FACULTAD DE INGENIERIA Y ARQUITECTURA

CARRERA:

INGENIERIA EN DESARROLLO DE SOFTWARE

ASIGNATURA:

DESARROLLO DE APLICACIONES MÓVILES AVANZADAS SECCIÓN B

DOCENTE:

ING. LUIS HUMBERTO RIVAS RODRÍGUEZ

ALUMNO:

MAURICIO MOISES CARRANZA VIERA

TEMA:

PROGRAMACIÓN ORIENTADA A OBJETOS EN KOTLIN

LINK DEL REPOSITORIO: https://github.com/u20210127/DESARROLLO-DE-APLICACIONES-MOVILES-AVANZADAS-SECCION-B.git

FECHA ENTREGA:

02/02/2025

1. Explique con sus propias palabras para que sirve el siguiente fragmento de código en la aplicación explicada en la práctica.

Se ejecuta cuando se crea la actividad (onCreate).

- super.onCreate(savedInstanceState): Llama al método onCreate de la clase base (Activity o AppCompatActivity) para asegurarse de que la actividad se inicialice correctamente. Esto es necesario para que Android maneje adecuadamente el ciclo de vida de la actividad.

- setContentView(R.layout.activity_juego): Establece el diseño de la interfaz de usuario para esta actividad. En este caso, utiliza el archivo de diseño activity_juego.xml, que define cómo se verá la pantalla.

- val cuadrilla = findViewById<GridLayout>(R.id.gridLayout): Obtiene una referencia a un GridLayout (un diseño que organiza sus elementos en una cuadrícula) que está definido en el archivo de diseño activity_juego.xml. Este GridLayout se almacena en la variable cuadrilla.

- val textoNivel = findViewById<TextView>(R.id.levelText): Obtiene una referencia a un TextView (un componente que muestra texto) que también está definido en el archivo de diseño. Este TextView se almacena en la variable textoNivel y probablemente se usa para mostrar el nivel actual del juego.

- val circulos = List(size = 12) { crearCirculo() }: Crea una lista de 12 elementos, donde cada elemento es un círculo generado por la función crearCirculo(). Esta función probablemente devuelve un View personalizado o un ImageView que representa un círculo en la interfaz.

- circulos.forEach { cuadrilla.addView(it) }: Recorre la lista de círculos y añade cada uno de ellos al GridLayout (cuadrilla). Esto significa que los círculos se mostrarán en la pantalla dentro de la cuadrícula.

- iniciarNivel(): Llama a una función llamada iniciarNivel(), que probablemente se encarga de configurar y comenzar un nuevo nivel del juego. Esto podría incluir inicializar variables, establecer el texto del nivel en textoNivel, o cualquier otra lógica necesaria para empezar el juego.

Resumen: Este código inicializa la interfaz de usuario de un juego, crea una cuadrícula con 12 círculos y comienza un nuevo nivel. La actividad utiliza un diseño definido en activity_juego.xml, que incluye un GridLayout para organizar los círculos y un TextView para mostrar el nivel actual. La función iniciarNivel() se encarga de preparar el juego para que el usuario pueda empezar a jugar.

2. Explique brevemente que proceso realiza el siguiente código.

Es una función llamada mostrarProximoCirculoVerde(). Esta función se encarga de mostrar círculos verdes de manera secuencial en la interfaz del juego, con un retraso entre cada aparición. Además, maneja la transición entre niveles y la finalización del juego cuando se completan todos los niveles.

- Condición inicial (if (circulosVerdes < circulosPorNivel)): Verifica si el número de círculos verdes mostrados (circulosVerdes) es menor que la cantidad total de círculos que deben mostrarse en el nivel actual (circulosPorNivel).

- Uso de manejador.postDelayed: El manejador es un objeto que permite ejecutar código después de un retraso específico. En este caso, se programa la ejecución de un bloque de código después de un retraso de 1000 milisegundos (1 segundo).

- Bloque de código dentro de postDelayed:

- cambiarCirculoAleatorio(): Llama a una función que cambia el color de un círculo aleatorio a verde (o realiza alguna acción para resaltar un círculo).

- circulosVerdes++: Incrementa el contador de círculos verdes mostrados en el nivel actual.

- mostrarProximoCirculoVerde(): Llama recursivamente a la misma función para mostrar el siguiente círculo verde después de otro retraso de 1 segundo. Esto crea un efecto en el que los círculos verdes aparecen uno tras otro, con un intervalo de 1 segundo entre cada uno.

- Condición else:
- Si ya se han mostrado todos los círculos verdes del nivel actual (circulosVerdes >= circulosPorNivel), se ejecuta este bloque.

- Aquí se verifica si el nivel actual (nivelActual) es igual a 2:
- Si es el nivel 2 (if (nivelActual == 2)):
- Llama a la función finalizarJuego(completado = true), lo que indica que el juego ha sido completado exitosamente.

- Si no es el nivel 2 (else):
- Incrementa el nivel actual (nivelActual++).
- Llama a la función iniciarNivel() para comenzar el siguiente nivel.

Resumen:

La función mostrarProximoCirculoVerde() se encarga de Mostrar círculos verdes uno por uno, con un retraso de 1 segundo entre cada uno. Llevar un conteo de cuántos círculos verdes se han mostrado en el nivel actual. Avanzar al siguiente nivel una vez que se han mostrado todos los círculos verdes del nivel actual. Finalizar el juego cuando se completa el nivel 2. Esta función es clave para la progresión del juego, ya que controla la aparición de los círculos verdes y la transición entre niveles.

3. Explique con sus propias palabras para que nos sirve el código que se está utilizando en el archivo ResultadosActivity.kt.

Se encarga de mostrar los resultados del juego al usuario después de que este ha terminado.

Esta actividad muestra:

1. Si el usuario ha completado el juego o ha perdido.
2. El puntaje obtenido.
3. Un botón para reiniciar el juego y volver a la pantalla principal.

Explicación detallada:

1. **Herencia y configuración inicial:**
    - La clase ResultadosActivity hereda de AppCompatActivity, lo que significa que es una actividad compatible con versiones anteriores de Android.
    - En el método onCreate, se llama a super.onCreate(savedInstanceState) para inicializar la actividad correctamente.
    - Se establece el diseño de la interfaz de usuario con setContentView(R.layout.activity_resultados), que utiliza el archivo de diseño activity_resultados.xml.
2. **Obtención de datos del intent:**
    - Se obtienen los datos enviados desde la actividad anterior (probablemente la actividad del juego) a través del Intent:
        - **puntaje**: El puntaje obtenido por el usuario (valor entero).
        - **completado**: Un valor booleano que indica si el usuario completó el juego o no.
        - **nivel**: El nivel en el que el usuario terminó el juego (valor entero).
3. **Referencias a vistas:**
    - Se obtienen referencias a dos TextView del diseño:
        - **textoResultado**: Para mostrar un mensaje que indica si el usuario ganó o perdió.
        - **textoPuntaje**: Para mostrar el puntaje obtenido.
4. **Lógica para mostrar el resultado:**
    - Dependiendo del valor de completado y nivel, se muestra un mensaje diferente en textoResultado:
        - Si el juego fue completado (completado = true):
            - Si el nivel es mayor que 1, se muestra un mensaje de felicitación por completar todos los niveles.

        - Si el nivel es 1, se muestra un mensaje indicando que el nivel fue superado.
    - Si el juego no fue completado (completado = false), se muestra un mensaje de "¡Perdiste!".

5. **Mostrar el puntaje:**
    - El puntaje obtenido se muestra en textoPuntaje con el formato: "Puntaje: X", donde X es el valor de puntaje.

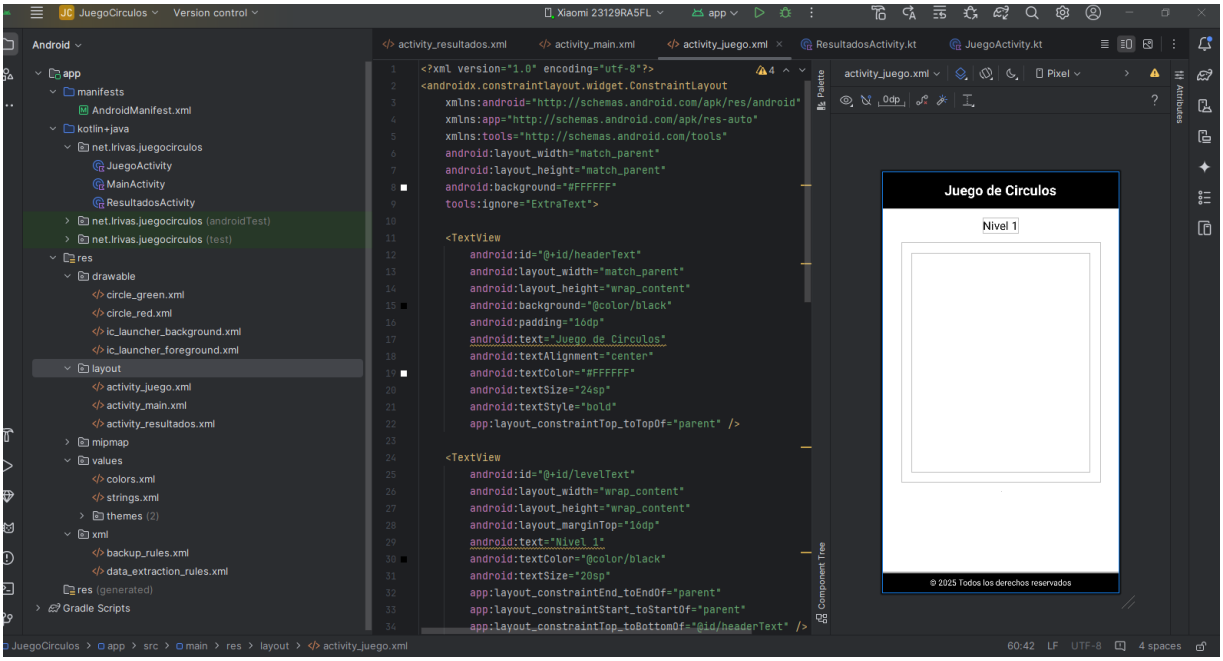
6. **Botón "Nuevo Juego":**
    - Se obtiene una referencia a un Button con el ID nuevoJuego.
    - Se configura un OnClickListener para este botón:
        - Cuando el usuario hace clic en el botón, se inicia la actividad principal (MainActivity) mediante un Intent.
        - Se llama a finish() para cerrar la actividad actual (ResultadosActivity) y evitar que el usuario regrese a ella presionando el botón "Atrás".

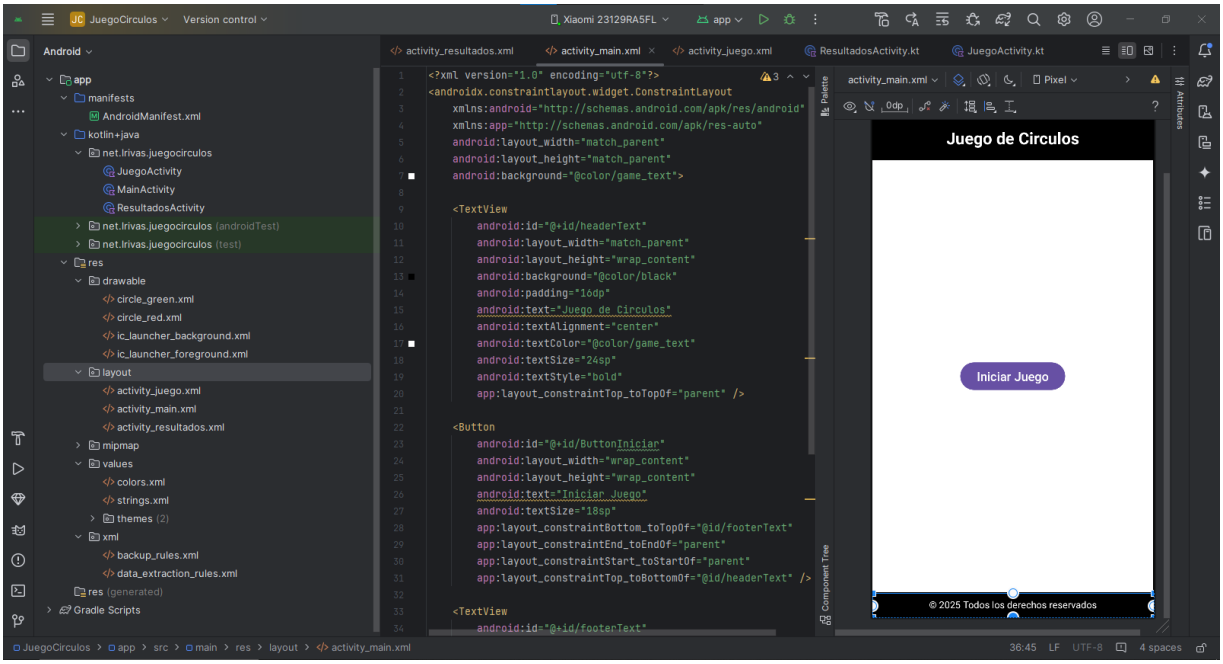
Resumen:

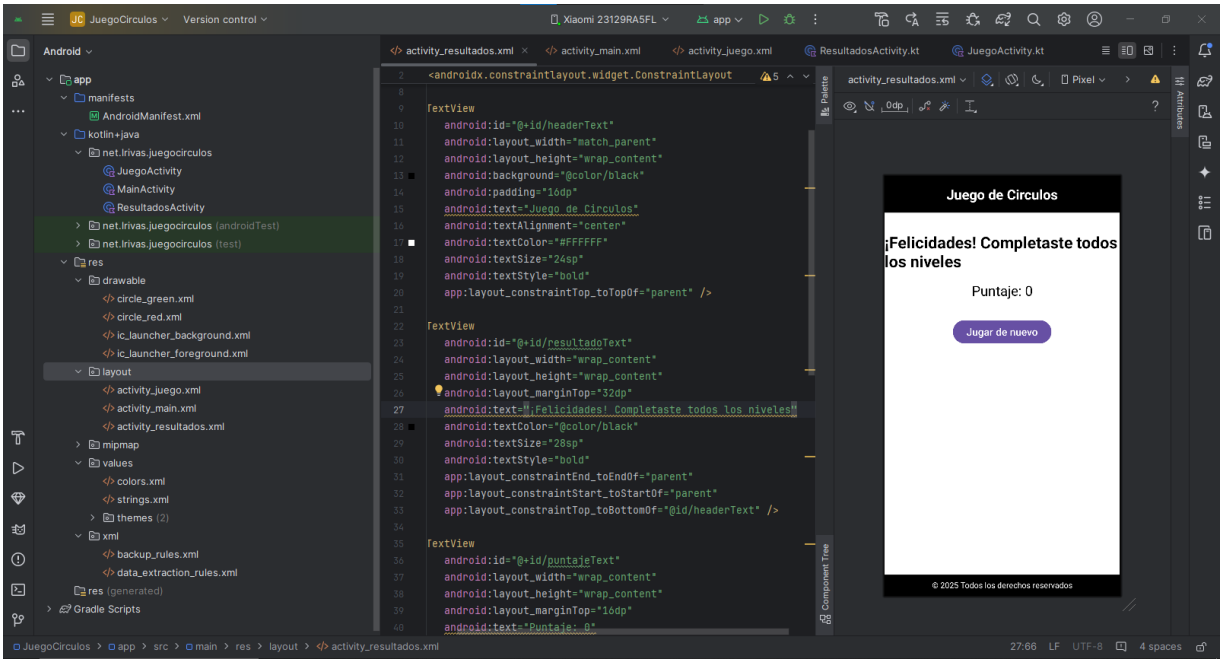
El archivo ResultadosActivity.kt es responsable de:

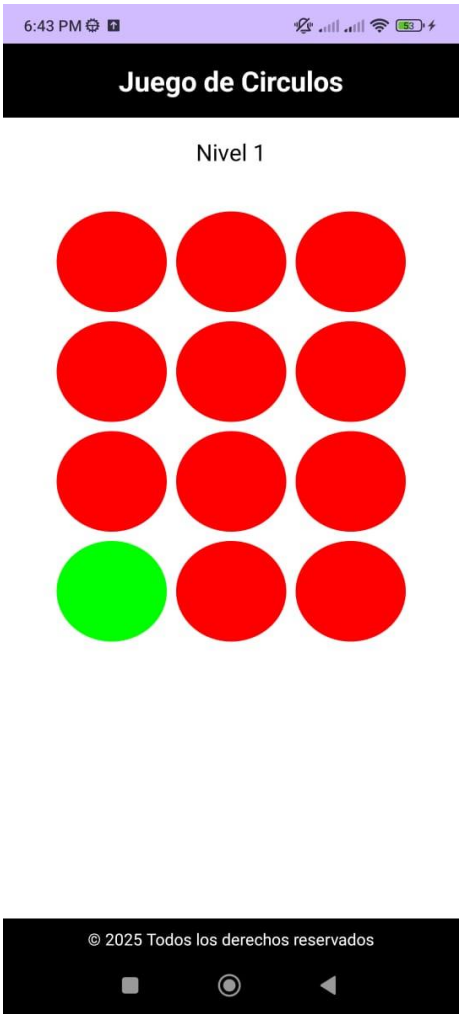
1. Mostrar los resultados del juego (si el usuario ganó o perdió y su puntaje).
2. Permitir al usuario reiniciar el juego y volver a la pantalla principal.
3. Cerrar la actividad actual para evitar comportamientos no deseados al navegar entre pantallas.

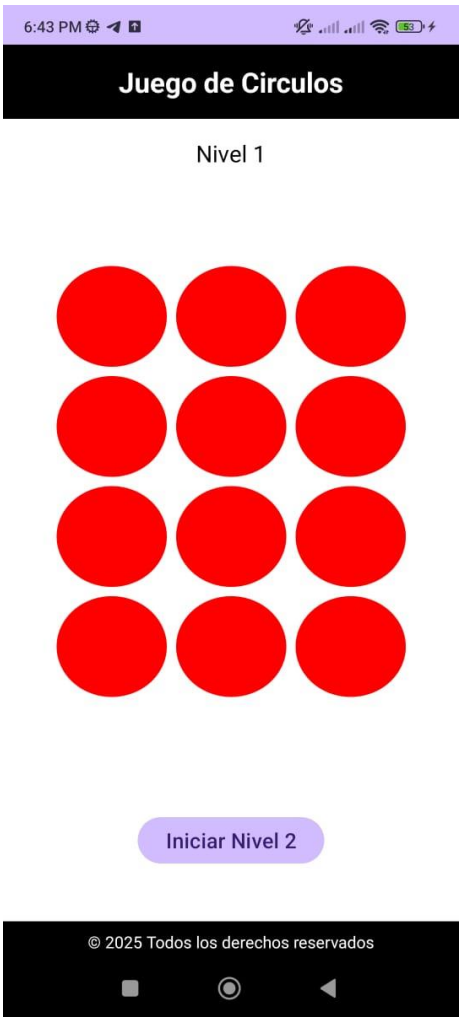
Esta actividad es esencial para proporcionar retroalimentación al usuario sobre su desempeño en el juego y permitirle reiniciar la experiencia si lo desea.

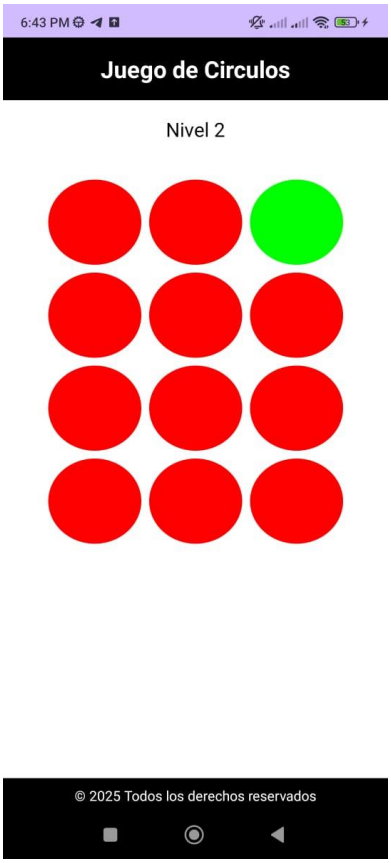

JuegoCirculos
Version control
Xiaomi 23129RA5FL
app
Android
activity_resultados.xml
activity_main.xml
activity_juego.xml
ResultadosActivity.kt
JuegoActivity.kt
app
manifests
AndroidManifest.xml
kotlin+java
net.lrivas.juegocirculos
JuegoActivity
MainActivity
ResultadosActivity
net.lrivas.juegocirculos (androidTest)
net.lrivas.juegocirculos (test)
res
drawable
circle_green.xml
circle_red.xml
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
activity_juego.xml
activity_main.xml
activity_resultados.xml
mipmap
values
colors.xml
strings.xml
themes (2)
xml
backup_rules.xml
data_extraction_rules.xml
res (generated)
Gradle Scripts
<?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"
android:background="#FFFFFF"
tools:ignore="ExtraText">
<TextView
android:id="@+id/headerText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@color/black"
android:padding="16dp"
android:text="Juego de Circulos"
android:textAlignment="center"
android:textColor="#FFFFFF"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintTop_toTopOf="parent" />
<TextView
android:id="@+id/levelText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:text="Nivel 1"
android:textColor="@color/black"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/headerText" />
Palette
Component Tree
Attributes
Pixel
Juego de Circulos
Nivel 1
© 2025 Todos los derechos reservados
JuegoCirculos › app › src › main › res › layout › activity_juego.xml
60:42
LF
UTF-8
4 spaces


JuegoCirculos
Version control
Xiaomi 23129RA5FL
app
Android
activity_resultados.xml
activity_main.xml
activity_juego.xml
ResultadosActivity.kt
JuegoActivity.kt
app
manifests
AndroidManifest.xml
kotlin+java
net.lrivas.juegocirculos
JuegoActivity
MainActivity
ResultadosActivity
net.lrivas.juegocirculos (androidTest)
net.lrivas.juegocirculos (test)
res
drawable
circle_green.xml
circle_red.xml
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
activity_juego.xml
activity_main.xml
activity_resultados.xml
mipmap
values
colors.xml
strings.xml
themes (2)
xml
backup_rules.xml
data_extraction_rules.xml
res (generated)
Gradle Scripts
<?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"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/game_text">
<TextView
android:id="@+id/headerText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@color/black"
android:padding="16dp"
android:text="Juego de Circulos"
android:textAlignment="center"
android:textColor="@color/game_text"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintTop_toTopOf="parent" />
<Button
android:id="@+id/ButtonIniciar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Iniciar Juego"
android:textSize="18sp"
app:layout_constraintBottom_toTopOf="@id/footerText"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/headerText" />
<TextView
android:id="@+id/footerText"
Palette
Component Tree
Attributes
Pixel
Juego de Circulos
Iniciar Juego
© 2025 Todos los derechos reservados
JuegoCirculos › app › src › main › res › layout › activity_main.xml
36:45
LF
UTF-8
4 spaces

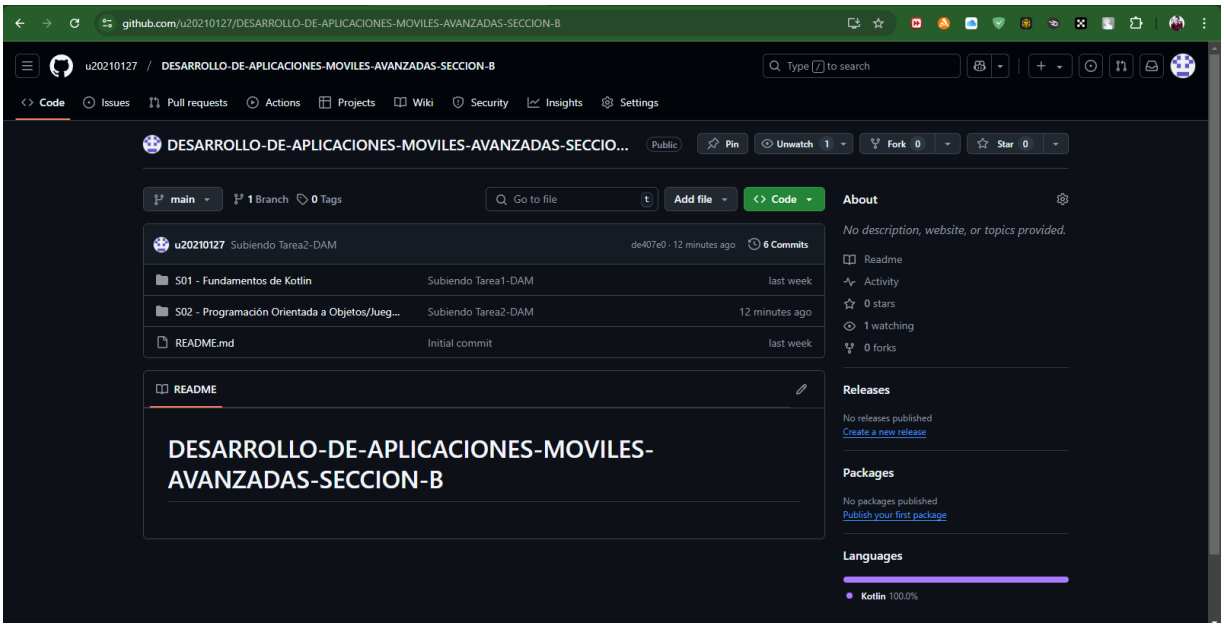

JuegoCirculos
Version control
Xiaomi 23129RA5FL
app
Android
app
manifests
AndroidManifest.xml
kotlin+java
net.lrivas.juegocirculos
JuegoActivity
MainActivity
ResultadosActivity
net.lrivas.juegocirculos (androidTest)
net.lrivas.juegocirculos (test)
res
drawable
circle_green.xml
circle_red.xml
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
activity_juego.xml
activity_main.xml
activity_resultados.xml
mipmap
values
colors.xml
strings.xml
themes (2)
xml
backup_rules.xml
data_extraction_rules.xml
res (generated)
Gradle Scripts
activity_resultados.xml
activity_main.xml
activity_juego.xml
ResultadosActivity.kt
JuegoActivity.kt
<androidx.constraintlayout.widget.ConstraintLayout
TextView
android:id="@+id/headerText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@color/black"
android:padding="16dp"
android:text="Juego de Circulos"
android:textAlignment="center"
android:textColor="#FFFFFF"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintTop_toTopOf="parent" />
TextView
android:id="@+id/resultadoText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="32dp"
android:text="¡Felicidades! Completaste todos los niveles!"
android:textColor="@color/black"
android:textSize="28sp"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/headerText" />
TextView
android:id="@+id/puntajeText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:text="Puntaje: 0"
Palette
Component Tree
Attributes
Pixel
Juego de Circulos
¡Felicidades! Completaste todos los niveles
Puntaje: 0
Jugar de nuevo
© 2025 Todos los derechos reservados
JuegoCirculos > app > src > main > res > layout > activity_resultados.xml
27:66
LF
UTF-8
4 spaces


6:43 PM
Juego de Circulos
Nivel 1
© 2025 Todos los derechos reservados


6:43 PM
Juego de Circulos
Nivel 1
Iniciar Nivel 2
© 2025 Todos los derechos reservados


6:43 PM
Juego de Circulos
Nivel 2
© 2025 Todos los derechos reservados


6:44 PM
Juego de Circulos
¡Felicidades! Completaste todos los niveles
Puntaje: 7
Jugar de nuevo
© 2025 Todos los derechos reservados


github.com/u20210127/DESARROLLO-DE-APLICACIONES-MOVILES-AVANZADAS-SECCION-B
u20210127 / DESARROLLO-DE-APLICACIONES-MOVILES-AVANZADAS-SECCION-B
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
DESARROLLO-DE-APLICACIONES-MOVILES-AVANZADAS-SECCIO...
Public
Pin
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Add file
Code
u20210127 Subiendo Tarea2-DAM
de407e0 · 12 minutes ago
6 Commits
S01 - Fundamentos de Kotlin
Subiendo Tarea1-DAM
last week
S02 - Programación Orientada a Objetos/Jueg...
Subiendo Tarea2-DAM
12 minutes ago
README.md
Initial commit
last week
README
DESARROLLO-DE-APLICACIONES-MOVILES-AVANZADAS-SECCION-B
About
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Kotlin 100.0%