

Capítulo 4

Diseño de la Interfaz de Usuario con el Robot

Dr. Raúl Marín Prades

Septiembre 2023

Tabla de Contenidos

Resumen

PASO 1: Lanzar Android Studio

PASO 2: Abrir ventana principal

PASO 3: Añadir botón

PASO 4: Pestaña «Text»

PASO 5: Ejecutar aplicación

PASO 6: Atributo «Text»

PASO 7: Propiedades botón

PASO 8: Propiedad «ID»

PASO 9: Crear botones

PASO 10: Ejecutar aplicación

PASO 11: Añadir botones conexión/desconexión

PASO 12: Declaración variables gráficas

PASO 13: Importación librería «widget»

PASO 14: Enlazar código a recursos gráficos

PASO 15: Variable etiqueta (Estado)

PASO 16: Añadir método al botón

PASO 17: Asignar método a evento

PASO 18: Evento en función «onCreate»

Conclusión

Resumen

En este capítulo vamos a continuar con el proyecto de diseñar una aplicación Android simple, que permita interactuar de forma gráfica con un robot Arduino, a través de Bluetooth. Así, diseñaremos una primera versión de la interfaz gráfica, con botones simples, con el objetivo de introducir los conceptos básicos de interacción hombre-máquina (e.g. Botones), así como los métodos en código Java que permitirán dar respuesta a la interacción del usuario.

Mejoras a esta interfaz, utilizando los sensores Pan/Tilt del dispositivo Android, se plantearán mas adelante.

PASO 1: Lanzar Android Studio

Ejecutar la aplicación Android Studio y abrir la aplicación básica desarrollada en el capítulo anterior (i.e. «MyFirstApp»).



Figura 1: Inicialización Android Studio

PASO 2: Abrir ventana principal

Abrir el fichero «activity_main.xml», el cual guarda la descripción gráfica de la ventana principal, seleccionando la pestaña «Design» (ver Figura 2). Por el momento podemos encontrar un objeto de tipo «TextView» con el texto «Hello World».

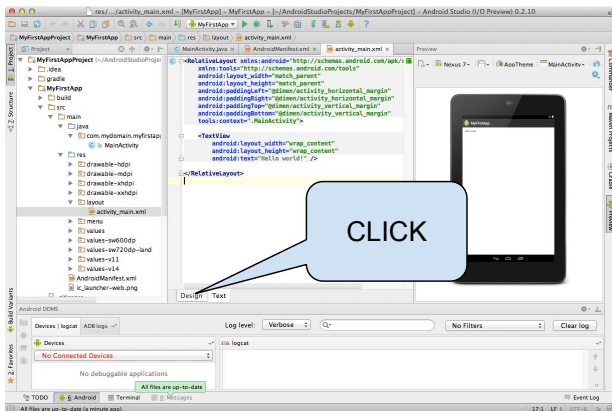


Figura 2: Visualizar el fichero «activity_main.xml»

PASO 3: Añadir botón

Seleccionar el componente «Button» en la paleta de componentes, y arrastrarlo al «layout» gráfico de la ventana. Al hacer click en este botón haremos que el robot se mueva hacia adelante, por lo cual es conveniente situarlo en la parte superior-central de la ventana (ver Figura 3).

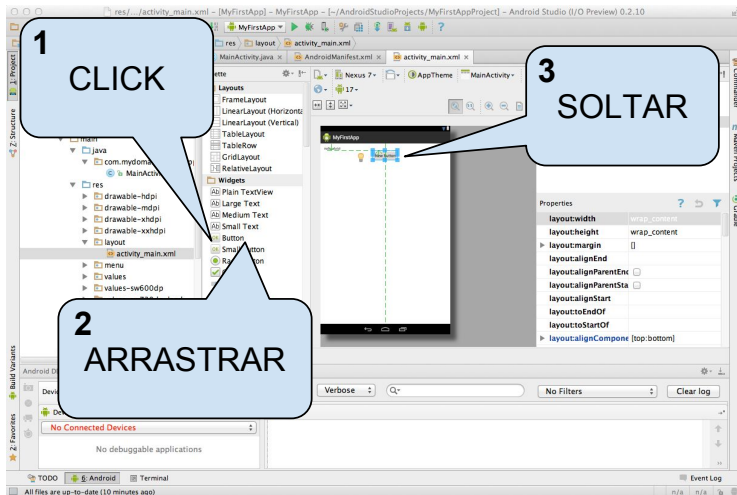


Figura 3: Añadir botón para mover el robot hacia adelante (i.e. «Forward»)

PASO 4: Pestaña «Text»

Hacer click en el tabulador «Text» para ver el fichero de texto que ha sido modificado, expresando la presencia del nuevo botón en la ventana principal (ver Figura 4).

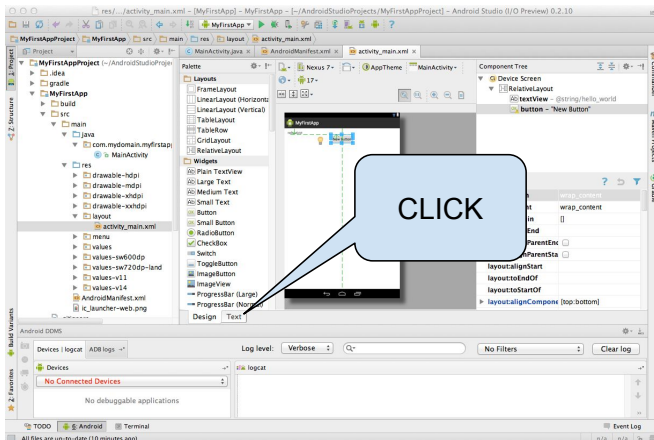


Figura 4: Seleccionar pestaña «Text»

PASO 5: Ejecutar aplicación

Ejecutar la aplicación en el simulador o el dispositivo Android, con el objetivo de ver el primer botón en acción (ver Figura 5).

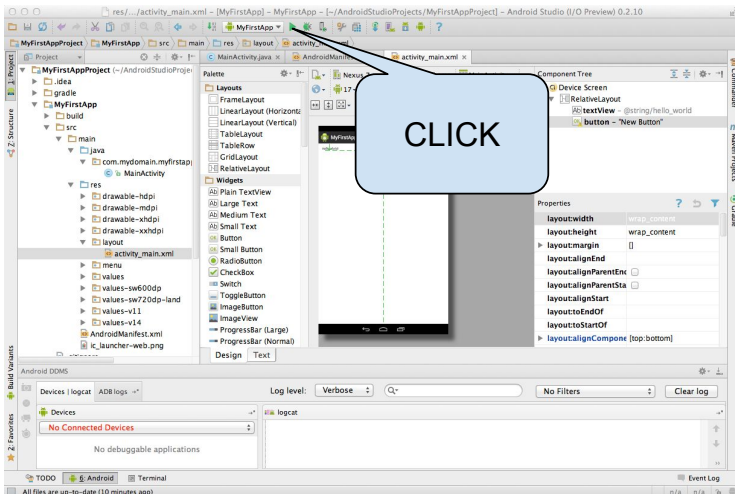


Figura 5: Ejecutar la aplicación

PASO 6: Atributo «Text»

En la ventana «Properties» buscar el atributo «text», hacer click en el espacio para rellenar su contenido, y escribir «Forward», indicando al usuario que al hacer click en este botón el robot se moverá hacia adelante (ver Figura 6).

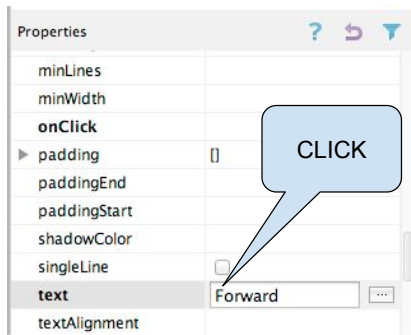


Figura 6: Rellenar propiedad «text» del botón

PASO 7: Propiedades botón

Estudiar las diferentes propiedades que se pueden asignar al botón, como por ejemplo «height» (i.e. altura), «width» (i.e. anchura), «textStyle» (i.e. estilo del texto), y «textSize» (i.e. tamaño del texto), con el objetivo de encontrar un formato adecuado para la aplicación (ver Figura 7).

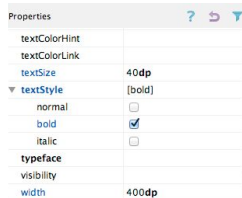


Figura 7: Ajustar las propiedades del botón

- Dependiendo de la versión de Android Studio instalada las propiedades del texto pueden presentar un orden diferente, como por ejemplo el atributo «textAppearance», o el «textSize».

PASO 8: Propiedad «ID»

Una de las propiedades de los componentes gráficos que mas importancia tiene es «ID», la cual da nombre a la variable, con el objetivo de acceder a sus métodos y propiedades desde el código Java. Actualizar el valor de la propiedad «ID» del botón «forButton», tal cual se puede apreciar en la Figura 8.

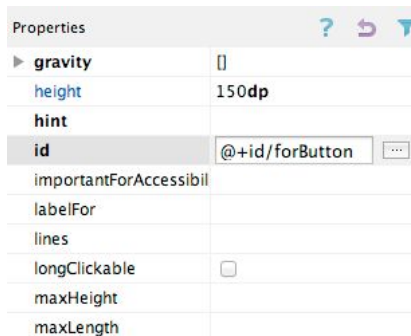


Figura 8: Dar valor a la propiedad «ID»

PASO 9: Crear botones

Create the corresponding “backward”, “left”, and “right” buttons, following the instructions above.

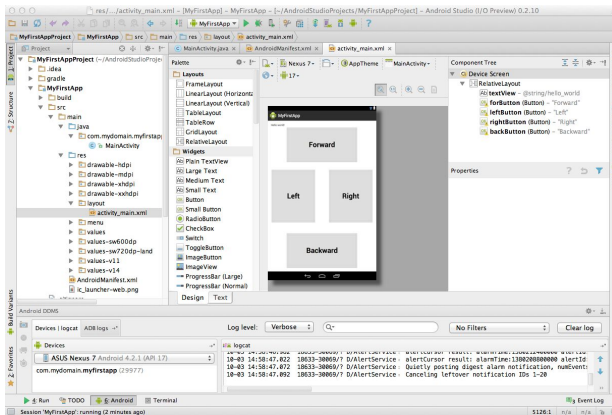


Figura 9: Crear los botones que permitirán mover el robot hacia adelante, atrás, izquierda y derecha

PASO 10: Ejecutar aplicación

Ejecutar la aplicación en el simulador o la tablet, comprobando que los botones funcionan correctamente, aunque todavía no hemos asignado acciones a los mismos.

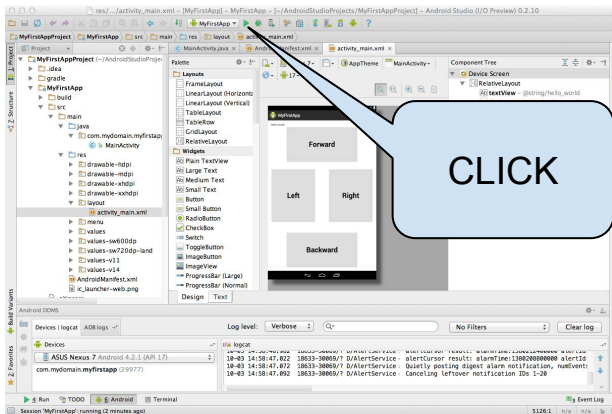


Figura 10: Ejecutar el estado actual de la aplicación

PASO 11: Añadir botones conexión/desconexión

Añadir dos nuevos botones para permitir a la aplicación conectar y desconectar con el robot a través del bluetooth, según se puede ver en la Figura 11.

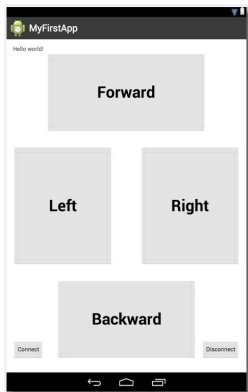


Figura 11: Añadir dos botones para conectar y desconectar el dispositivo Android al robot, a través de Bluetooth (i.e. «Connect» y «Disconnect»)

PASO 12: Declaración variables gráficas

En este paso vamos a empezar el estudio de la gestión de los componentes gráficos (e.g. Botones) desde el código Java. Para ello, dentro del fichero «MainActivity.java» vamos a definir una variable de tipo «Button», para cada uno de los botones gráficos, tal cual se muestra en el siguiente ejemplo.

```
public class MainActivity extends Activity {  
  
    //User Interface Variables  
    Button connectButton;  
    Button disconnectButton;  
    Button forwardButton;  
    Button backwardButton;  
    Button turnLeftForwardButton;  
    Button turnRightForwardButton;
```

De esta manera la lógica del programa se guarda en el fichero Java, mientras que la representación gráfica, que puede variar de un dispositivo Android a otro, se aloja en fichero XML correspondiente. Esto permite desacoplar el programa de la visualización de los datos.

PASO 13: Importación librería «widget»

Añadir la siguiente instrucción al principio del archivo «MainActivity.java», con el objetivo de poder usar los componentes gráficos, como por ejemplo la clase «Button».

```
import android.widget.*;
```

PASO 14: Enlazar código a recursos gráficos

Añadir el siguiente código en el método «onCreate» del fichero «MainActivity.java», con el objetivo de enlazar los componentes gráficos, identificados con la propiedad «ID», con las variables Java que permitirán programar la lógica de la aplicación. La variable «R» enlaza a los recursos actuales del proyecto.

```
connectButton = (Button) findViewById(R.id.connectButton);
disconnectButton = (Button) findViewById(R.id.disconnectButton);
forwardButton = (Button) findViewById(R.id.forButton);
backwardButton= (Button) findViewById(R.id.backButton);
turnLeftForwardButton= (Button) findViewById(R.id.leftButton);
turnRightForwardButton= (Button) findViewById(R.id.rightButton);
```

PASO 15: Variable etiqueta (Estado)

Ahora es el momento de programar una acción (i.e. evento) a cada botón. Para ello vamos a hacer uso de la etiqueta (o variable) de tipo «TextView», donde se representaba en el ejemplo anterior la cadena «Hola Mundo/Hello World». Cuando se haga click en uno de los botones (e.g. «Connect»), el texto a mostrar en la etiqueta será por ejemplo «Botón Conectar seleccionado/Connect button pressed». Para ello, es necesario asociar también una variable de tipo «TextView» al objeto gráfico. Añadir la siguiente línea de código al método «onCreate», previa inicialización en la sección de atributos de la clase «MainActivity.java» (i.e. TextView statusLabel).

```
statusLabel = (TextView) findViewById(R.id.textView);
```

PASO 16: Añadir método al botón

Añadir un nuevo método a la clase «MainActivity» que responda a la selección del botón «Connect», según el siguiente ejemplo. En caso de necesidad, añadir la línea de importación «import android.view.View».

```
public void onClickConnectButton(View view){  
    statusLabel.setText("Connect pressed");  
}
```

NOTA: En versiones recientes de Android Studio es necesario añadir la siguiente línea al fichero activity_main.xml para que se visualice el parámetro "onClick" en la ventana de propiedades:

```
<Button  
    ...  
    android:onClick="onClickConnectButton"  
    ...  
>
```

PASO 17: Asignar método a evento

Asignar el método «onClickConnectButton» a la propiedad «onclick» del botón. Esta opción debería aparecer de forma automática en la ventana de propiedades del botón, según se muestra en la Figura 12).



Figura 12: Seleccionar el método «onClickConnectButton» en la propiedad «onClick»

PASO 18: Evento en función «onCreate»

De forma alternativa, es posible añadir un método de respuesta al evento «onClick» añadiendo el siguiente código al método «MainActivity.onCreate()»:

```
connectButton.setOnClickListener(new View.OnClickListener(){  
    public void onClick(View v){  
        Log.e(TAG, "connectButton Clicked");  
        statusLabel.setText("Connect button pressed");  
    }  
});
```

Conclusión

En este capítulo hemos estudiado el diseño de la interfaz de usuario que nos permitirá controlar de forma remota el robot. Para ello hemos aprendido a añadir componentes gráficos a un fichero XML de «layout» (e.g. `activity_mail.xml`), así como declarar las variables Java para su gestión, añadiendo los eventos correspondientes.

En los siguientes capítulos vamos a estudiar el código Java necesario para interactuar con el robot, primeramente a través de la interfaz Bluetooth, y luego utilizando la interfaz TCP/UDP de Internet.