

MODULE

Android Touch Event Handling

*Nawfal Ali**Updated 25 May 2020*

In the previous weeks, we learn how to translate a touch on the screen into an action. In this week, we will learn how to intercept touch events in an Activity or View and execute the appropriate behaviour.

Assignment Project Exam Help

What is Gesture?

A gesture is simply a sequence of touch events. Each touch event comes with x and y coordinates. A gesture starts with the touch-down event, continues as the system tracks the position of the user's finger(s), and end by the touch-up event.

Intercepting Touch Events

Touch events can be intercepted by a view through the overriding of `onTouchEvent()` method or the registration of an **`onTouchListener`** and the implementation of `onTouch()` callback method.

What is MotionEvent?

Object used to report movement (mouse, pen, finger, trackball) events. Motion events may hold either absolute or relative movements and other data, depending on the type of device.

It is an object that is passed through to the onTouch() callback method. It is the key to obtaining information about motion events such as the location of the touch within the view, the type of the event, and others.

Types of Events

Assignment Project Exam Help

There are several types the MotionEvent object can report and here are some of them:

<https://tutorcs.com>

- **MotionEvent.ACTION_DOWN**: This event is generated when the first touch on a view occurs.
- **MotionEvent.ACTION_UP**: This event is generated when the touch is lifted from the screen.
- **MotionEvent.ACTION_MOVE**: Any motion of the touch between the ACTION_DOWN and ACTION_UP events will be represented by this event.

WeChat: cstutorcs

Overriding onTouchEvent() method

If you want to listen to all the touch events that might occur on your activity (if none of the child views handle it), it is possible to implement onTouchEvent() callback method as shown in the following example.

```
1. public class MainActivity extends AppCompatActivity {  
2.  
3.     private static final String DEBUG_TAG = "WEEK10_TAG";
```

```
4.
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.     }
10.
11.     @Override
12.     public boolean onTouchEvent(MotionEvent event) {
13.         int action = event.getActionMasked();
14.         switch(action) {
15.             case (MotionEvent.ACTION_DOWN) :
16.                 Log.d(DEBUG_TAG, "Action was DOWN");
17.                 return true;
18.             case (MotionEvent.ACTION_MOVE) :
19.                 Log.d(DEBUG_TAG, "Action was MOVE");
20.                 return true;
21.             case (MotionEvent.ACTION_UP) :
22.                 Log.d(DEBUG_TAG, "Action was UP");
23.                 return true;
24.             default :
25.                 return false;
26.         }
27.     }
28. }
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

The activity implements `onTouchEvent()` method at line@12. The input parameter to this callback method is of type `MotionEvent`. Using this input parameter, we can extract the type of the event as shown in line@13. The `getActionMasked()` method returns an integer number (constant) that represents the type of the current event. The action is then compared with pre-defined constant values to determine the type of the current event as implemented by the switch case in lines@14-26.

Registering onTouchListener

This approach allows us to listen to a specific view instead of the entire layout.

```
1. public class MainActivity extends AppCompatActivity {
2.
3.     private static final String DEBUG_TAG = "WEEK10_TAG";
4.
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.         View view=findViewById(R.id.my_layout);
10.        view.setOnClickListener(new View.OnClickListener() {
11.            @Override
12.            public boolean onTouch(View v, MotionEvent event) {
13.                int action = event.getActionMasked();
14.                switch(action) {
15.                    case (MotionEvent.ACTION_DOWN) :
16.                        Log.d(DEBUG_TAG, "Action was DOWN");
17.                        return true;
18.                    case (MotionEvent.ACTION_MOVE) :
19.                        Log.d(DEBUG_TAG, "Action was MOVE");
20.                        return true;
21.                    case (MotionEvent.ACTION_UP) :
22.                        Log.d(DEBUG_TAG, "Action was UP");
23.                        return true;
24.                    default :
25.                        return false;
26.                }
27.            }
28.        });
29.    }
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```
30.  
31. }
```

As shown in the code above, the activity retrieves a reference to the layout that has `id=my_layout` at line@9. The code then registers a touch listener by calling `setOnTouchListener()` method and providing an anonymous instance of `View.OnTouchListener()` that implements `onTouch()` callback Event (line@12).

The callback method `onTouch()` accepts as input the `MotionEvent` object that provides the type of the event and its coordinates.

What does the return true statement indicate?

It indicates that you have handled the touch event.

Assignment Project Exam Help

<https://tutorcs.com>

What does the return false statement indicate?

It indicates that you have not handled the current event.

WeChat: estutores

MotionEvent Position

In order to query the position of the current event, we must call `getX()` and `getY()` methods to retrieve the absolute coordinates relative to the View, that dispatched them.

```
1. public class MainActivity extends AppCompatActivity {  
2.  
3.     private static final String DEBUG_TAG = "WEEK10_TAG";  
4.  
5.     @Override
```

```

6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.activity_main);
9.         View view=findViewById(R.id.my_layout);
10.        view.setOnTouchListener(new View.OnTouchListener() {
11.            @Override
12.            public boolean onTouch(View v, MotionEvent event) {
13.                int x=(int)event.getX();
14.                int y=(int)event.getY();
15.
16.                int action = event.getActionMasked();
17.                switch(action) {
18.                    case (MotionEvent.ACTION_DOWN) :
19.                        Log.d(DEBUG_TAG,"Action was DOWN at x="+x+ " and
20.                            y="+y);
21.                        return true;
22.                    case (MotionEvent.ACTION_MOVE) :
23.                        Log.d(DEBUG_TAG,"Action was MOVE at x="+x+ " and
24.                            y="+y);
25.                        return true;
26.                    case (MotionEvent.ACTION_UP) :
27.                        Log.d(DEBUG_TAG,"Action was UP at x="+x+ " and
28.                            y="+y);
29.                        return true;
30.                    default :
31.                        return false;
32.                }
33.            }
34.        });
35.    }

```

Assignment Project Exam Help
<https://tutores.com>
 WeChat: cstutorcs

Lines 13 and 14 retrieve the x and y coordinates of the current event. The return type of both methods `getX()` and `getY()` is float and this is due to sub-pixel accuracy.

Now, let's deploy the code above into an emulator and test it.

1. D/WEEK10_TAG: Action was DOWN at x=334 and y=653
2. D/WEEK10_TAG: Action was UP at x=334 and y=653

and here is the log when some move events generated:

1. D/WEEK10_TAG: Action was DOWN at x=416 and y=431
2. D/WEEK10_TAG: Action was MOVE at x=416 and y=436
3. D/WEEK10_TAG: Action was MOVE at x=420 and y=440
4. D/WEEK10_TAG: Action was MOVE at x=420 and y=448
5. D/WEEK10_TAG: Action was MOVE at x=422 and y=451
6. D/WEEK10_TAG: Action was MOVE at x=422 and y=456
7. D/WEEK10_TAG: Action was MOVE at x=422 and y=458
8. D/WEEK10_TAG: Action was MOVE at x=424 and y=462
9. D/WEEK10_TAG: Action was MOVE at x=425 and y=460
10. D/WEEK10_TAG: Action was UP at x=425 and y=460

As you can see the first and last events are ACTION_DOWN and ACTION_UP respectively.

Is there a way to get the X and Y coordinates relative to the device screen?

YES. You have to use `getRawX()` and `getRawY()`.

To test the two new methods, let's listen to the touch events in a view in a layout.

1. `<?xml version="1.0" encoding="utf-8"?>`

```
2. <androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:app="http://schemas.android.com/apk/res-auto"
4.     xmlns:tools="http://schemas.android.com/tools"
5.     android:id="@+id/my_layout"
6.     android:layout_width="match_parent"
7.     android:layout_height="match_parent"
8.     tools:context=".MainActivity">
9.
10.    <FrameLayout
11.        android:id="@+id/frame_layout_id"
12.        android:layout_width="209dp"
13.        android:layout_height="329dp"
14.        android:background="#4AE61F"
15.        app:layout_constraintBottom_toBottomOf="parent"
16.        app:layout_constraintEnd_toEndOf="parent"
17.        app:layout_constraintStart_toStartOf="parent"
18.        app:layout_constraintTop_toTopOf="parent">
19.
20.    </FrameLayout>
21.
22.    <androidx.constraintlayout.widget.Guideline
23.        android:id="@+id/guideline"
24.        android:layout_width="wrap_content"
25.        android:layout_height="wrap_content"
26.        android:orientation="horizontal"
27.        app:layout_constraintGuide_percent="0.5909713" />
28.
29.    <TextView
30.        android:id="@+id/action_type"
31.        android:layout_width="0dp"
32.        android:layout_height="wrap_content"
33.        android:layout_marginTop="36dp"
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs


```
34.         android:textSize="30sp"
35.         app:layout_constraintEnd_toEndOf="parent"
36.         app:layout_constraintHorizontal_bias="0.5"
37.         app:layout_constraintStart_toEndOf="@+id/textView2"
38.         app:layout_constraintTop_toTopOf="@+id/guideline" />
39.
40.     <TextView
41.         android:id="@+id/textView2"
42.         android:layout_width="wrap_content"
43.         android:layout_height="wrap_content"
44.         android:layout_marginStart="8dp"
45.         android:layout_marginEnd="8dp"
46.         android:text="Action Type"
47.         android:textSize="30sp"
48.         app:layout_constraintEnd_toStartOf="@+id/action_type"
49.         app:layout_constraintHorizontal_bias="0.5"
50.         app:layout_constraintStart_toStartOf="parent"
51.         app:layout_constraintTop_toTopOf="@+id/action_type" />
52.
53.     <TextView
54.         android:id="@+id/textView3"
55.         android:layout_width="227dp"
56.         android:layout_height="wrap_content"
57.         android:layout_marginStart="8dp"
58.         android:layout_marginEnd="8dp"
59.         android:text="getX(),getY()"
60.         android:textSize="24sp"
61.         app:layout_constraintEnd_toStartOf="@+id/get_x_y_id"
62.         app:layout_constraintHorizontal_bias="0.5"
63.         app:layout_constraintStart_toStartOf="parent"
64.         app:layout_constraintTop_toTopOf="@+id/get_x_y_id" />
65.
66.     <TextView
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```
67.         android:id="@+id/get_x_y_id"
68.         android:layout_width="0dp"
69.         android:layout_height="wrap_content"
70.         android:layout_marginStart="8dp"
71.         android:layout_marginTop="32dp"
72.         android:textSize="24sp"
73.         app:layout_constraintEnd_toEndOf="parent"
74.         app:layout_constraintHorizontal_bias="0.5"
75.         app:layout_constraintStart_toEndOf="@+id/textView3"
76.         app:layout_constraintTop_toBottomOf="@+id/action_type" />
77.
78.     <TextView
79.         android:id="@+id/textView5"
80.         android:layout_width="227dp"
81.         android:layout_height="wrap_content"
82.         android:layout_marginTop="32dp"
83.         android:text="getRawX, getRawY()"
84.         android:textSize="24sp"
85.         app:layout_constraintEnd_toStartOf="@+id/get_raw_x_y_id"
86.         app:layout_constraintHorizontal_bias="0.5"
87.         app:layout_constraintStart_toStartOf="parent"
88.         app:layout_constraintTop_toBottomOf="@+id/textView3" />
89.
90.     <TextView
91.         android:id="@+id/get_raw_x_y_id"
92.         android:layout_width="0dp"
93.         android:layout_height="wrap_content"
94.         android:layout_marginStart="8dp"
95.         android:textSize="24sp"
96.         app:layout_constraintEnd_toEndOf="parent"
97.         app:layout_constraintHorizontal_bias="0.5"
98.         app:layout_constraintStart_toEndOf="@+id/textView5"
99.         app:layout_constraintTop_toTopOf="@+id/textView5" />
```

Assignment Project Exam Help

<https://tutores.com>

WeChat: cstutorcs

100. `</androidx.constraintlayout.widget.ConstraintLayout>`

and here is the activity controller:

```
1. public class MainActivity extends AppCompatActivity {
2.
3.
4.     TextView actionType;
5.     TextView getX;
6.     TextView getRawXY;
7.
8.     @Override
9.     protected void onCreate(Bundle savedInstanceState) {
10.         super.onCreate(savedInstanceState);
11.         setContentView(R.layout.activity_main);
12.         View view=findViewById(R.id.frame_layout_id);
13.         actionType=findViewById(R.id.action_type);
14.         getX=findViewById(R.id.get_x_y_id);
15.         getRawXY=findViewById(R.id.get_raw_x_y_id);
16.         view.setOnClickListener(new View.OnClickListener() {
17.             @Override
18.             public boolean onTouch(View v, MotionEvent event) {
19.                 int x=(int)event.getX();
20.                 int y=(int)event.getY();
21.                 int rawX=(int)event.getRawX();
22.                 int rawY=(int)event.getRawY();
23.                 getX.setText(x+", "+y);
24.                 getRawXY.setText(rawX+", "+rawY);
25.
26.                 int action = event.getActionMasked();
27.                 switch(action) {
28.                     case (MotionEvent.ACTION_DOWN) :
29.                         actionType.setText("Down");
```

```
30.         return true;
31.     case (MotionEvent.ACTION_MOVE) :
32.         actionTypes.setText("MOVE");
33.         return true;
34.     case (MotionEvent.ACTION_UP) :
35.         actionTypes.setText("UP");
36.         return true;
37.     default :
38.         return false;
39.     }
40. }
41. });
42. }
43. }
```

Assignment Project Exam Help

Let's test it: <https://tutorcs.com>

WeChat: cstutorcs

0:00 / 2:14

References:

- Android Studio 3.5 Development Essentials – Java Edition
- <https://developer.android.com/>

Copyright © Monash University, unless otherwise stated. All Rights Reserved, except for individual components (or items) marked with their own licence restrictions



Copyright © 2021 Monash University, unless otherwise stated

<https://tutorcs.com>

Disclaimer and Copyright
Privacy

Service Status

WeChat: cstutorcs