Text

Description automatically generated

|  |  |
| --- | --- |
| **Assignment No** | Assignment No 06 |

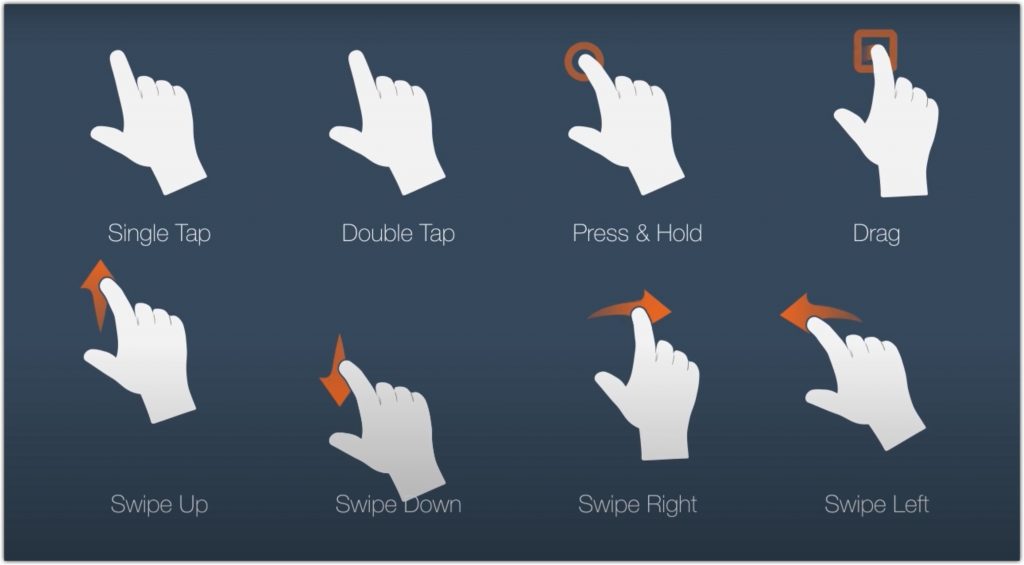
Assignment Number - 06

**Title :** Flutter InteractiveViewer and GestureDetector widget

with Double Tap zoom in an out image

**Theory :**

**Gestures**

****

**Figure 1: Gesture by using Pointer**

Gestures represent semantic actions (for example, tap, drag, and scale) that are recognized from multiple individual pointer events, potentially even multiple individual pointers.

Gestures can dispatch multiple events, corresponding to the lifecycle of the gesture (for example, drag start, drag update, and drag end). Gestures define any physical action or movement of a user in the intention of specific control of the mobile device.

The gesture system in Flutter has two separate layers.

* The first layer has raw pointer events that describe the location and movement of pointers (for example, touches, mice, and styli) across the screen.
* The second layer has *gestures* that describe semantic actions that consist of one or more pointer movements.

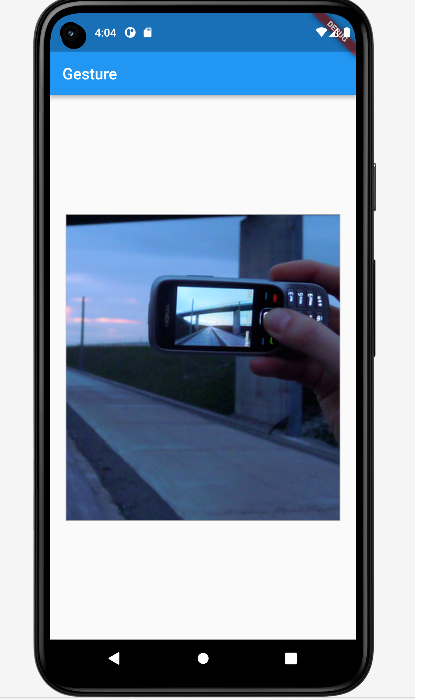
**Pointer:**

* Pointers represent raw data about the user’s interaction with the device’s screen.
* Pointers are the first layer that represents the raw data about user interaction.
* It has events, which describe the **location** and **movement** of pointers such as touches, mice, and style across the screens.
* Flutter does not provide any mechanism to cancel or stop the pointer-events from being dispatched further.
* Flutter provides a **Listener**widget to listen to the pointer-events directly from the widgets layer.

**Source Code:-**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Flutter Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: MyHomePage(),  
 );  
 }  
}  
  
class MyHomePage extends StatefulWidget {  
  
 @override  
 State<MyHomePage> createState() => \_MyHomePageState();  
}  
class \_MyHomePageState extends State<MyHomePage> {  
  
 final zoomTransformationController = TransformationController();  
  
 bool \_zoomedIn = false;  
  
 void \_DoubleTapZoomIn(){  
 zoomTransformationController.value = Matrix4.identity()..scale(3.0);  
 }  
 void \_DoubleTapZoomOut(){  
 zoomTransformationController.value = Matrix4.identity();  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 double screenWidth = MediaQuery.*of*(context).size.width;  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Gesture'),  
 ),  
 body: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 GestureDetector(  
 onDoubleTap: (){  
 if(\_zoomedIn == false){  
 \_DoubleTapZoomIn();  
 }else {  
 \_DoubleTapZoomOut();  
 }  
 setState(() {  
 \_zoomedIn = !\_zoomedIn;  
 });  
 },  
 child: Container(  
 width: screenWidth,  
 height: screenWidth,  
 margin: EdgeInsets.all(20),  
 decoration: BoxDecoration(  
 border: Border.all(width: 1, color: Colors.*grey*)  
 ),  
 child: InteractiveViewer(  
 boundaryMargin: EdgeInsets.*zero*,  
 transformationController: zoomTransformationController,  
 maxScale: 3.0,  
 minScale: 1,  
 child : Image.network('https://upload.wikimedia.org/wikipedia/commons/thumb/b/b6/Image\_created\_with\_a\_mobile\_phone.png/1024px-Image\_created\_with\_a\_mobile\_phone.png'),  
  
 ),  
 ),  
 ),  
 ],  
 ),  
 );  
 }  
}

**Output:**



**Conclusion :** in this assignment I have learn about flutter InteractiveViewer and GestureDetector widgets and their working If you want to use both the InteractiveViewer and GestureDetector widgets together to achieve double-tap zoom functionality, you can use a combination of the two