

Práctica 3:

Nuevos Paradigmas de Interacción

Aplicación Android, 3 sensores:

- Cámara
- MultiTouch
- Gestos

Autores:

- Eva María Almansa Aránega
- Luis Alberto Segura Delgado
- Samuel López Liñán

Pantalla de inicio de la App: MegaLauncher



1. Desliza con dos dedos:
 - Hacia arriba abre la cámara de fotos.
 - Hacia abajo abre la cámara de vídeo.
2. Dibuja una letra (en caso de que no este instalada alguna de las aplicaciones y es elegida, redirecciona a la aplicación Play Store de Android):
 - t acceso a Twitter.
 - F acceso a Facebook.
 - W acceso a WhatsApp.



Configuración proyecto:

- Solicitar permisos al dispositivo Android, modificación archivo AndroidManifest.xml:

```
<manifest... >
```

```
...
```

```
<!--- Permisos para la camara, audio, video y SD >-->
```

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

```
<uses-permission android:name="android.permission.RECORD_VIDEO"/>
```

```
<uses-permission
```

```
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<uses-feature android:name="android.hardware.camera" />
```

```
</manifest>
```

Control Cámara/Vídeo:

Class CameraController.java

- Variables para el control de la actividad:

```
//Control de solicitud de actividad
```

```
private static final int CAMERA_CAPTURE_IMAGE_REQUEST_CODE = 100;
```

```
private static final int CAMERA_CAPTURE_VIDEO_REQUEST_CODE = 200;
```

```
public static final int MEDIA_TYPE_IMAGE = 1;
```

```
public static final int MEDIA_TYPE_VIDEO = 2;
```

```
private Context c;
```

- Constructor:

```
public CameraController(Context cont)
```

```
public void captureImage()
```

Gestión Cámara/Vídeo:

Class CameraController.java

- Capturar una imagen:

```
public void captureImage()
```

- Grabar un vídeo:

```
public void recordVideo()
```

- Actividad que se produce una vez finalizado la Captura/Grabación de la cámara:

```
protected void onActivityResult(int requestCode, int  
resultCode, Intent data)
```

- En la ruta preestablecida en los dos primeros métodos anteriores, almacenar en la SD la imagen/vídeo:

```
private void previewCapturedImage()
```

Gestión MultiTouch:

Class MultiTouchView.java

- Capturar movimiento del puntero (dedo):

```
public boolean onTouchEvent(MotionEvent event)
```

- Tres acciones distintas sobre la variable del tipo `MotionEvent` que contiene toda la información del puntero:

1) **Movimiento del puntero:** `MotionEvent.ACTION_MOVE`

2) **Pulsación del puntero:**

`MotionEvent.ACTION_POINTER_DOWN` 0

`MotionEvent.ACTION_POINTER_DOWN`

3) **Puntero deja de estar en contacto con el puntero:**

`MotionEvent.ACTION_POINTER_UP` 0

`MotionEvent.ACTION_POINTER_UP` 0

`MotionEvent.ACTION_CANCEL`

Gestión MultiTouch:

Class MultiTouchView.java

- Dibujar en pantalla los dos dedos detectados:
protected void onDraw(Canvas canvas)
- Si se detecta uno de los dos movimientos posibles, llamada a la clase encargada de la cámara:
 - 1) Detecta movimiento hacia arriba, uso de la cámara de fotos: `cam.captureImage();`
 - 2) Detecta movimiento hacia abajo, uso de la cámara de vídeo: `cam.recordVideo();`

Gestión OneTouch: Class MainActivity.java

- Crear una biblioteca de gestos:
 - 1) Para crear la biblioteca de gestos, se puede usar una aplicación en la página de Android, gestureBuilder y almacenar en la carpeta /raw con el nombre de gestures.
 - 2) Acceso desde la función principal Oncreate:

```
protected void onCreate(Bundle  
savedInstanceState) {  
...  
gesturelib =  
GestureLibraries.fromRawResource(this,  
R.raw.gestures);
```


Gestión OneTouch:

Class MainActivity.java

- Crear una biblioteca de gestos:
 - 1) Crear una vista en layout/activity_main.xml y añadir al menos dos características (definir la capa de gestos con un id y para que se pueda detectar múltiples trazas):

```
<android.gesture.GestureOverlayView
...
    android:id="@+id/gesture_view"
    android:gestureStrokeType="multiple"
...
</android.gesture.GestureOverlayView>
```

- 2) En MainActivity.java para tener acceso al id anterior:

```
GestureOverlayView gestureview =
    (GestureOverlayView) findViewById(R.id.gesture_view);
gestureview.setGestureStrokeType(GestureOverlayView.GESTURE
_STROKE_TYPE_MULTIPLE);
```

Gestión OneTouch:

Class MainActivity.java

- En la función Oncreate, añadir la siguiente línea para asignar a la vista el Listener de gestos para cada acción:

```
gestureview.addOnGesturePerformedListener(gesturelistene  
r);
```

- Crear el Listener de Gestos, encargado de interpretar y asignar acciones:

```
private GestureOverlayView.OnGesturePerformedListener  
gesturelistener = new  
GestureOverlayView.OnGesturePerformedListener() {
```