

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**ALUMNO:**

**ALDAVERA GALLAGA IVÁN**

**APPLICATION DEVELOPMENT FOR MOBILE DEVICES**

**PROFESOR:**

**ALEJANDRO SIGFRIDO CIFUENTES ÁLVAREZ**

**TÍTULO DEL REPORTE**

**CANVAS**

**FECHA DE REALIZACIÓN:**

**18 – JUNIO – 2020**

**VERSIÓN DEL REPORTE:**

**V1**

**ÍNDICE**

**OBJETIVO……………………………………………………………1**

**CONCEPTOS………………………………………………………...1**

**DESARROLLO……………………………………………………….2**

**LISTADO DE SOFTWARE…………………………………..2**

**LISTADO DE HARDWARE………………………………….2**

**PRUEBAS…………………………………………………………….3**

**EJEMPLO 1……………………………………………………3**

**EJEMPLO 2……………………………………………………4**

**CONCLUSIONES……………………………………………………4**

**BIBLIOGRAFÍA………………………………………………………4**

**OBJETIVO**

Aprender a dibujar gráficas y figuras mediante el canvas en Android, de manera similar a como se podía realizar en MIDLETs.

**CONCEPTOS**

El ambiente android.graphics divide el dibujo en dos áreas: Qué dibujar, a cargo del Canvas y cómo dibujar, a cargo de Paint.

**Canvas** proporciona un método para trazar una línea, mientras que Paint proporciona métodos para definir el color de esa línea. El Canvas tiene un método para dibujar un rectángulo, mientras que Paint define si rellena ese rectángulo con un color o dejarlo vacío. El Canvas define las formas que se pueden dibujar en la pantalla, mientras que Paint define el color, estilo, fuente, y así sucesivamente para cada figura que se dibuja.

**Manejo de Eventos**: para elaborar adecuadamente la vista personalizada, se necesita conocer su tamaño. Las vistas personalizadas complejos a menudo necesitan realizar múltiples cálculos de diseño, dependiendo del tamaño y la forma de su área de la pantalla. Nunca hacer suposiciones sobre el tamaño de la vista en la pantalla. Incluso si sólo una aplicación utiliza el View, esa aplicación debe manejar diferentes tamaños de pantalla, múltiples densidades de pantalla y diferentes relaciones de aspecto, tanto en modo vertical y horizontal.

**Dibujo**: Una vez creados los objetos y definido el código de medición, se implanta onDraw(). Cada View implanta a onDraw ( ) de forma diferente, pero hay algunas operaciones comunes que la mayoría de Views comparten:

* Dibujar texto usando drawText(). Especificar el tipo de letra con setTypeface() y el color del texto con setColor().
* Dibujar formas primitivas utilizando drawRect(), drawOval(), y drawArc(). Con setStyle() se cambian las formas si se rellenan, perfilan, o ambos.
* Dibujar formas más complejas usando la clase Path. Definir una forma agregando líneas y curvas a un objeto Path, y a continuación, dibujar la forma usando drawPath(). Al igual que con las formas primitivas, las rutas pueden perfilarse, rellenarse, o ambos dependiendo de setStyle().
* Definir rellenos degradados creando objetos LinearGradient. Invocar a setShader() para utilizar el LinearGradient en formas rellenas.
* Dibujar mapas de bits utilizando drawBitmap().

**DESARROLLO**

Para el desarrollo de estos ejercicios, se retomó los conceptos ya manejados anteriormente en MIDLETs pero ahora aplicados a Android, tales como el área para dibujar o el pincel y las propiedades de éste.

Para el primer ejercicio, se tuvo que dividir la pantalla en cuatro partes, de manera que quedara como un plano cartesiano, para poder introducir en ella varias figuras como un círculo, un óvalo o un arco.

Para el segundo ejercicio, esta vez además de dividir la pantalla en cuatro partes, se debió dibujar ondas senoidales y cosenoidales sobre la división horizontal, además de ampliar o reducir los espectros de frecuencia de éstas.

**LISTADO DE SOFTWARE:**

* Java Development Kit 8: Necesario para el desarrollo de aplicaciones Android.
* Android Studio 4.0: IDE para el desarrollo de aplicaciones Android
* Windows 10: Sistema operativo sobre el que corren los anteriores programas.
* Android 10. Sistema Operativo sobre el que corre la aplicación

**LISTADO DE HARDWARE:**

* Laptop HP con GB RAM DDR4 y procesador Intel Core i3
* Dispositivo Móvil Motorola One con 4GB de RAM y procesador SoC Qualcomm Snapdragon 625
* Cable tipo C para la transferencia de datos. PRUEBAS

**PRUEBAS**

**EJEMPLO 1:**

|  |  |
| --- | --- |
|  |  |
| Aquí se muestra el ejemplo 1 tal cual viene en el código proporcionado. | Y aquí con las cuatro figuras que se piden para dibujar. |

**EJEMPLO 2:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Así se ve en el canvas de las ondas senoidales y cosenoidales tal cual como se proporcionó en el código. | Aquí se puede ver la modificación del inciso 1, la cual pide modificar la amplitud y el periodo de las dos señales, seno y coseno, del inciso | Y aquí la otra modificación, la cual pide acotar las magnitudes de la amplitud y periodo, en los ejes coordenados del inciso |

**CONCLUSIONES**

El canvas nos permite dibujar todo tipo de figuras en Android, y es útil al momento de querer representar algo de manera gráfica.

**BIBLIOGRAFÍA**

* Archivo Android Canvas, Cifuentes, A. IPN-ESCOM