import 'dart:math';

import 'package:flutter/material.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

home: Start(),

);

}

}

class Start extends StatefulWidget {

const Start({Key? key}) : super(key: key);

@override

State<Start> createState() => \_StartState();

}

class \_StartState extends State<Start> {

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.black26,

appBar: AppBar(

backgroundColor: Colors.black12,

title: Text("Guess Game"),

),

body: Center(

child: MaterialButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(builder: (context) => const Select()),

);

},

child: Container(

height: 50,

width: 200,

decoration: BoxDecoration(

color: Colors.green, borderRadius: BorderRadius.circular(30)),

child: Center(

child: Text(

"Start",

style: TextStyle(fontSize: 30, color: Colors.white),

))),

),

),

);

}

}

class Select extends StatefulWidget {

const Select({Key? key}) : super(key: key);

@override

State<Select> createState() => \_SelectState();

}

class \_SelectState extends State<Select> {

@override

int a = 0, b = 0;

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.red,

title: Text("Select Range"),

),

body: Column(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Text(

"Initial Value :",

style: TextStyle(fontSize: 30.0, color: Colors.orangeAccent),

),

Container(

child: Slider(

value: a.toDouble(),

onChanged: (double value) {

setState(() {

this.a = value.toInt();

});

},

min: 0,

max: 100,

divisions: 100,

),

padding: EdgeInsets.only(left: 30, right: 30),

),

Text(

a.toString(),

style: TextStyle(fontSize: 25.0),

),

Container(

height: 200,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: <Widget>[

Text(

"Final Value :",

style: TextStyle(fontSize: 30.0, color: Colors.orangeAccent),

),

Container(

child: Slider(

value: b.toDouble(),

onChanged: (double value) {

setState(() {

this.b = value.toInt();

});

},

min: 0,

max: 100,

divisions: 100,

),

padding: EdgeInsets.only(left: 30, right: 30),

),

Text(

b.toString(),

style: TextStyle(fontSize: 25.0),

),

],

),

),

SizedBox(

width: 400,

height: 60,

child: MaterialButton(

color: Colors.red,

child: Text(

"Next",

style: TextStyle(fontSize: 30, color: Colors.white),

),

shape:

OutlineInputBorder(borderRadius: BorderRadius.circular(15)),

onPressed: () {

Navigator.push(context,

MaterialPageRoute(builder: (BuildContext context) {

return ShowNumber(

i: a,

f: b,

);

}));

},

),

)

],

),

);

}

}

class ShowNumber extends StatefulWidget {

int i, f;

ShowNumber({Key? key, required this.i, required this.f}) : super(key: key);

@override

State<ShowNumber> createState() => \_ShowNumberState();

}

class \_ShowNumberState extends State<ShowNumber> {

@override

int \_randomNumber1 = 0;

int \_randomNumber2 = 0;

int \_randomNumber3 = 0;

List<gussModel> items = [];

gussModel correctNumber = gussModel(0, -1, false);

gussModel selectdNumber = gussModel(0, -1, false);

int score = 0;

bool isCorrect = false;

Random random = Random();

void first() {

var min = widget.i;

var max = widget.f;

\_randomNumber1 = min + random.nextInt(max - min);

gussModel a = gussModel(1, \_randomNumber1, false);

setState(() {

items.add(a);

});

}

void second() {

var min = widget.i;

var max = widget.f;

\_randomNumber2 = min + random.nextInt(max - min);

gussModel a = gussModel(1, \_randomNumber2, false);

setState(() {

items.add(a);

});

}

void third() {

var min = widget.i;

var max = widget.f;

\_randomNumber3 = min + random.nextInt(max - min);

gussModel a = gussModel(1, \_randomNumber3, false);

setState(() {

items.add(a);

});

}

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.red.shade50,

appBar: AppBar(

backgroundColor: Colors.red,

title: Text("GENERATE NUMBER AND SELECT"),

),

body: Column(

crossAxisAlignment: CrossAxisAlignment.stretch,

mainAxisAlignment: MainAxisAlignment.center,

children: [

Padding(

padding: const EdgeInsets.all(15.0),

child: SizedBox(

width: 200,

height: 60,

child: MaterialButton(

color: Colors.amber,

child: Text("Generate Number and Guass the Number true"),

shape:

OutlineInputBorder(borderRadius: BorderRadius.circular(15)),

onPressed: () {},

),

),

),

Padding(

padding: const EdgeInsets.all(15.0),

child: SizedBox(

width: 100,

height: 60,

child: MaterialButton(

color: Colors.blueGrey,

child: Text(

"Click me",

style: TextStyle(fontSize: 30, color: Colors.white),

),

shape:

OutlineInputBorder(borderRadius: BorderRadius.circular(15)),

onPressed: () {

items = [];

correctNumber = gussModel(0, -1, false);

first();

second();

third();

correctNumber = items[0 + random.nextInt(2)];

},

),

),

),

Column(

children: items!.map((item) {

return Padding(

padding: const EdgeInsets.all(15.0),

child: GestureDetector(

onTap: () {

setState(() {

selectdNumber = item;

selectdNumber.isCorrect = correctNumber.num == item.num;

});

},

child: Container(

height: 50,

width: 150,

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(12),

color: item.num == selectdNumber.num

? ((selectdNumber.isCorrect ?? false)

? Colors.green

: Colors.red)

: Colors.black),

child: Center(

child: Text(

item.num.toString(),

style: TextStyle(fontSize: 25, color: Colors.white),

),

)),

));

}).toList(),

),

],

),

);

}

}

class gussModel {

int? id;

int? num;

bool? isCorrect;

gussModel(this.id, this.num, this.isCorrect);

}