



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



Application Development For Mobile Devices

Semana 2

Emiliano González López

11/Junio/2020

Canvas

El paso más importante en la elaboración de una vista personalizada es reemplazar el método `onDraw()`. El parámetro de `onDraw()` es un objeto `Canvas` que el `View` puede utilizar para dibujarse. La clase `Canvas` define métodos para el dibujo de texto, líneas, mapas de bits, y muchas otras primitivas gráficas. Se pueden utilizar estos métodos en `onDraw()` para crear la interfaz de usuario personalizada (IU). Antes de llamar a cualquier método de dibujo, es necesario crear un objeto `Paint`

El ambiente `android.graphics` divide el dibujo en dos áreas:

- Qué dibujar, a cargo del `Canvas`.
- Cómo dibujar, a cargo de `Paint`.

Por tanto, el `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

Ejercicio 1. Dibujar figuras básicas en el centro de cada cuadrante.

La implementación de la practica es realmente sencilla, declaramos un objeto pant que será donde pintaremos y todo eso será agregado a nuestro canvas.

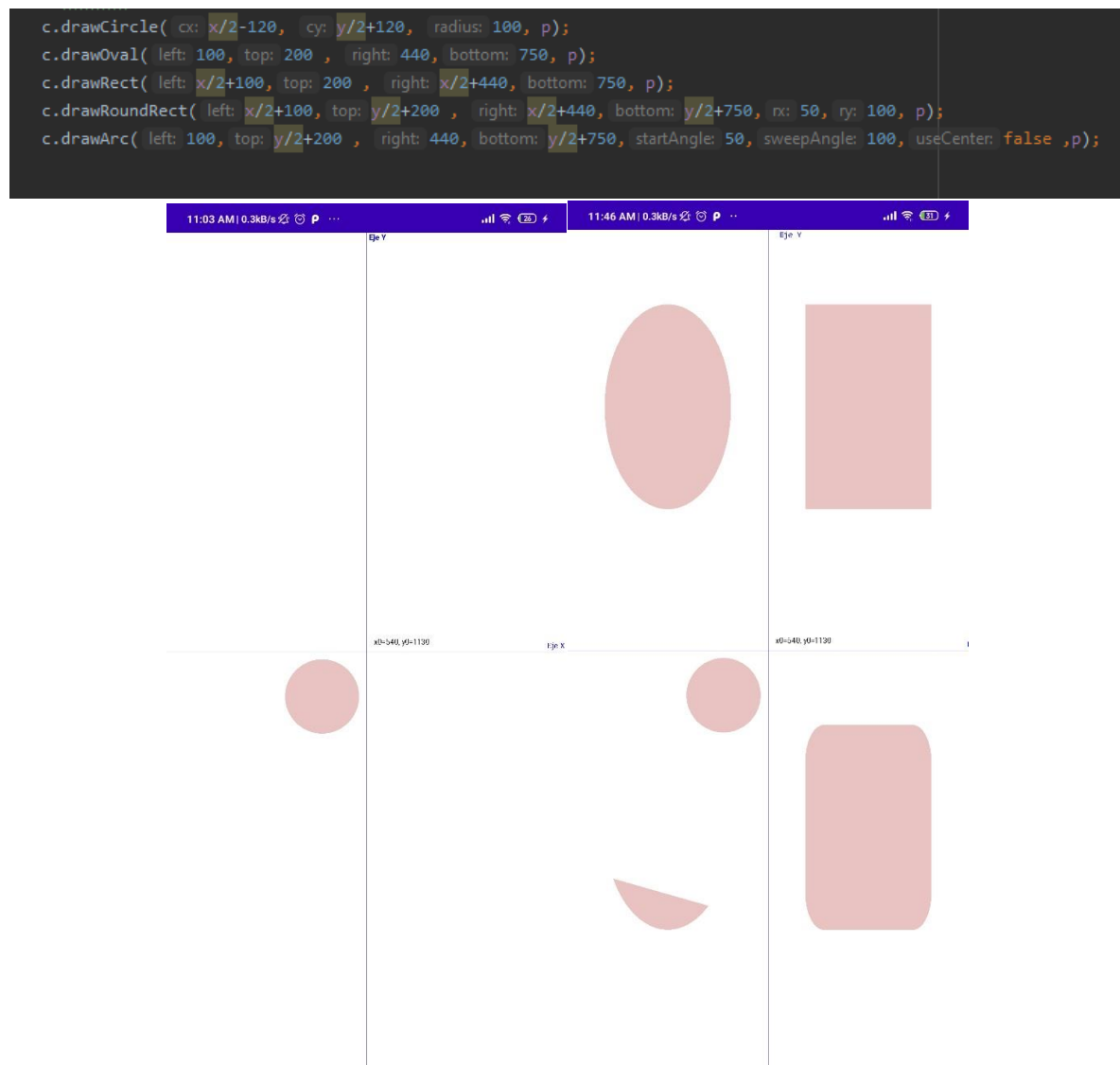
Dibujaremos el color del fondo, nuestro texto del centro, los ejes, y finalmente procederemos a dibujar las figuras como se muestra en el siguiente Código.

Drawcircle para nuestro circulo

Drawrec para el rectángulo

DrawRoundRec para el rectángulo redondeado

DrawArc para el arco



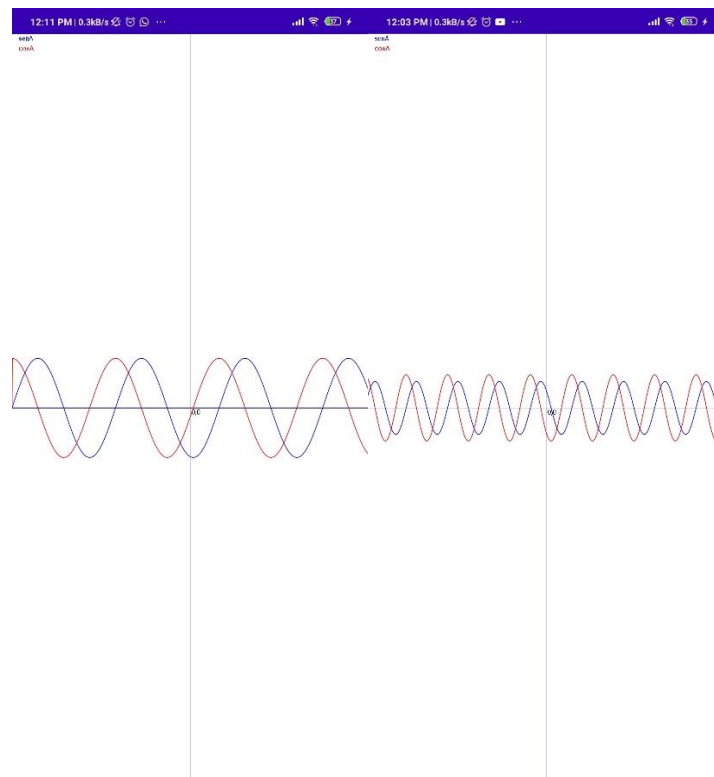
Ejercicio 2.

Dibujar líneas senoidales y cosenoidales.

En este ejercicio se realizaron 2 ondas, esto gracias a la función `lineTo`, que traza un camino desde 1 hasta `x`, donde `x` es el ancho de nuestra pantalla, el ejercicio a realizar era cambiar la amplitud y la frecuencia.

En las siguientes imágenes podemos ver el código mencionado y las salidas.

```
r.moveTo( x: 0, y: 0); p.setColor(Color.BLUE);  
for(int i=1; i<x; i++) r.lineTo(i, y: (float) Math.sin(i / 50f) * (-150f));  
r.offset( dx: 0, y0); c.drawPath(r, p);  
r = new Path();  
r.moveTo( x: 0, y: 0); p.setColor(Color.RED);  
for(int i=1; i<x; i++) r.lineTo(i, y: (float) Math.cos(i / 50f) * (-150f));  
r.offset( dx: 0, y0);  
c.drawPath(r, p);
```



Conclusión.

En este caso no tuve ningún problema con la realización, ya que prácticamente es igual que en los midlets o en java "normal", de la misma manera no hubo problema respecto al código del pdf o con los ejercicios pedidos ya que la documentación estaba muy clara respecto a las diferentes formas.

Referencias

<https://developer.android.com/reference/android/graphics/Canvas>