

## ▼ Лабораторная работа No2 «Виджеты»

### Вариант 5.

используя 20 виджет и 4й виджет реализовать три варианта вращения красного, зеленого и синего объекта с разной скоростью, в пунктах меню должно быть написано red, green, blue

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'dart:math' as math;

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

class ActionSheetApp extends StatelessWidget {

  ActionSheetApp({super.key});

  @override
  Widget build(BuildContext context) {
    return CupertinoApp(
      theme: const CupertinoThemeData(brightness: Brightness.light),
      home: ActionSheetWithState(),
    );
  }
}

class ActionSheetWithState extends StatefulWidget{
  @override
  State<StatefulWidget> createState() => ActionSheetExample();
}

class ActionSheetExample extends State<ActionSheetWithState> {
  var text = "Choose in actions";
  var color = Colors.cyan;

  // This shows a CupertinoModalPopup which hosts a CupertinoActionSheet.
  void _showActionSheet(BuildContext context) {
    showCupertinoModalPopup<void>(
      context: context,
      builder: (BuildContext context) => CupertinoActionSheet(
        title: const Text('Title'),
        message: const Text('Message'),
        actions: <CupertinoActionSheetAction>[
          CupertinoActionSheetAction(
            /// This parameter indicates the action would be a default
            /// default behavior, turns the action's text to bold text.
            isDefaultAction: true,
            onPressed: () {
```

```

        setState(() {
          text = "GREEN";
          color = Colors.green;
        });
        Navigator.pop(context);
      },
      child: const Text('GREEN'),
    ),
    CupertinoActionSheetAction(
      isDefaultAction: true,
      onPressed: () {
        setState(() {
          text = "BLUE";
          color = Colors.blue;
        });
        Navigator.pop(context);
      },
      child: const Text('BLUE'),
    ),
    CupertinoActionSheetAction(
      /// This parameter indicates the action would perform
      /// a destructive action such as delete or exit and turns
      /// the action's text color to red.
      isDefaultAction: true,
      onPressed: () {
        setState(() {
          text = "RED";
          color = Colors.red;
        });
        Navigator.pop(context);
      },
      child: const Text('RED'),
    ),
  ],
),
);
}

@override
Widget build(BuildContext context) {
  // return MyStatefulWidget(text, color);
  return CupertinoPageScaffold(
    child: Center(
      child: CupertinoButton(
        child: MyStatefulWidget(text, color),
        onPressed: () => _showActionSheet(context),
      ),
    ),
  );
}

class MyStatefulWidget extends StatefulWidget {
  final String widgetText;
  final MaterialColor widgetColor;

```

```

const MyStatefulWidget(this.widgetText, this.widgetColor, {super.key});

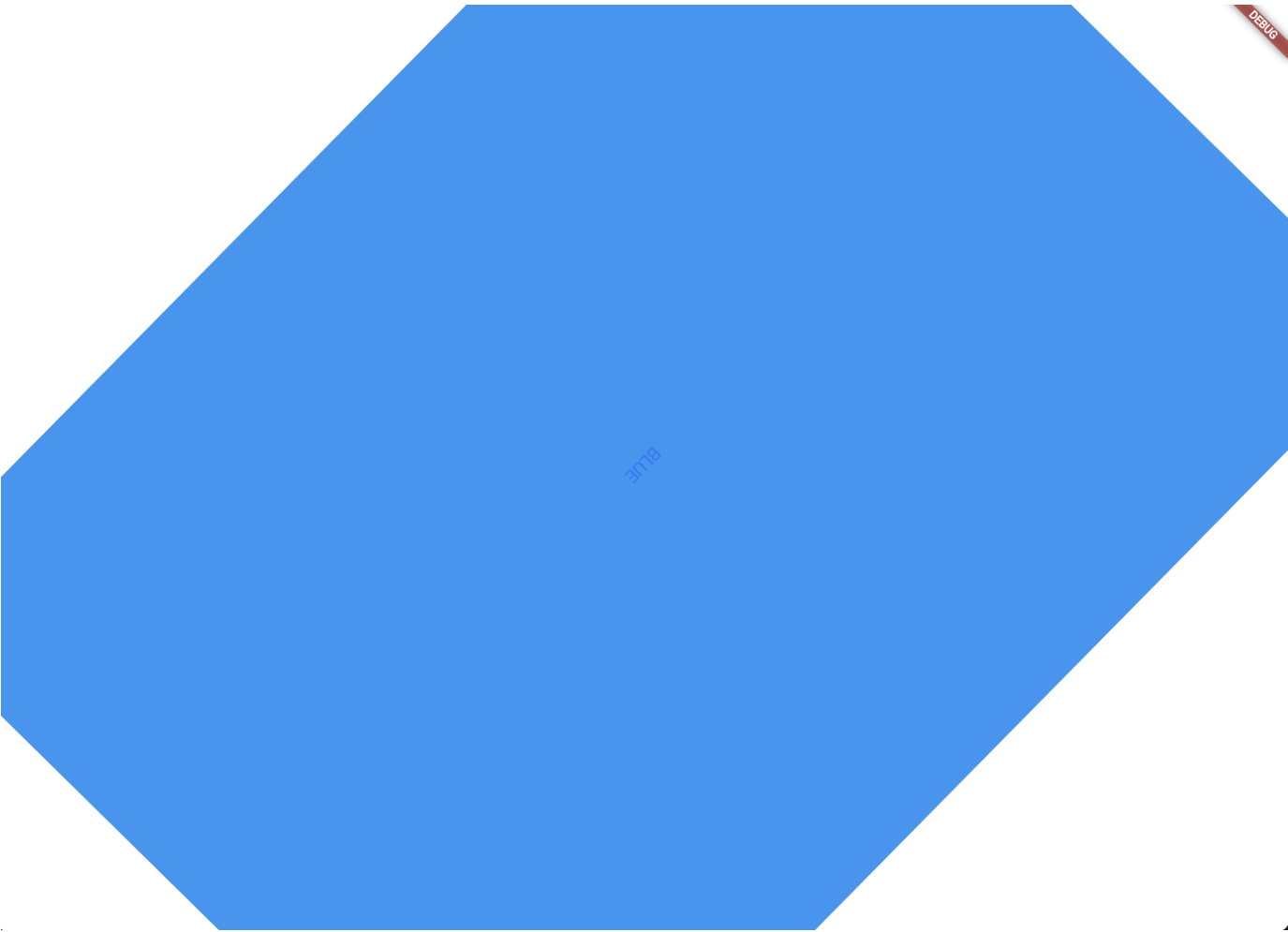
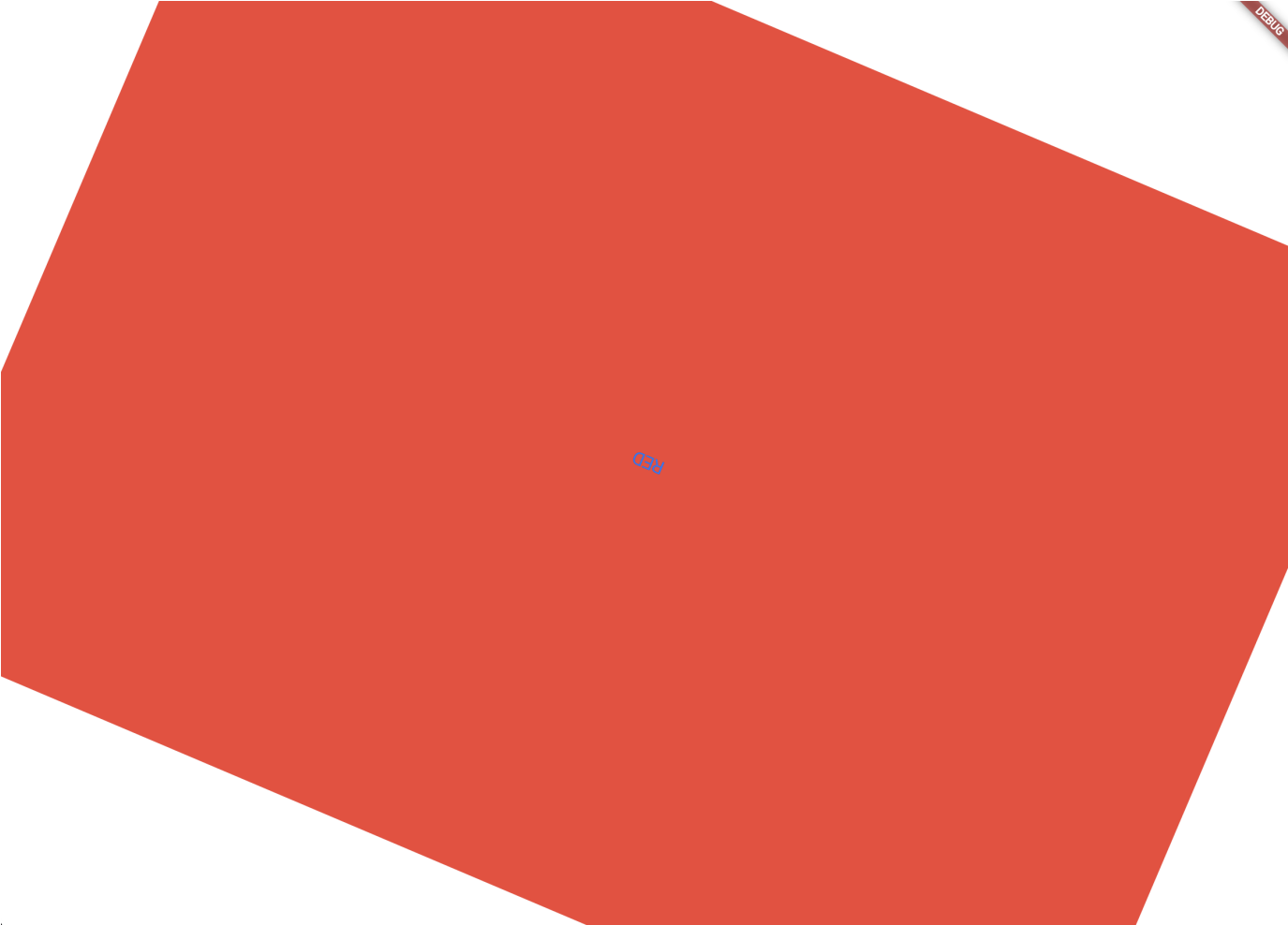
@override
State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}

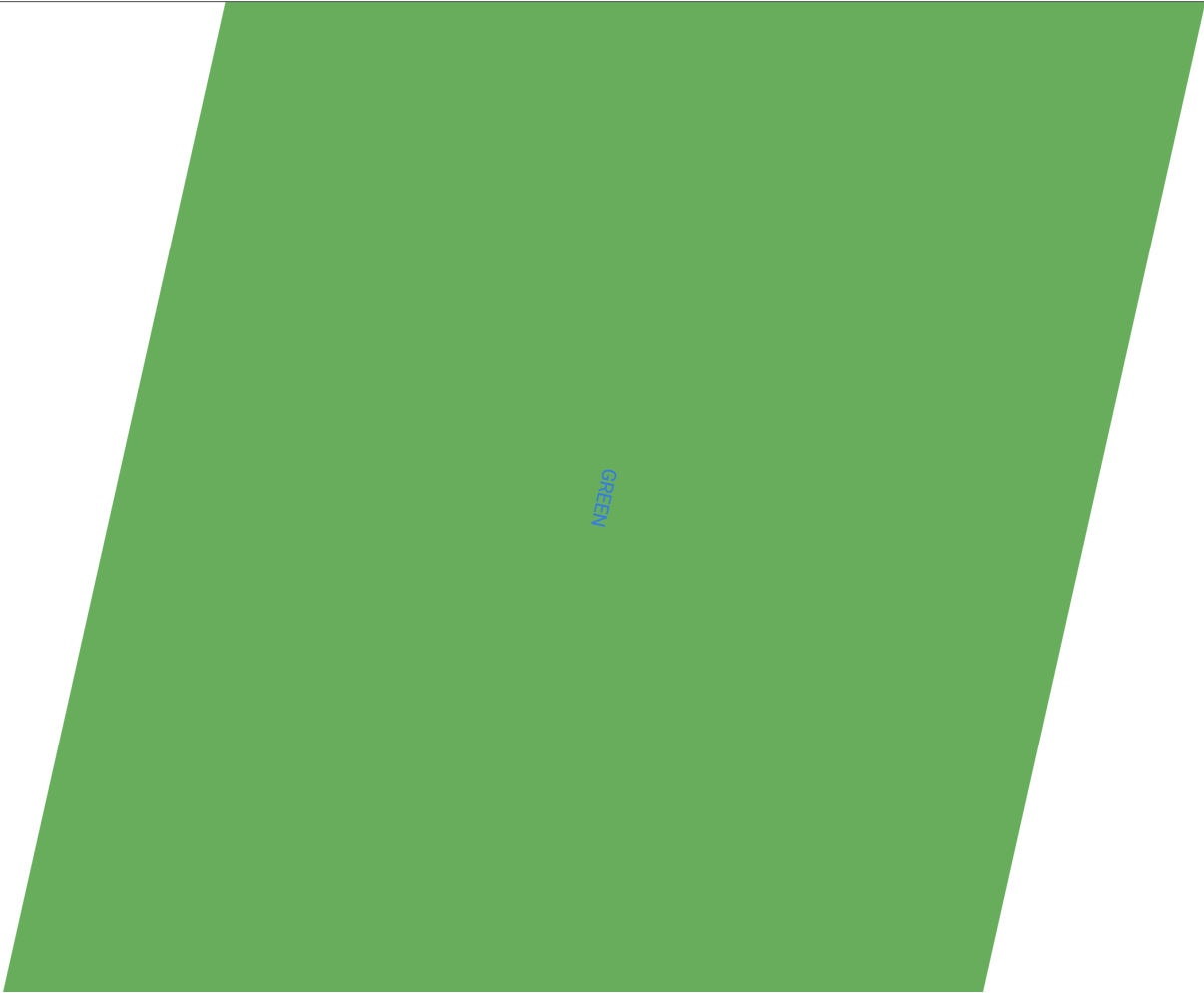
/// AnimationControllers can be created with `vsync: this` because of TickerProvide
class _MyStatefulWidgetState extends State<MyStatefulWidget>
  with TickerProviderStateMixin {
  late final AnimationController _controller = AnimationController(
    duration: const Duration(seconds: 100), // <---- скорость
    vsync: this,
  )..repeat();

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

  @override
  Widget build(BuildContext context) {
    print(widget.widgetText);
    print(widget.widgetColor);
    return AnimatedBuilder(
      animation: _controller,
      child: Container(
        //width: 100.0,
        //height: 100.0,
        color: widget.widgetColor,
        child: Center(
          child: Text(widget.widgetText),
        ),
      ),
      builder: (BuildContext context, Widget? child) {
        return Transform.rotate(
          angle: _controller.value * 2.0 * math.pi,
          // <-----угол поворота на 2 пи
          child: child,
        );
      },
    );
  }
}

```





Message

[Платные продукты Colab](#) - [Отменить подписку](#)

