

Лабораторная работа № 3

Проверка усвоенного материала по теме использования библиотек работы с 3D объектами

Задача

Реализовать мобильное приложение выводящее трехмерный объект по вариантам. Использование библиотеки на усмотрение программиста из рассмотренных на лекции.

Реализовать приложение моделирования движения глаз человека влево и вправо, модель головы человека можно взять из лекции. Движение глаз реализуется ползунком.

```
import 'dart:math';

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Cube',
      theme: ThemeData.dark(),
      home: MyHomePage(title: 'Flutter Cube Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key? key, this.title}) : super(key: key);

  final String? title;

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage>
  with SingleTickerProviderStateMixin {
  late Scene _scene;
  Object? _bunny;
```

```
Object? eye1;
Object? eye2;
late AnimationController _controller;
double _ambient = 0.1;
double _diffuse = 0.8;
double rotate = 0.0;
double _specular = 0.5;

void _onSceneCreated(Scene scene) {
  _scene = scene;
  scene.camera.position.z = 10;
  scene.light.position.setFrom(Vector3(0, 10, 10));
  scene.light.setColor(Colors.purple, _ambient, _diffuse, _specular);
  _bunny = Object(
    position: Vector3(0, -1.0, 0),
    scale: Vector3(10.0, 10.0, 10.0),
    lighting: true,
    fileName: 'assets/skull/12140_Skull_v3_L2.obj');
  eye1 = Object(
    position: Vector3(0.9, -3.0, 2.5),
    scale: Vector3(1.0, 1.0, 1.0),
    lighting: true,
    fileName: 'assets/eye.obj');
  eye2 = Object(
    position: Vector3(-0.8, -3.0, 2.5),
    scale: Vector3(1.0, 1.0, 1.0),
    lighting: true,
    fileName: 'assets/eye.obj');
  scene.world.add(_bunny!);
  scene.world.add(eye1!);
  scene.world.add(eye2!);
}

@override
void initState() {
  super.initState();
  _controller = AnimationController(
    duration: Duration(milliseconds: 30000), vsync: this)
    ..addListener(() {
      if (_bunny != null) {
        _bunny!.updateTransform();
        _scene.update();
      }
      if (eye1 != null) {
        eye1!.updateTransform();
        _scene.update();
      }
      if (eye2 != null) {
        eye2!.updateTransform();
        _scene.update();
      }
    })
    ..repeat();
}
```

```

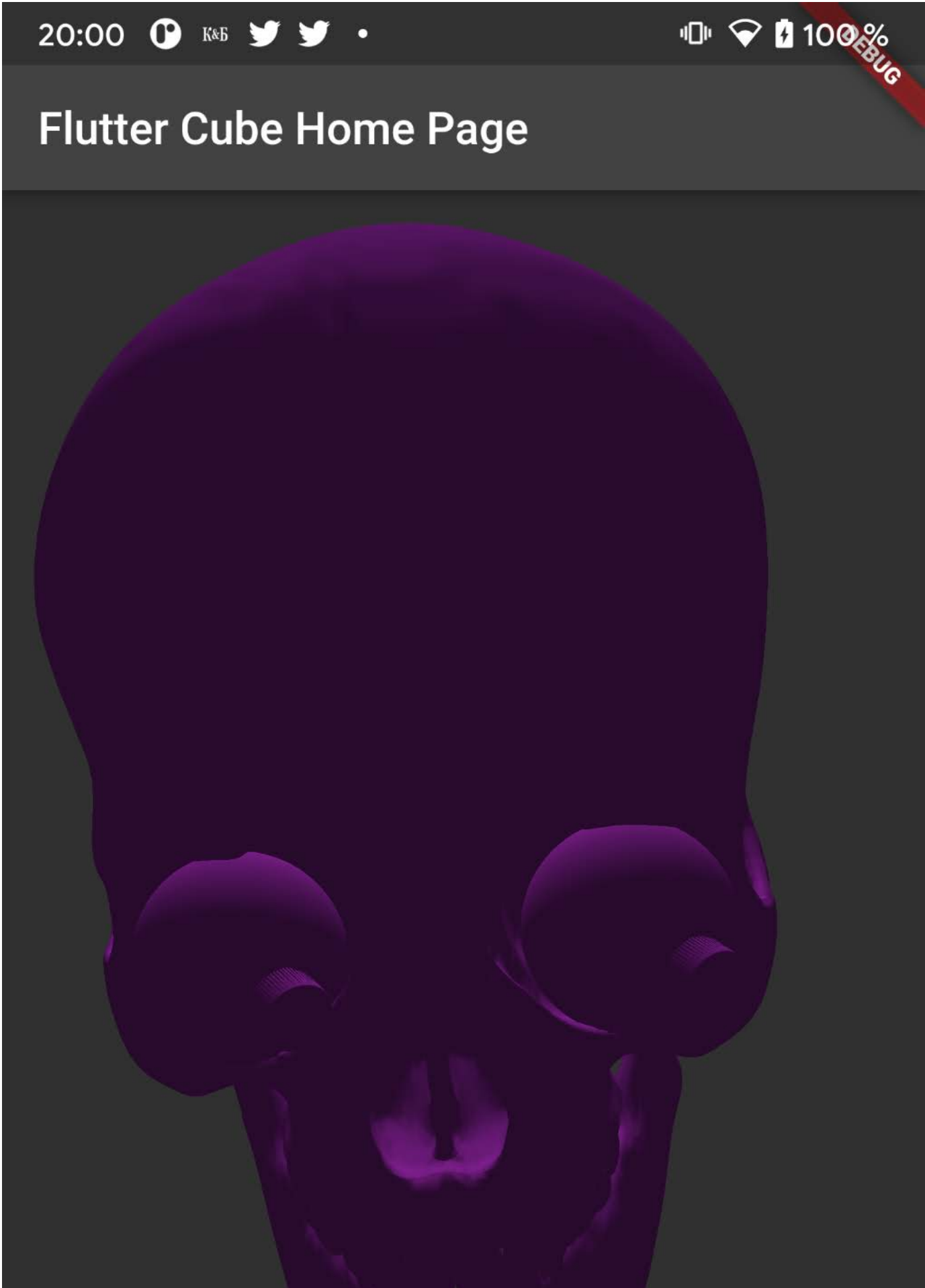
@override
void dispose() {
  _controller.dispose();
  super.dispose();
}

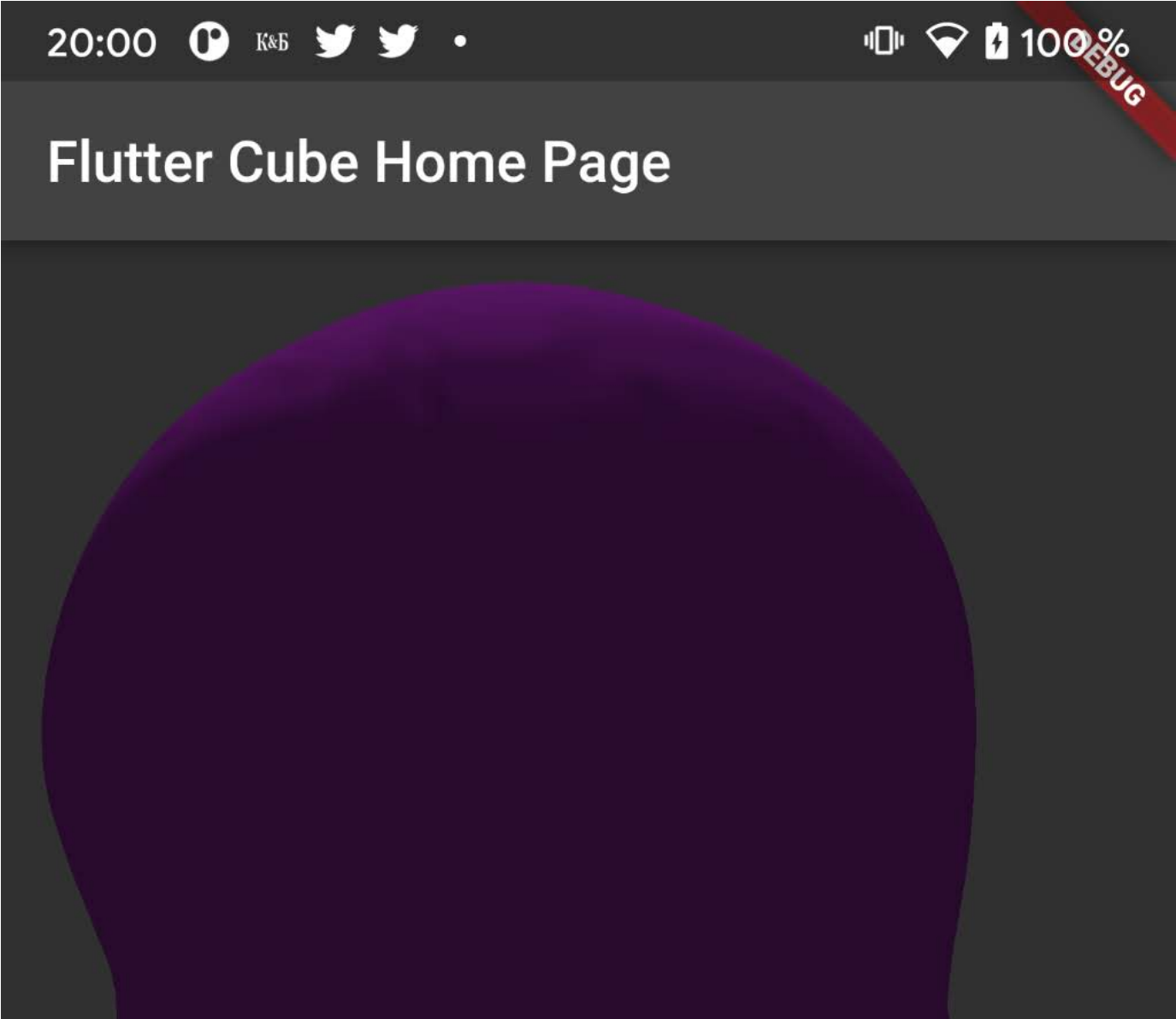
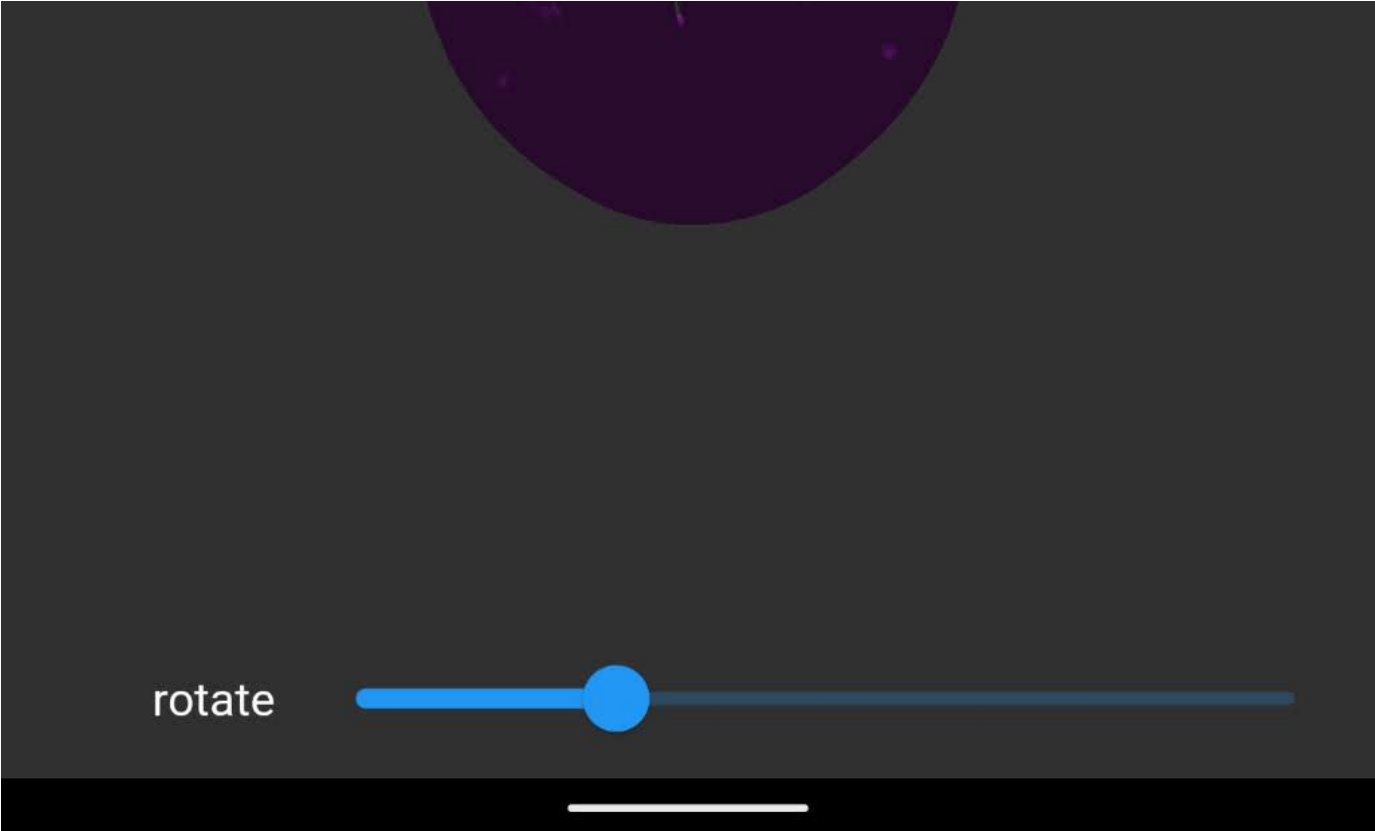
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title!),
    ),
    body: Stack(
      children: <Widget>[
        Cube(onSceneCreated: _onSceneCreated),
        Column(
          mainAxisAlignment: MainAxisAlignment.end,
          children: <Widget>[
            Row(
              mainAxisAlignment: MainAxisAlignment.end,
              children: <Widget>[
                Flexible(flex: 2, child: Text('rotate')),
                Flexible(
                  flex: 8,
                  child: Slider(
                    value: rotate,
                    min: 0.0,
                    max: 350.0,
                    divisions: 350,
                    onChanged: (value) {
                      setState(() {
                        rotate = value;
                        eye1!.rotation.z = _controller.value * rotate;
                        eye2!.rotation.z = _controller.value * rotate;
                      });
                    },
                  ),
                ),
              ],
            ),
          ],
        ),
      ],
    ),
    // Row(
    //   mainAxisAlignment: MainAxisAlignment.end,
    //   children: <Widget>[
    //     Flexible(flex: 2, child: Text('diffuse')),
    //     Flexible(
    //       flex: 8,
    //       child: Slider(
    //         value: _diffuse,
    //         min: 0.0,
    //         max: 1.0,
    //         divisions: 100,
    //         onChanged: (value) {
    //           setState(() {
    //             _diffuse = value;
    //             scene.light.setColor(

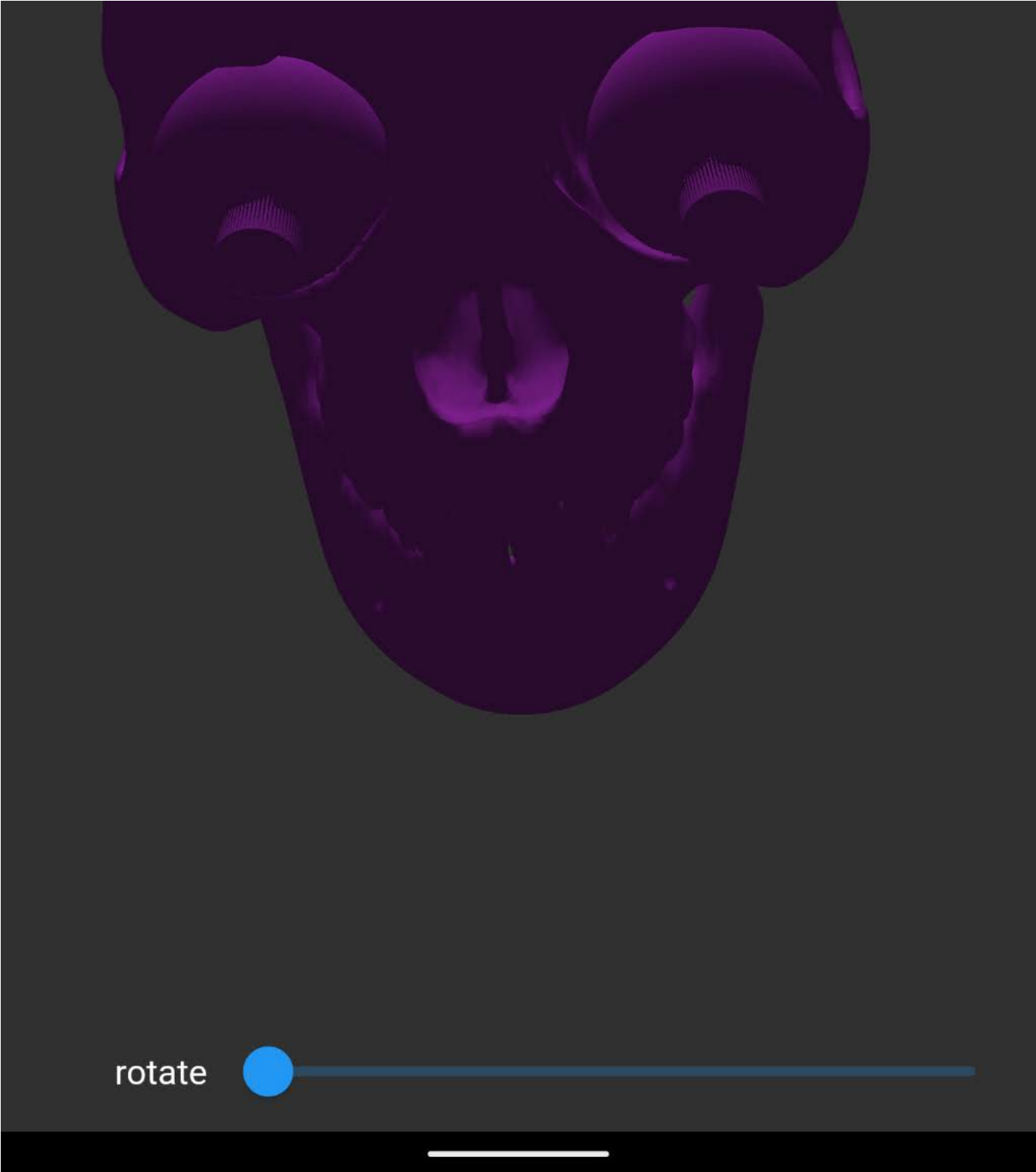
```

resea

```
}  
}  
}
```







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

