import 'package:flutter/material.dart';

import 'dart:ui' as ui;

void main() {

runApp(DrawingApp());

}

class DrawingApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

home: DrawingPage(),

debugShowCheckedModeBanner: false,

);

}

}

class DrawingPage extends StatefulWidget {

@override

\_DrawingPageState createState() => \_DrawingPageState();

}

class \_DrawingPageState extends State<DrawingPage> {

List<Offset?> \_points = [];

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text("Simple Drawing App"),

),

body: GestureDetector(

onPanUpdate: (details) {

setState(() {

// Add current finger position to points list

RenderBox renderBox = context.findRenderObject() as RenderBox;

Offset localPosition = renderBox.globalToLocal(details.globalPosition);

\_points.add(localPosition);

});

},

onPanEnd: (details) {

setState(() {

\_points.add(null); // Add a null to mark the end of a stroke

});

},

child: CustomPaint(

painter: DrawingPainter(points: \_points),

size: Size.infinite,

child: Container(

color: Colors.white, // Set background to white

),

),

),

);

}

}

class DrawingPainter extends CustomPainter {

final List<Offset?> points;

DrawingPainter({required this.points});

final Paint \_paint = Paint()

..color = Colors.black

..strokeWidth = 4.0

..strokeCap = StrokeCap.round

..isAntiAlias = true;

@override

void paint(Canvas canvas, Size size) {

for (int i = 0; i < points.length - 1; i++) {

if (points[i] != null && points[i + 1] != null) {

canvas.drawLine(points[i]!, points[i + 1]!, \_paint);

}

}

}

@override

bool shouldRepaint(DrawingPainter oldDelegate) => true;

}