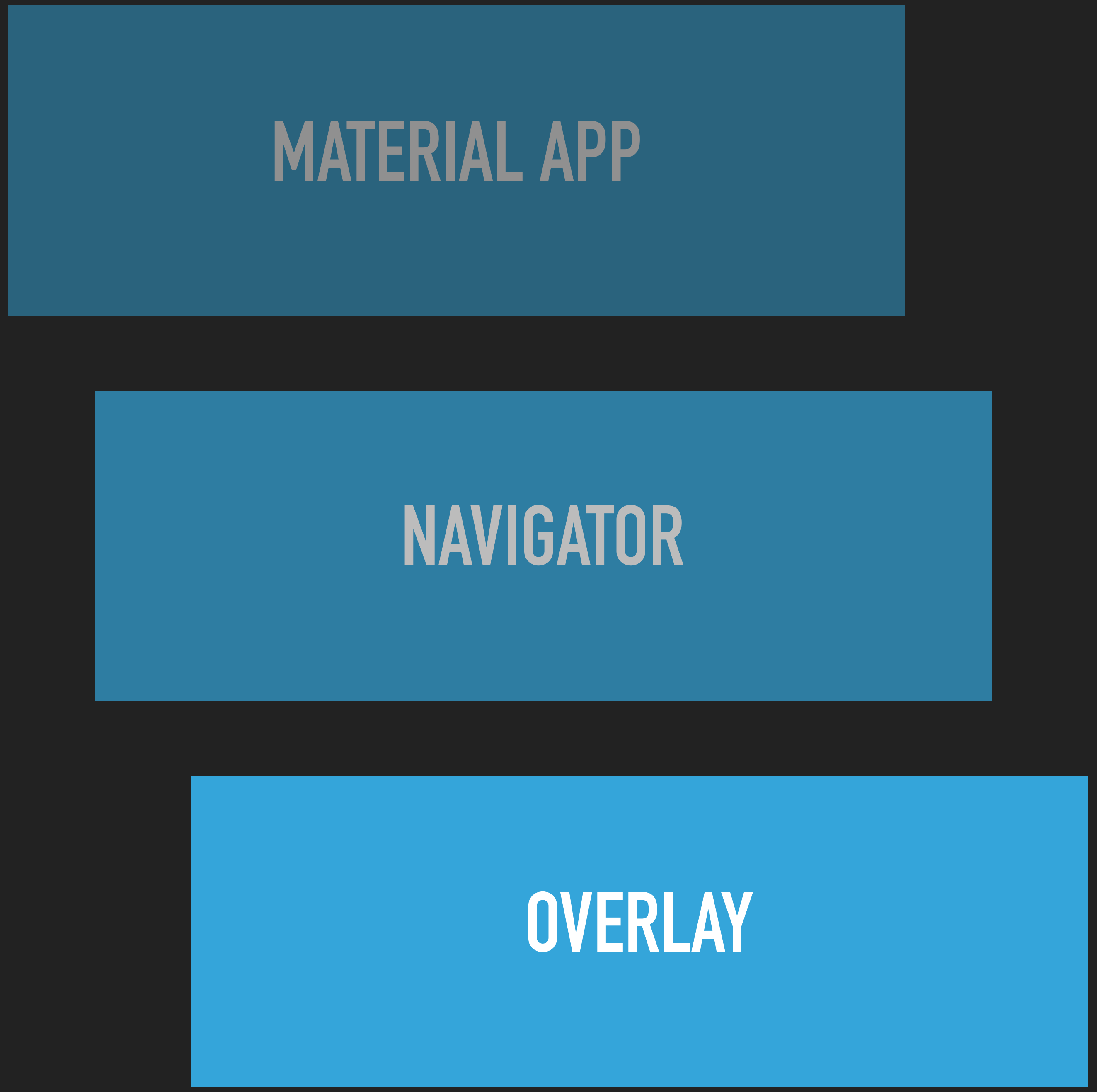
OVERLAY

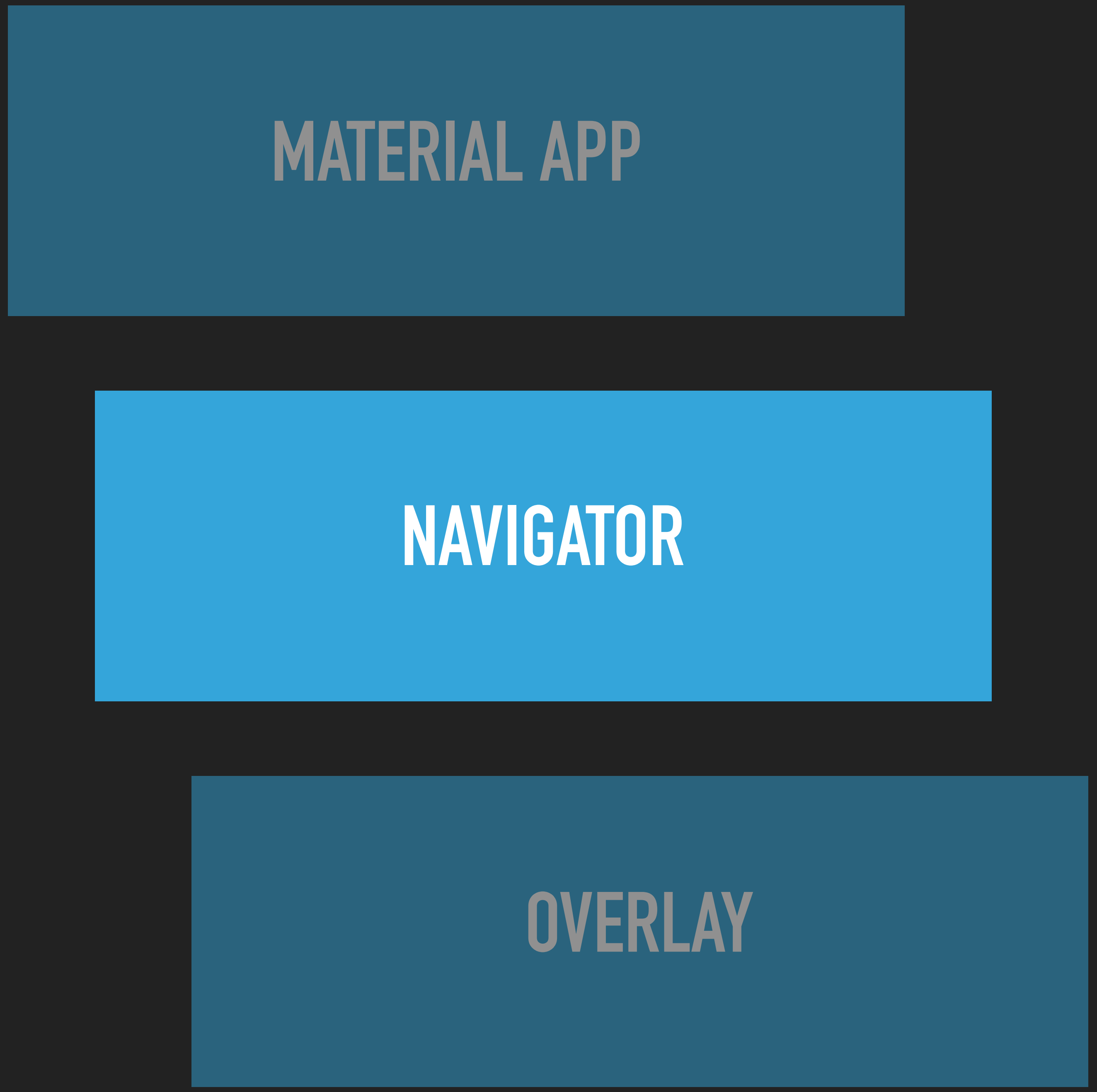


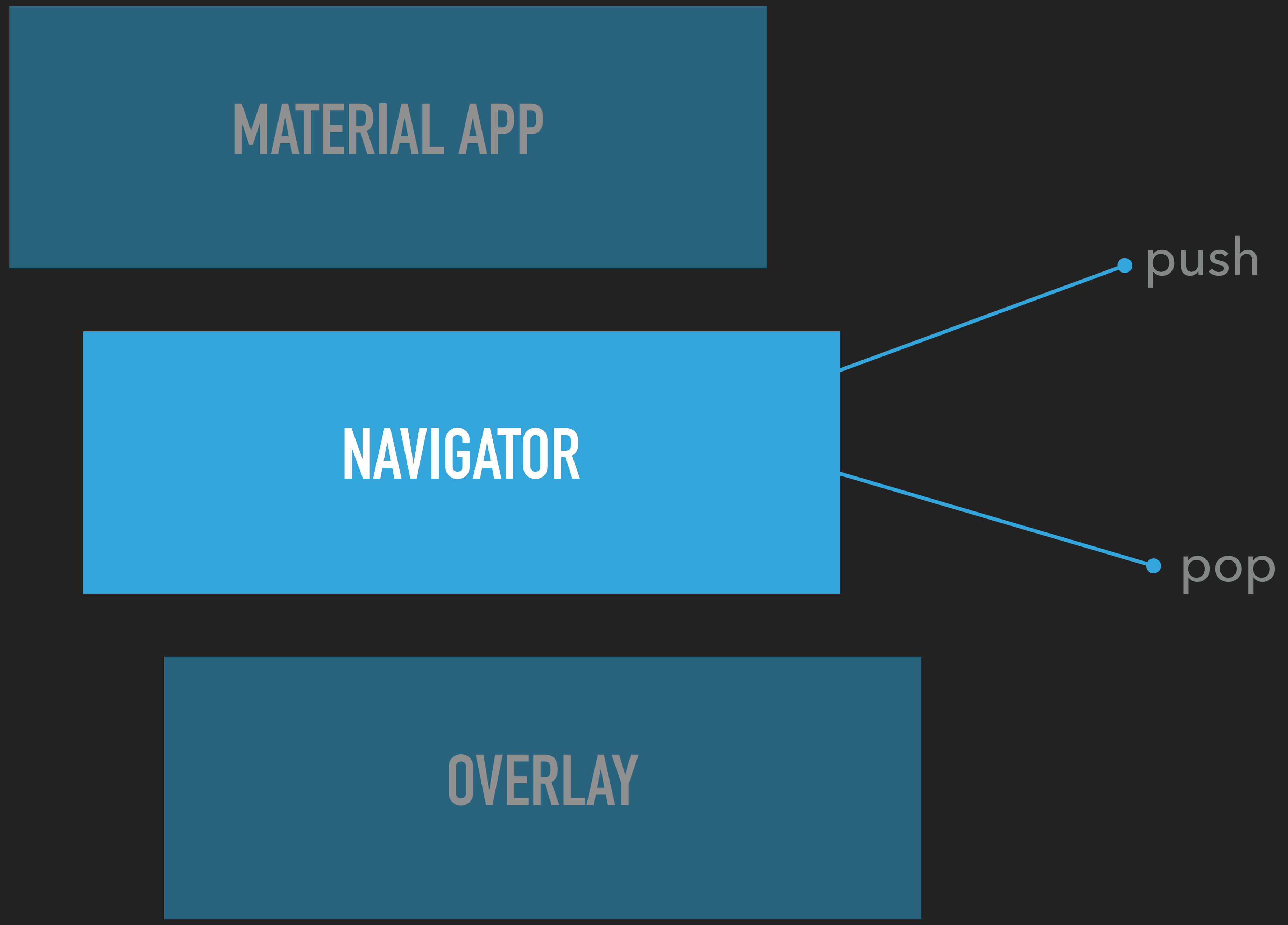


MATERIAL APP


MATERIAL APP
NAVIGATOR


MATERIAL APP
NAVIGATOR
OVERLAY


MATERIAL APP
NAVIGATOR
OVERLAY


MATERIAL APP
NAVIGATOR
push
pop
OVERLAY

Помедленеее

PAGE ROUTE

- Сборка widget-а экрана
- Анимирование перехода
- Добавление экрана в Overlay

NAVIGATOR

- Предоставляет Overlay
- Управление жизненным циклом PageRoute
- Хранение истории PageRoute

```
Navigator.push(context,
               MaterialPageRoute(builder: (context) {
                  return SecondScreen();
                }));
```

CONTEXT

- Объект класса BuildContext
- Управляет расположением widget-а в дереве

main.dart

```dart
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation',
      theme: ThemeData.dark(),
      home: MyHomePage(),
    );
  }
}
```

main.dart

```
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation',
      theme: ThemeData.dark(),
      home: MyHomePage(),
    );
  }
}
```

```
class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Navigation'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
      title: 'Navigation',
      theme: ThemeData.dark(),
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Navigation'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
```

main.dart

```
            RaisedButton(
              color: Color(0xFFFDCB56),
              child: Text(
                'NEXT',
                style: TextStyle(color: Colors.black),
              ),
              onPressed: () {
                Navigator.push(context, MaterialPageRoute(builder: (context) {
                  return SecondScreen();
                }));
              },
            ),
          ]
```

main.dart

```dart
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation',
      theme: ThemeData.dark(),
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Navigation'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
              RaisedButton(
                color: Color(0xFFFDCB56),
                child: Text(
                  'NEXT',
                  style: TextStyle(color: Colors.black),
                ),
                onPressed: () {
                  Navigator.push(context, MaterialPageRoute(builder: (context) {
                    return SecondScreen();
                  }));
                },
              ),
            ]
```

ROUTING TABLE

```
Navigator.push(context,
               MaterialPageRoute(builder: (context) {
                  return SecondScreen();
                }));
```

ROUTING TABLE

```
~~Navigator.push(context, MaterialPageRoute(builder: (context) {~~
                  ~~return SecondScreen();~~
                ~~}));~~
```

```
Navigator.pushNamed(context, '/second');
```

ROUTING TABLE

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation',
      theme: ThemeData.dark(),
```

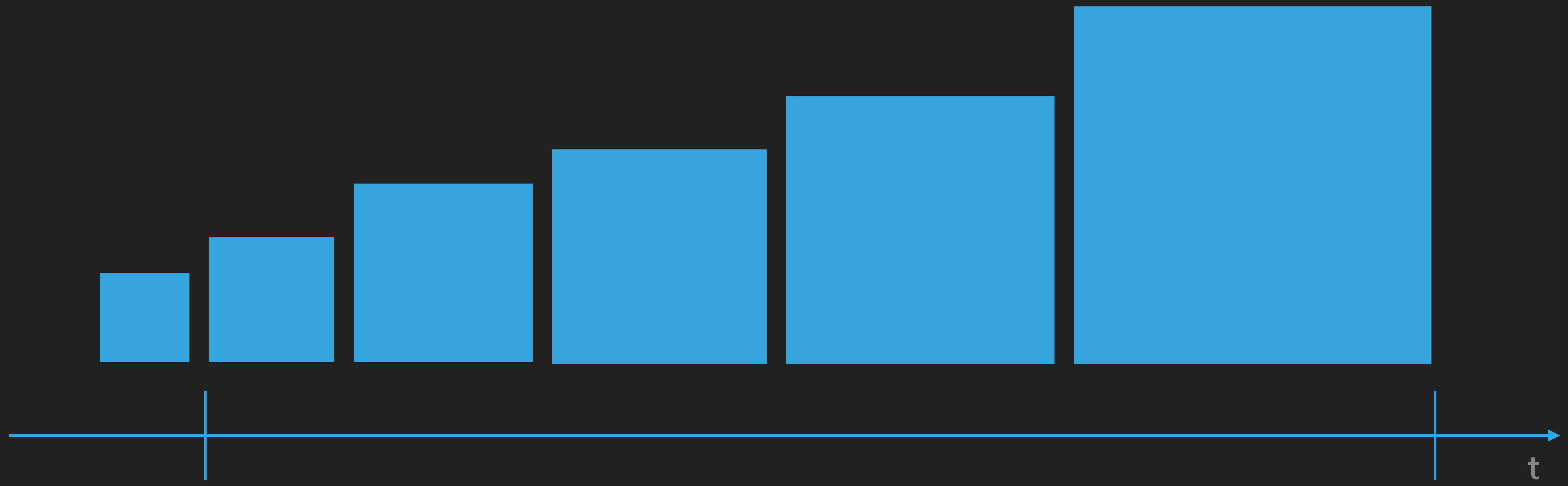
ROUTING TABLE

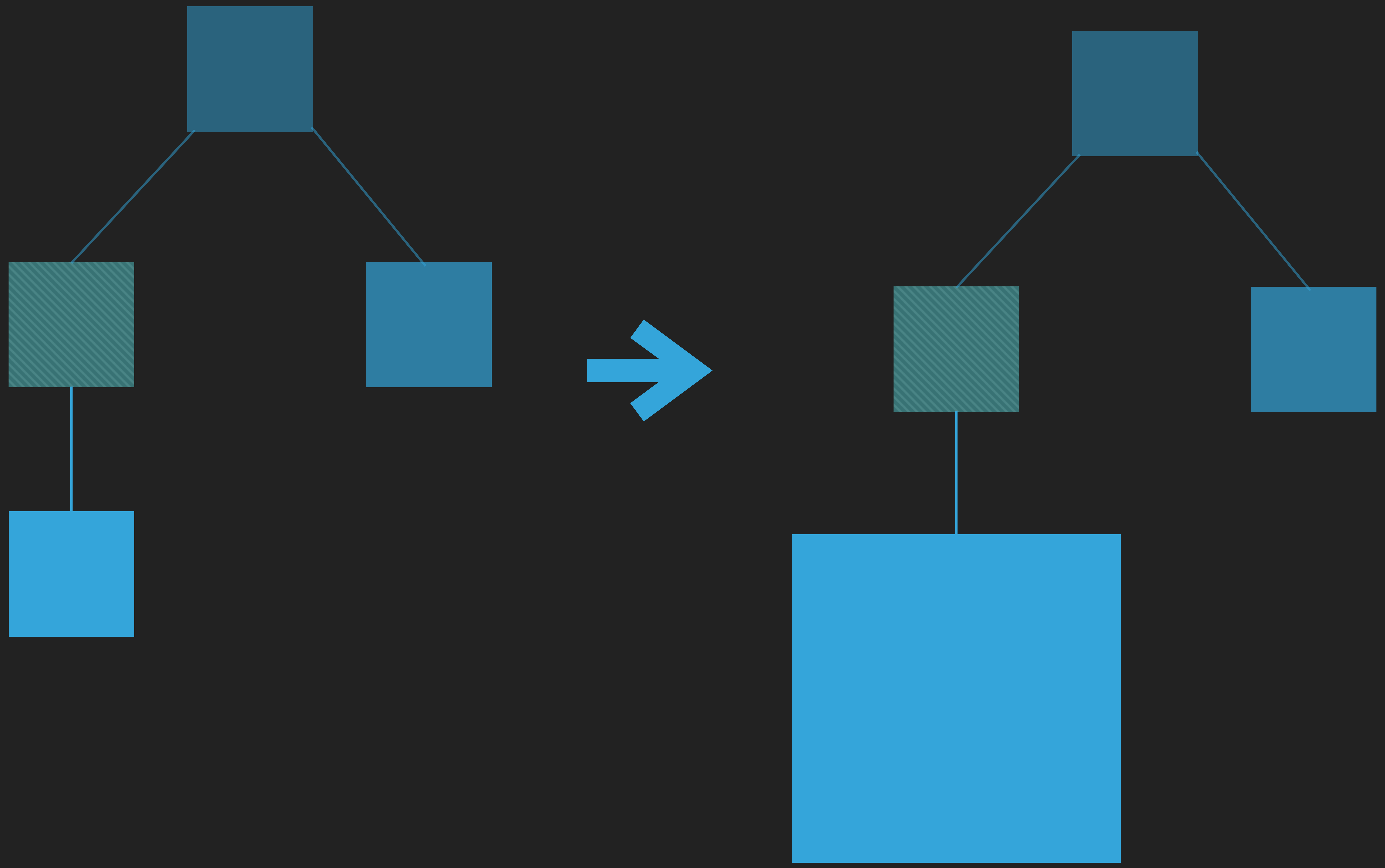
```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation',
      theme: ThemeData.dark(),
      // home: MyHomePage(),
      routes: {
        '/': (context) => MyHomePage(),
        '/second': (context) => SecondScreen()
      },
```

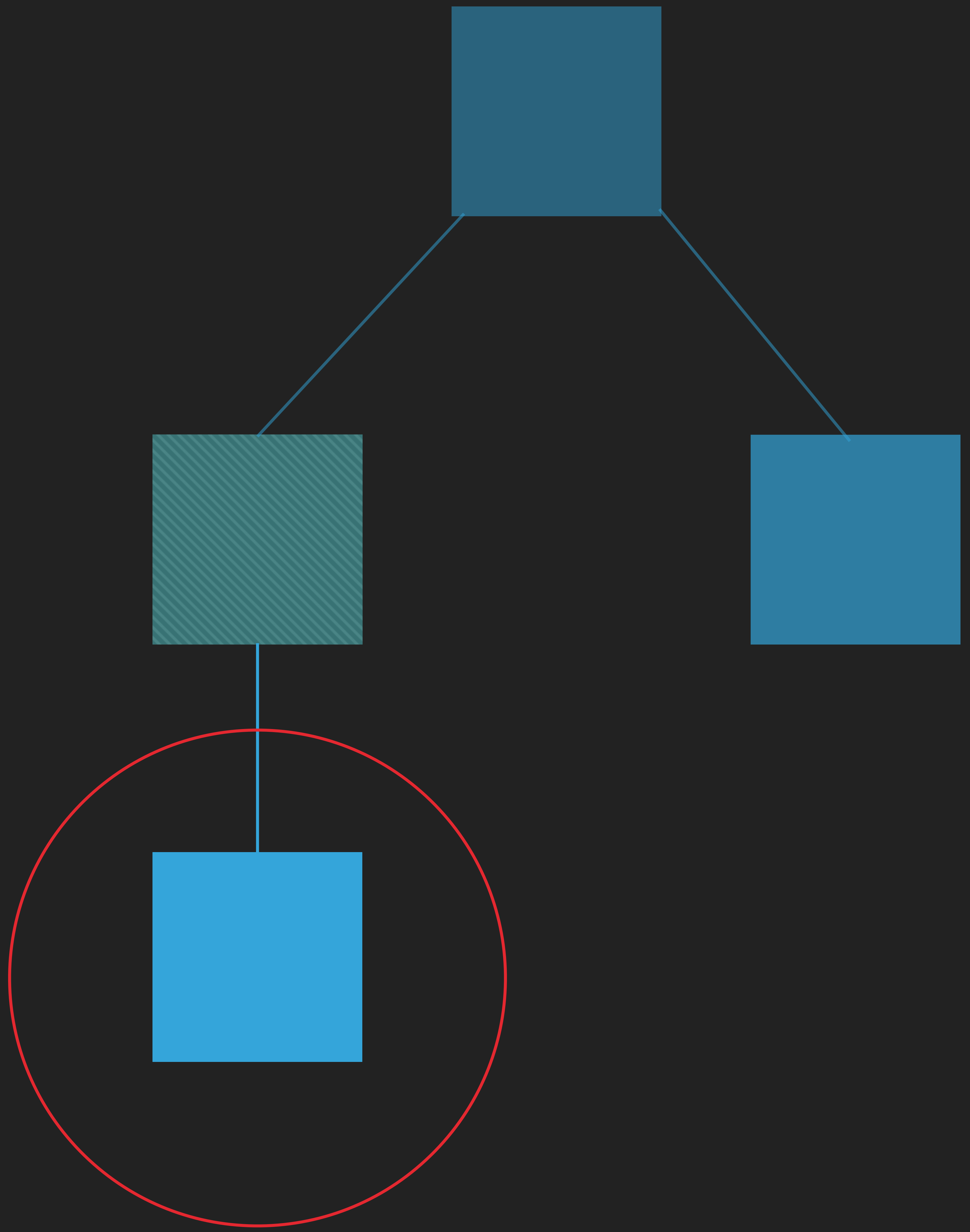
ANIMATIONS

ПЛАН

- Что такое анимации?
- Как они работают?
- Как делать анимации во Flutter?

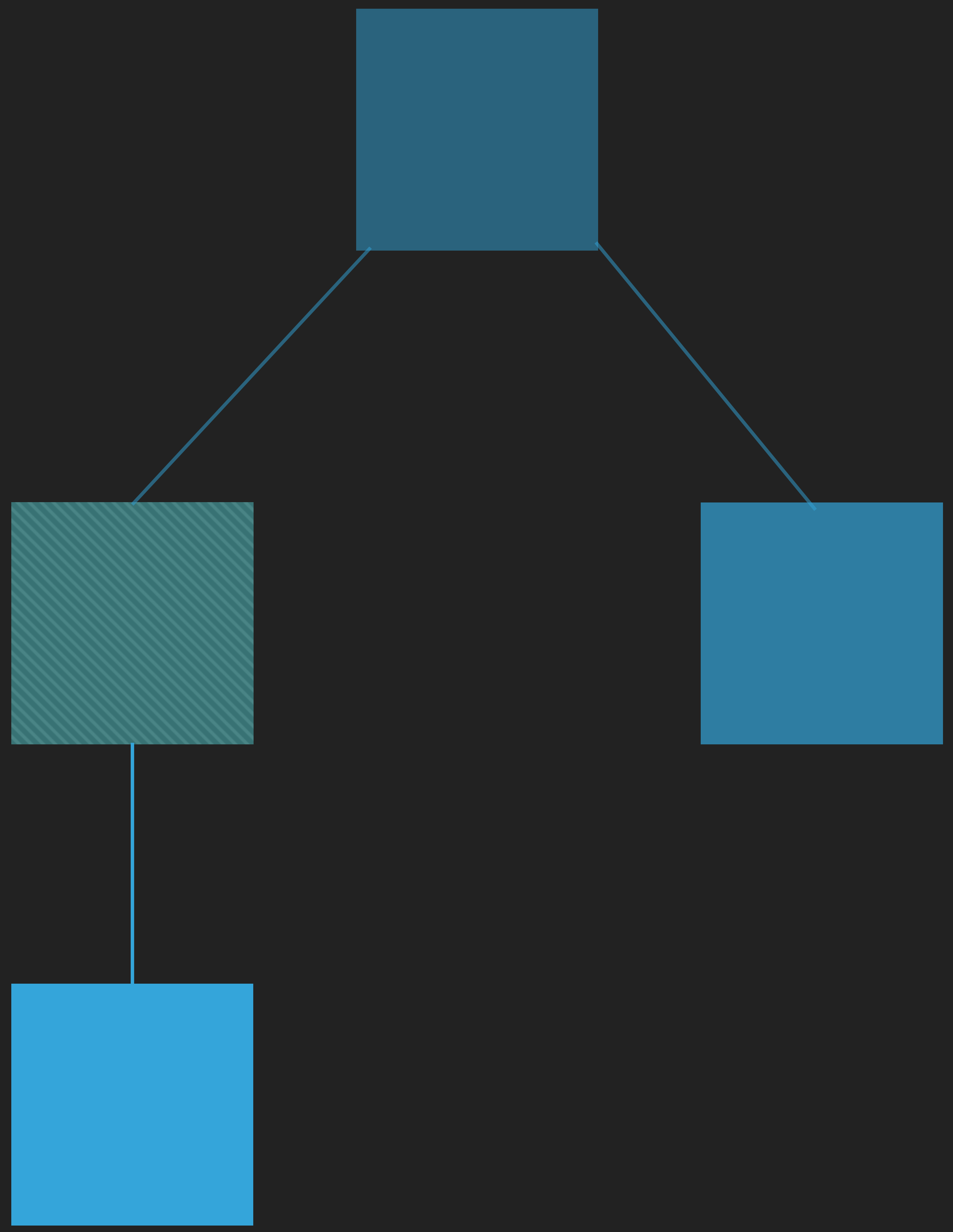

t

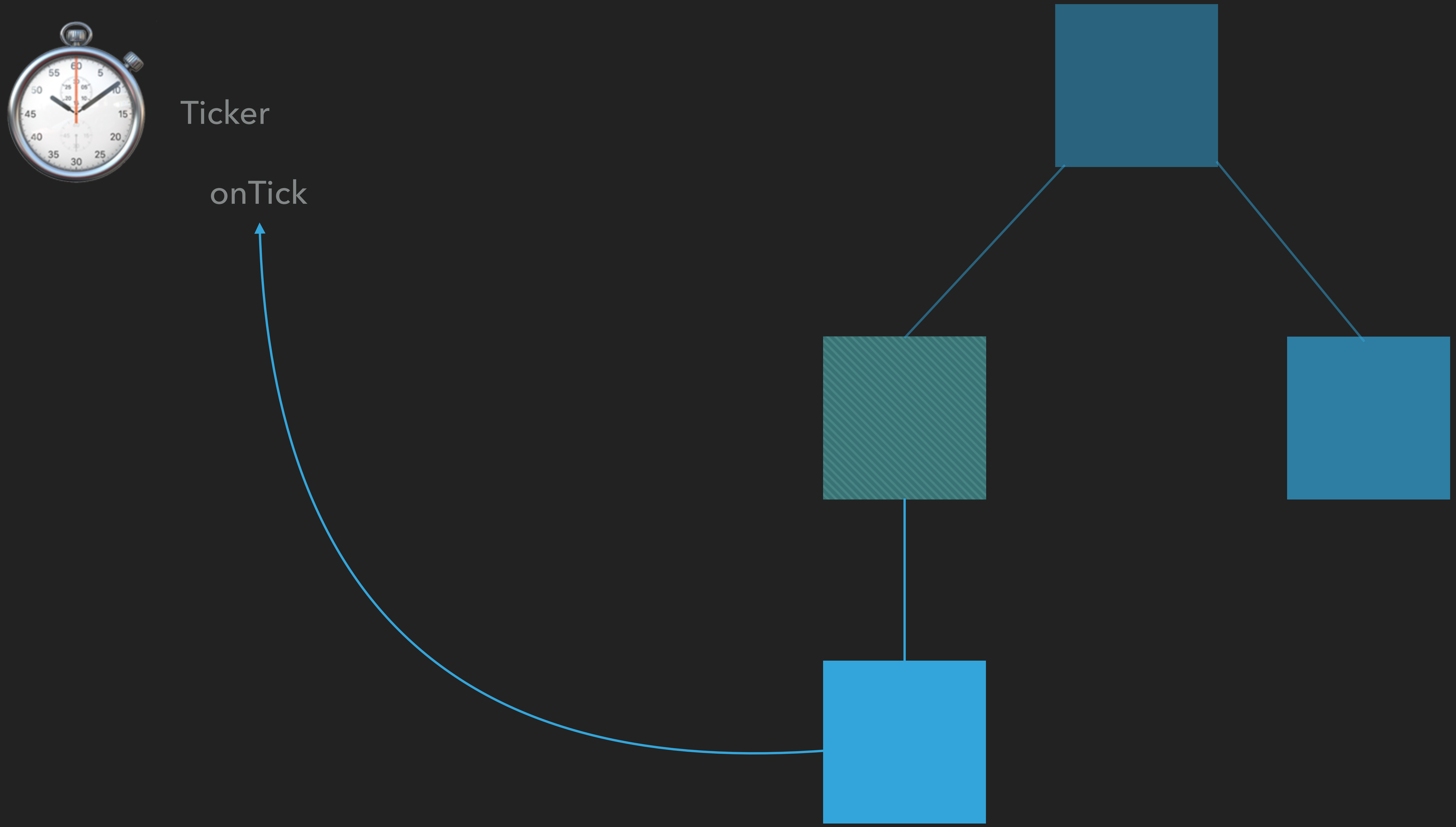


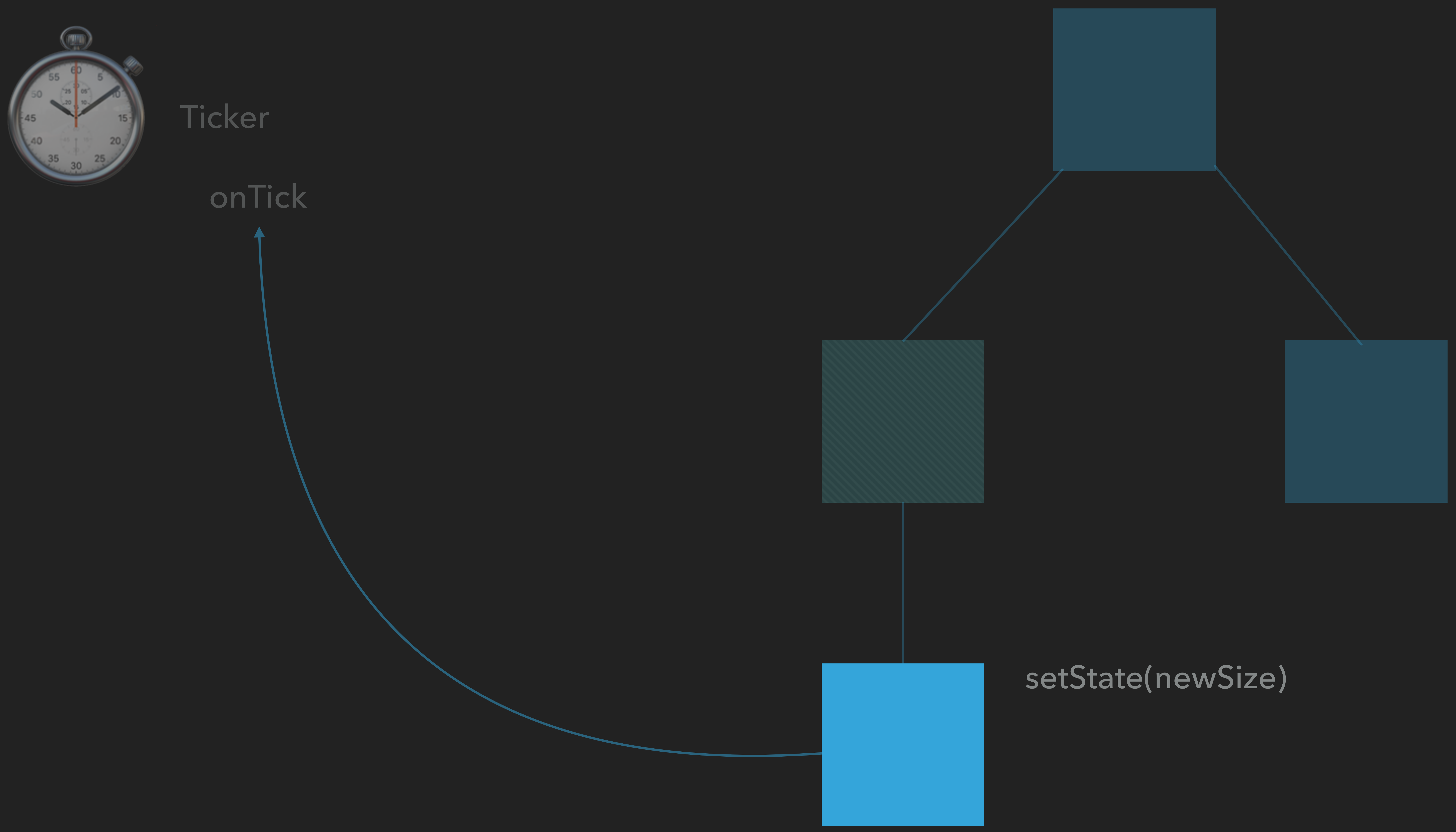


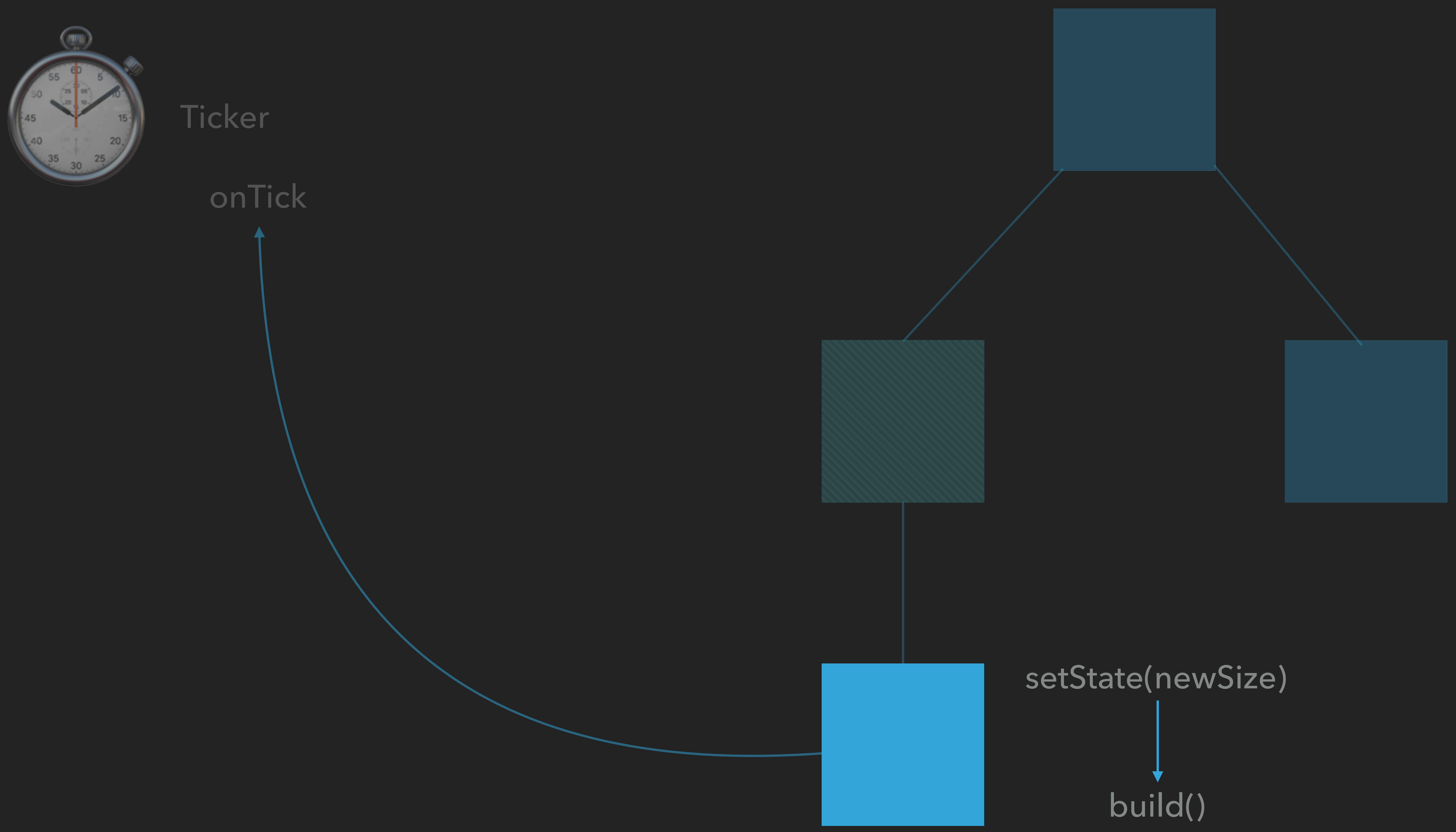
Ticker

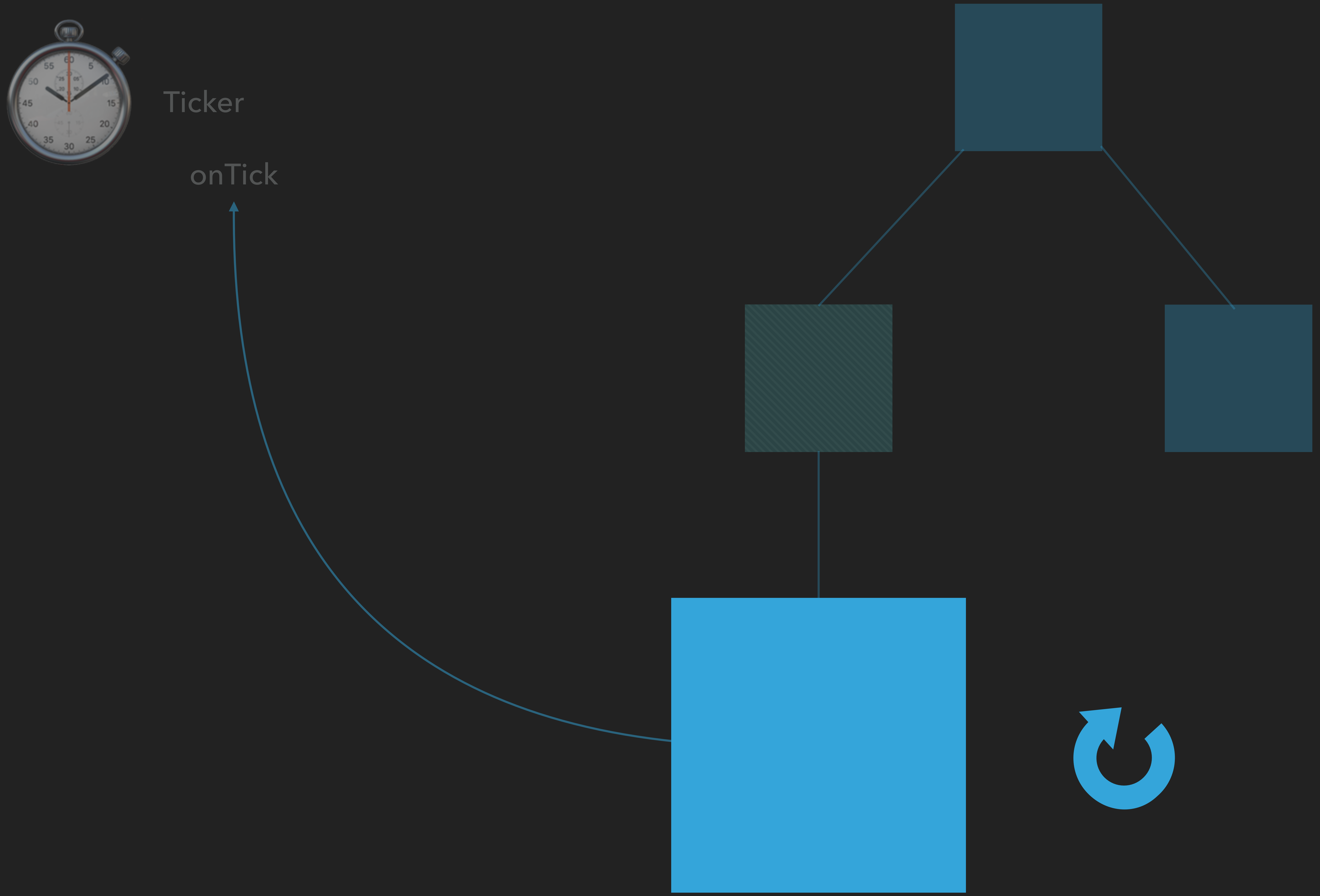
onTick



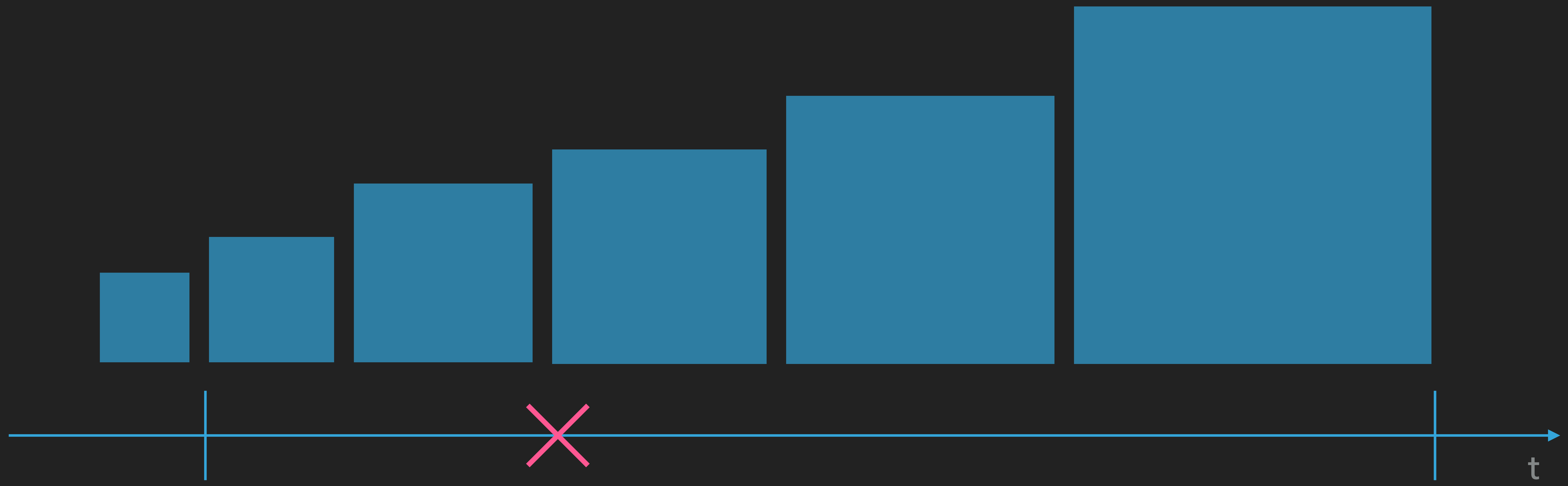

Ticker
onTick


Ticker
onTick
setState(newSize)


Ticker
onTick
setState(newSize)
build()

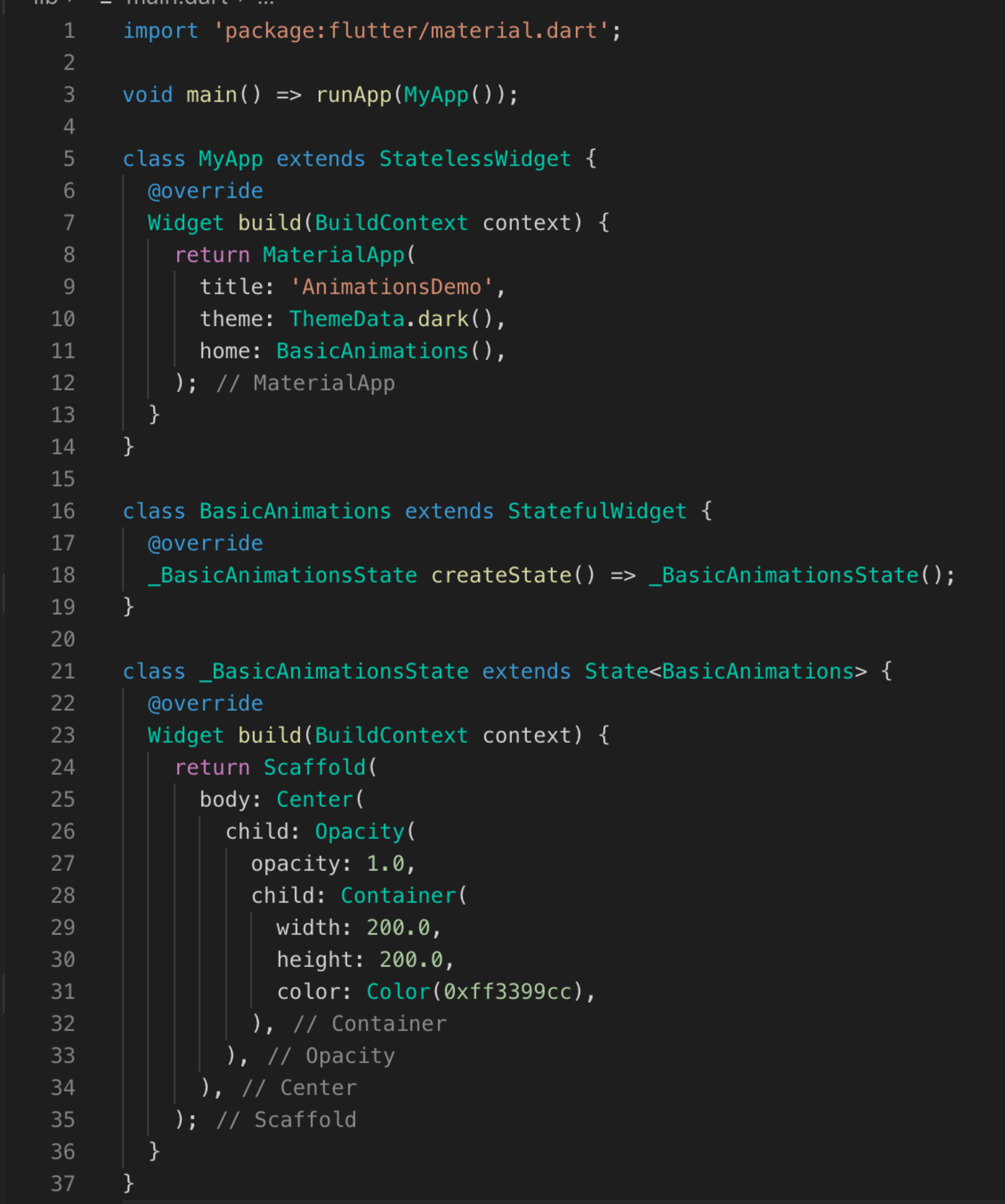

Ticker
onTick

Как менять значения?

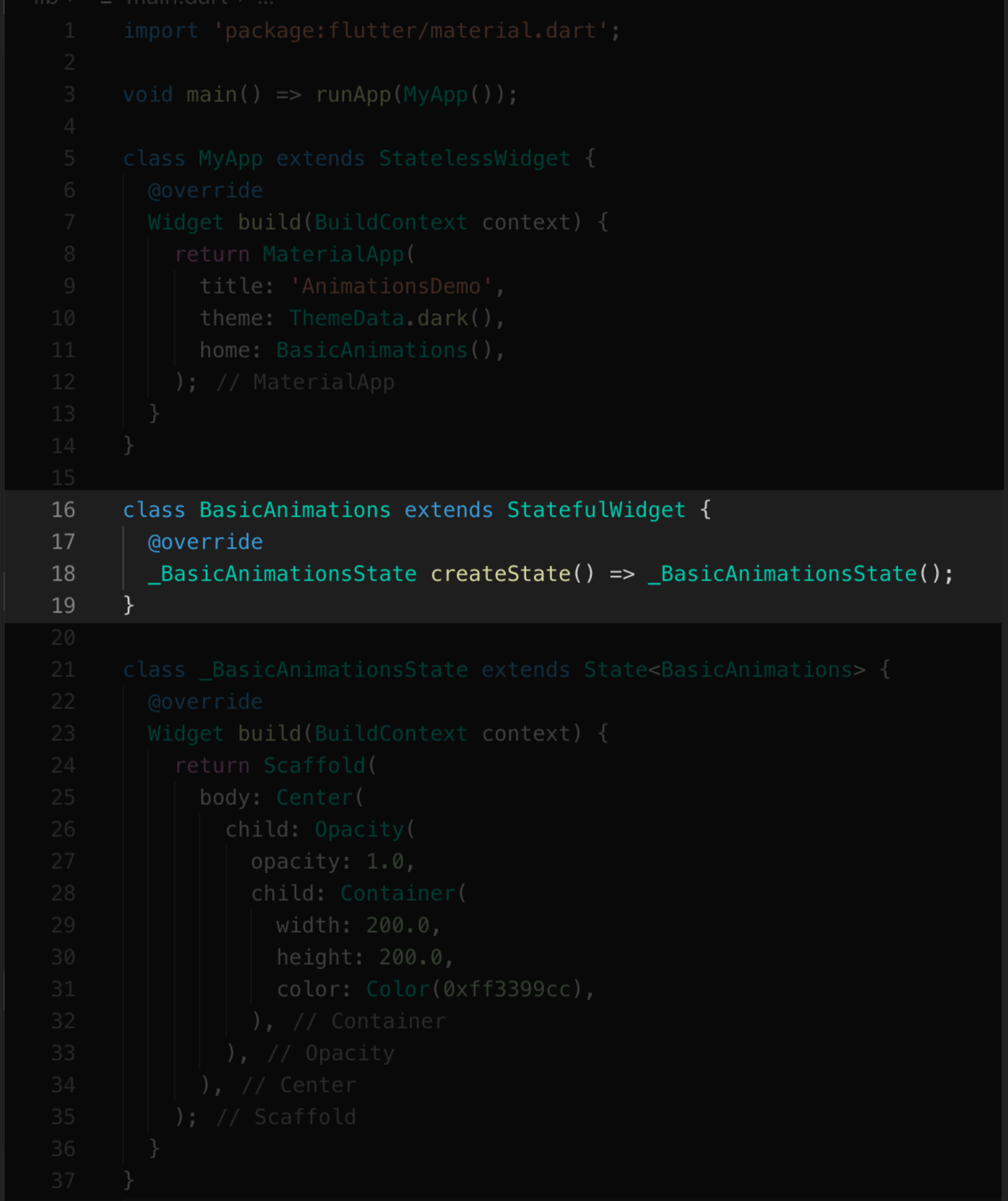


ANIMATION STATE

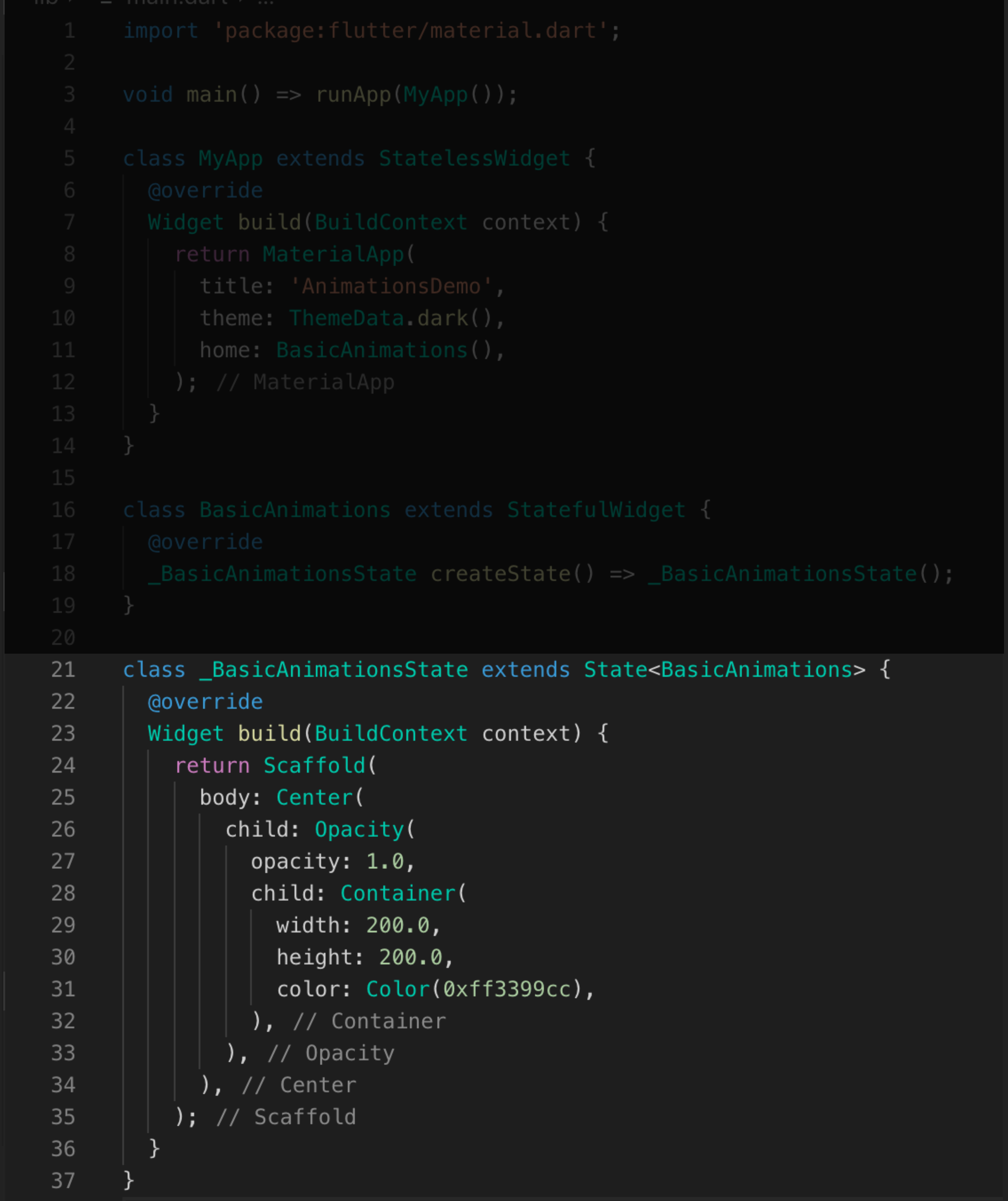
- Текущее значение
- Диапазон значений
- Продолжительность
- Относительное время



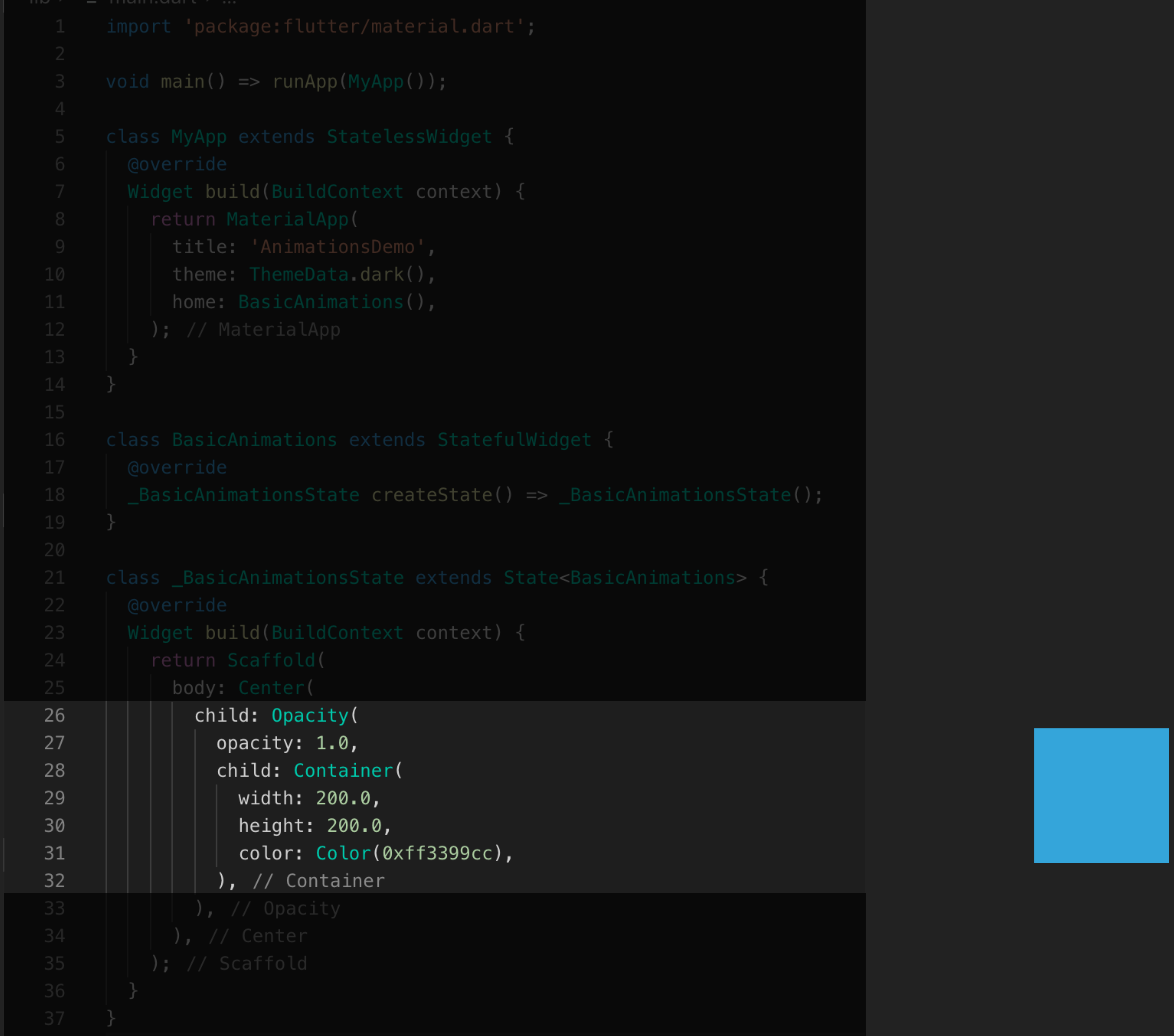
```dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'AnimationsDemo',
      theme: ThemeData.dark(),
      home: BasicAnimations(),
    ); // MaterialApp
  }
}

class BasicAnimations extends StatefulWidget {
  @override
  _BasicAnimationsState createState() => _BasicAnimationsState();
}

class _BasicAnimationsState extends State<BasicAnimations> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Opacity(
          opacity: 1.0,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ), // Container
        ), // Opacity
      ), // Center
    ); // Scaffold
  }
}
```



```dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'AnimationsDemo',
      theme: ThemeData.dark(),
      home: BasicAnimations(),
    ); // MaterialApp
  }
}

class BasicAnimations extends StatefulWidget {
  @override
  _BasicAnimationsState createState() => _BasicAnimationsState();
}

class _BasicAnimationsState extends State<BasicAnimations> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Opacity(
          opacity: 1.0,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ), // Container
        ), // Opacity
      ), // Center
    ); // Scaffold
  }
}
```



```dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'AnimationsDemo',
      theme: ThemeData.dark(),
      home: BasicAnimations(),
    ); // MaterialApp
  }
}

class BasicAnimations extends StatefulWidget {
  @override
  _BasicAnimationsState createState() => _BasicAnimationsState();
}

class _BasicAnimationsState extends State<BasicAnimations> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Opacity(
          opacity: 1.0,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ), // Container
        ), // Opacity
      ), // Center
    ); // Scaffold
  }
}
```



```dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'AnimationsDemo',
      theme: ThemeData.dark(),
      home: BasicAnimations(),
    ); // MaterialApp
  }
}

class BasicAnimations extends StatefulWidget {
  @override
  _BasicAnimationsState createState() => _BasicAnimationsState();
}

class _BasicAnimationsState extends State<BasicAnimations> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Opacity(
          opacity: 1.0,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ), // Container
        ), // Opacity
      ), // Center
    ); // Scaffold
  }
}
```

Ticker

```
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
```

ANIMATION CONTROLLER

- Реализует абстрактный класс Animation
- Подписывается на Ticker
- По тику выдает значения от 0 до 1
- Задает продолжительность анимации
- Запуск/Остановка анимации

Ticker

```
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
```

Ticker

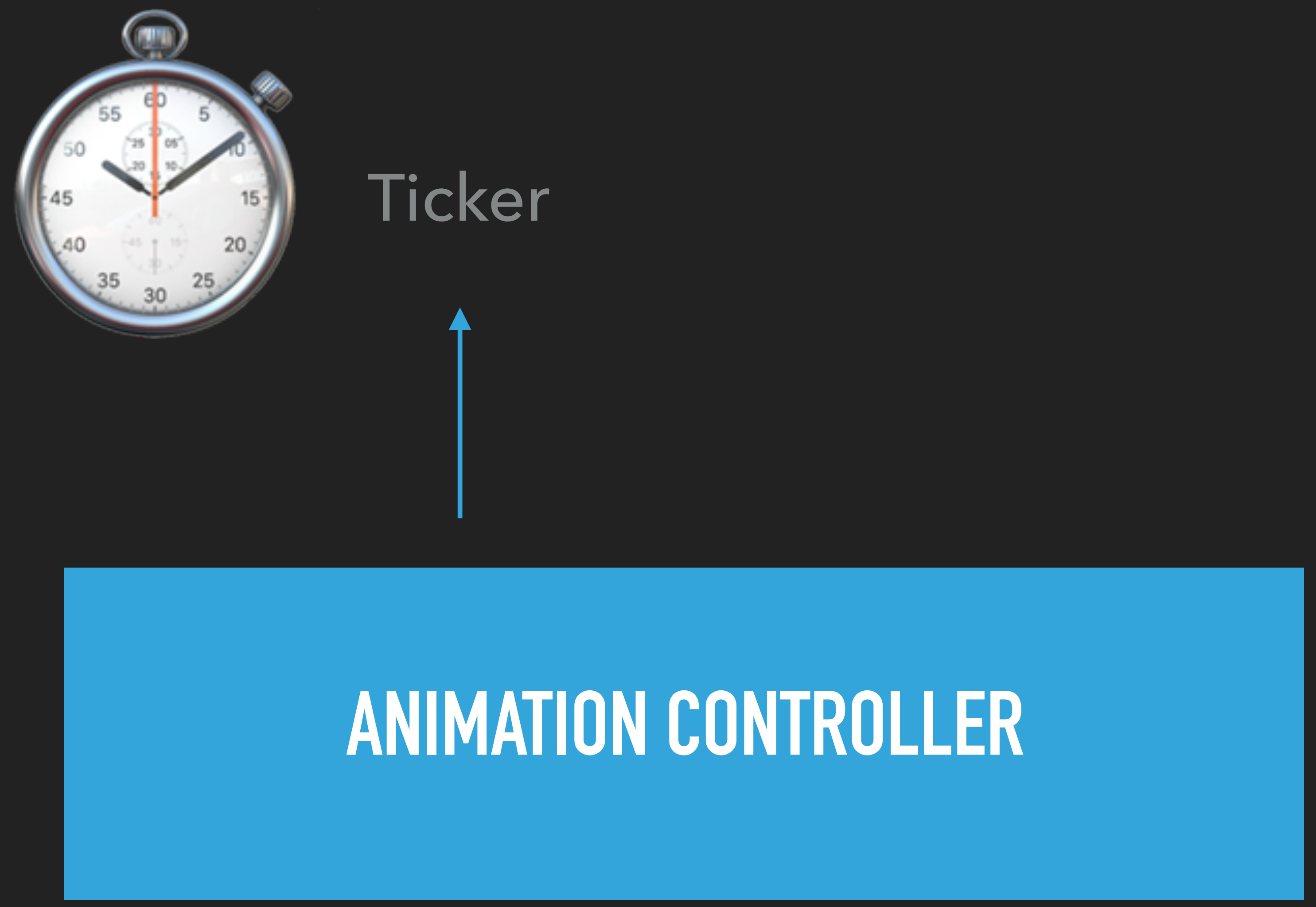
```
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
```

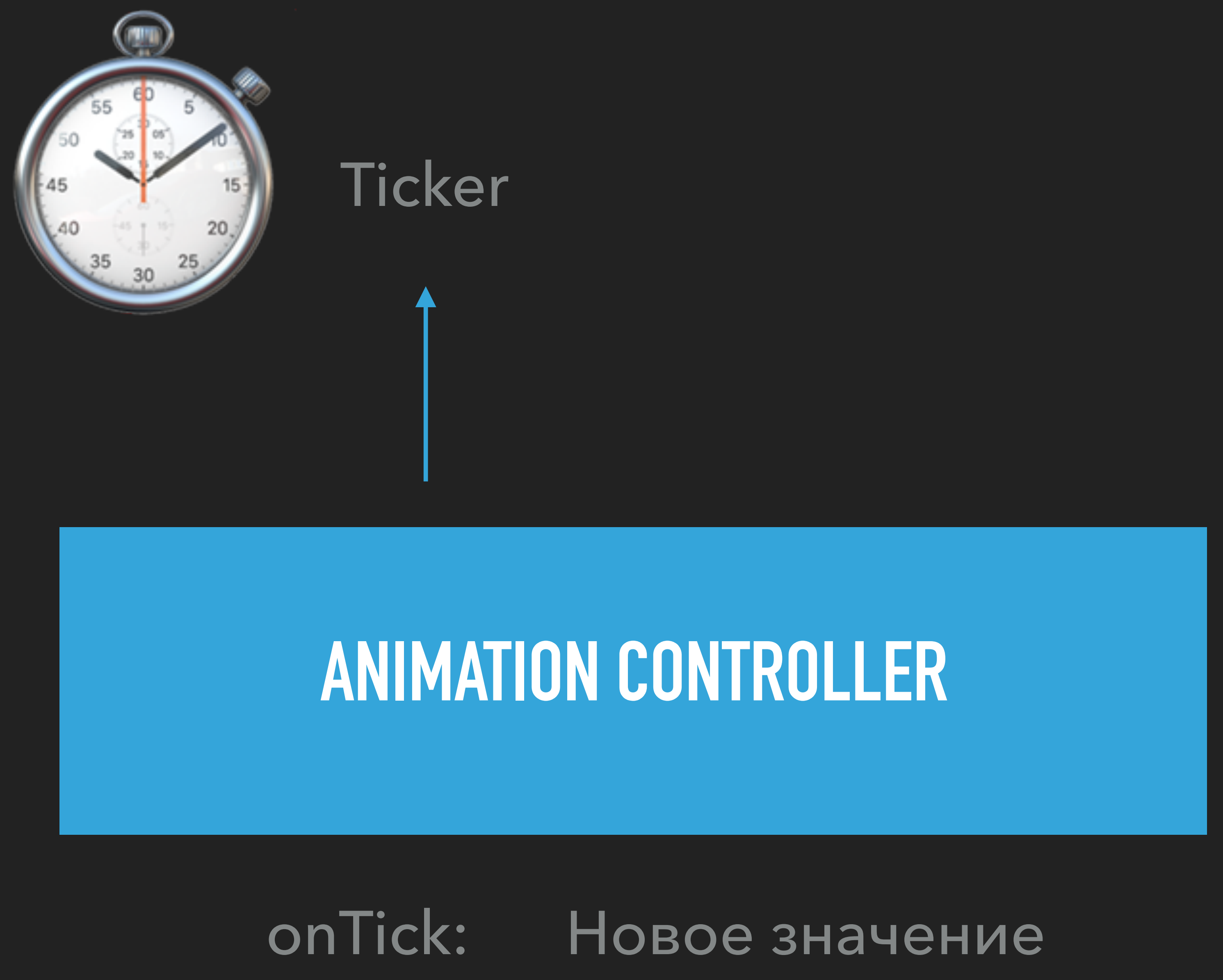
```
@override
void initState() {
  super.initState();
  _controller =
      AnimationController(duration: Duration(seconds: 2), vsync: this);
}
```

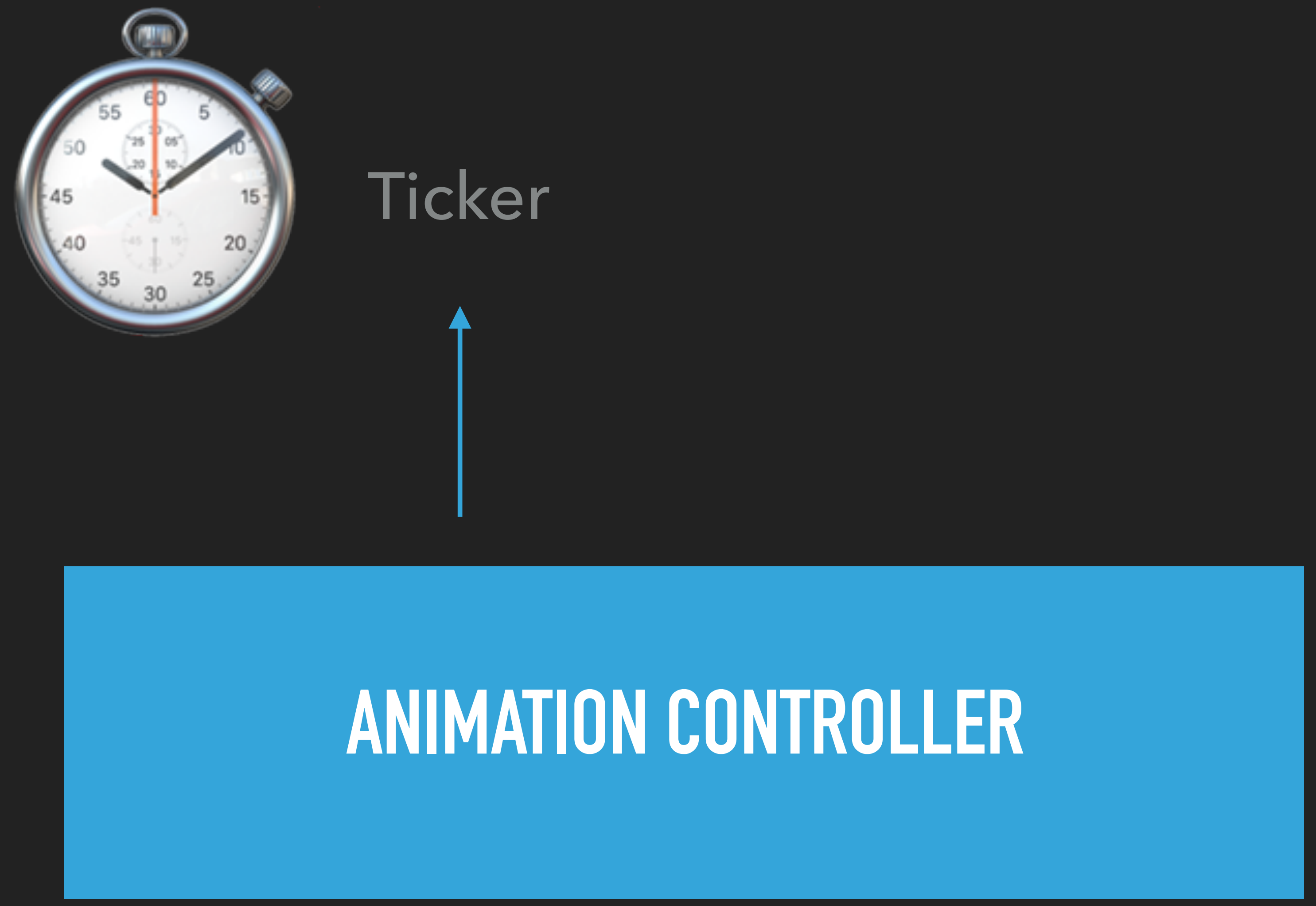
```dart
abstract class Animation<T> extends Listenable implements ValueListenable<T> {
```

`Listenable`

Ticker

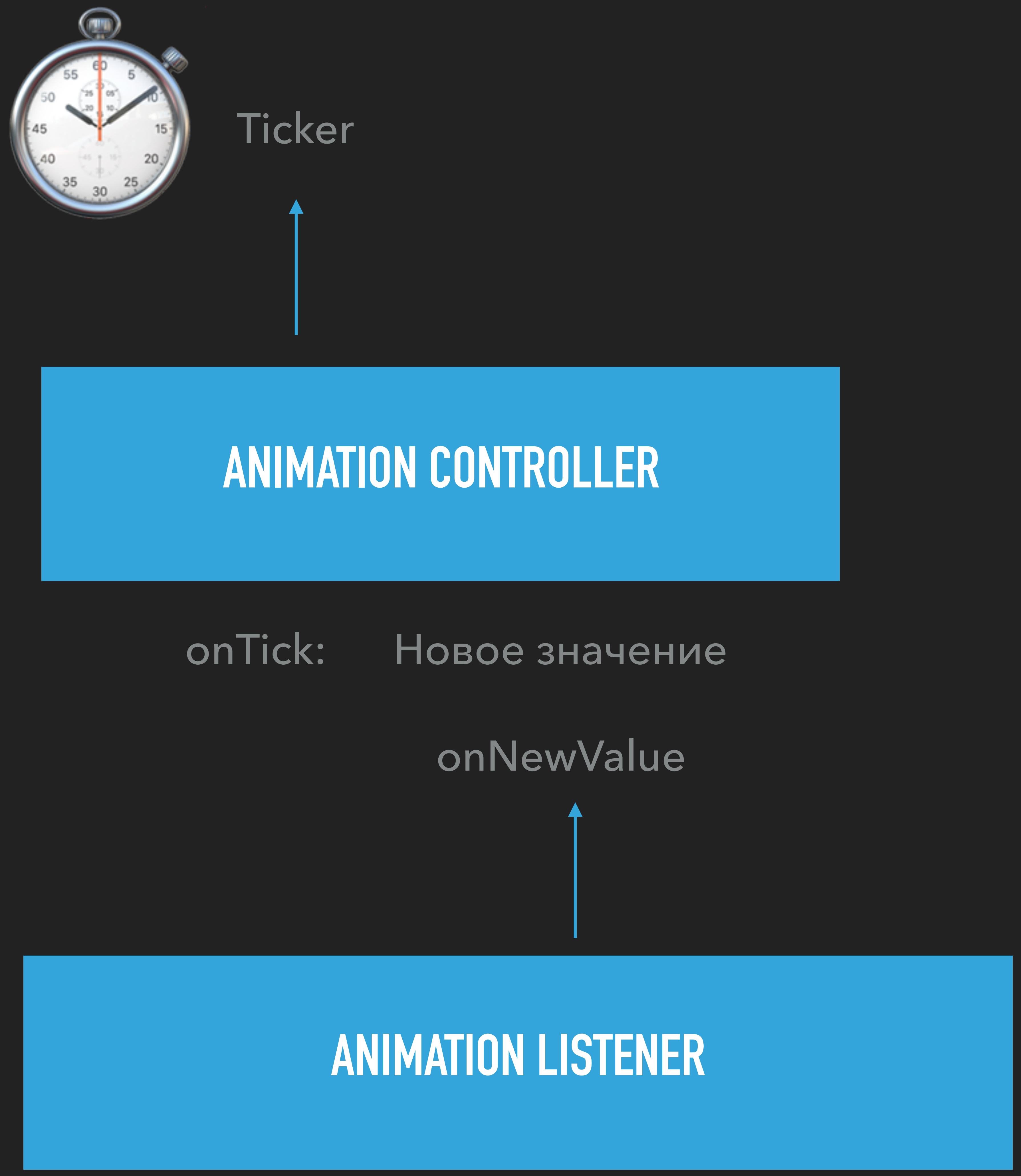

Ticker
ANIMATION CONTROLLER


Ticker
ANIMATION CONTROLLER
onTick:
Новое значение



onTick: Новое значение

onNewValue


Ticker
ANIMATION CONTROLLER
onTick:
Новое значение
onNewValue
ANIMATION LISTENER

```
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;

  @override
  void initState() {
    super.initState();
    _controller =
        AnimationController(duration: Duration(seconds: 2), vsync: this)
          ..addListener(() {
            setState(() {});
          });
  }
```

```
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
  @override
  void initState() {
    super.initState();
    _controller =
        AnimationController(duration: Duration(seconds: 2), vsync: this)
          ..addListener(() {
            setState(() {});
          });
  }
```

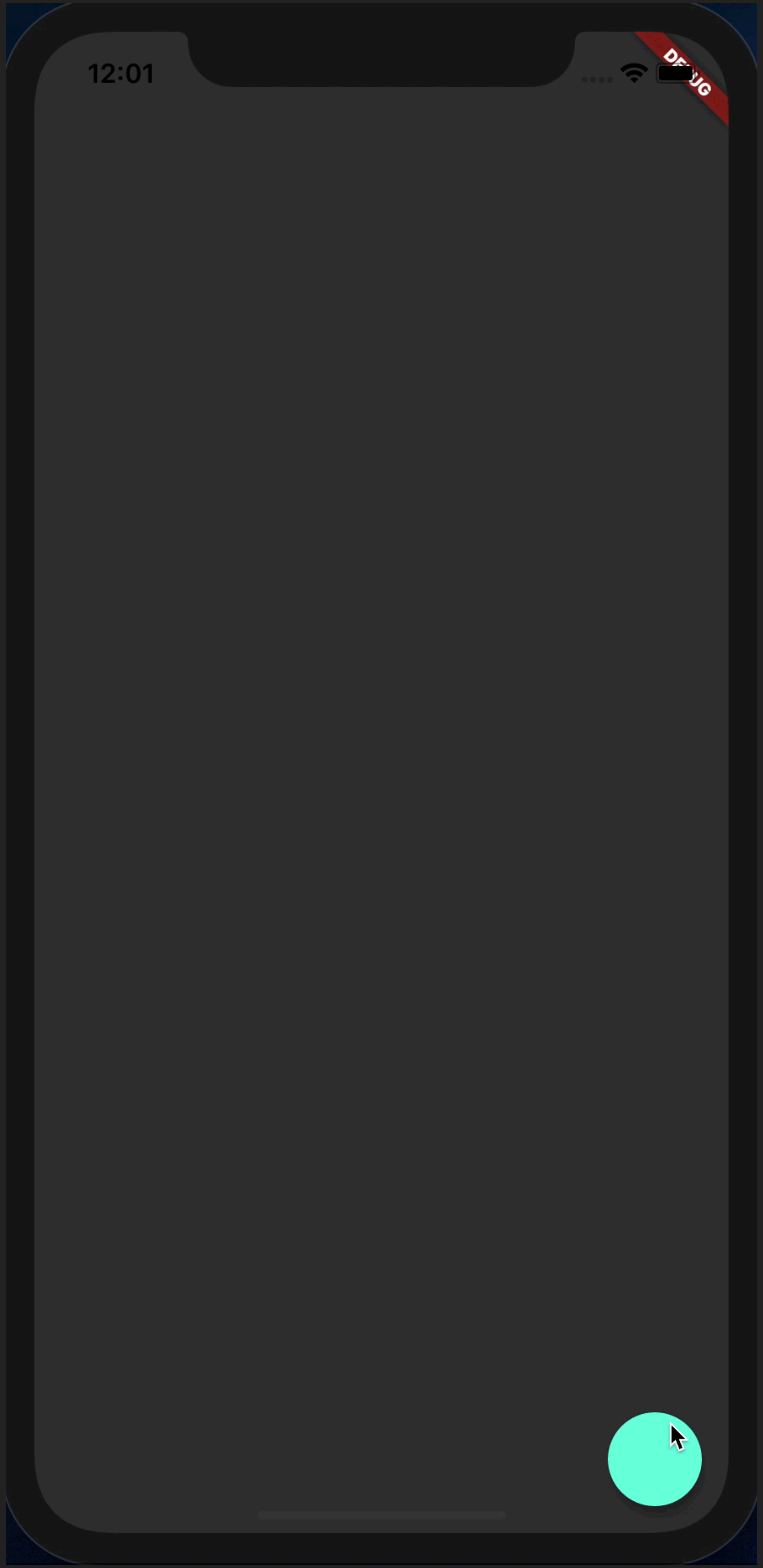
Rect разработчик: “Почему setState() пустой?”

```
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
 @override
 void initState() {
   super.initState();
   _controller =
       AnimationController(duration: Duration(seconds: 2), vsync: this)
         ..addListener(() {
           setState(() {});
         });
 }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Opacity(
          opacity: 1.0,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ),
        ),
```

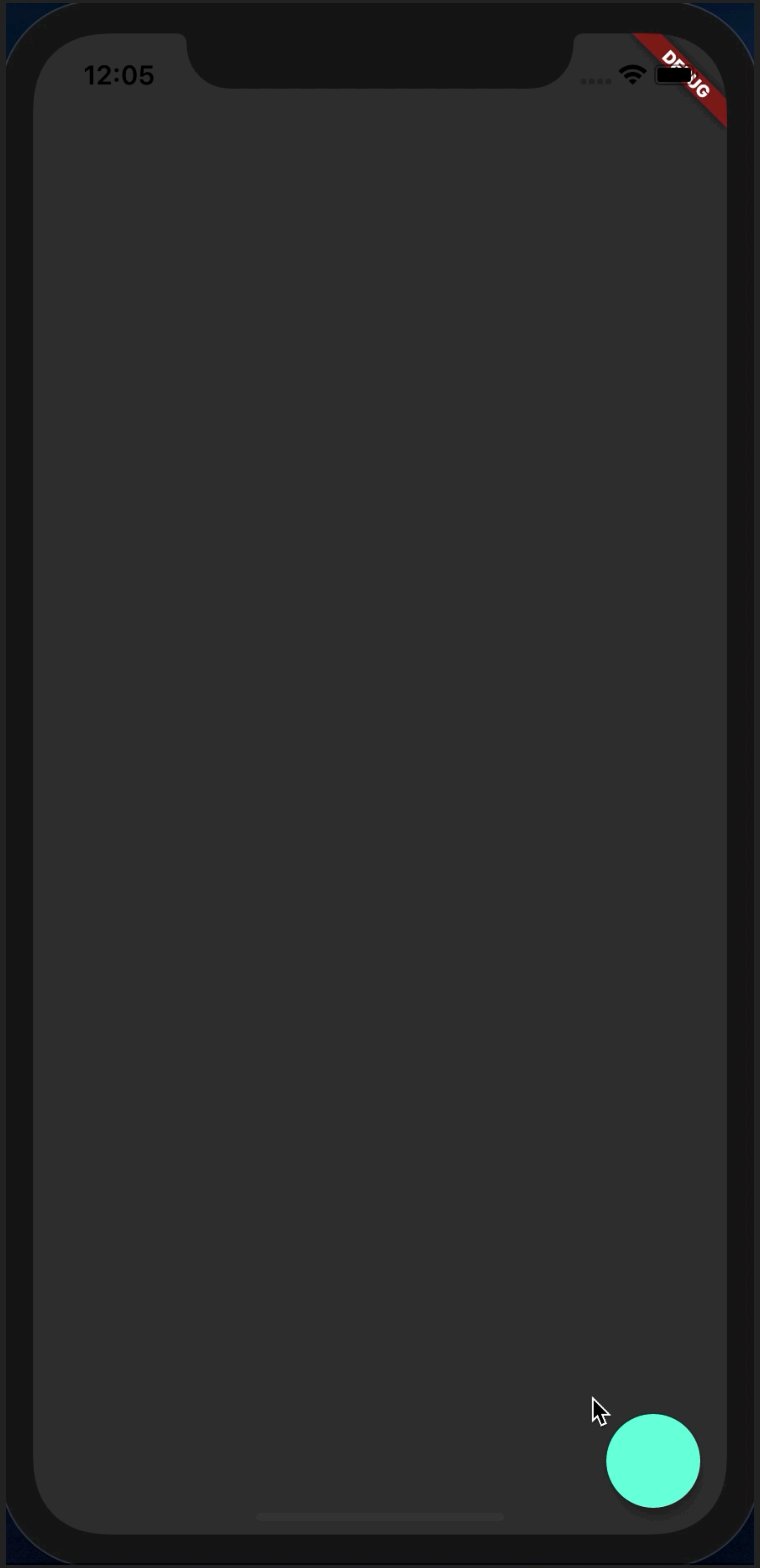
```dart
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
  @override
  void initState() {
    super.initState();
    _controller =
        AnimationController(duration: Duration(seconds: 2), vsync: this)
          ..addListener(() {
            setState(() {});
          });
  }
```

```dart
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Opacity(
          opacity: _controller.value,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ),
        ),
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
```

```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(
        onPressed: _startAnimation,
      ),
      body: Center(
        child: Opacity(
          opacity: _controller.value,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ),
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;


  void _startAnimation() {
    _controller.forward();
  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(
        onPressed: _startAnimation,
      ),
      body: Center(
        child: Opacity(
          opacity: _controller.value,
          child: Container(
            width: 200.0,
            height: 200.0,
            color: Color(0xff3399cc),
          ),
```


12:01
DEBUG

```dart
  void _startAnimation() {
    if (_controller.status == AnimationStatus.completed) {
      _controller.reverse();
    } else {
      _controller.forward();
    }
}
```

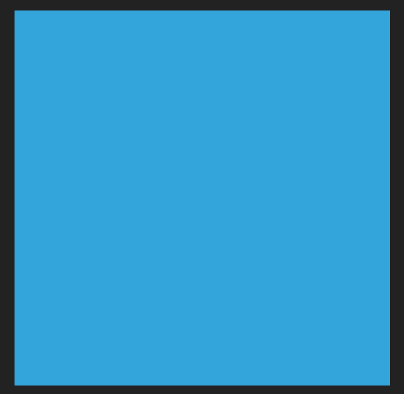

12:05
DEBUG

```
child: Opacity(
  opacity: _controller.value,
child: Opacity(
  opacity: _controller.value,
```

0…1

Как поменять ширину?

```
child: Container(
  width: 200.0,
  height: 200.0,
  color: Color(0xff3399cc),
),
```



Как поменять ширину?

```
child: Container(
  width: 200.0,
  height: 200.0,
  color: Color(0xff3399cc),
),
```

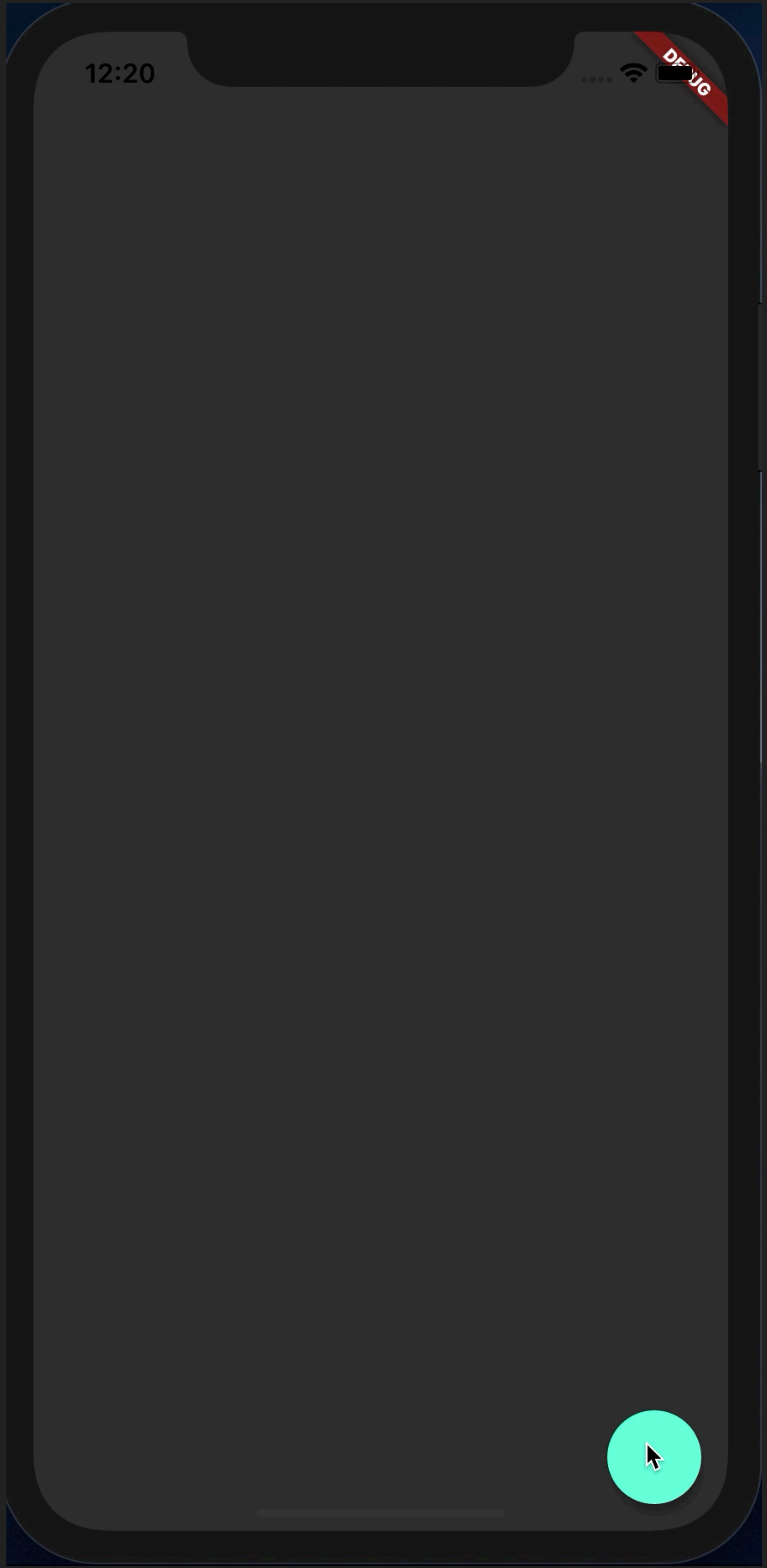
200...400

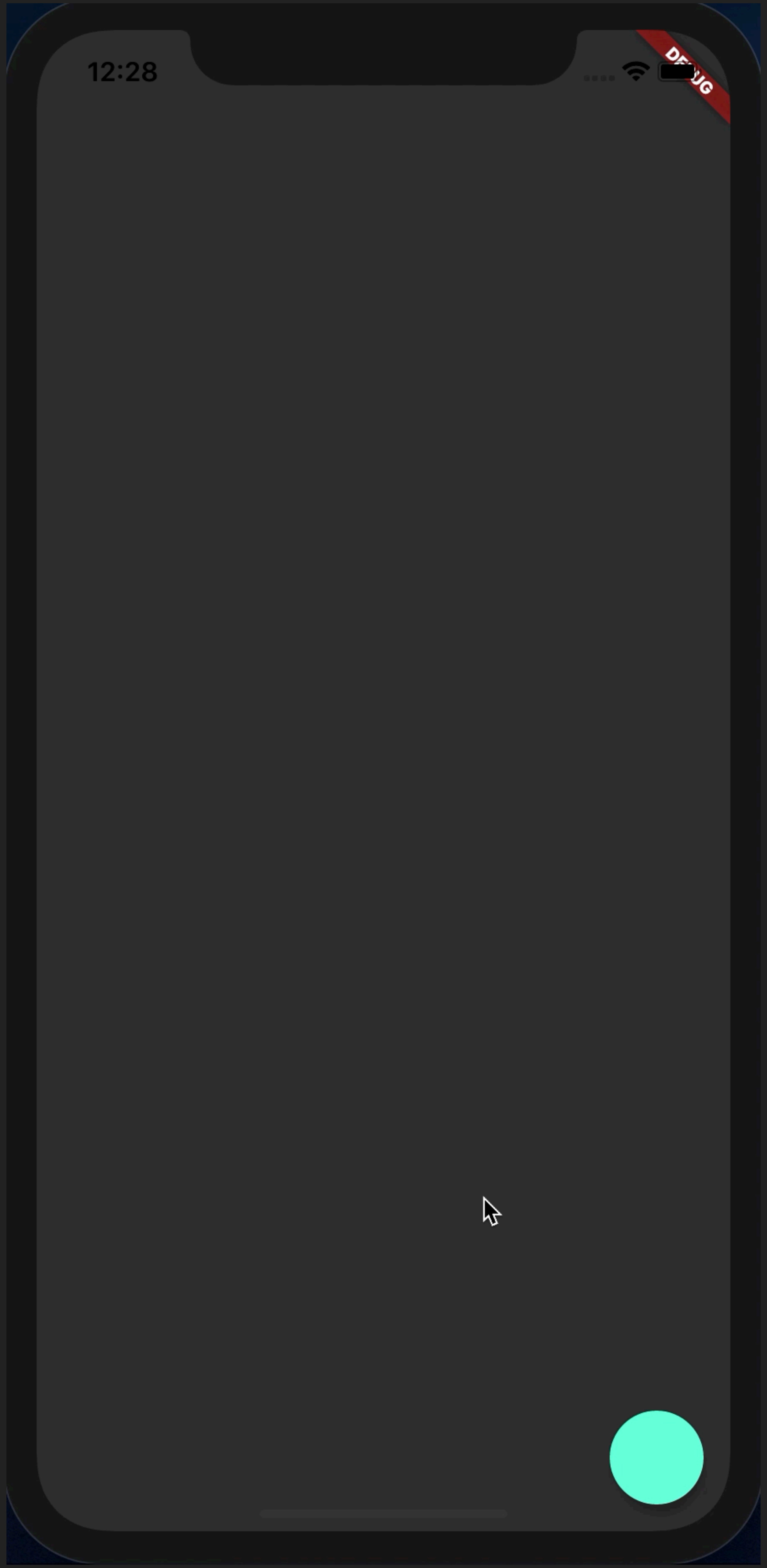
TWEEN

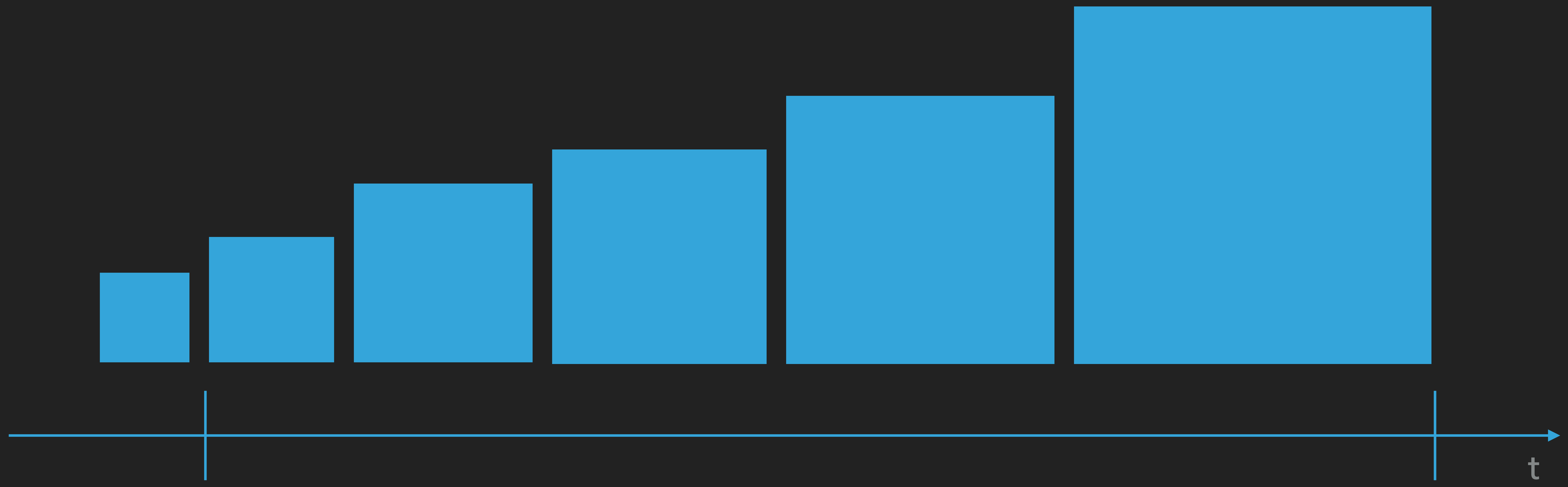
- Генерирует значения из любого диапазона
- Примеры: ColorTween, DecorationTween…
- Stateless
- Конвертируется в анимацию с помощью animate()

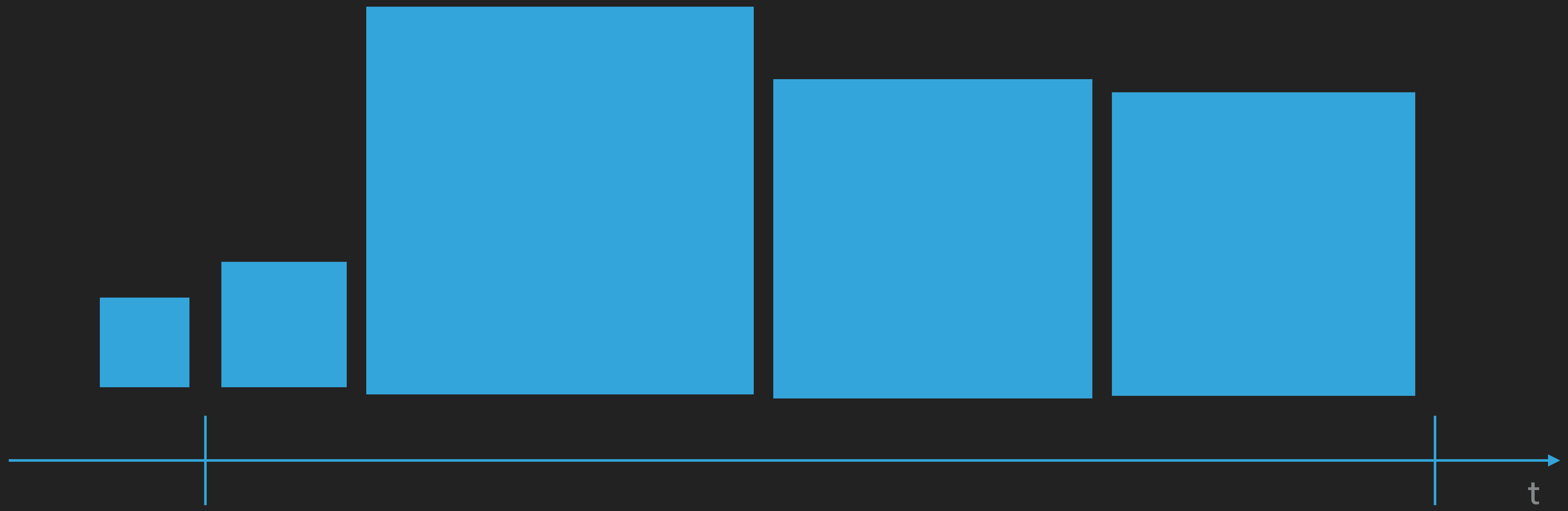
```dart
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
  Animation _width;

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

    _width = Tween(begin: 200.0, end: 400.0).animate(_controller);
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    floatingActionButton: FloatingActionButton(
      onPressed: _startAnimation,
    ),
    body: Center(
      child: Opacity(
        opacity: _controller.value,
        child: Container(
                    width: _width.value,
                    height: 200.0,
                    color: Color(0xff3399cc),
                  ),
```


12:20
DEBUG

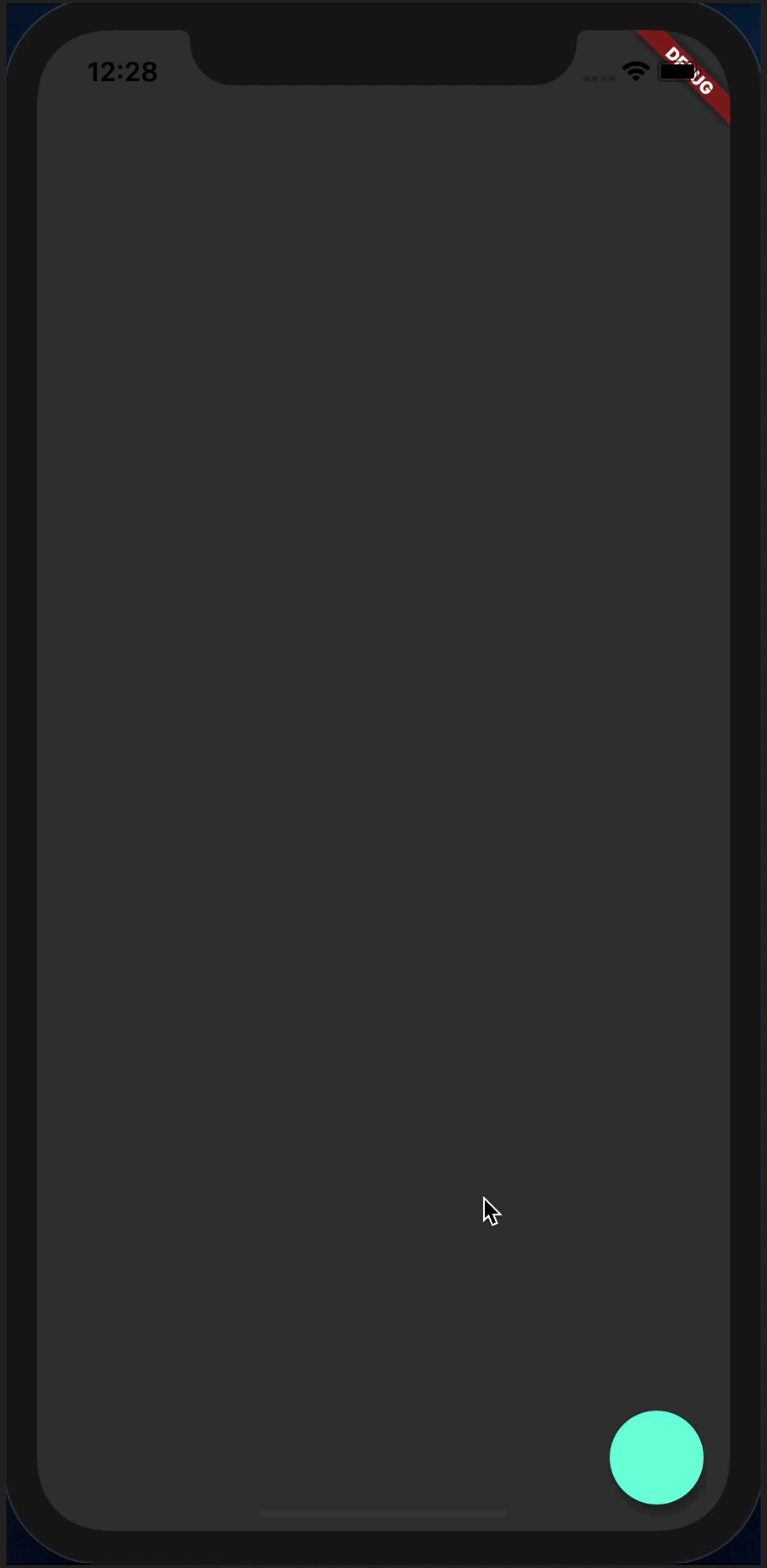

12:28
DEBUG

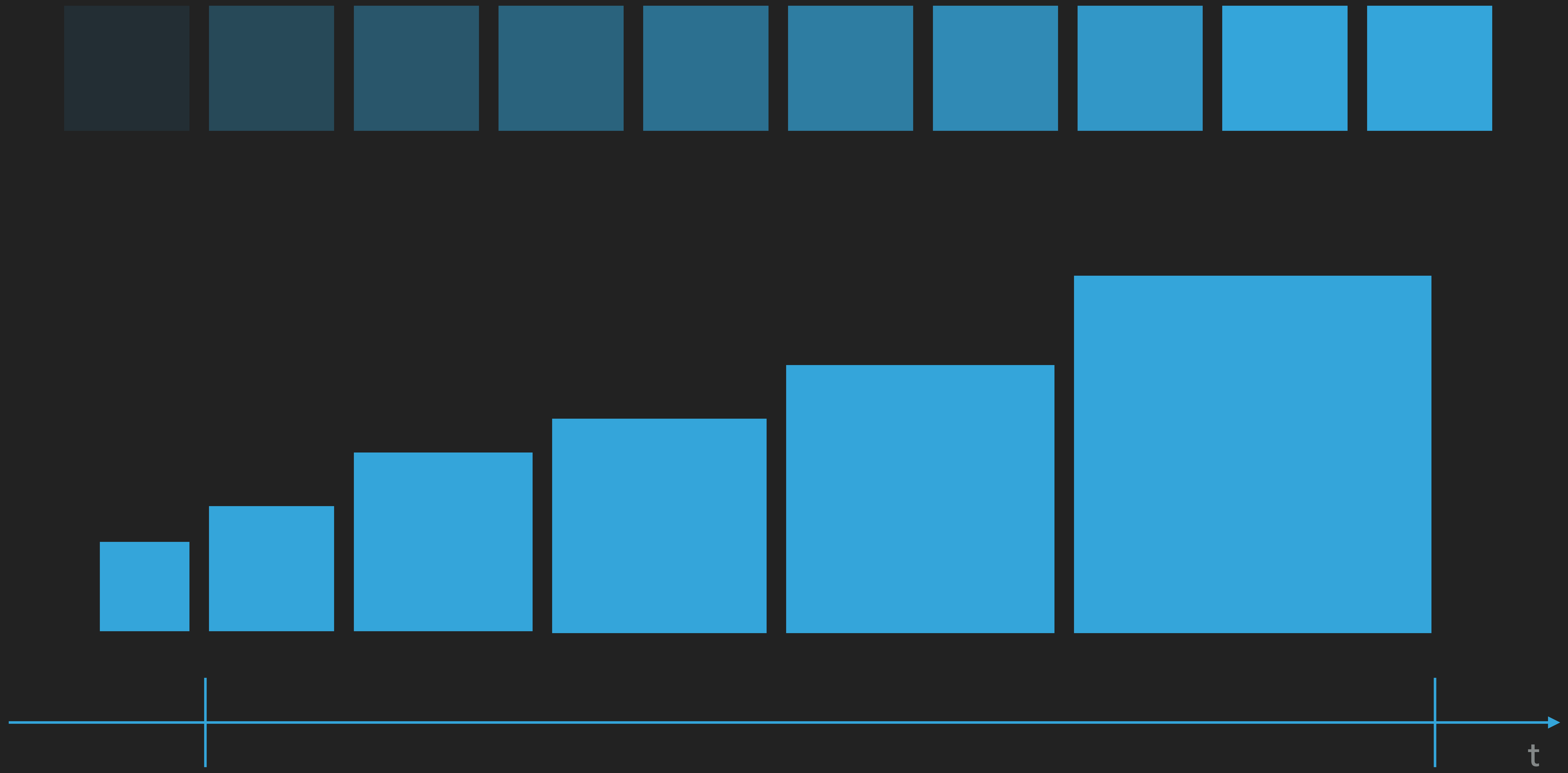

t

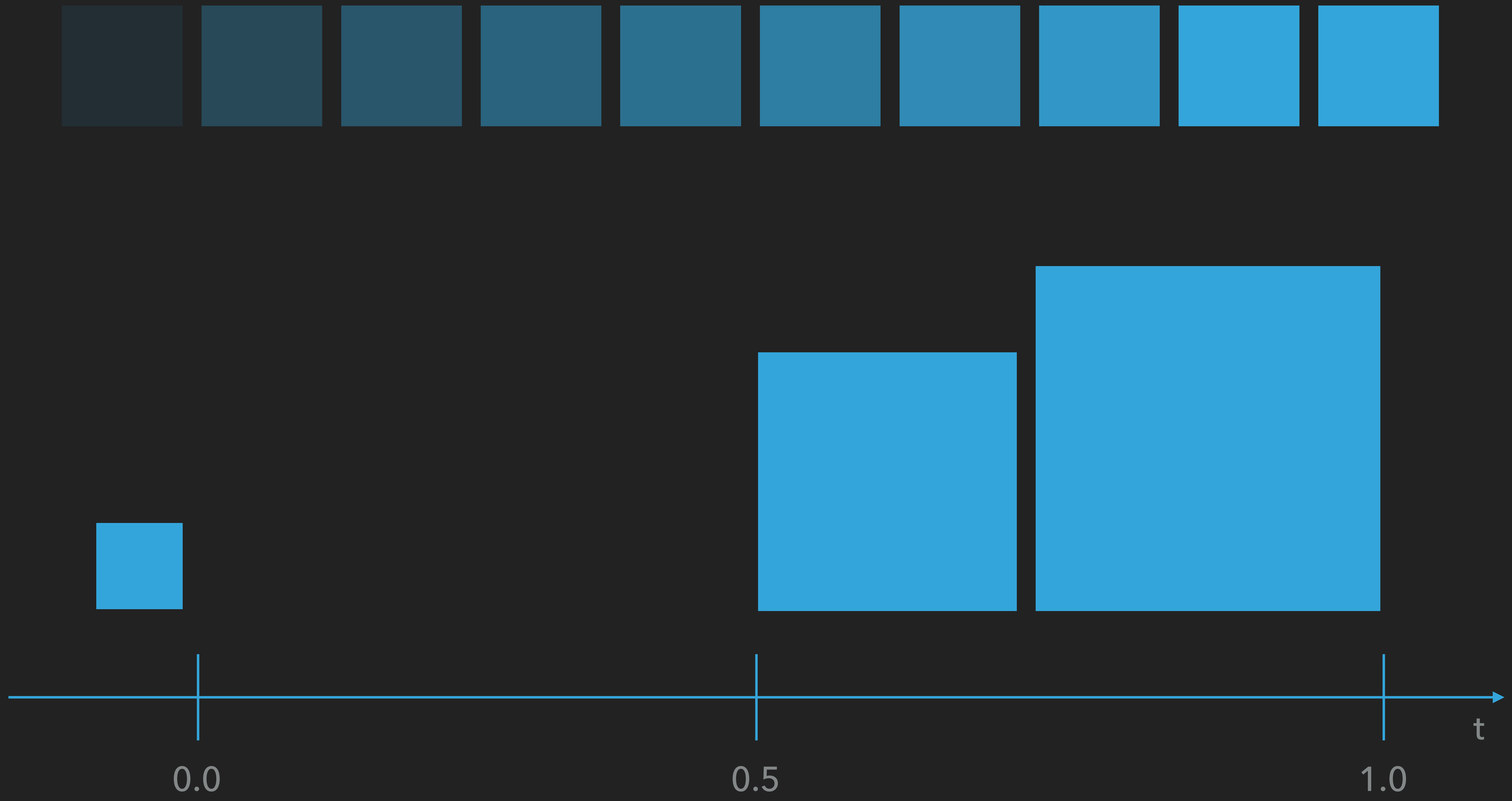

t

```dart
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
  Animation _width;

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

    _width = Tween(begin: 200.0, end: 400.0).animate(_controller);
```

```dart
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
   Animation _width;

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

    _width = Tween(begin: 200.0, end: 400.0)
        .animate(
          CurvedAnimation(
            curve: Curves.bounceOut,
            parent: _controller));
```

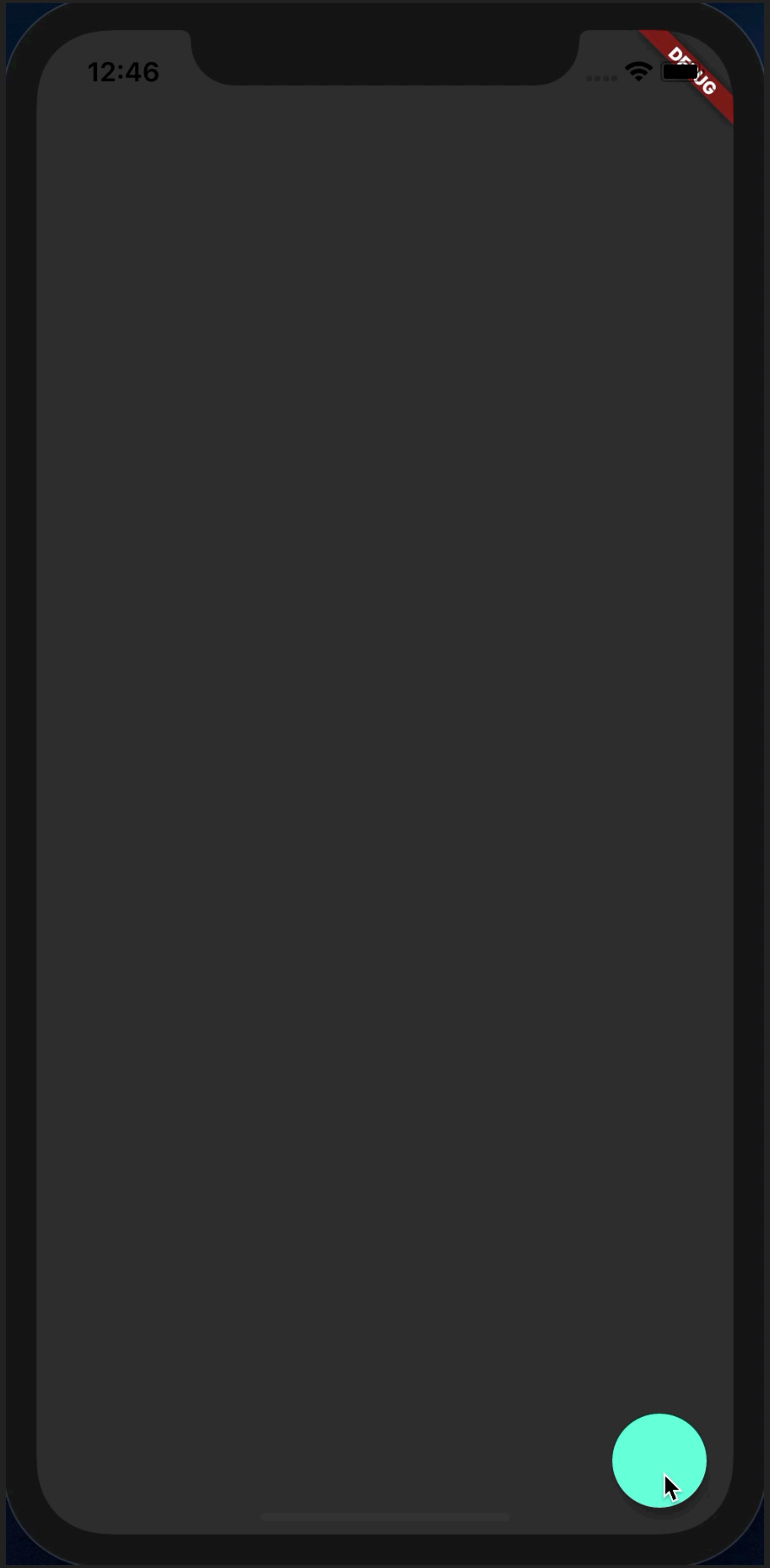

12:28
DEBUG


t


t
0.0
0.5
1.0

```dart
class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
  Animation _width;

  Animation _height;

  @override
  void initState() {
    super.initState();
    _controller =
        AnimationController(duration: Duration(seconds: 2), vsync: this)
          ..addListener(() {
            setState(() {});
          });
    _width = Tween(begin: 200.0, end: 400.0)
        .animate(CurvedAnimation(curve: Curves.bounceOut, parent: _controller));

    _height = Tween(begin: 200.0, end: 400.0)
        .animate(
          CurvedAnimation(
            curve: Interval(0.5, 1.0),
            parent: _controller));
```

```
@override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(
        onPressed: _startAnimation,
      ),
      body: Center(
        child: Opacity(
          opacity: _controller.value,
          child: Container(
            color: Color(0xff3399cc),
            width: _width.value,
            height: _height.value,
```

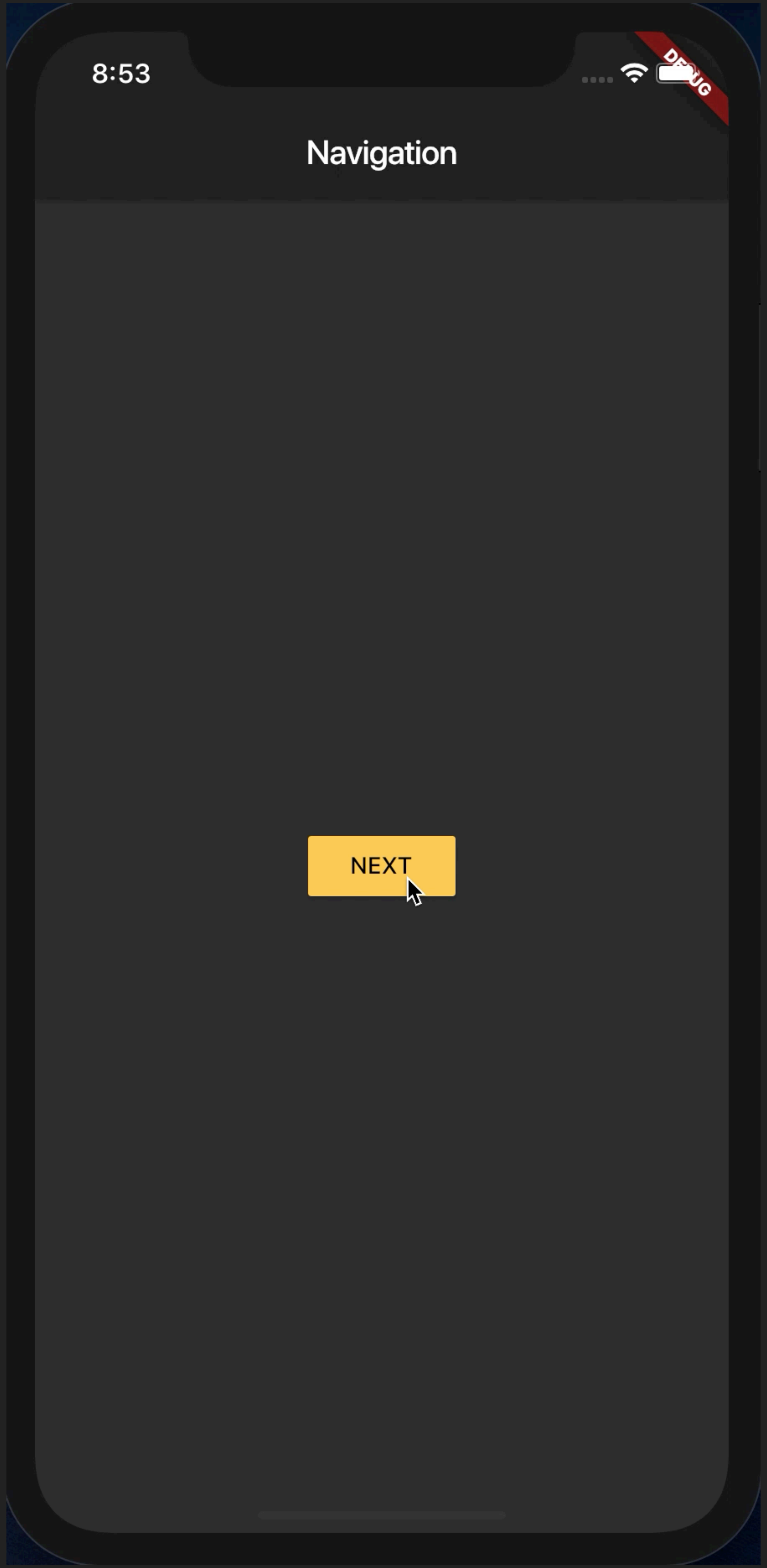

12:46
DEBUG

```
class BasicAnimations extends StatefulWidget {
  @override
  _BasicAnimationsState createState() => _BasicAnimationsState();
}

class _BasicAnimationsState extends State<BasicAnimations>
    with SingleTickerProviderStateMixin {
  AnimationController _controller;
  Animation _width;
  Animation _height;

  @override
  void initState() {
    super.initState();
    _controller =
        AnimationController(duration: Duration(seconds: 2), vsync: this)
          ..addListener(() {
            setState(() {});
          });
    _width = Tween(begin: 200.0, end: 400.0)
        .animate(CurvedAnimation(curve: Curves.bounceOut, parent: _controller));

    _height = Tween(begin: 200.0, end: 400.0).animate(
        CurvedAnimation(curve: Interval(0.5, 1.0), parent: _controller));
  }

  void _startAnimation() {
    if (_controller.status == AnimationStatus.completed) {
      _controller.reverse();
    } else {
      _controller.forward();
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: FloatingActionButton(
        onPressed: _startAnimation,
      ),
      body: Center(
```

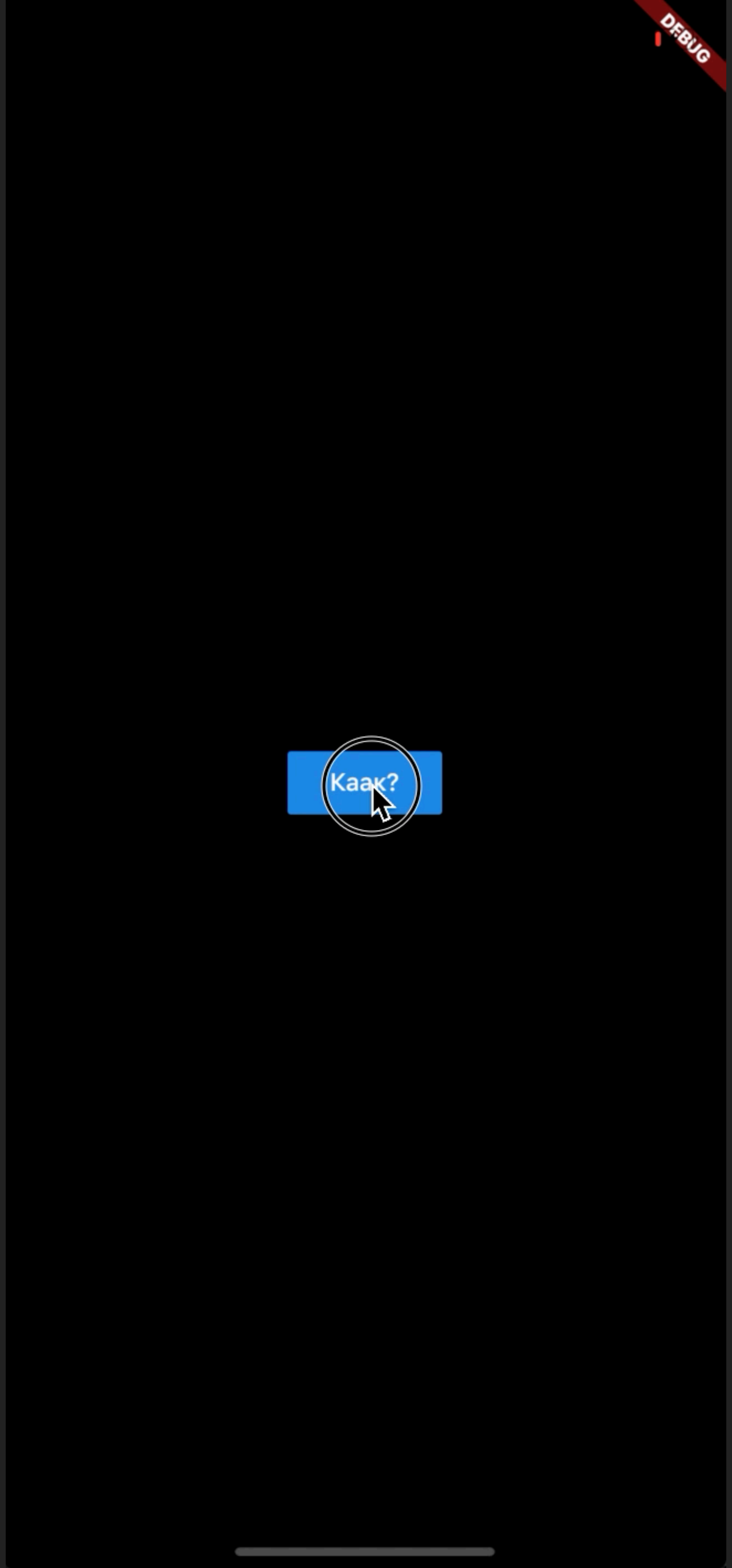
TRANSITONS

- AnimatedWidget
- FadeTransition, ScaleTransition…

ANIMATE NAVIGATION


8:53
DEBUG
Navigation
NEXT

СЛОЖНО ЛИ КАСТОМИЗИРОВАТЬ?


DEBUG
Каак?

PAGE ROUTE

- Сборка widget-а экрана
- Анимирование перехода
- Добавление экрана в Overlay

```dart
class SpinPageRoute extends PageRouteBuilder {
  final Widget widget;
  SpinPageRoute({@required this.widget})
      : super(pageBuilder: (context, animation, secondaryAnimation) {
          return widget;
        }, transitionsBuilder: (context, animation, secondary, child) {
          return RotationTransition(
              turns: Tween(begin: 0.0, end: 20.0).animate(animation),
              child: ScaleTransition(
                scale: animation,
                child: child,
              ));
        });
}
```

```dart
class SpinPageRoute extends PageRouteBuilder {
  final Widget widget;
  SpinPageRoute({@required this.widget})
      : super(pageBuilder: (context, animation, secondaryAnimation) {
          return widget;
        }, transitionsBuilder: (context, animation, secondary, child) {
          return RotationTransition(
              turns: Tween(begin: 0.0, end: 20.0).animate(animation),
              child: ScaleTransition(
                scale: animation,
                child: child,
              ));
        });
}
```

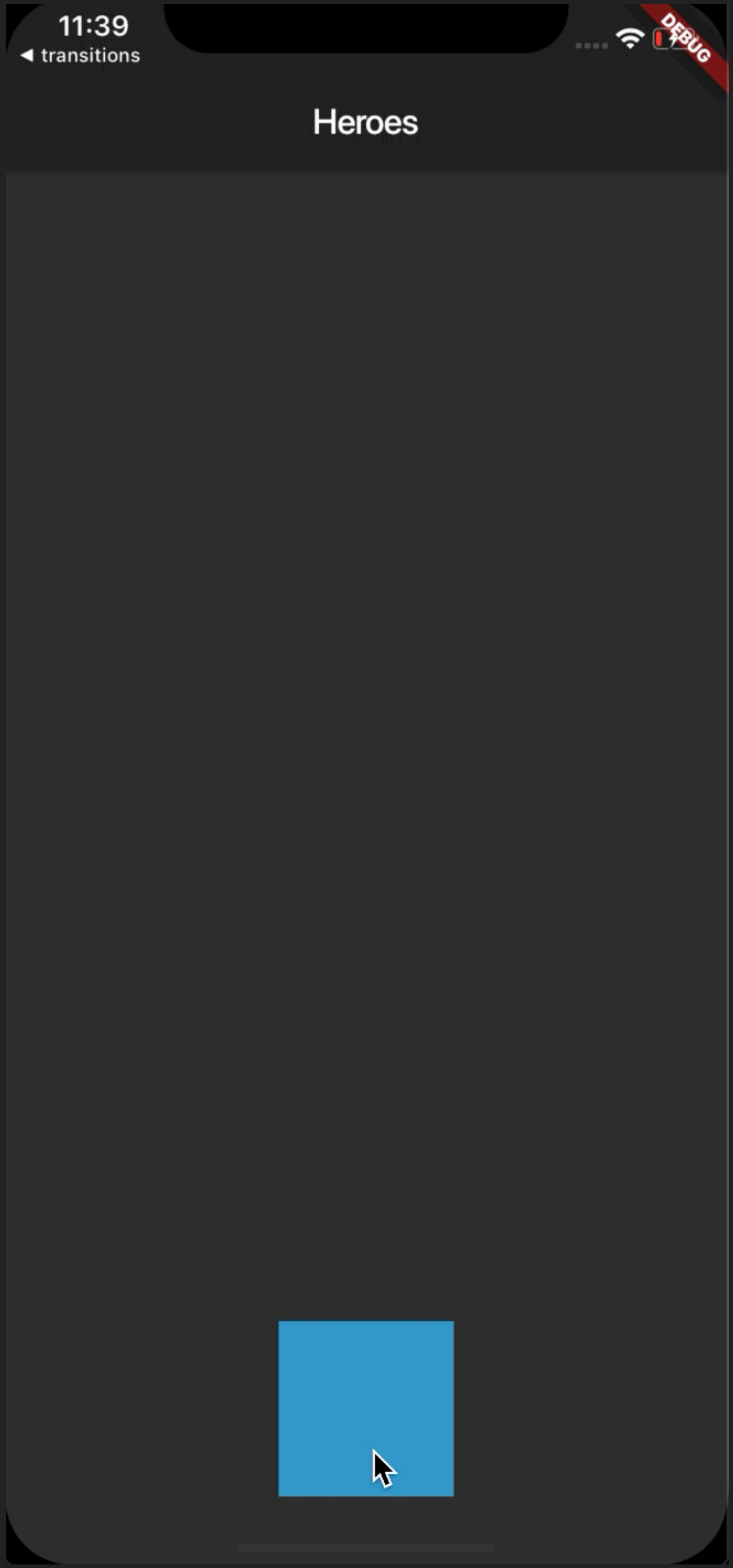
```
class SpinPageRoute extends PageRouteBuilder {
  final Widget widget;
  SpinPageRoute({@required this.widget})
      : super(pageBuilder: (context, animation, secondaryAnimation) {
          return widget;
        }, transitionsBuilder: (context, animation, secondary, child) {
          return RotationTransition(
              turns: Tween(begin: 0.0, end: 20.0).animate(animation),
              child: ScaleTransition(
                scale: animation,
                child: child,
              ));
        });
}
```

```dart
Navigator.push(context, SpinPageRoute(widget: SpinnedScreen()));
```

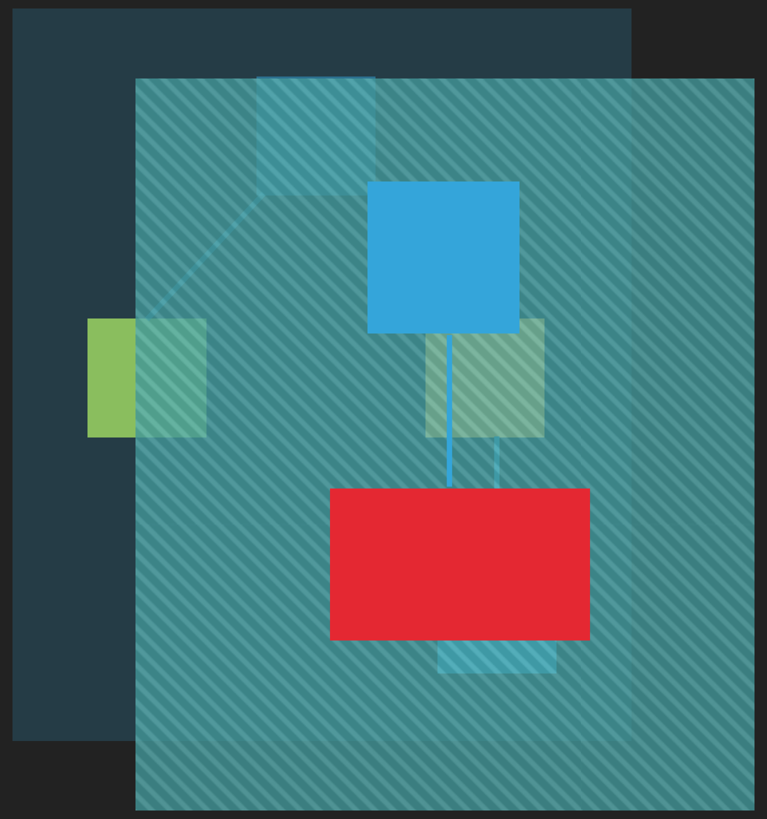
АНИМИРОВАНИЕ ВИДЖЕТОВ МЕЖДУ ЭКРАНАМИ

```
class StartWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GestureDetector(
        onTap: () {
          Navigator.pushNamed(context, '/second');
        },
        child:
          Container(width: 100.0, height: 100.0, color: Color(0xff3399cc)));
  }
}
```

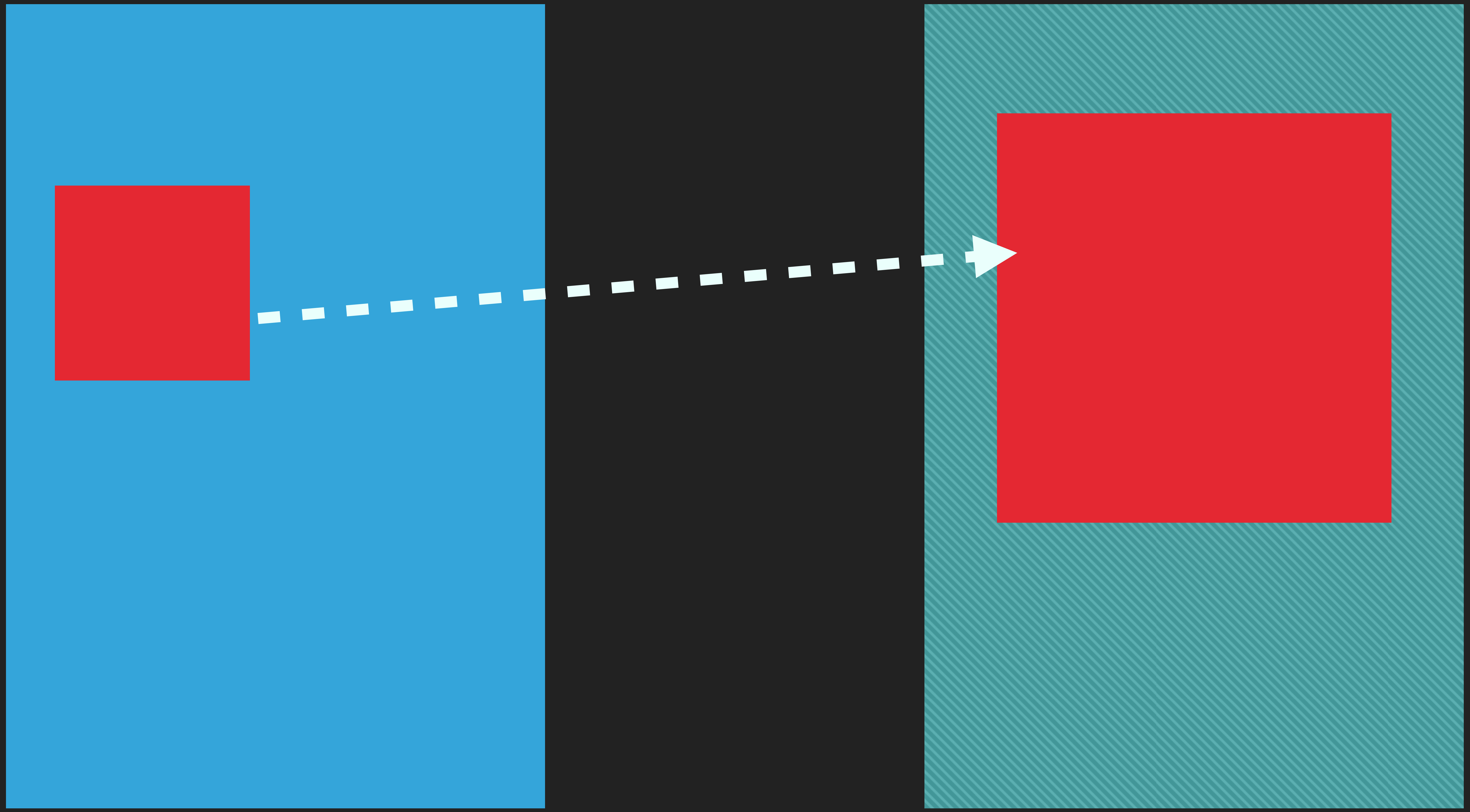
```dart
class Second extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Hero Home'),
      ),
      body: Center(
          child: Column(
        children: <Widget>[
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: Container(
                width: 400.0, height: 400.0, color: Color(0xffE42832)),
          ),
        ],
      )),
    );
  }
}
```



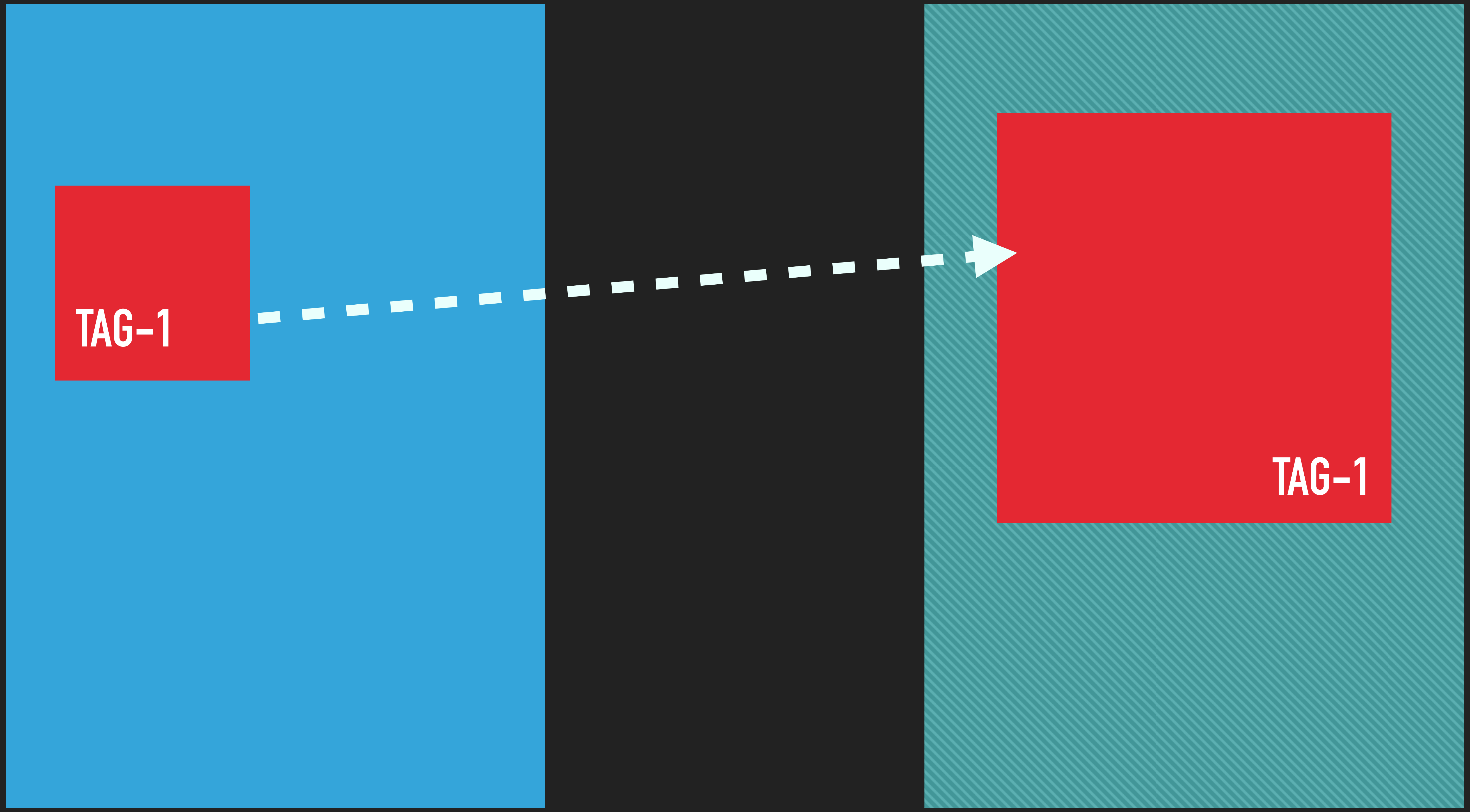
Heroes

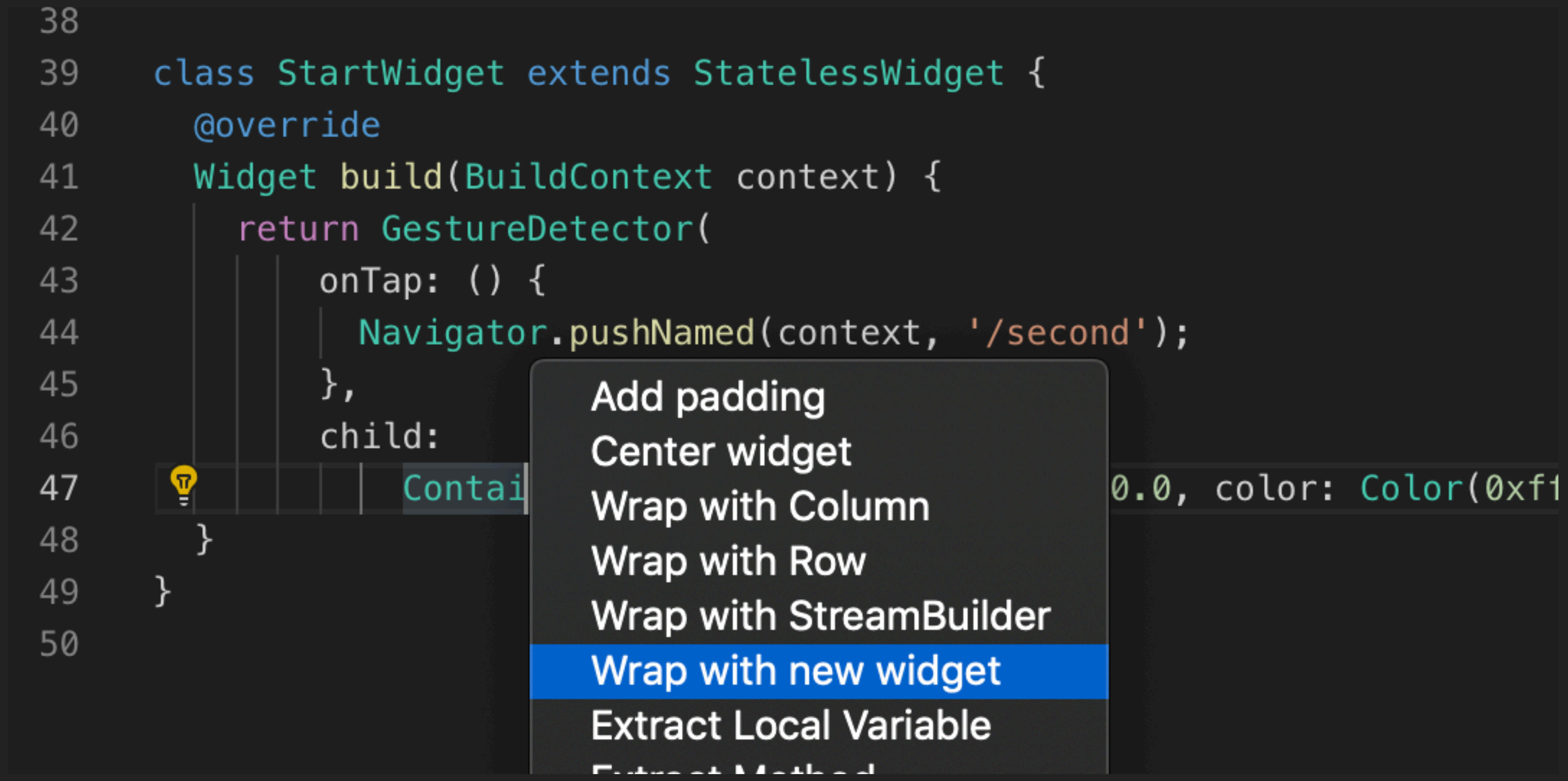


HERO



HERO

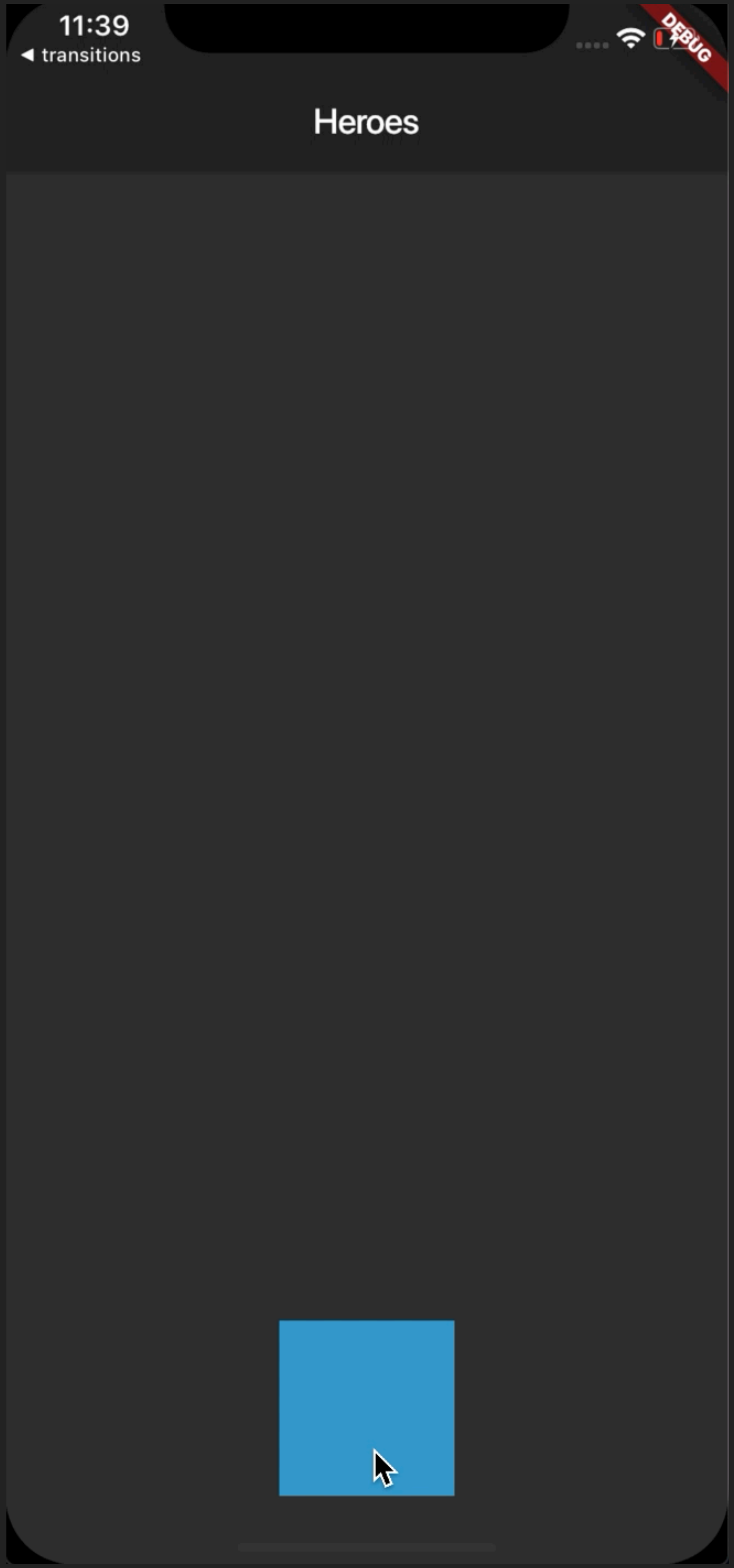
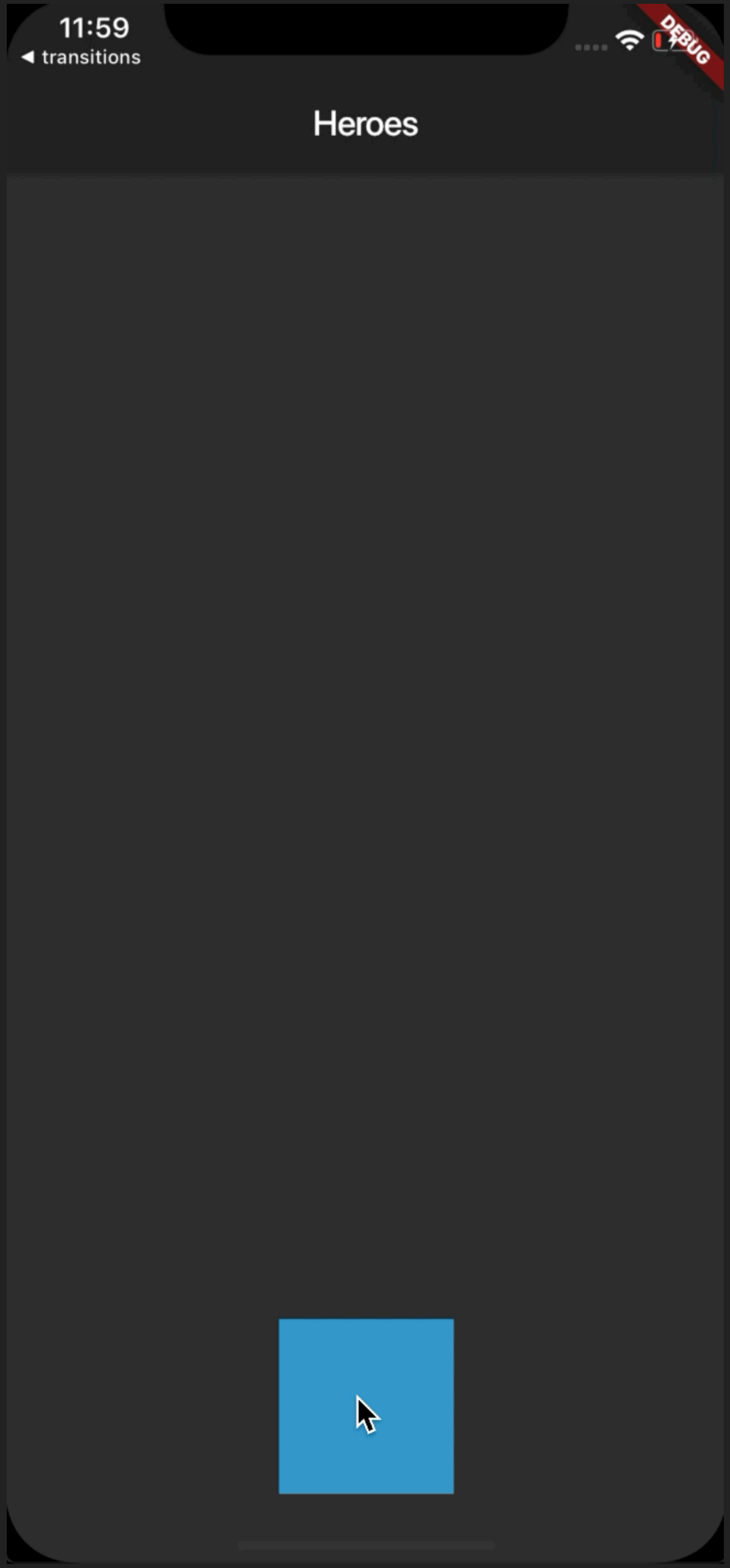




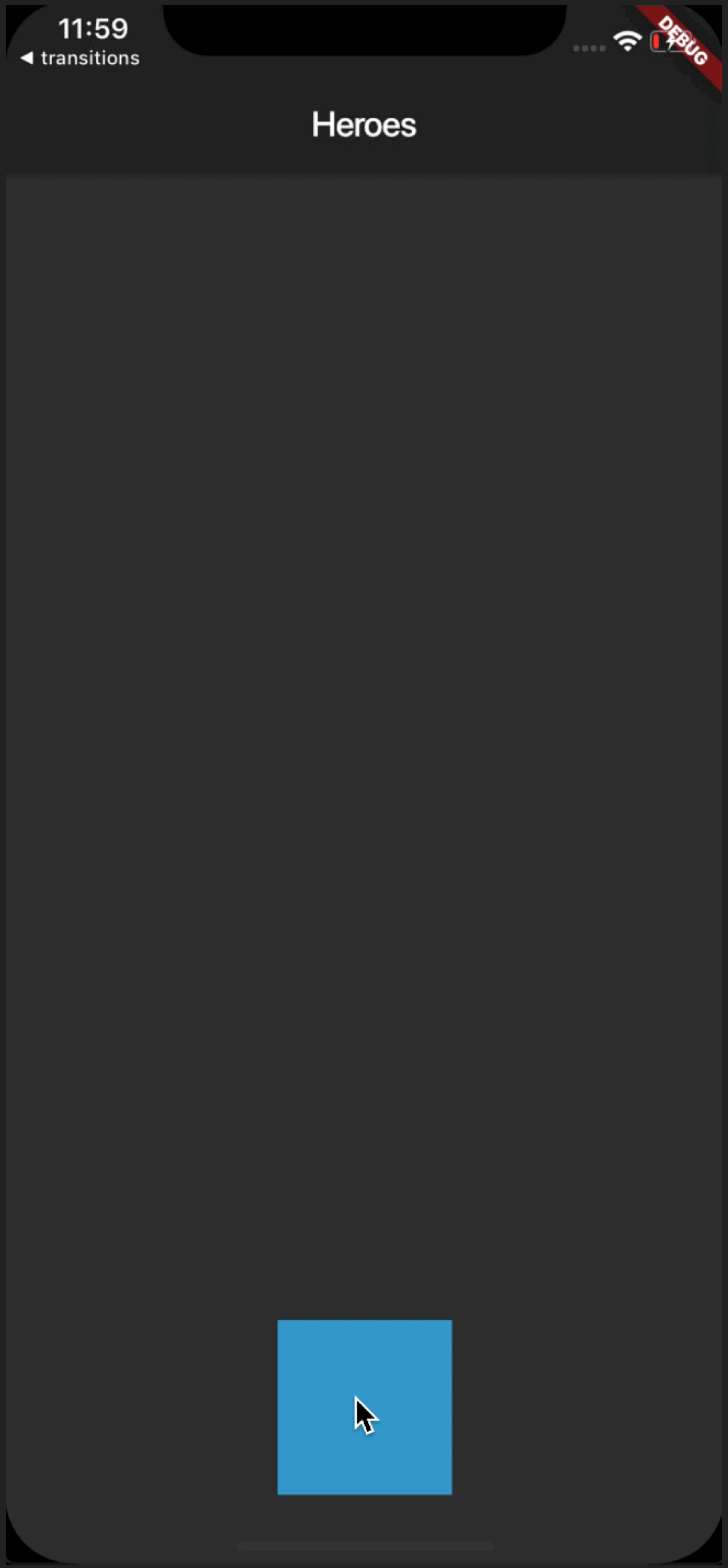
```
class StartWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: () {
        Navigator.pushNamed(context, '/second');
      },
      child:
        Contai                                  0.0, color: Color(0xf1
  }
}
```

```dart
class StartWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GestureDetector(
        onTap: () {
          Navigator.pushNamed(context, '/second');
        },
        child:
                    Hero(tag: 'tag-1',
                      child: Container(
                        width: 100.0,
                        height: 100.0,
                        color: Color(0xff3399cc))));
```

```
class Second extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Hero Home'),
      ),
      body: Center(


          Hero(
            tag: 'tag-1',
            child: Container(
              width: 400.0,
              height: 400.0,
              color: Color(0xffE42832)),
            )
```


11:39
◀ transitions
DEBUG
Heroes

11:59
◀ transitions
DEBUG
Heroes

```
      Hero(tag: 'tag-1',
           flightShuttleBuilder: (context, animation, direction, from, to) {
             return RotationTransition(
                 turns: Tween(begin: 0.0, end: 20.0).animate(animation),
                 child: to.widget,
               );
           },
           child: Container(
             width: 100.0,
             height: 100.0,
             color: Color(0xff3399cc))));
  }
```

11:39
◀ transitions
DEBUG
Heroes

11:59
◀ transitions
DEBUG
Heroes

12:08
◀ transitions
DEBUG
Heroes