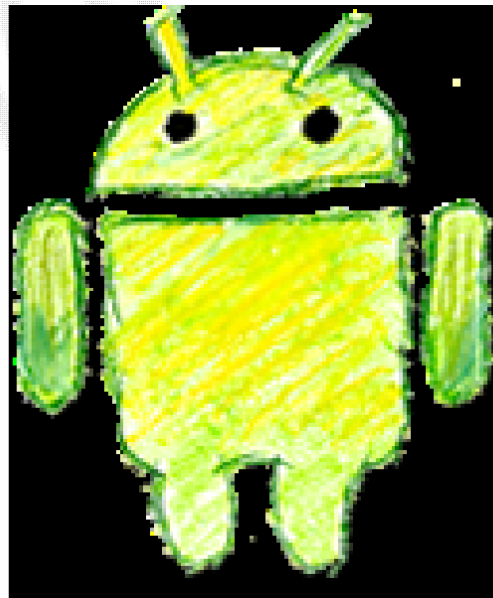


# Basic 2D Graphics in Android



# Outline

## 1. Android Graphics Programming

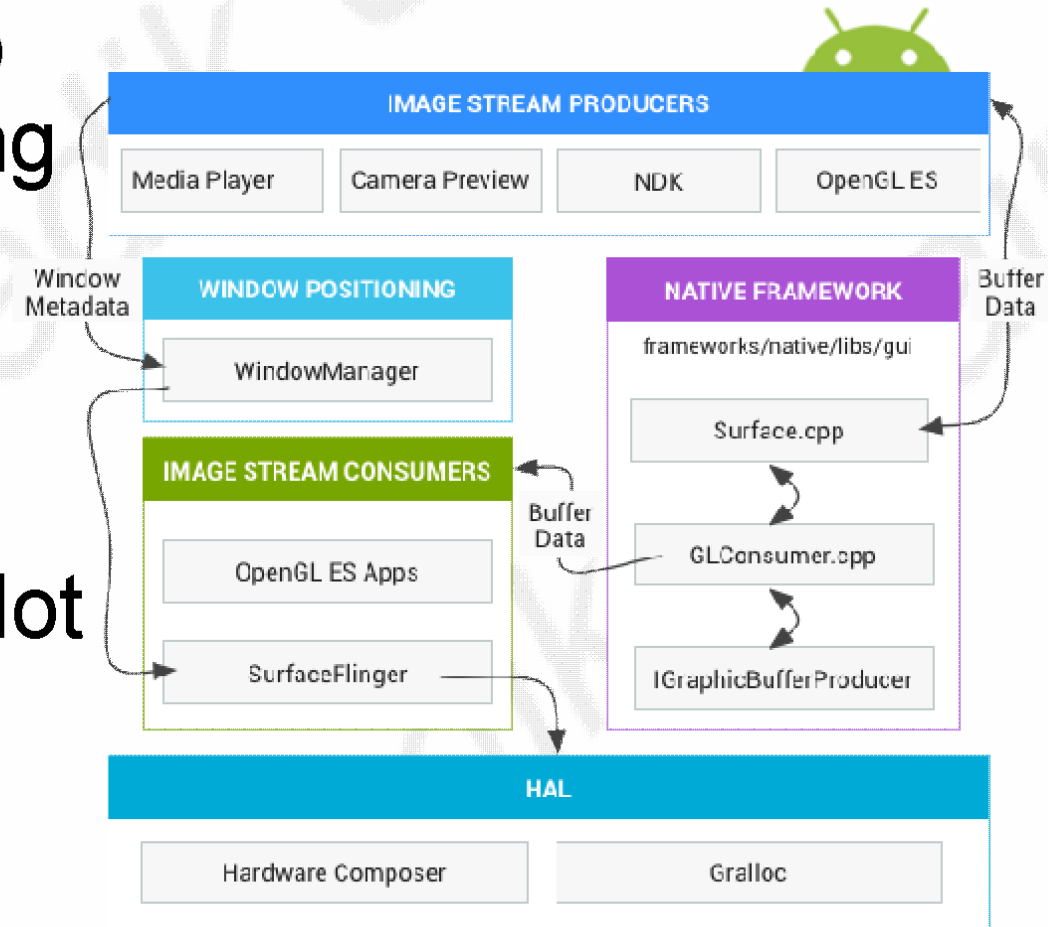
1. Drawing on a View
2. Drawing Rectangle on Canvas

## 2. Digital Image

1. BMP- Bitmap
2. RGB to Grayscale Conversion

# 1. Android Graphics Programming

- There are many ways to do graphics programming in Android
  - 2D vs. 3D
  - static vs. dynamic
- Many of them require a lot of knowledge of the underlying graphics libraries.



# 1.1 Drawing on a View

- Visible elements in an Android UI are called **Views**
- Each View has an associated **Canvas**
- When the View is shown, its ***onDraw*** method is automatically called by Android
- It uses the Canvas to render the different things it wants to display
- We can create our own View with our own ***onDraw*** method to display basic objects using the Canvas

# Canvas and Paint

- **Canvas** has methods for drawing Arcs, Bitmaps, Circles, Lines, Ovals, Paths, Rectangles, etc.
- Also methods to rotate, scale, skew, translate
- **Paint** has methods for setting the alpha, color, shade, stroke, etc.

# Let's Create a New Project!

- In Eclipse, go to File → New → Project
- Then select “Android Project”
- Name the project “**DrawingOnView**”
- Next, create your own custom View class inside the public Class **MainActivity**



# Creating Your Own View Class

1. Create a new Java class that extends View
2. Implement the necessary constructors
3. Implement the *onDraw* method and use the Canvas parameter to draw using a Paint object
4. Add your View to the application's Layout

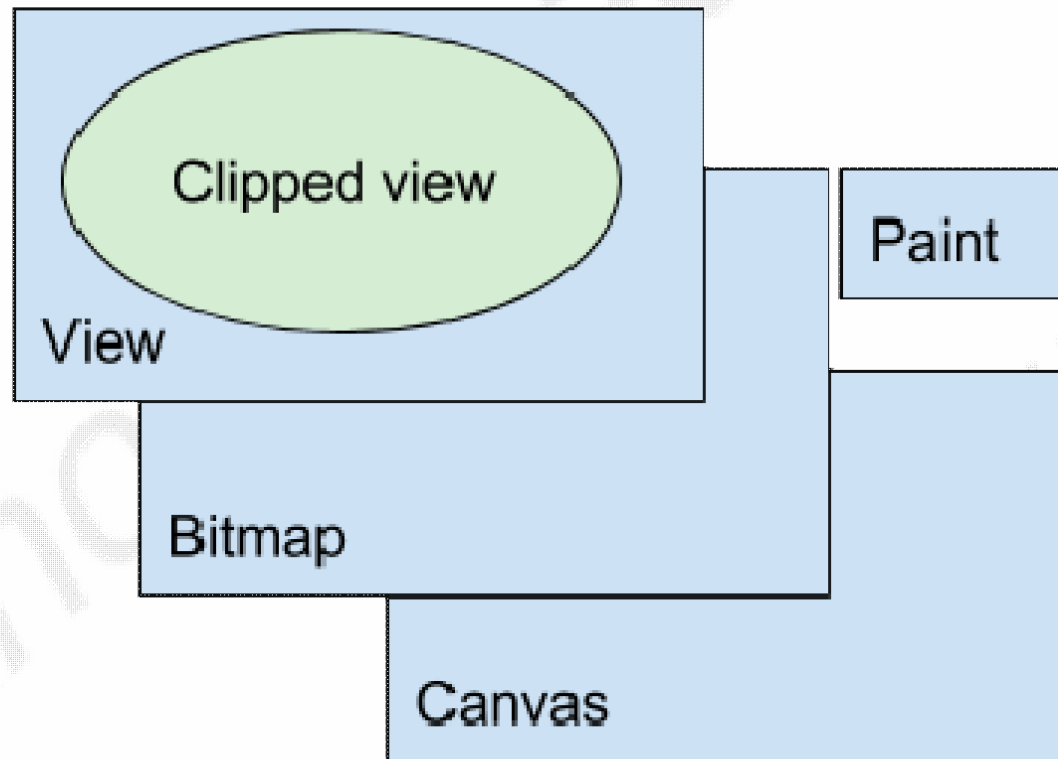
## 1.2. Drawing Rectangle on Canvas

- To draw dynamic 2D graphics where in your application needs to regularly re draw itself, drawing on a canvas is a better option. A Canvas works for you as an interface, to the actual surface upon which your graphics will be drawn.
- If you need to create a new Canvas, then you must define the bitmap upon which drawing will actually be performed. The Bitmap is always required for a Canvas.
- The below example explains to draw a rectangle as explained.



# Android Objects

- The figure below shows all the pieces required to draw to a canvas.



## 2. Digital Image

### 1. Colour

- CMYK (Cyan, Magenta, Yellow, Black)
- RGB (Red, Green, Blue)
- Grayscale (0-255)
- BW (Black White)

### 2. Graphic Resolution

Describes the amount of detail of the stored image. Resolution is often used as the number of pixels in digital image imaging.

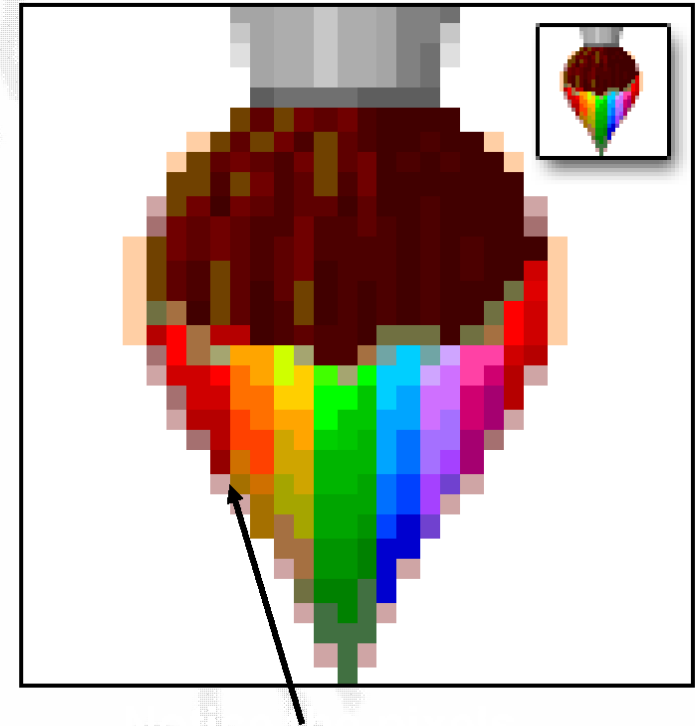
# Raster Graphics

Also called **bitmap** graphics

Consist of grids of tiny dots called pixels

Have a fixed resolution and cannot be resized without altering image quality

Edited in paint programs



# Common Raster Formats



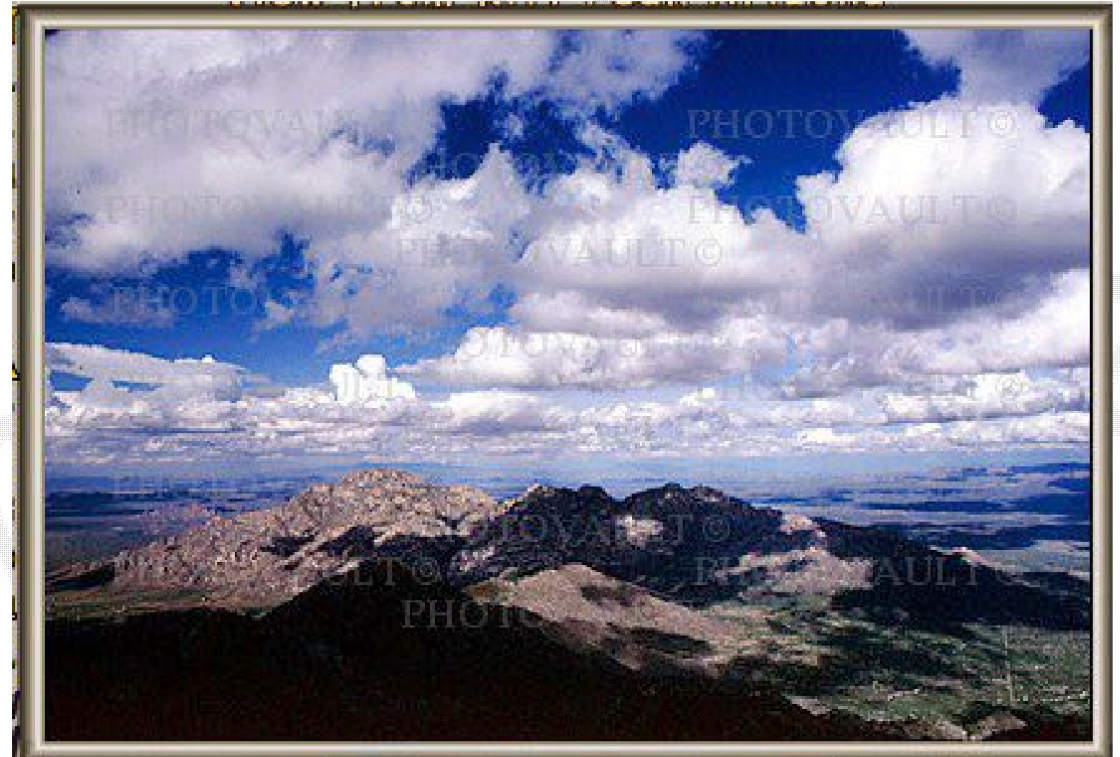
GIF

JPEG

BMP

PNG

TIFF



## 2.1 BMP - Bitmap

- why BMP file --
  - *Simplicity*
  - *Highly standardized*
  - *Extremely widespread*
- BMP file created by Microsoft and IBM 1986 (Windows BMP and Device Independent Bitmap)
- Different BMP format because of different bits stored per pixel. Such as 4bits(16 color bitmap), 8bits(256 color), 16bits(high color bitmap), 24bits(true color), 32bits etc.

# The Pixel Data

- In the 8-bit format, each pixel is represented by a single byte of data, that byte is an index into the Color Table. In the 24-bit format, each pixel is represented by three consecutive bytes of data that specify the blue, green, and red component values respectively.
- The pixel data in a BMP file is scan line padded to a 32-bit (4-byte) boundary. What this means is that if you have a 1071x363 24-bit image, that each scan line (one row of data in the image) consists of 363 pixels each of which requires 3-bytes (24-bits) to encode. Thus you have 1,089 bytes of data per line. The format requires that scan lines be multiples of 4-bytes so 3 null bytes (value is zero) are added to the end of the data for each line to make a total of 1,092 (4x273) bytes per line.



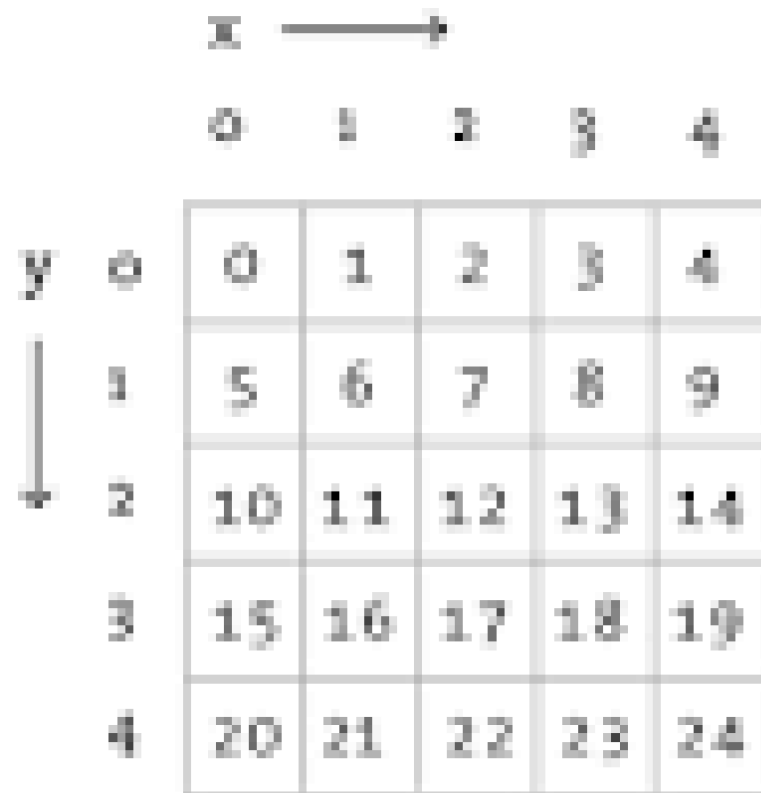
## 2.2 RGB to Grayscale Conversion

- Now we will convert an color image into a grayscale image. There are two methods to convert it. Both has their own merits and demerits. The methods are:
  - **Average method**
  - Weighted method or luminosity method
- Average method is the most simple one. You just have to take the average of three colors. Since its an RGB image, so it means that you have add r with g with b and then divide it by 3 to get your desired grayscale image.

# Average Method

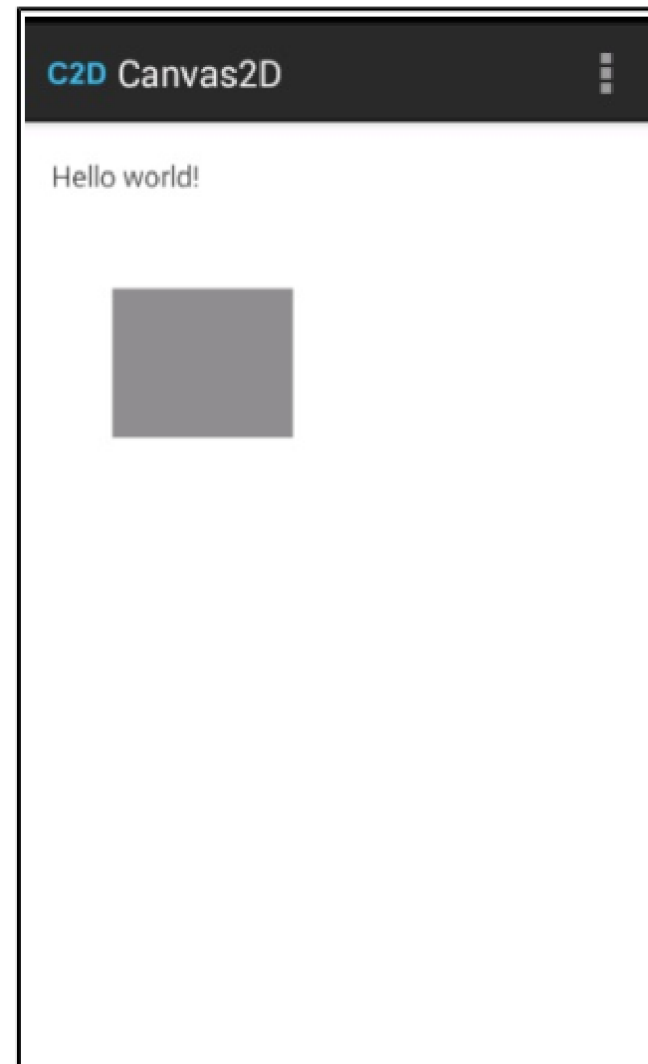
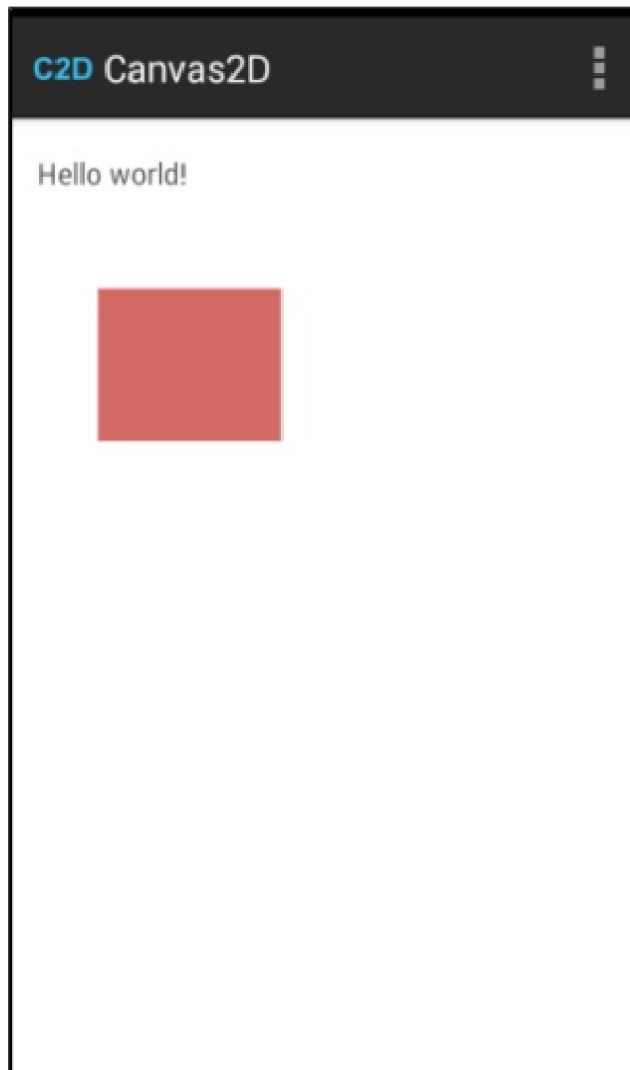
- Its done in this way.

$$\text{Grayscale} = (R + G + B / 3)$$



		$x \longrightarrow$				
		0	1	2	3	4
$y \downarrow$	0	0	1	2	3	4
	1	5	6	7	8	9
	2	10	11	12	13	14
	3	15	16	17	18	19
	4	20	21	22	23	24

# Perubahan Warna



**- Thank You -**

## **Referensi:**

Erwin S., Pemrograman android dengan menggunakan eclipse & staruml, 2018, bab 10.