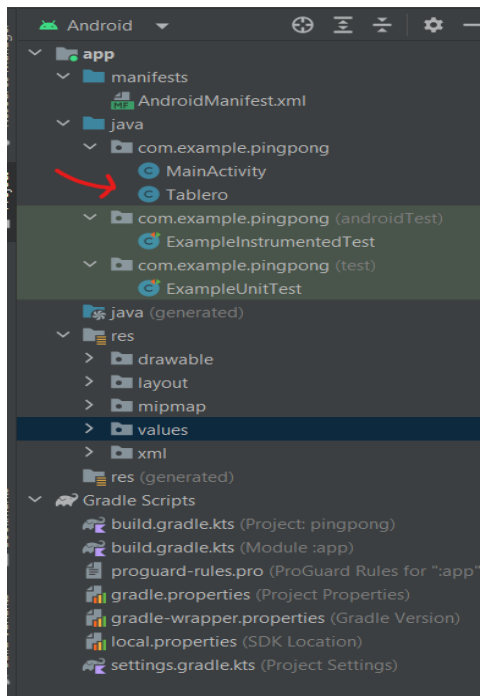


PINBALL ANDROID STUDIO

Javier Sancho Pérez



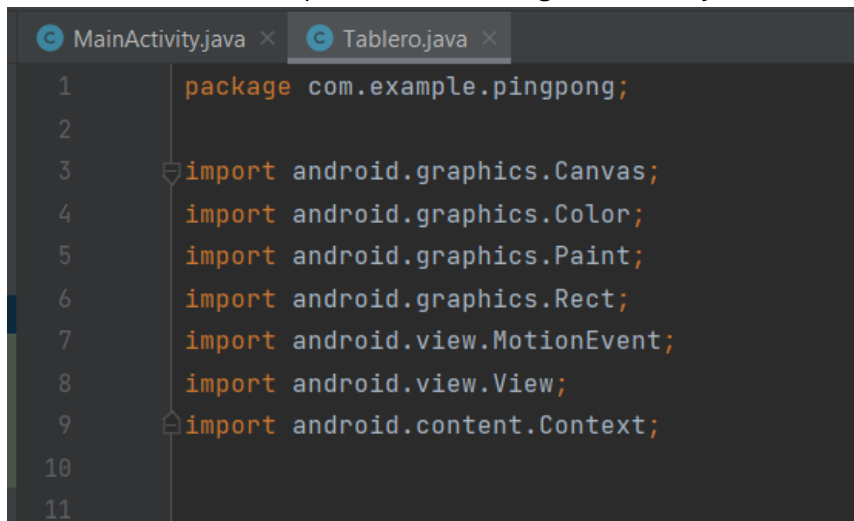
Comenzamos creando una nueva clase llamada Tablero.java la cual va a ser una vista (View) que será llamada por nuestra MainActivity:



Lo siguiente es llamar a la clase Tablero desde el ActivityMain.java. Esta será la única vez que añadamos código a ActivityMain.java


```
MainActivity.java x Tablero.java x
1 package com.example.pingpong;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        //setContentView(R.layout.activity_main);
13        setContentView(new Tablero( context: this));
14    }
15 }
```

En la clase Tablero, lo primero que vamos a necesitar es importar todas las librerías necesarias. También lo podemos hacer según nos lo vaya solicitando el proceso.



```
1 package com.example.pingpong;
2
3 import android.graphics.Canvas;
4 import android.graphics.Color;
5 import android.graphics.Paint;
6 import android.graphics.Rect;
7 import android.view.MotionEvent;
8 import android.view.View;
9 import android.content.Context;
10
11
```

Lo siguiente es declarar extender la clase Tablero de la clase View y crear las variables o atributos de la clase Tablero:



```
12 public class Tablero extends View {
13     Paint pintar = new Paint();
14     int x=100, y=100;
15     int xBarra=0;
16     boolean dirXP=true, dirYP=true;
17     boolean dirXBarra=true;
18
19     int anchoPantalla=0;
20
21     int puntuacion=0;
```

Explicación variables (atributos):

Paint pintar: Instanciamos el objeto pintar de la clase Paint para dibujar en pantalla.

x, y: Será la posición de la pelota en la horizontal y la vertical.

xBarra; Será la posición de la barrita en la horizontal, ya que en la vertical no es necesario

dirXP, dirYP: Dirección de la pelota en los ejes X e Y.

dirXBarra: Guardaremos la dirección de la barra.

anchoPantalla: Guardaremos el ancho de la pantalla.

puntuación: Son los puntos acumulados en el juego los cuales se inician a 0.

Posteriormente creamos el constructor de la clase y le añadimos una imagen como fondo de pantalla de la app:

```
22
23     public Tablero(Context context){
24         super(context);
25         setBackgroundResource(R.drawable.pinball1);
26
27     }
28
```

Ahora creamos el método `onDraw(Canvas canvas)`, que es el método que dibuja los elementos en la pantalla y donde vamos a programar toda la lógica para que la pelota y la barra se mueva.

Primero pintamos la pelota de color amarillo, la situamos en el eje X e Y (valen 100 desde su declaración) tamaño será 70, este tamaño se puede modificar si queremos que la pelota sea mas grande o mas pequeña.

Lo mismo hacemos con la barrita, y finalmente con el letrero de los puntos

```
29     @Override
30     public void onDraw(Canvas canvas){
31         try{
32             anchoPantalla=canvas.getWidth();
33             pintar.setColor(Color.parseColor("colorString: "#D7D00A"));
34             canvas.drawCircle(x,y, radius: 70,pintar);
35
36             pintar.setColor(Color.parseColor("colorString: "#87C8C7"));
37             canvas.drawRect(xBarra, top: canvas.getHeight()-50, right: xBarra+200, canvas.getHeight(), pintar);
38             pintar.setTextSize(90);
39             pintar.setColor(Color.BLACK);
40             canvas.drawText(text: "Puntuacion: "+puntuacion, x: 100, y: 100, pintar);

```

A continuación, creamos las colisiones con los bordes laterales y superior, siendo una pelota de radio 70px.

Si la pelota toca el borde izquierdo (X), cambia de dirección a la derecha.

Si toca el borde derecho (`canvas.getWidth()-70`), cambia a la izquierda.

Si toca el borde superior (Y), baja

```
41         if(x<70){
42             dirXP=true; //Derecha
43         }
44         if(x>canvas.getWidth()-70){
45             dirXP=false; //Izquierda
46         }
47         if(y<70){
48             dirYP=true; //Abajo
49         }

```

Si la pelota toca el suelo, cambia de dirección y se restan 20 puntos.
Esta cantidad se puede modificar.

```
51         if(y>canvas.getHeight()-70){
52             dirYP=false; //Arriba
53             puntuacion-=20;
54         }
55
```

Lo siguiente es el movimiento de la pelota que, en este caso se ha programado para que avance 28px tanto en el eje X como en el Y.

Esto también se puede variar en caso de querer que la pelota recorra más distancia o menos. A mayor cantidad de Px, mayor distancia, y por lo tanto, mayor dificultad.

```
56         if(dirXP==true){
57             x+=28;
58         }else{
59             x-=28;
60         }
61
62         if(dirYP==true){
63             y+=28;
64         }else{
65             y-=28;
66         }
67
```

La colisión con la barra se hace de la siguiente manera:

Primero creamos un rectángulo que representa la pelota de medidas 70x70x70x70 y luego la barra.

```
68
69         Rect pelota =new Rect( left: x-70, top: y-70, right: x+70, bottom: y+70);
70         Rect barra =new Rect(xBarra, top: canvas.getHeight()-50, right: xBarra+200, canvas.getHeight());
```

Ahora detectamos la colisión con la función intersect(otroRect) que devuelve true si los dos rectángulos se superponen.

Si esto ocurre y los 2 rectángulos se “chocan” la dirección de la pelota cambia hacia arriba y la puntuación suma 5 a la cantidad actual.

Esto último también se puede modificar.

```
71         if(pelota.intersect(barra)){
72             dirYP=false;
73             puntuacion+=5;
74         }
75
```

El movimiento de la barra lo haremos de la siguiente manera, si dirXBarra es true, la barra se mueve a la derecha, mientras que si es false se mueve a la izquierda.

La cantidad de movimiento actual es de 20px. Esto se puede modificar para que la barra vaya más rápido o mas despacio.

En mi caso la barra se mueve más lento (20px) que la pelota (28px), por decisión propia.

```
75
76     if(dirXBarra==true){
77         if(xBarra<canvas.getWidth()-200) {
78
79             xBarra += 20;
80         }
81     }else{
82         if(xBarra>0) {
83             xBarra -= 20;
84         }
85     }
86
```

La parte final del método onDraw() es crear el hilo y dormirlo durante 1 milisegundo. Esto también es editable, a mayor cantidad menor velocidad del juego y por lo tanto menor dificultad.

Finalmente invalidate() vuelve a llamar a onDraw() para actualizar el juego.

```
86
87     Thread.sleep( millis: 1);
88     invalidate();
89 }catch(Exception e){
90
91 }
92
93
```

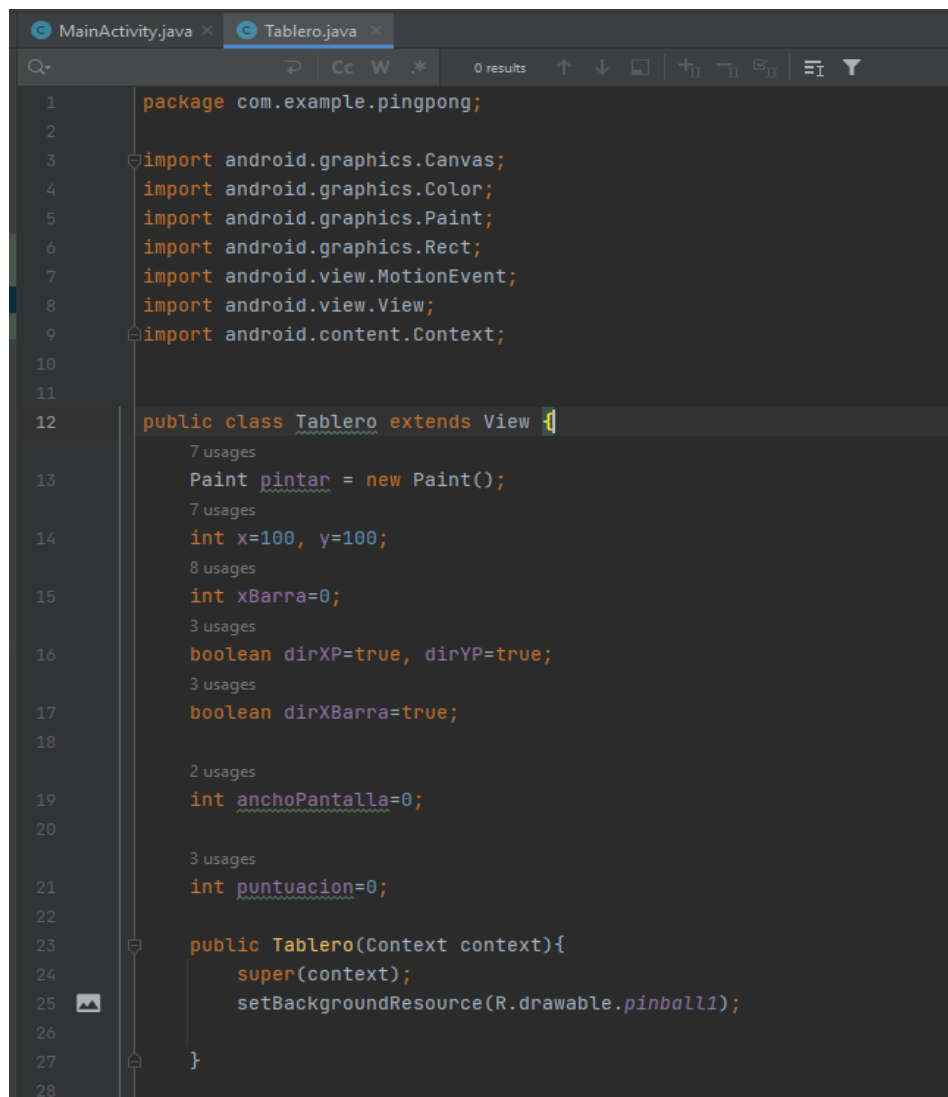
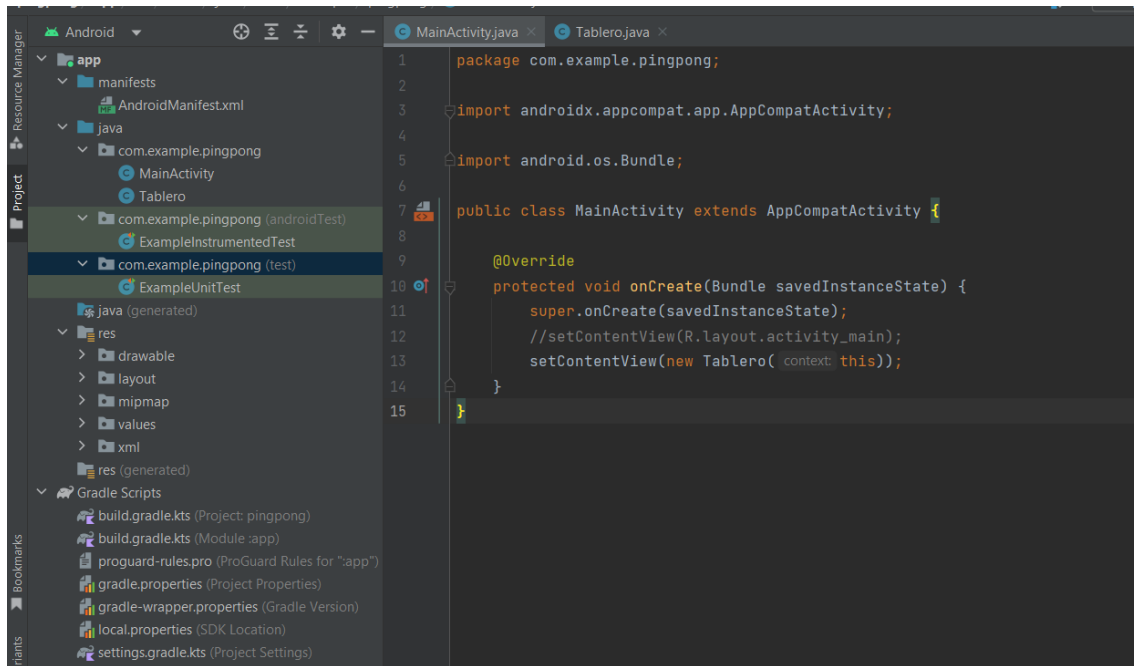
Por último, y fuera del método onDraw() creamos el método onTouchEvent(MotionEvent e) que detecta los toques en la pantalla.

Si el usuario toca la izquierda de la pantalla, la barra se mueve a la izquierda.

Si toca la derecha, la barra se mueve a la derecha.

```
94     @Override
95
96     public boolean onTouchEvent(MotionEvent e){
97         float posX=e.getX();
98         if(posX<anchoPantalla/2){
99             dirXBarra=false;
100         }else{
101             dirXBarra=true;
102         }
103
104         return true;
105     }
106
107 }
```

CÓDIGO COMPLETO



```

28
29
30 @Override
31 public void onDraw(Canvas canvas){
32     try{
33         anchoPantalla=canvas.getWidth();
34         pintar.setColor(Color.parseColor( colorString: "#07000A"));
35         canvas.drawCircle(x,y, radius: 70,pintar);
36
37         pintar.setColor(Color.parseColor( colorString: "#87C8C7"));
38         canvas.drawRect(xBarra, top: canvas.getHeight()-50, right: xBarra+200, canvas.getHeight(), pintar);
39         pintar.setTextSize(90);
40         pintar.setColor(Color.BLACK);
41         canvas.drawText( text: "Puntuacion: "+puntuacion, x: 100, y: 100, pintar);
42         if(x<70){
43             dirXP=true; //Derecha
44         }
45         if(x>canvas.getWidth()-70){
46             dirXP=false; //Izquierda
47         }
48         if(y<70){
49             dirYP=true; //Abajo
50         }
51
52         if(y>canvas.getHeight()-70){
53             dirYP=false; //Arriba
54             puntuacion-=20;
55         }
56
57         if(dirXP==true){
58             x+=28;
59         }else{
60             x-=28;
61         }
62
63         if(dirYP==true){
64             y+=28;
65         }else{
66             y-=28;

```

```

66     }
67
68
69     Rect pelota =new Rect( left: x-70, top: y-70, right: x+70, bottom: y+70);
70     Rect barra =new Rect(xBarra, top: canvas.getHeight()-50, right: xBarra+200, canvas.getHeight());
71     if(pelota.intersect(barra)){
72         dirYP=false;
73         puntuacion+=5;
74     }
75
76     if(dirXBarra==true){
77         if(xBarra<canvas.getWidth()-200) {
78             xBarra += 20;
79         }
80     }else{
81         if(xBarra>0) {
82             xBarra -= 20;
83         }
84     }
85
86     Thread.sleep( millis: 1);
87     invalidate();
88 }catch(Exception e){
89 }
90 }
91
92 @Override
93
94 public boolean onTouchEvent(MotionEvent e){
95     float posX=e.getX();
96     if(posX<anchoPantalla/2){
97         dirXBarra=false;
98     }else{
99         dirXBarra=true;
100     }
101     return true;
102 }
103 }
104

```