Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторная работа № 8**

**по дисциплине**

**«Технологии разработки моильных приложений»**

**Тема: Камера**

Выполнила:

ст. гр. ПРИ-117

А.И. Рощина

Принял:

Козаев А.Т.

Владимир, 2020

1. ЦЕЛЬ РАБОТЫ

Разработать мобильное приложение, которое будет использовать камеру устройства для вывода изображения на экран. Научиться обрабатывать событие поворота устройства и подгонять изображение под экран.

1. ВЫПОЛНЕНИЕ РАБОТЫ

Репозиторий проекта:

[https://github.com/flamebrier/tdma/tree/main/lab8](https://github.com/flamebrier/tdma/tree/main/lab11)

***Задания***

1. Выполнить пример из лабораторной работы.
2. Сделать снимок

***Ход работы***

Код:

import 'dart:io';

import 'package:camera/camera.dart';

import 'package:flutter/material.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

final cameras = await availableCameras();

final firstCamera = cameras.first;

runApp(MaterialApp(

title: 'Лабораторная 8',

theme: ThemeData(

primarySwatch: Colors.amber,

visualDensity: VisualDensity.adaptivePlatformDensity,

),

home: CameraScreen(camera: firstCamera),

));

}

class CameraScreen extends StatefulWidget {

final CameraDescription camera;

const CameraScreen({Key key, this.camera}) : super(key: key);

@override

\_CameraScreenState createState() => \_CameraScreenState();

}

class \_CameraScreenState extends State<CameraScreen> {

CameraController \_controller;

Future<void> \_initializeControllerFuture;

bool isFull = false;

String imagePath = "";

@override

void initState() {

super.initState();

\_controller = CameraController(

widget.camera,

ResolutionPreset.medium,

);

\_initializeControllerFuture = \_controller.initialize();

}

@override

void dispose() {

\_controller.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.black,

floatingActionButton: Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

FloatingActionButton(

heroTag: "btn1",

child: Icon(isFull

? Icons.fullscreen\_exit\_rounded

: Icons.fullscreen\_rounded),

backgroundColor: Colors.white,

onPressed: () {

if (mounted)

setState(() {

isFull = !isFull;

});

},

),

FloatingActionButton(

heroTag: "btn2",

child: Icon(Icons.camera\_alt),

onPressed: () async {

try {

await \_initializeControllerFuture;

final image = await \_controller.takePicture();

if (mounted)

setState(() {

imagePath = image?.path ?? "";

});

} catch (e) {

print(e);

}

},

),

FloatingActionButton(

heroTag: "btn3",

child: ClipOval(child: (imagePath ?? "").isNotEmpty

? Center(child: Container(width: double.infinity, child: Image.file(File(imagePath))))

: Icon(Icons.image\_rounded)),

backgroundColor: Colors.white,

onPressed: ((imagePath ?? "").isNotEmpty)

? () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => DisplayPictureScreen(

imagePath: imagePath,

),

),

);

}

: null,

)

],

),

floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,

body: FutureBuilder<void>(

future: \_initializeControllerFuture,

builder: (context, snapshot) {

if (snapshot.connectionState == ConnectionState.done) {

if (isFull) {

return Transform.scale(

scale: \_controller.value.aspectRatio,

child: Center(

child: CameraPreview(\_controller),

),

);

}

return Center(child: CameraPreview(\_controller));

} else {

return Center(child: CircularProgressIndicator());

}

}));

}

}

class DisplayPictureScreen extends StatelessWidget {

final String imagePath;

const DisplayPictureScreen({Key key, this.imagePath}) : super(key: key);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Display the Picture')),

backgroundColor: Colors.black,

body: Center(child: Image.file(File(imagePath))),

);

}

}



Рисунок 1 - Вывод изображения с камеры



Рисунок 2 - Подгонка изображения под размер экрана

За подгонку изображения под размер экрана отвечает следующий код:

if (isFull) {

return Transform.scale(

scale: \_controller.value.aspectRatio,

child: Center(

child: CameraPreview(\_controller),

),

);

}

return Center(child: CameraPreview(\_controller));

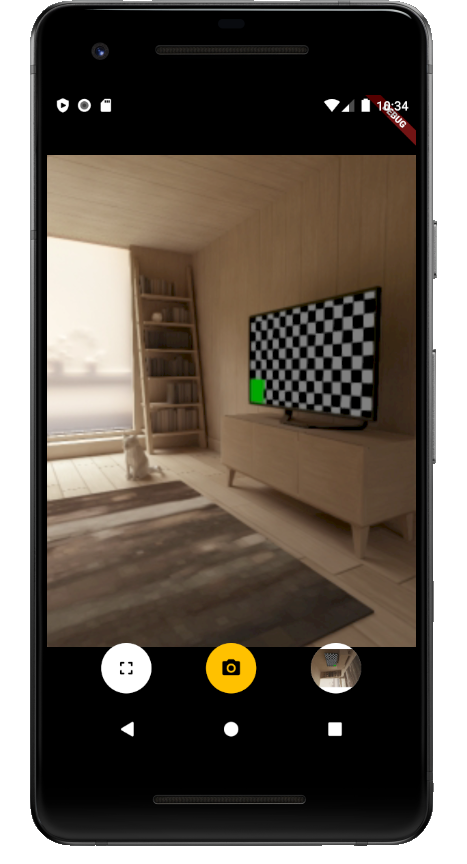


Рисунок 3 - Сделано фото



Рисунок 4 - Просмотр фото

1. ВЫВОД

В процессе выполнения лабораторной работы было разработано приложение, использующее камеру устройства для вывода изображения на экран.