

20. Música en tu dispositivo

Para reproducir una pieza de música vamos a añadir la siguiente clase `Music` a nuestro proyecto:

`/src/Music.java`

```
package es.uam.eps.android.ccc20;

import android.content.Context;
import android.media.MediaPlayer;

public class Music {
    private static MediaPlayer player;

    public static void play (Context context, int id){
        player = MediaPlayer.create(context, id);
        player.setLooping(true);
        player.start();
    }

    public static void stop (Context context){
        if(player != null){
            player.stop();
            player.release();
            player = null;
        }
    }
}
```

La clase `MediaPlayer` se utiliza para controlar la ejecución de ficheros de audio y vídeo. El método `create()` crea un objeto `MediaPlayer` asociado a un identificador que apunta a un cierto fichero como, por ejemplo, un fichero `mp3`.

Si el argumento del método `setLooping()` es `true`, la reproducción del fichero se repite indefinidamente.

El método `start()` arranca o reanuda la reproducción. Tras parar la reproducción, es importante llamar inmediatamente al método `release()` de tal forma que se liberen los recursos utilizados por el motor del reproductor interno asociado con el objeto `MediaPlayer`.

Si deseamos reproducir música mientras que la actividad se encuentre en primer plano, deberemos situar la llamada `play()` dentro del método `onResume()`, pues este método del ciclo de vida de la actividad se ejecuta cuando ésta empieza a interactuar con el usuario:

```
protected void onResume() {
    super.onResume();
    Music.play(this, R.raw.sampleaudio);
}
```

Recuerda que el método `onPause()` del ciclo de vida de la actividad se ejecuta automáticamente cuando el sistema está a punto de lanzar otra actividad o reanudar una actividad previa.

Por lo tanto, si queremos que la música se detenga cuando la actividad deje el primer plano, el objeto `MediaPlayer` deberá ser parado en el método `onPause()`:

```
protected void onPause() {
    super.onPause();
    Music.stop(this);
}
```

El fichero `MainActivity.java` del proyecto `CCC20` añade los métodos `onResume()` y `onPause()` al correspondiente fichero Java del proyecto de la unidad 18 (`CCC18`):

`/src/MainActivity.java`

```
package es.uam.eps.android.ccc20;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.RadioButton;
import android.widget.Toast;

public class MainActivity extends Activity implements OnClickListener{
    Game game;
    static final int SIZE = 7;

    private final int ids [][] = {
        {0, 0, R.id.f1, R.id.f2, R.id.f3, 0, 0},
        {0, 0, R.id.f4, R.id.f5, R.id.f6, 0, 0},
        {R.id.f7, R.id.f8, R.id.f9, R.id.f10, R.id.f11, R.id.f12, R.id.f13},
        {R.id.f14, R.id.f15, R.id.f16, R.id.f17, R.id.f18, R.id.f19, R.id.f20},
        {R.id.f21, R.id.f22, R.id.f23, R.id.f24, R.id.f25, R.id.f26, R.id.f27},
        {0, 0, R.id.f28, R.id.f29, R.id.f30, 0, 0},
        {0, 0, R.id.f31, R.id.f32, R.id.f33, 0, 0}};

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        registerListeners();
        game = new Game();
        setFigureFromGrid();
    }

    private void registerListeners () {
        RadioButton button;

        for (int i=0; i<SIZE; i++)
            for (int j=0; j<SIZE; j++)
                if (ids[i][j]!=0){
                    button = (RadioButton) findViewById(ids[i][j]);
                    button.setOnClickListener(this);
                }
    }

    public void onClick (View v){
        int id = ((RadioButton) v).getId();

        for (int i=0; i<SIZE; i++)
            for (int j=0; j<SIZE; j++)
                if (ids[i][j] == id) {
                    game.play(i, j);
                    break;
                }

        setFigureFromGrid();
        if (game.isGameFinished())
            Toast.makeText(this, R.string.gameOverTitle, Toast.LENGTH_LONG).show();
    }

    private void setFigureFromGrid () {
        RadioButton button;

        for (int i=0; i<SIZE; i++)
            for (int j=0; j<SIZE; j++)
```

```

        if (ids[i][j] != 0){
            int value = game.getGrid(i, j);
            button = (RadioButton) findViewById(ids[i][j]);

            if (value == 1)
                button.setChecked(true);
            else
                button.setChecked(false);
        }
    }

    protected void onResume() {
        super.onResume();
        Music.play(this, R.raw.funkandblues);
    }

    protected void onPause() {
        super.onPause();
        Music.stop(this);
    }
}

```

El archivo `funkandblues.mp3` debe situarse en una carpeta de nombre `/res/raw/`. Su identificador se pasa como segundo argumento del método `play()` en `onResume()`. Puedes encontrar archivos sin derechos de autor en el siguiente enlace:

<http://incompetech.com/m/c/royalty-free/>.

Resumiendo, el proyecto `ccc20` cuenta ya con tres archivos Java:

- `Game.java`: lógica del juego.
- `MainActivity.java`: gestión de las pulsaciones del tablero.
- `Music.java`: evoltorio de la clase `MediaPlayer` de Android.

El archivo de diseño `activity_main.xml` se puede copiar del proyecto `ccc18`.