Desarrollo de Juegos Android

Programación multimedia y dispositivos móviles

Desarrollo de Aplicaciones Multiplataforma

Actividad

Desarrollo de Juegos Android.

**Objetivos**

* Aplicar motores de juegos.
* Aprender la estructura de un juego para móviles.
* Implementar gráficos y eventos gestuales.

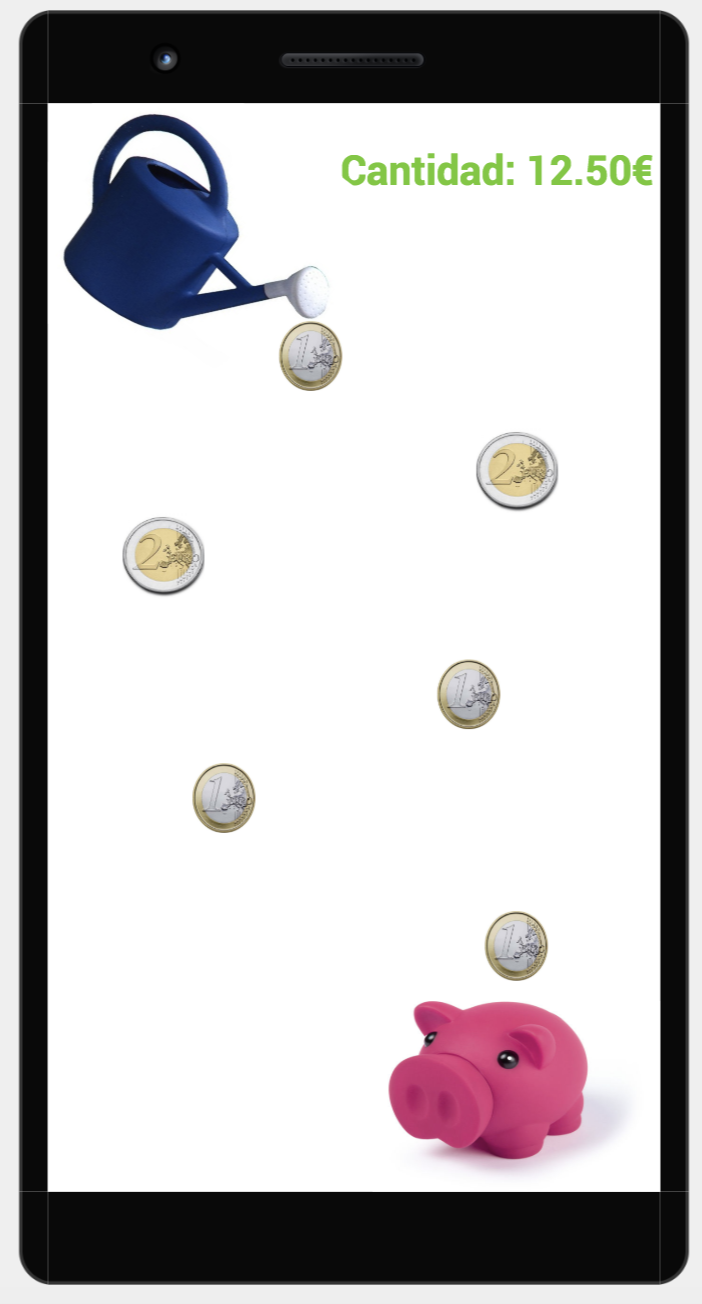
|  |
| --- |
| **¿Cómo lo hago?** |
| 1. Rellena los datos que se piden en la tabla “Antes de empezar”. 2. Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc. 3. Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar. 4. Recuerda entregar la actividad en formato PDF a no ser que el profesor o profesora indique lo contrario. 5. Recuerda nombrar el archivo siguiendo estas indicaciones:  * Ciclo\_Módulo o crédito\_Tema\_ACT\_número actividad\_Nombre y apellido   + Ejemplo: AF\_M01\_T01\_ACT\_01\_Maria Garcia |

|  |  |
| --- | --- |
| **Antes de empezar…** | |
| Nombre | Jacobo |
| Apellidos | Montero Gil |
| Módulo/Crédito | M07 |
| UF (solo ciclos LOE) |  |
| Título de la actividad | Juego Android |

Actividad 4

***Juego Android***

Programar en Android un juego consistente en recoger monedas y que van cayendo desde la parte superior de la pantalla.



**Lógica del juego:**

* Las monedas saldrán de un elemento ubicado en la parte superior derecha de la pantalla (en la imagen como ejemplo está la regadera).
* Elegir un fondo de pantalla personalizado.
* En función del tipo de moneda que se vaya recogiendo, la cantidad monetaria irá cambiando. Las monedas que serán de 10, 20, 50 céntimos y de 1, 2 euros.
* Para recoger las monedas, la pantalla tendrá un elemento en la parte inferior, que el usuario podrá ir moviendo de izquierda a derecha (en la imagen como ejemplo está el cerdito).
* Permitir elegir entre diferentes niveles de dificultad (mayor/menor velocidad de caída de monedas).
* Incorporar elementos que no se deben recoger y que por lo tanto restaran la cantidad de dinero recogido, como por ejemplo frutas, animales …

Contenidos

[Solución 7](#_Toc61174481)

[1. El juego. 7](#_Toc61174482)

[ Pantalla principal: 7](#_Toc61174483)

[ Pantalla de juego: 9](#_Toc61174484)

[ Desarrollo del juego 9](#_Toc61174485)

[2. El código 10](#_Toc61174486)

[ AndroidManifest.xml 10](#_Toc61174487)

[ MainActivity.java 11](#_Toc61174488)

[ ActivityPantalla.java 12](#_Toc61174489)

[ Juego.java 15](#_Toc61174490)

[ activity\_main.xml 18](#_Toc61174491)

[ activity\_pantalla.xml 20](#_Toc61174492)

[ menuprincipal.xml 21](#_Toc61174493)

[3. Capturas de pantalla 21](#_Toc61174494)

[ Pantalla principal 22](#_Toc61174495)

[ Pantalla de juego 22](#_Toc61174496)

[ Pantalla final 23](#_Toc61174497)

# Solución

# El juego.

Se han implementado dos pantallas.

## Pantalla principal:

1-Se ha implementado un menú en la parte superior. Con los siguientes elementos:

1\_1-Botón para salir

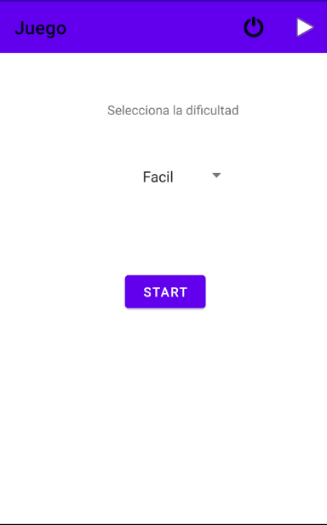
1\_2-Botón para iniciar

2-Se ha colocado un spinner para que el usuario seleccione la dificultad

1\_2

1\_1

3-Botón para iniciar



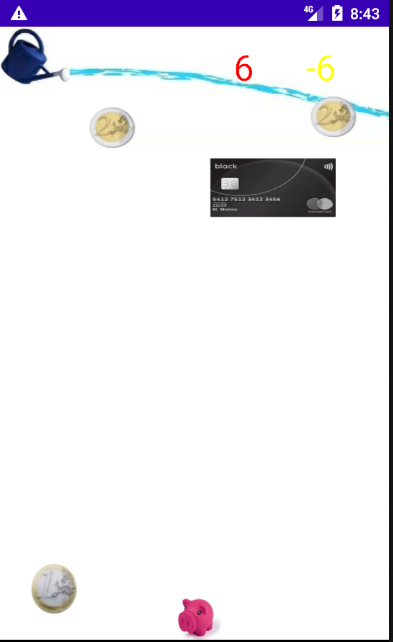
3

2

1

## Pantalla de juego:

6

1-Monedas con valor de 1

5

2-Monedas con valor de 2

3-Tarjeta black que resta 10

2

4-Hucha

5-Contador de tiempo

3

6-Contador de puntuación

1

4

## Desarrollo del juego

El tiempo fijado se ha establecido en 10”. La velocidad de caída es variable en función de la dificultad seleccionada. Al llegar al límite de tiempo el juego se detiene y se muestra un toast con la puntuación.

# El código

## AndroidManifest.xml

<?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">

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="228dp"

android:onClick="onClickIniciar"

android:text="@string/btStart"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent" />

<Spinner

android:id="@+id/spinner"

android:layout\_width="131dp"

android:layout\_height="20dp"

android:layout\_marginBottom="368dp"

android:entries="@array/dificultad"

android:textAlignment="center"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent" />

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="440dp"

android:text="@string/tvSelDificultad"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.539"

app:layout\_constraintStart\_toStartOf="parent" />

<androidx.appcompat.widget.Toolbar

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/barra"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="?attr/colorPrimary"

android:minHeight="?attr/actionBarSize"

android:theme="?attr/actionBarTheme"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

## MainActivity.java

public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 androidx.appcompat.widget.Toolbar barra = (androidx.appcompat.widget.Toolbar)  
 findViewById(R.id.*barra*);  
 setSupportActionBar(barra);  
  
 }  
  
 *//Acción a ejecutar desde el botón* public void onClickIniciar (View view) {  
 Intent intent = new Intent(this, ActivityPantalla.class);  
 Spinner sp=(Spinner) findViewById(R.id.*spinner*);  
 intent.putExtra("dificultad",sp.getSelectedItem().toString());  
 startActivity(intent);  
 }  
 *//Metodo para el botón del menú* public void onClickIniciar\_M () {  
 Intent intent = new Intent(this, ActivityPantalla.class);  
 Spinner sp=(Spinner) findViewById(R.id.*spinner*);  
 intent.putExtra("dificultad",sp.getSelectedItem().toString());  
 startActivity(intent);  
 }  
  
 *//Generación del menú y elementos* @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*menuprincipal*,menu);  
 return true;  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 if (item.getItemId()==R.id.*idSalir*){  
 finish();  
 }  
 else if(item.getItemId()==R.id.*idIniciar*){  
 onClickIniciar\_M();  
 }  
 return true;  
 }  
  
  
}

## ActivityPantalla.java

public class ActivityPantalla extends AppCompatActivity {  
 public Juego juego;  
 private Handler handler = new Handler();  
 private Random random = new Random();  
 String dificultad;  
 Timer timer;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 *//Inicio la vista de juego* juego = new Juego(this);  
 setContentView(juego);  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_pantalla*);  
 *//defino la vista juego sobre el visor* juego = (Juego) findViewById(R.id.*Pantalla*);  
 ViewTreeObserver obs = juego.getViewTreeObserver();  
  
 *//Obtengo los parametros de dificultad* Bundle parametros = this.getIntent().getExtras();  
 dificultad=parametros.getString("dificultad");  
  
 *//Declaro e inicio las variables para contar el tiempo discurrido* final long[] fin = new long[1];  
 long inicio = System.*currentTimeMillis*();  
 Intent intent = new Intent(this,MainActivity.class);  
 obs.addOnGlobalLayoutListener(new ViewTreeObserver.OnGlobalLayoutListener() {  
 @Override  
 public void onGlobalLayout() {  
 *// Establezco las posiciones y valores iniciales* juego.ancho = juego.getWidth();  
 juego.alto = juego.getHeight();  
 juego.posX=juego.ancho/2;  
 juego.posY=juego.alto-75;  
  
 juego.radio=75;  
*//* juego.posMonedaY1=250;  
 juego.posMonedaX1=190;  
  
 juego.posMonedaY2=250;  
 juego.posMonedaX2= 340;  
  
 juego.posMonedaY3=250;  
 juego.posMonedaX3= 490;  
  
 juego.posMonedaY4=250;  
 juego.posMonedaX4= 640;  
 });  
 *//Ejecutamos cada 20 milisegundos* timer = new Timer();  
 timer.schedule(new TimerTask() {  
 @Override  
 public void run() {  
 handler.post(new Runnable() {  
 public void run() {  
 *//Cada x segundos movemos la moneda Xdp en función de la dificultad* switch(dificultad){  
 case "Facil":  
 juego.posMonedaY1+=10;  
 juego.posMonedaY2+=10;  
 juego.posMonedaY3+=10;  
 juego.posMonedaY4+=10;  
 break;  
 case "Dificil":  
 juego.posMonedaY1+=20;  
 juego.posMonedaY2+=20;  
 juego.posMonedaY3+=20;  
 juego.posMonedaY4+=20;  
 break;  
  
 case "Muy dificil":  
 juego.posMonedaY1+=30;  
 juego.posMonedaY2+=30;  
 juego.posMonedaY3+=30;  
 juego.posMonedaY4+=30;  
 break;  
  
 }  
 *//Calculo el tiempo transcurrido* fin[0] = System.*currentTimeMillis*();  
  
 juego.tiempo=(int)((fin[0] -inicio)/1000);  
  
 *//Una vez se cumplen 10 segundo se para el hilo, aparece un toast con un mensaje y se resetean las posiciones de los elementos  
 //Se para también la reproducción del sonido* if(fin[0] -inicio>=10000){  
 cancel();  
 *//Mensaje* Toast toast1 =  
 Toast.*makeText*(getApplicationContext(),  
 "El juego ha terminado. Tu puntuación es :"+juego.puntuacion, Toast.*LENGTH\_SHORT*);  
  
 toast1.show();  
  
 *//Se detiene la reproducción del sonido* juego.gameloop.stop();  
  
 *//Se recolocan los elementos al finalizar* juego.posMonedaY1+=juego.alto;  
 juego.posMonedaY2+=juego.alto;  
 juego.posMonedaY3+=juego.alto;  
 juego.posMonedaY4+=juego.alto;  
  
 }  
 juego.invalidate();  
 }  
 });  
 }  
 }, 0, 20);  
  
 }  
  
  
}

## Juego.java

*// Esta es la clase que utilizaremos para dibujar*public class Juego extends View {  
 *//Declaraciones  
  
 //Tamaño pantalla* public int ancho,alto;  
  
 *//Posicion de los objetos* public int posX,posY,radio,posMonedaX1,posMonedaY1;  
 public int posMonedaX2,posMonedaY2;  
 public int posMonedaX3,posMonedaY3;  
 public int posMonedaX4,posMonedaY4;  
  
 *//Rectangulos en los que incrustamos las imagenes y detectamos los impactos* private RectF rectCesta;  
 private RectF rectMoneda1;  
 private RectF rectMoneda2\_1;  
 private RectF rectMoneda2\_2;  
 private RectF rectTarjeta;  
  
 *//Puntuación y tiempo* Integer puntuacion=0;  
 Integer tiempo;  
  
 *//Generación de numeros aleatorios* private Random random = new Random();  
  
 *//Bitmaps* Bitmap bmpMoneda2;  
 Bitmap bmpMoneda1;  
 Bitmap bmpTarjeta;  
  
  
 MediaPlayer gameloop = new MediaPlayer();  
  
  
 public Juego(Context context) {  
 super(context);  
 }  
  
 public Juego(Context context, AttributeSet attrs) {  
 super(context, attrs);  
 gameloop = MediaPlayer.*create*(context,R.raw.*mario*);  
 gameloop.start();  
 *//mantiene el loop del soundtrack* gameloop.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {  
 @Override  
 public void onCompletion(MediaPlayer mp) {  
 gameloop.start();  
 }  
 });  
 }  
  
 *//Sección que capta los eventos del usuario* @Override  
 public boolean onTouchEvent(MotionEvent event) {  
 *// you may need the x/y location* switch (event.getAction()) {  
 case MotionEvent.*ACTION\_DOWN*:  
 break;  
 case MotionEvent.*ACTION\_UP*:  
 break;  
 case MotionEvent.*ACTION\_MOVE*:  
  
 *//Elimino el movimiento vertical  
 //posY=(int)event.getY();* posX=(int)event.getX();  
  
 *//Encuadramos dentro de los limites el movimiento del cerdito* if(posX<50){  
 posX=50;  
 }else if(posX>1030){  
 posX=1030;  
 }  
  
  
 *// invalidate llama al onDraw y vuelve a pintar la bola* this.invalidate();  
 }  
 return true;  
 }  
 public Juego(Context context, AttributeSet attrs, int defStyle) {  
 super(context, attrs, defStyle);  
 }  
 @Override  
 protected void onDraw(Canvas canvas) {  
 super.onDraw(canvas);  
  
 *//Definimos los objetos a pintar* Paint fondo = new Paint();  
 Paint puntos = new Paint();  
 Paint tiempopaint = new Paint();  
  
 *//Bitmaps* Bitmap bmpRegadera = BitmapFactory.*decodeResource*(getResources(),  
 R.drawable.*regadera*);  
 Bitmap bmpHucha = BitmapFactory.*decodeResource*(getResources(),  
 R.drawable.*pig*);  
  
 *//definimos colores , estilos y tamaño letra* fondo.setColor(Color.*WHITE*);  
 fondo.setStyle(Paint.Style.*FILL\_AND\_STROKE*);  
 puntos.setTextSize(100);  
 puntos.setColor(Color.*YELLOW*);  
 tiempopaint.setTextSize(100);  
 tiempopaint.setColor(Color.*RED*);  
 *//Defino el rectangulo de la hucha* rectCesta= new RectF((posX-50),(posY-50),(posX+75),(posY+75));  
 *//Pinto rectángulo fondo* canvas.drawRect(new Rect(0,0,(ancho),(alto)),fondo);  
  
 *//Pinto la regadera* canvas.drawBitmap(bmpRegadera,0,0,null);  
 *// Pinto el rectangulo de la hucha* canvas.drawBitmap(bmpHucha,null,rectCesta,null);  
  
  
 *//Defino la posicion de las monedas de forma aleatoria a cada ejecución , dentro de  
 // los limites de la pantalla* if (posMonedaY1>alto) {  
 int ram=random.nextInt(ancho-radio\*2);  
 posMonedaY1=250;  
 posMonedaX1=ram;  
 *//Verifico que queda todo el objeto dentro de la pantalla* if(ram<radio\*2){  
  
 posMonedaX1=radio\*2;  
 }  
 }  
 if (posMonedaY2>alto) {  
 int ram=random.nextInt(ancho-radio\*2);  
 posMonedaY2=250;  
 posMonedaX2=ram;  
 if(ram<radio\*2){  
 posMonedaX2=radio\*2;  
 }  
 }  
 if (posMonedaY3>alto) {  
 int ram=random.nextInt(ancho-radio\*2);  
 posMonedaY3=250;  
 posMonedaX3=ram;  
 if(ram<radio\*2){  
  
 posMonedaX3=radio\*2;  
 }  
 }  
 if (posMonedaY4>alto) {  
 int ram=random.nextInt(ancho-350);  
 posMonedaY4=250;  
 posMonedaX4=ram;  
 if(ram<350){  
 posMonedaX4=350;  
 }  
  
 }  
 *//Inicio los bitmaps con la imagen correspondiente* bmpMoneda1= BitmapFactory.*decodeResource*(getResources(),  
 R.drawable.*moneda1png*);  
 rectMoneda1= new RectF((posMonedaX1-radio), (posMonedaY1-radio), (posMonedaX1+radio), (posMonedaY1+radio));  
 canvas.drawBitmap(bmpMoneda1,null,rectMoneda1,null);  
  
 bmpMoneda2= BitmapFactory.*decodeResource*(getResources(),  
 R.drawable.*moneda2png*);  
  
 rectMoneda2\_1= new RectF((posMonedaX2-radio),(posMonedaY2-radio),(posMonedaX2+radio),(posMonedaY2+radio));  
 canvas.drawBitmap(bmpMoneda2,null,rectMoneda2\_1,null);  
  
 rectMoneda2\_2= new RectF((posMonedaX3-radio),(posMonedaY3-radio),(posMonedaX3+radio),(posMonedaY3+radio));  
 canvas.drawBitmap(bmpMoneda2,null,rectMoneda2\_2,null);  
  
 bmpTarjeta= BitmapFactory.*decodeResource*(getResources(),  
 R.drawable.*black*);  
 rectTarjeta= new RectF((posMonedaX4),(posMonedaY4+5),(posMonedaX4+350),(posMonedaY4+170));  
 canvas.drawBitmap(bmpTarjeta,null,rectTarjeta,null);  
  
 *// Calculo intersección y en función del valor sumo o resto valor* if (RectF.*intersects*(rectCesta,rectMoneda1)) {  
 puntuacion += 1;  
 *//fuerzo la desaparición de la moneda después de la interseccion* posMonedaY1=alto+10;  
  
 }  
 if (RectF.*intersects*(rectCesta,rectMoneda2\_1)) {  
 int ram=random.nextInt(ancho);  
  
 puntuacion += 2;  
 posMonedaY2=alto+10;  
;  
  
 }  
 if (RectF.*intersects*(rectCesta,rectMoneda2\_2)){  
 int ram=random.nextInt(ancho);  
  
 puntuacion += 2;  
 posMonedaY3=alto+10;  
  
  
 }  
 if (RectF.*intersects*(rectCesta,rectTarjeta)) {  
 int ram=random.nextInt(ancho);  
  
 puntuacion -= 10;  
 posMonedaY4=alto+10;  
  
  
 }  
 *//Pinto puntuación y tiempo* canvas.drawText(puntuacion.toString(), 850,150,puntos);  
 canvas.drawText(tiempo.toString(), 650,150,tiempopaint);  
 }  
}

## activity\_main.xml

<?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">

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="228dp"

android:onClick="onClickIniciar"

android:text="@string/btStart"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent" />

<Spinner

android:id="@+id/spinner"

android:layout\_width="131dp"

android:layout\_height="20dp"

android:layout\_marginBottom="368dp"

android:entries="@array/dificultad"

android:textAlignment="center"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent" />

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="440dp"

android:text="@string/tvSelDificultad"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.539"

app:layout\_constraintStart\_toStartOf="parent" />

<androidx.appcompat.widget.Toolbar

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/barra"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="?attr/colorPrimary"

android:minHeight="?attr/actionBarSize"

android:theme="?attr/actionBarTheme"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

## activity\_pantalla.xml

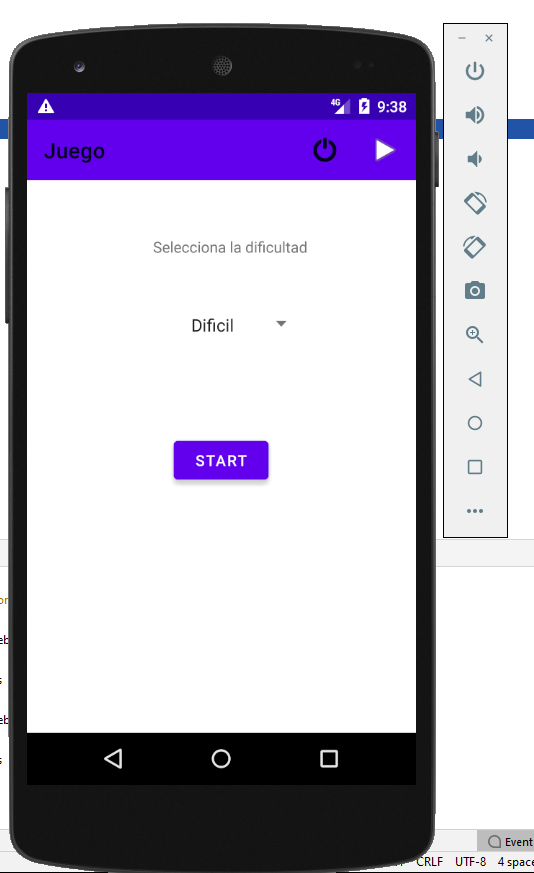
*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ActivityPantalla">  
 <view  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 class="com.example.juego.Juego"  
 android:id="@+id/Pantalla"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 />  
</androidx.constraintlayout.widget.ConstraintLayout>

## menuprincipal.xml

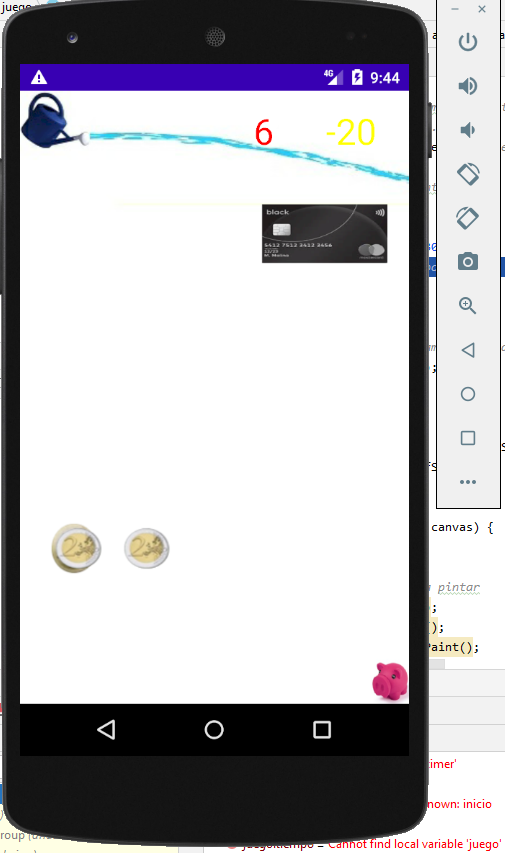
*<?*xml version="1.0" encoding="utf-8"*?>*<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
 >  
  
  
 <item android:id="@+id/idSalir"  
 android:icon="@android:drawable/ic\_lock\_power\_off"  
 android:title="Salir"  
 app:showAsAction="ifRoom"/>  
 <item android:id="@+id/idIniciar"  
 android:icon="@android:drawable/ic\_media\_play"  
 android:title="Salir"  
 app:showAsAction="ifRoom"/>  
  
</menu>

# Capturas de pantalla

## Pantalla principal



## Pantalla de juego



## Pantalla final

