PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

Tarea 4: Elementos multimedia

JESÚS MARTÍN ROBLES

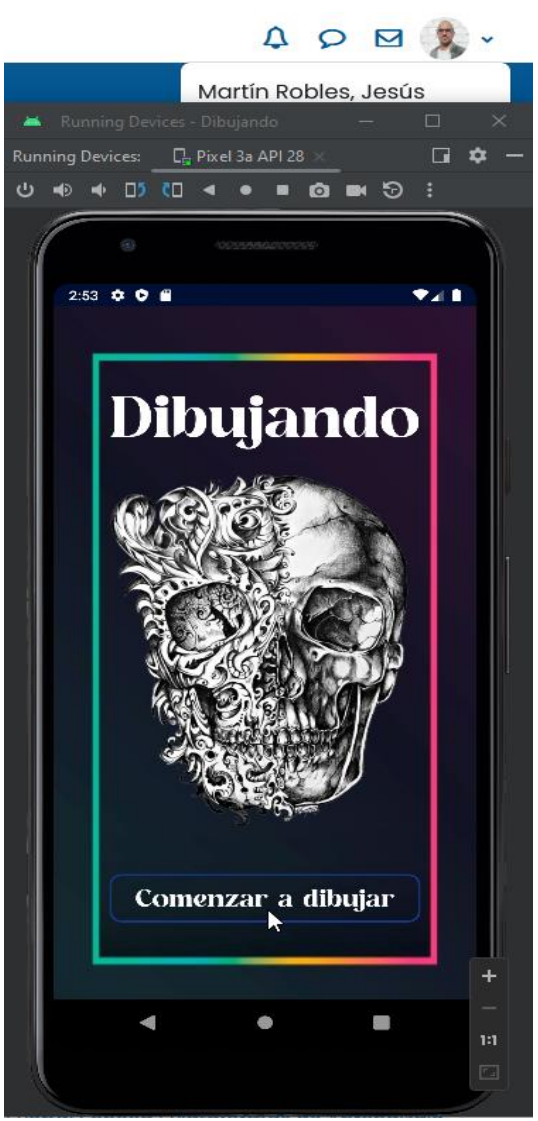
DAM a distancia Curso 23-24

ÍNDICE:

Descripción de Componentes

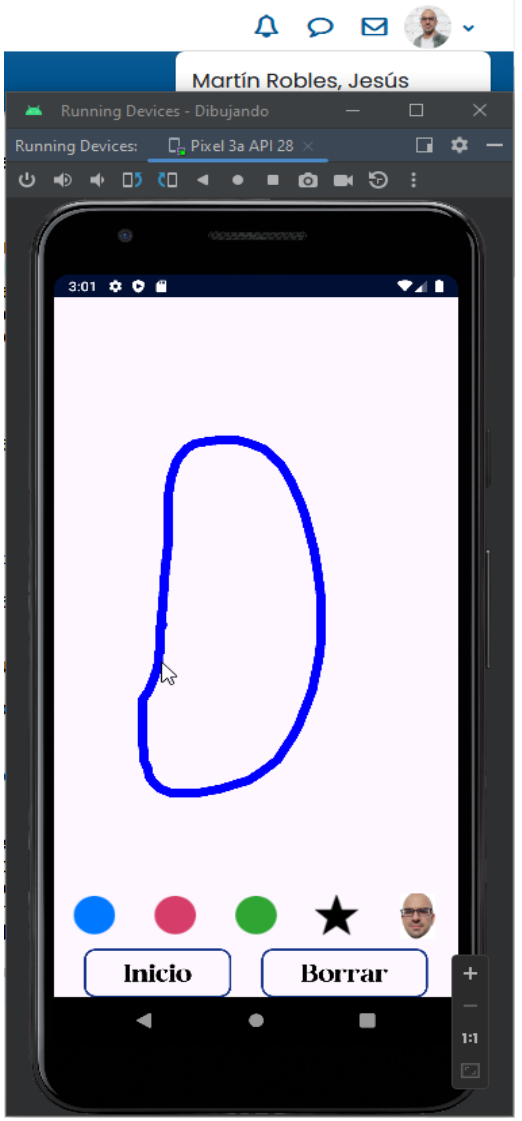
- Mi proyecto consta principalmente de cuatro clases java:
    - La clase `CanvasView` es una vista personalizada en Android que se utiliza para dibujar en un lienzo. Las partes más importantes son: `currentPaint` que es el pincel actual para dibujar, `currentBitmap` que es el mapa de bits actual para dibujar imágenes, `paths` que es una lista de caminos que se han dibujado, `onDraw` que es el método que se llama para dibujar en el lienzo, `onTouchEvent` que es el método que se llama cuando se toca la pantalla y `setBrushBitmap`, `setPaintColor` y `clearCanvas` que son métodos para establecer el mapa de bits del pincel, establecer el color del pincel y borrar el lienzo respectivamente. La clase en general permite dibujar en un lienzo con un pincel de un color específico o con una imagen específica, y también permite borrar el lienzo.
    - La clase `DrawingActivity` es una actividad en Android que se utiliza para manejar la interacción del usuario con la aplicación de dibujo. Las partes más importantes son: `canvasView` que es la vista del lienzo donde se dibuja, `paint` que es el pincel para dibujar, `bitmap` que es el mapa de bits para dibujar imágenes, `onCreate` que es el método que se llama cuando se crea la actividad y los métodos `setOnClickListener` que se llaman cuando se hace clic en los botones. La clase en general permite al usuario dibujar en un lienzo con un pincel de un color específico o con una imagen específica, cambiar el color del pincel, borrar el lienzo y volver a la actividad principal.
    - La clase `MainActivity` es la actividad principal en Android que se utiliza para iniciar la aplicación. Las partes más importantes son: `onCreate` que es el método que se llama cuando se crea la actividad y `startActivity` que es el método que se utiliza para iniciar otras actividades. La clase en general sirve como punto de entrada para la aplicación y puede iniciar otras actividades como `DrawingActivity`.
    - La clase `Quadruple` es una estructura de datos que almacena cuatro elementos. Las partes más importantes son sus cuatro campos, que pueden ser de cualquier tipo, y sus métodos para obtener estos campos. La clase en general se utiliza para agrupar cuatro elementos relacionados en una sola entidad, lo que puede ser útil para devolver cuatro valores desde un método o para almacenar cuatro elementos relacionados en una colección.
- Y de dos archivos XML:
    - El archivo `activity_main.xml` de la aplicación 'Dibujando' define la interfaz de usuario de la actividad principal, que incluye un `ConstraintLayout` con un `TextView`, un `ImageView` y un `Button`. Estos elementos se personalizan y se posicionan utilizando varios atributos, como `android:layout_width`, `android:layout_height`, `app:layout_constraintEnd_toEndOf`, `android:text`, `android:textColor`, `android:textSize` y `app:fontFamily`.
    - El archivo `activity_drawing.xml` define la interfaz de usuario de la actividad de dibujo en la aplicación 'Dibujando'. Este archivo utiliza un `RelativeLayout` como contenedor principal que contiene un `CanvasView` para el área de dibujo, un `View` que actúa como lienzo, y dos `LinearLayouts` que contienen botones. El primer `LinearLayout` (`five_buttons_layout`) contiene cinco botones que permiten al usuario seleccionar diferentes opciones de dibujo, como cambiar el color del pincel o seleccionar una imagen para dibujar. El segundo `LinearLayout` (`bottom_buttons_layout`) contiene dos botones para regresar a la pantalla principal y borrar el lienzo, respectivamente. Los botones se distribuyen equitativamente en el `LinearLayout` utilizando el atributo `android:layout_weight`.
- Aparte de varias imágenes, un archivo de sonido en mp3 y un nuevo tipo de fuente.

Capturas de Pantalla



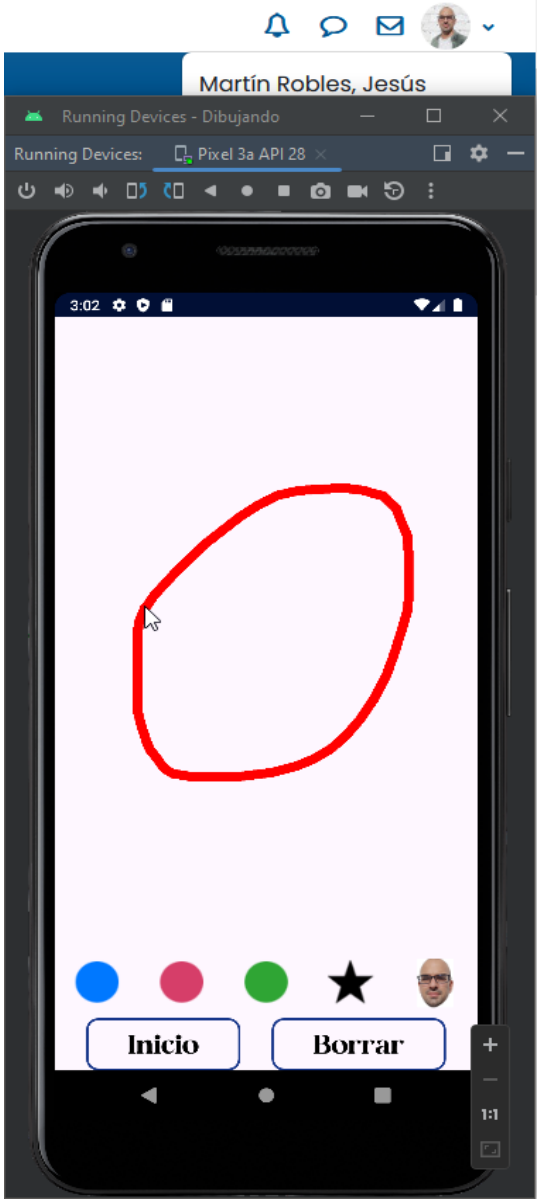
Pantalla inicia

Pantalla del lienzo en blanco

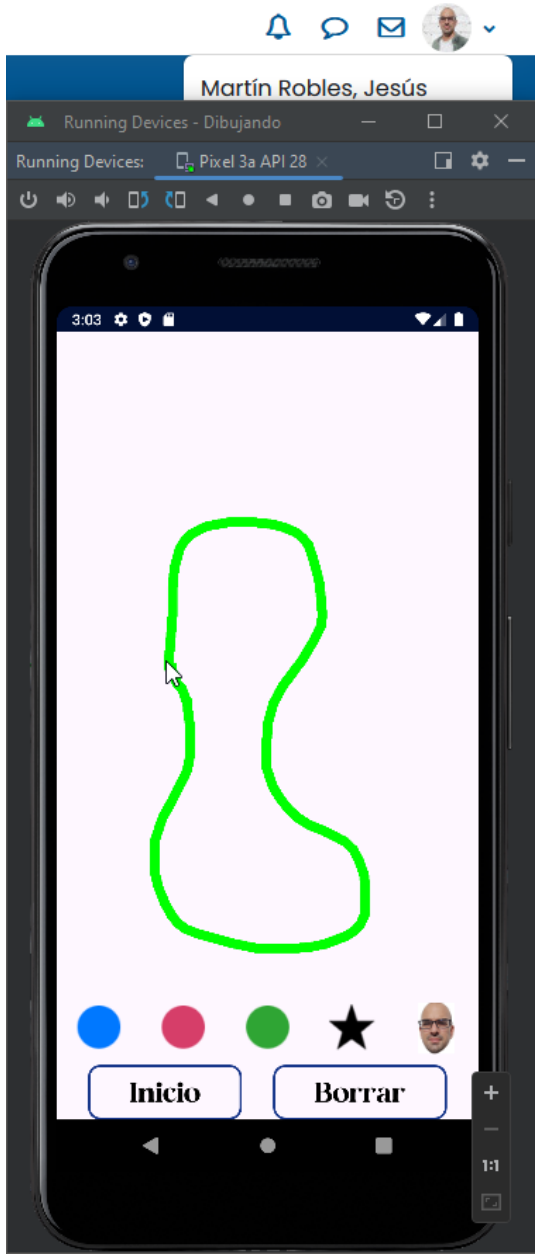


Trazo azul

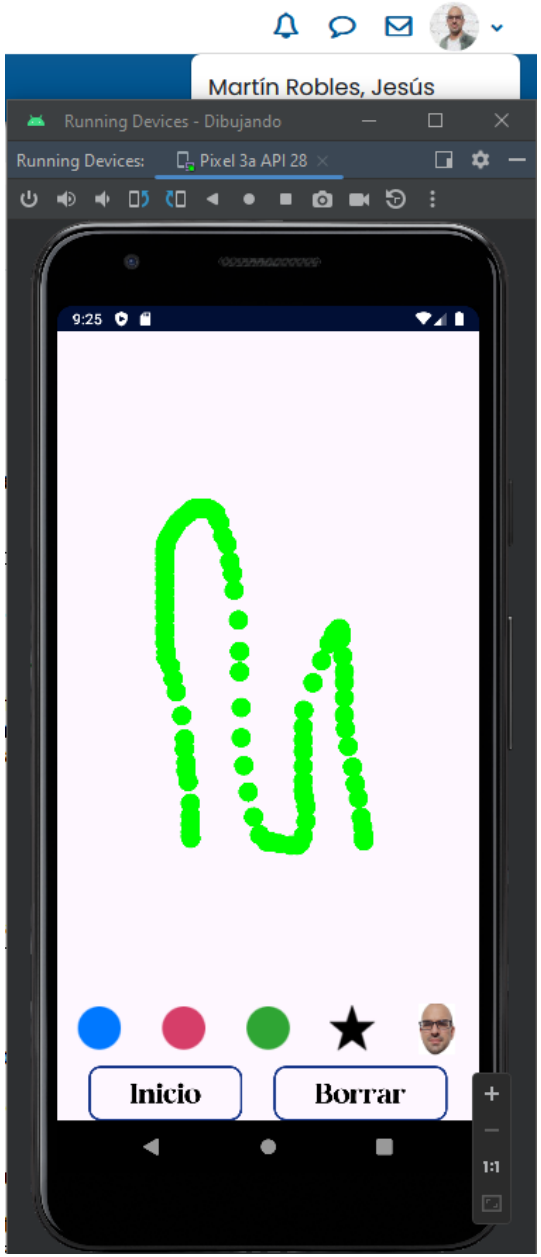

Martín Robles, Jesús
Running Devices - Dibujando
Running Devices:
Pixel 3a API 28
9:22
Inicio
Borrar
1:1


Martín Robles, Jesús
Running Devices - Dibujando
Running Devices:
Pixel 3a API 28
3:02
Inicio
Borrar
1:1


Martín Robles, Jesús
Running Devices - Dibujando
Running Devices:
Pixel 3a API 28
9:24
Inicio
Borrar
1:1



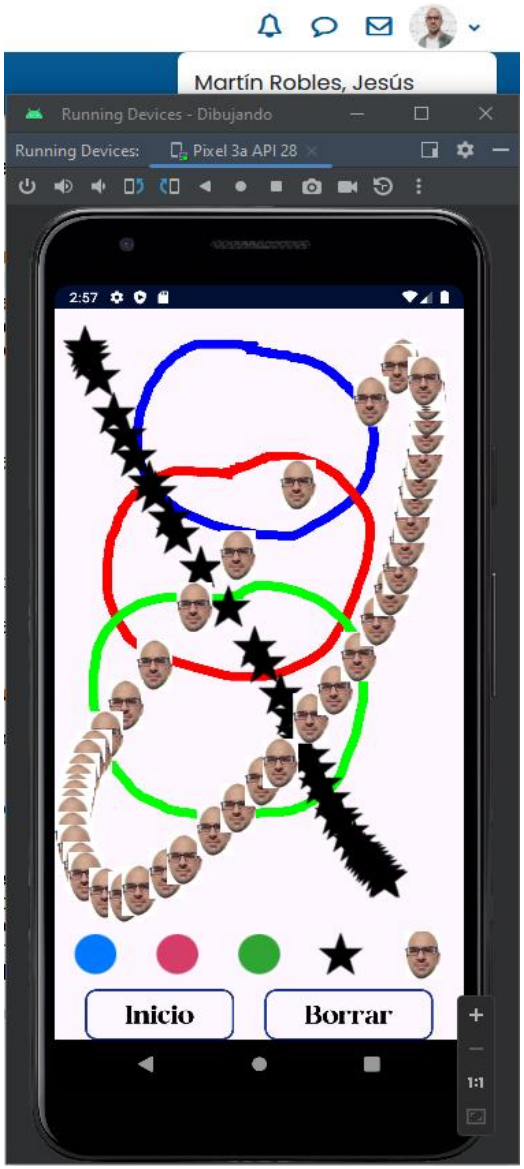
Trazo verde

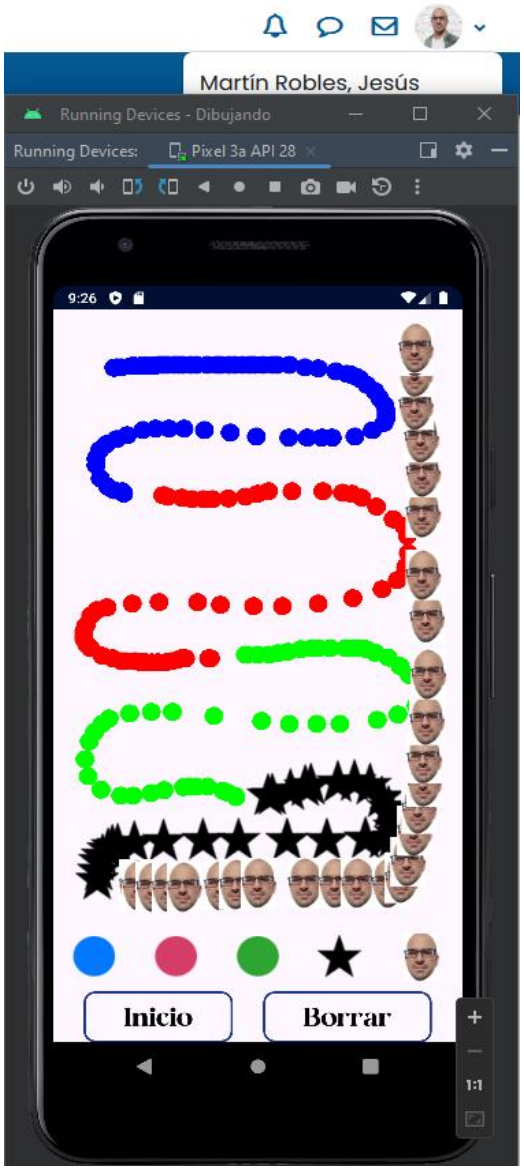


Trazo verde de círculos

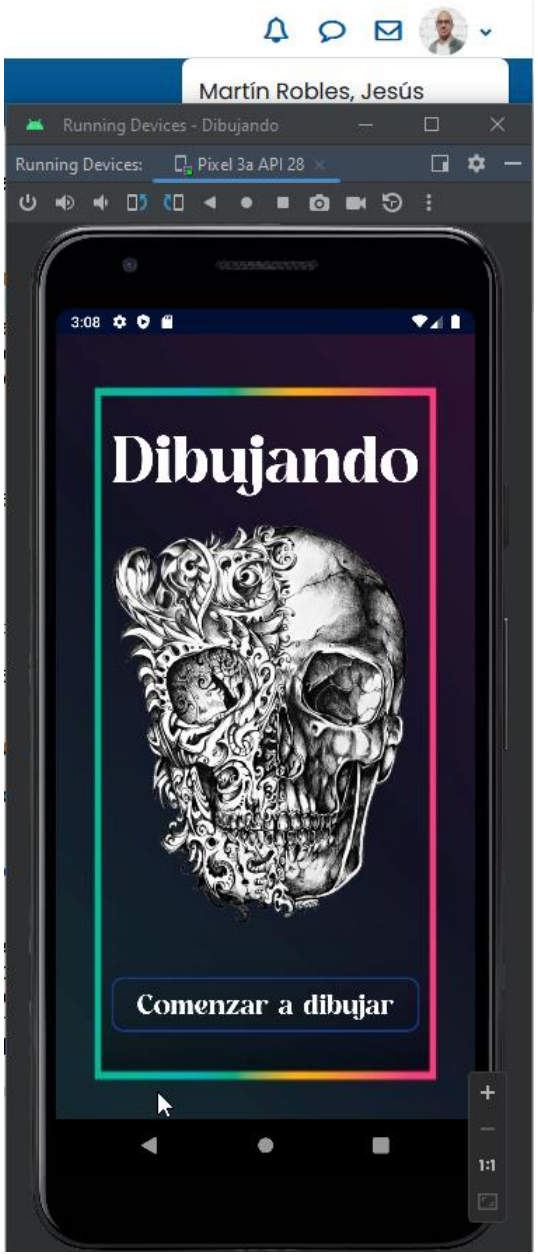
Trazo de estrellas


Martín Robles, Jesús
Running Devices - Dibujando
Running Devices:
Pixel 3a API 28
3:04
Inicio
Borrar


Martín Robles, Jesús
Running Devices - Dibujando
Running Devices:
Pixel 3a API 28
2:57
Inicio
Borrar


Martín Robles, Jesús
Running Devices - Dibujando
Running Devices:
Pixel 3a API 28
9:26
Inicio
Borrar

Pantalla borrada



Vuelta a la pantalla de inicio

Dificultades

Durante la fase de desarrollo de la aplicación, la principal dificultad me surgió al implementar la funcionalidad de los cinco botones para dibujar en la segunda pantalla. Estos botones se dividían en tres para trazos de colores y dos para trazos de imágenes.

1. Problemas con Trazos de Colores:
    - Inicialmente, al dibujar trazos de colores, se presentaba un inconveniente en el cual todo el trazo existente en el lienzo, independientemente del color, automáticamente adoptaba el color del último trazo dibujado. Este comportamiento no deseado generaba confusión en la visualización de los dibujos.
2. Fallas en Trazos de Imágenes:
    - En la implementación de los trazos de imágenes, se enfrentaron desafíos adicionales. A pesar de que los trazos de colores funcionaban correctamente, los trazos de imágenes no presentaban el comportamiento esperado. La aplicación experimentaba cierres inesperados, regresiones a la pantalla de inicio o el lienzo quedaba en blanco al intentar dibujar con estos pinceles.
3. Problemas de Estabilidad:
    - A lo largo del proceso de creación de los pinceles, se detectaron múltiples incidentes en los que la aplicación experimentaba fallos, cierres repentinos o comportamientos inesperados al intentar realizar acciones de dibujo. Estas instancias generaban una experiencia de usuario insatisfactoria y requerían una solución.
4. Adaptación del Tamaño de Imágenes de Fondo:
    - Se enfrentaron desafíos relacionados con el tamaño de la imagen de fondo, especialmente en diferentes emuladores. La adaptación del tamaño se volvía problemática, lo que me llevó a la decisión de reducir el tamaño de las imágenes y utilizar la propiedad 'fill_parent' para garantizar una visualización adecuada en diversas pantallas.

Además, para abordar el problema de los trazos, implementé una solución en la clase CanvasView. Esta clase maneja el dibujo en el lienzo manteniendo un registro de todos los trazos dibujados y redibujándolos cada vez que se llama a onDraw(). Esta estrategia asegura que todos los trazos se respeten unos a otros, ya que cada trazo se dibuja encima del anterior, garantizando una representación precisa de la obra de arte.

Ante las dudas que me han surgido a raíz de leer los foros, respecto a cómo tenían que ser los trazos que realizara el pincel, he decidido presentar el código de la aplicación y el documento explicativo con dos opciones en el caso de los pinceles de colores, una en la que el trazo del pincel se realiza como una línea continua y otra en el que la línea se realiza como una consecución de círculos del color correspondiente.