

Reporte de la herramienta llamada Area of Circles de geogebra hecha en android studio

Luis Rodrigo Loredó Tavaréz*
*Ingeniería en Tecnologías de la Información
Universidad Politécnica de Victoria

Resumen—Una escuela desea implementar una herramienta educativa para sus estudiantes en forma de aplicación móvil en Android. La herramienta, desarrollada en Java usando Android Studio, tiene el propósito de enseñar a los estudiantes conceptos básicos de geometría, específicamente relacionados con el círculo y las propiedades de sus circunferencias. Se ha observado que muchos estudiantes tienen dificultades para comprender la relación entre el radio de un círculo y la longitud de su circunferencia. Adicionalmente, quieren explorar conceptos de segmentación y polígonos usando controles interactivos en la pantalla.

I. INTRODUCCIÓN

En el ámbito educativo, una comprensión profunda y aplicada de la geometría es esencial para desarrollar habilidades matemáticas más avanzadas y capacidades de resolución de problemas. La geometría, como rama de las matemáticas, ha sido estudiada a lo largo de los siglos y su importancia en la educación ha sido ampliamente reconocida [1]. Sin embargo, muchos estudiantes encuentran desafiantes ciertos conceptos geométricos, particularmente aquellos que requieren una comprensión visual y espacial. Uno de estos desafíos observados en el aula es la relación entre el radio de un círculo y la longitud de su circunferencia, un concepto fundamental en la geometría del círculo que ha sido bien documentado [2].

II. DESARROLLO EXPERIMENTAL

La aplicación diseñada busca abordar la problemática educativa que enfrentan los usuarios al aprender conceptos básicos de geometría, enfocándose principalmente en las propiedades de las circunferencias y la relación entre el radio de un círculo y la longitud de su circunferencia, así como en la exploración de segmentación y polígonos. A continuación se describen las formas en que la aplicación resuelve estos desafíos:

■ Interactividad y Visualización:

- La aplicación proporciona una interfaz gráfica interactiva que permite a los usuarios manipular y visualizar geometrías en tiempo real.
- Utiliza controles deslizantes (SeekBar) para permitir a los usuarios ajustar valores como el radio del círculo, y visualizar cómo estos cambios afectan la longitud de la circunferencia y otras propiedades geométricas.
- A través del desplazamiento de los controles deslizantes, los usuarios pueden observar cómo la línea se desenvuelve alrededor del círculo, proporcionando

una representación visual de la relación entre el radio y la circunferencia.

■ Exploración de Conceptos Geométricos Avanzados:

- La aplicación introduce una forma de explorar la segmentación y los polígonos, permitiendo a los usuarios visualizar cómo se forman los polígonos al segmentar un círculo.
- A través de la manipulación interactiva, los usuarios pueden variar el número de segmentos y observar cómo cambian las propiedades geométricas, proporcionando una comprensión más profunda de estos conceptos.

■ Animación y Feedback Inmediato:

- La aplicación ofrece animaciones que ayudan a los usuarios a visualizar cómo las propiedades geométricas cambian dinámicamente con la variación de parámetros.
- Proporciona feedback inmediato a través de la visualización gráfica, lo que facilita la comprensión y el aprendizaje interactivo.

■ Ambiente de Aprendizaje Autodirigido:

- La aplicación proporciona un ambiente donde los usuarios pueden explorar y aprender a su propio ritmo, experimentando con diferentes configuraciones y observando los resultados en tiempo real.

■ Simplicidad y Accesibilidad:

- Al estar diseñada como una aplicación móvil, la herramienta es accesible para los usuarios, permitiendo el aprendizaje en cualquier momento y lugar.

■ Personalización del Aprendizaje:

- La aplicación ofrece a los usuarios la capacidad de explorar conceptos geométricos de manera individualizada, lo que puede ser beneficioso para diferentes estilos de aprendizaje.

EXPLICACIÓN DE CLASES, VARIABLES Y MÉTODOS:

1. Clase: DragAndDropView

Variables::

- mScaleFactor: Variable de tipo float que podría usarse para almacenar un factor de escala para el dibujo.
- mPosX y mPosY: Variables de tipo float para almacenar las coordenadas de posición.

- `thread`: Variable de tipo `DragAndDropThread` para gestionar el ciclo de dibujo y actualización de la vista.
- `figuras`: Lista de `ArrayList<Figura>` para almacenar las figuras geométricas que serán dibujadas.
- `seekBar`, `newSeekBar`, `thirdSeekBar`: Variables de tipo `SeekBar` para controles deslizables en la interfaz de usuario.
- `lap`, `extra`, `seekbar3`, `progress`: Variables de tipo `int` para almacenar valores de progreso de los `SeekBars` y otros valores relevantes.
- `checkBox`: Variable de tipo `CheckBox` para un control de casilla de verificación en la interfaz de usuario.

Métodos::

- `DragAndDropView(Context context)`: Constructor que inicializa la vista y configura el callback del `SurfaceHolder`.
- `initSeekBar(Context context)`: Método para inicializar y configurar los `SeekBars` y el `CheckBox`, y agregarlos al layout principal.
- `surfaceChanged(...)`, `surfaceCreated(...)`, `surfaceDestroyed(...)`: Métodos de callback del ciclo de vida de `SurfaceHolder`.
- `drawAnimation(Canvas canvas)`: Método para dibujar animaciones en el canvas proporcionado.
- `onDraw(Canvas canvas)`: Método para dibujar en el canvas proporcionado, llamando a `drawAnimation(canvas)`.

2. Clase: *DragAndDropThread*

Variables::

- `sh`: Variable de tipo `SurfaceHolder` para acceder y controlar el `Surface`.
- `view`: Variable de tipo `DragAndDropView` para referenciar la vista donde se realizará el dibujo.
- `run`: Variable de tipo `boolean` para controlar el ciclo de ejecución del hilo.

Métodos::

- `DragAndDropThread(SurfaceHolder sh, DragAndDropView view)`: Constructor que inicializa las variables `sh` y `view`.
- `setRunning(boolean run)`: Método para establecer el estado de ejecución del hilo.
- `run()`: Método sobrescrito de la clase `Thread`, donde se gestiona el ciclo de dibujo y actualización de la vista.

3. Clase: *Figura*

Variables::

- `id`: Variable de tipo `int` para almacenar un identificador único para la figura.
- `x` y `y`: Variables de tipo `float` para almacenar las coordenadas de posición de la figura.
- `color`: Variable de tipo `String` para almacenar el color de la figura.

Métodos::

- `getX()`, `setX(float x)`, `getY()`, `setY(float y)`: Métodos de acceso (getter y setter) para las coordenadas `x` e `y`.
- `getId()`: Método de acceso (getter) para obtener el identificador de la figura.
- `getColor()`, `setColor(String color)`: Métodos de acceso (getter y setter) para el color de la figura.

1. PLATAFORMA Y LENGUAJE DE PROGRAMACIÓN:

- **Plataforma:** Android
- **Lenguaje de Programación:** Java
- **Entorno de Desarrollo:** Android Studio

Esta aplicación se desarrolla en la plataforma Android, utilizando el lenguaje de programación Java. Android Studio, que es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android, proporciona las herramientas necesarias para construir, probar y depurar la aplicación.[3]

2. ESTRUCTURA DE LA APLICACIÓN:

La estructura central de la aplicación se compone de tres clases principales: `DragAndDropView`, `DragAndDropThread`, y `Figura`.

- `DragAndDropView`: Esta clase extiende de `SurfaceView` e implementa `SurfaceHolder.Callback`, siendo el punto principal donde se manejan las interacciones del usuario y se dibuja la interfaz gráfica.
- `DragAndDropThread`: Esta clase extiende de `Thread` y se encarga de gestionar el ciclo de dibujo y actualización de la vista.
- `Figura`: Es una clase abstracta que define las características comunes de las figuras geométricas que se utilizarán en la aplicación.

3. FUNCIONAMIENTO:

La aplicación funciona a través de una interfaz gráfica que permite a los usuarios interactuar con diferentes controles como `SeekBars` y `CheckBox` para explorar propiedades geométricas del círculo y otros polígonos. La clase `DragAndDropView` contiene la lógica principal de cómo se renderiza la geometría en el canvas y cómo se actualiza en respuesta a las interacciones del usuario. Los `SeekBars` permiten a los usuarios controlar diferentes parámetros geométricos, y el `CheckBox` proporciona una opción adicional para explorar la relación entre el radio y la circunferencia. El hilo `DragAndDropThread` garantiza que la vista se actualice y se vuelva a dibujar de manera continua, proporcionando una interacción fluida.

4. LIBRERÍAS Y FUNCIONALIDADES

SURFACEVIEW Y SURFACEHOLDER:

Uso: `SurfaceView` proporciona una superficie de dibujo dedicada, que se ejecuta en un hilo separado. Esto es ideal para operaciones de dibujo intensivas y actualizaciones de UI

que pueden ser manejadas fuera del hilo principal de la UI para evitar la lentitud o el atasco. `SurfaceHolder` proporciona el control y el acceso a esta superficie.

Por qué se usan: Se utilizan para proporcionar una área de dibujo eficiente y para manejar el ciclo de vida de la superficie de dibujo en respuesta a los eventos del sistema.

THREAD:

Uso: La clase `Thread` [4] se utiliza para crear un hilo separado de ejecución.

Por qué se usa: Se utiliza para manejar las operaciones de dibujo y actualización de la superficie en un hilo separado, permitiendo una interacción fluida y evitando la lentitud en la UI.

CANVAS:

Uso: `Canvas` [5] proporciona los métodos necesarios para dibujar texto, figuras y bitmaps en la superficie.

Por qué se usa: Se utiliza para dibujar las figuras geométricas y los controles de la interfaz de usuario en la superficie proporcionada por `SurfaceView`.

PAINT:

Uso: `Paint` [6] se utiliza para definir los estilos de dibujo como el color, el estilo del trazo, etc.

Por qué se usa: Se utiliza para definir cómo se dibujan las figuras geométricas y el texto en el `Canvas` [3].

SEEKBAR Y CHECKBOX:

Uso: `SeekBar` [7] proporciona un control deslizable que permite a los usuarios seleccionar un valor de un rango continuo. `CheckBox` [8] proporciona un control de casilla de verificación que permite a los usuarios seleccionar o deseleccionar una opción.

Por qué se usan: Se utilizan para proporcionar interactividad a los usuarios, permitiéndoles explorar las propiedades geométricas al ajustar los valores mediante controles deslizantes y casillas de verificación.

ARRAYLIST:

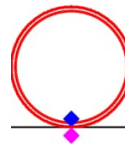
Uso: `ArrayList` [9] proporciona una estructura de datos de lista dinámica que puede crecer y disminuir de tamaño según sea necesario.

Por qué se usa: Se utiliza para almacenar y gestionar una lista de objetos de la clase `Figura`, que representan las figuras geométricas que se dibujarán en el `Canvas`.

III. RESULTADOS

La aplicación creada en Java mediante Android Studio, logra abordar los desafíos educativos en geometría, facilitando a los usuarios la comprensión de las relaciones entre las propiedades geométricas de circunferencias y polígonos. A través de una interfaz interactiva y visual, los alumnos pueden manipular geometrías en tiempo real, observando cómo las variaciones en parámetros como el radio afectan a otras propiedades. Las clases clave `DragAndDropView`, `DragAndDropThread`, y `Figura` gestionan la UI, el ciclo de dibujo y las representaciones geométricas, respectivamente.

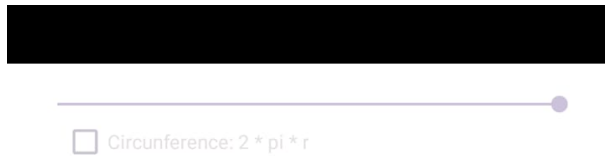
Pantalla completa del usuario:



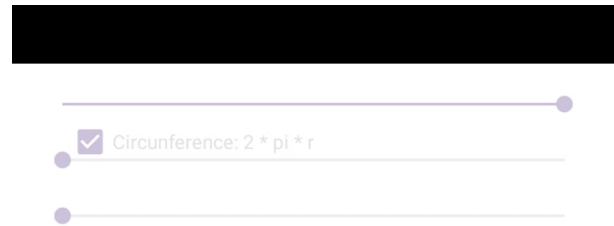
(a) Pantalla principal

Figura 1: En esta pantalla se puede visualizar algunos elementos de la pantalla

Esta pantalla muestra que con el seekbar, se realiza el movimiento de la linea roja, hace la animacion hacia la derecha, desenvolviendose del circulo y al llegar a 180, se visualiza el checkbox



se visualiza el checkbox presionado el cual agrega otras dos seekbar



(a) Pantalla principal

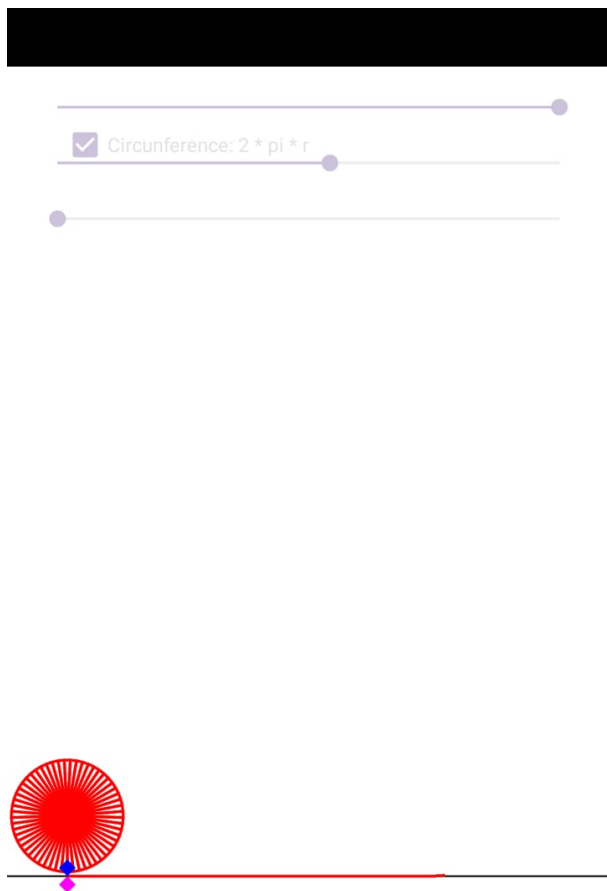
Figura 2: En esta pantalla se puede visualizar mas elementos de la pantalla asi como el funcionamiento de la linea roja



(a) Pantalla principal

Figura 3: Activacion de las dos seekbar mediante el checkbox

El funcionamiento del seekbar hace que se generen triángulos dentro del círculo



(a) Pantalla principal

Figura 4: Creación de los triángulos del círculo internos

IV. CONCLUSIÓN

La aplicación desarrollada aborda de manera efectiva los desafíos educativos identificados, facilitando a los usuarios el aprendizaje y la exploración de conceptos geométricos básicos relacionados con las circunferencias y polígonos. Utilizando Java en el entorno de Android Studio, se creó una herramienta interactiva y visual que permite a los usuarios manipular geometrías en tiempo real, promoviendo así una comprensión más profunda de cómo las propiedades geométricas se relacionan entre sí.

Las clases `DragAndDropView`, `DragAndDropThread`, y `Figura` constituyen la columna vertebral de la aplicación, gestionando la interfaz de usuario, el ciclo de dibujo y actualización, y las representaciones de las figuras geométricas respectivamente. La interactividad es una característica destacada de esta aplicación, proporcionada principalmente a través de controles deslizables (`SeekBar`) y una casilla de verificación (`CheckBox`), que permiten a los usuarios ajustar y experimentar con los parámetros geométricos de una manera intuitiva.

La utilización de `SurfaceView` y `SurfaceHolder` junto con un hilo separado para las operaciones de dibujo, garantiza una ejecución fluida y una respuesta rápida a las interacciones del usuario, lo que es crucial para mantener el interés y la participación de los usuarios. La clase `Canvas`, junto con la clase `Paint`, proporciona las herramientas necesarias para renderizar las geometrías y los controles de la UI de manera efectiva.

La capacidad de visualizar cómo las propiedades geométricas cambian dinámicamente en respuesta a la manipulación de controles interactivos, brinda a los usuarios una retroalimentación visual inmediata que es invaluable para el aprendizaje autodirigido. Esto, combinado con la accesibilidad y simplicidad de una aplicación móvil, crea un entorno de aprendizaje enriquecedor que es fácilmente accesible para los usuarios en cualquier momento y lugar.

REFERENCIAS

- [1] D. E. Smith. *History of Mathematics: Volume I - General Survey of the History of Elementary Mathematics*. https://www.hlevkin.com/hlevkin/90MathPhysBioBooks/mathHistory/Smith_History-Of-Mathematics-Vol1_1923.pdf. Consultado el 13-10-2023.
- [2] Licda. Lilliam Patricia Rojas Artavia. *Circunferencia*. https://www.mep.go.cr/sites/default/files/recursos/recursos-interactivos/educ_abierta/mate_primaria/areas/geometria/circunferencia.pdf.
- [3] Google. *Android Studio*. <https://developer.android.com/studio>.
- [4] AndroidDev. *Thread*. <https://developer.android.com/reference/java/lang/Thread>. Consultado el 13-10-2023.
- [5] AndroidDev. *Canvas*. <https://developer.android.com/reference/android/graphics/Canvas>. Consultado el 13-10-2023.
- [6] AndroidDev. *Paint*. <https://developer.android.com/reference/android/graphics/Paint>.

- [7] AndroidDev. *SeekBar*. <https://developer.android.com/reference/android/widget/SeekBar>.
- [8] AndroidDev. *CheckBox*. <https://developer.android.com/reference/android/widget/CheckBox>.
- [9] AndroidDev. *ArrayList*. <https://developer.android.com/reference/java/util/ArrayList>. Consultado el 13-10-2023.