



Asignatura: Application Development for Mobile Devices.

Tarea 12: Canvas y MotionEvent.

La clase MotionEvent.

Cuando el usuario toca la pantalla por primera vez, el sistema entrega un evento táctil al `View` con el código de acción `ACTION_DOWN` y un conjunto de valores que incluyen las coordenadas X e Y del toque e información sobre la presión, el tamaño y la orientación del área de contacto.

Los eventos de movimiento con `ACTION_MOVE` pueden agrupar múltiples muestras de movimiento dentro de un solo objeto. La clase `MotionEvent` proporciona muchos métodos para consultar la posición y otras propiedades de los cursores, por ejemplo `getX(int)`, `getY(int)`, `getAxisValue(int)`, `getPointerId(int)`, `getToolType(int)`, y muchos otros. La mayoría de estos métodos aceptan el índice del cursor como parámetro en lugar del `id` del cursor.

Un gesto comienza con un evento de movimiento `ACTION_DOWN` que proporciona la ubicación del primer cursor hacia abajo. Con cada cursor adicional que sube o baja, el marco generará, en consecuencia, un evento de movimiento con `ACTION_POINTER_DOWN` o `ACTION_POINTER_UP`. Los movimientos del cursor se describen mediante eventos de movimiento con `ACTION_MOVE`. Finalmente, un gesto finaliza cuando el cursor final sube, como lo representa un evento de movimiento con `ACTION_UP` o cuando el gesto se cancela con `ACTION_CANCEL`.

Los siguientes ejemplos muestran el uso de la clase `MotionEvent`, conjuntamente con sus métodos y sus constantes.

Ejemplo 1.

Paso 1.

Crear un nuevo proyecto. Abrir el archivo `MainActivity.java` y modificar su contenido con el código siguiente:

```
import android.content.Context;
import android.os.*;
import android.app.*;
import android.view.*;
import android.graphics.*;
public class MainActivity extends Activity{
    Lienzo l;
    public void onCreate(Bundle b){
        super.onCreate(b);
        l = new Lienzo(this);
        setContentView(l);
    }
    class Lienzo extends View{
        Path pt;
        Paint pn;
        String s;
        float x, y;
        public Lienzo(Context c){
            super(c);
            pt = new Path();
        }
        public void onDraw(Canvas c){
            pn = new Paint();
            pn.setStyle(Paint.Style.STROKE);
            pn.setStrokeWidth(3);
            pn.setColor(Color.BLACK);
            c.drawColor(Color.rgb(250, 250, 100));
            if(s == "00") pt.moveTo(x, y);
            if(s == "xy") pt.lineTo(x, y);
            c.drawPath(pt, pn);
        }
    }
}
```



```
}  
public boolean onTouchEvent(MotionEvent e){  
    x = e.getX();  
    y = e.getY();  
    if(e.getAction() == MotionEvent.ACTION_DOWN) s = "00";  
    if(e.getAction() == MotionEvent.ACTION_MOVE) s = "xy";  
    invalidate();  
    return true;  
}  
}  
}
```

Paso 2.

Abrir el archivo **activity_main.xml** y modificar su contenido con el código siguiente:

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity"  
    android:id="@+id/xl1">  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Hello World!"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintLeft_toLeftOf="parent"  
        app:layout_constraintRight_toRightOf="parent"  
        app:layout_constraintTop_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Paso 3.

Ejecutar el programa. Colocar el dedo en la pantalla del móvil y empezar a pintar sobre la pantalla, como se indica en la figura siguiente Figura 1.

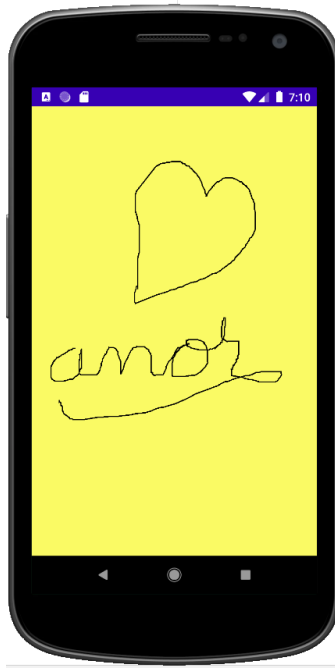


Figura 1. Dibujo táctil sobre la pantalla.

Ejemplo 2.

Paso 1.

Crear un nuevo proyecto. Abrir el archivo `MainActivity.java` y modificar su contenido con el código siguiente:

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;
import android.content.Context;
import android.graphics.*;
import android.os.Bundle;
import android.view.*;

public class MainActivity extends AppCompatActivity implements View.OnTouchListener{
    private int x, y;
    private Lienzo l;
    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
        x = 100;
        y = 100;
        ConstraintLayout cl = findViewById(R.id.xl1);
        l = new Lienzo(this);
        l.setOnTouchListener(this);
        cl.addView(l);
    }
    public boolean onTouch(View v, MotionEvent e) {
        x = (int) e.getX();
        y = (int) e.getY();
        l.invalidate();
        return true;
    }
    class Lienzo extends View {
        public Lienzo(Context c) {
            super(c);
        }
    }
}
```



```
protected void onDraw(Canvas c) {  
    c.drawRGB(255, 255, 0);  
    Paint p = new Paint();  
    p.setARGB(255, 255, 0, 0);  
    p.setStrokeWidth(4);  
    p.setStyle(Paint.Style.STROKE);  
    c.drawCircle(x, y, 20, p);  
}  
}  
}
```

Paso 2.

Abrir el archivo `activity_main.xml` y modificar su contenido con el código siguiente:

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity"  
    android:id="@+id/xl1">  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Hello World!"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintLeft_toLeftOf="parent"  
        app:layout_constraintRight_toRightOf="parent"  
        app:layout_constraintTop_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Paso 3.

Ejecutar el programa. Colocar el dedo en la pantalla del móvil y empezar a mover la circunferencia sobre la pantalla. También, digitar en otro punto de la pantalla para automáticamente posicionar la circunferencia en otra posición, como se indica en la figura siguiente, **Figura 2**.

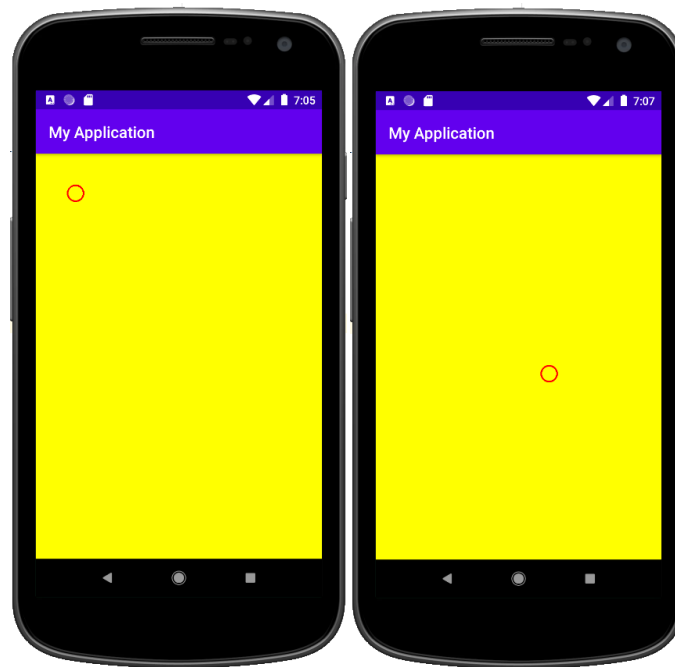


Figura 2. Arrastre táctil y posicionamiento de una figura sobre la pantalla.

EJERCICIO.

Diseñar una aplicación que permita unir, entre tres puntos en diferente posición, una línea continua dibujada con la mano, como se indica en la siguiente figura. Los puntos son fijos y están indicados con circunferencias de color rojo.

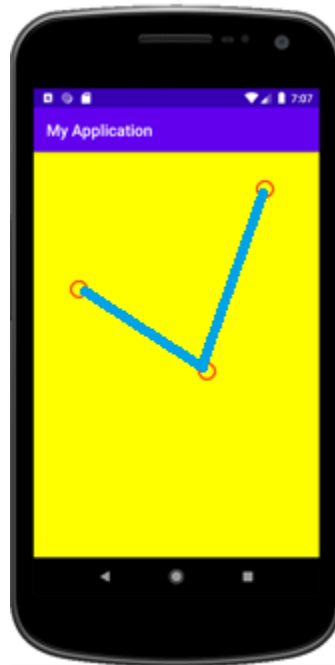


Figura 3. Rectas unidas dibujadas con la mano entre tres puntos.

NOTA. Obtener las imágenes de la ejecución de las aplicaciones y generar un reporte. Guardar el archivo con la sintaxis AlumnoTarea12Grupo.PDF. Enviarlo al sitio indicado por el profesor.