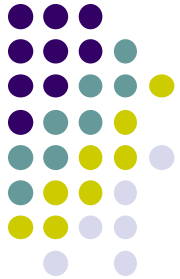
Mobile 3D Graphics

Introduction to Android Canvas

Android Canvas

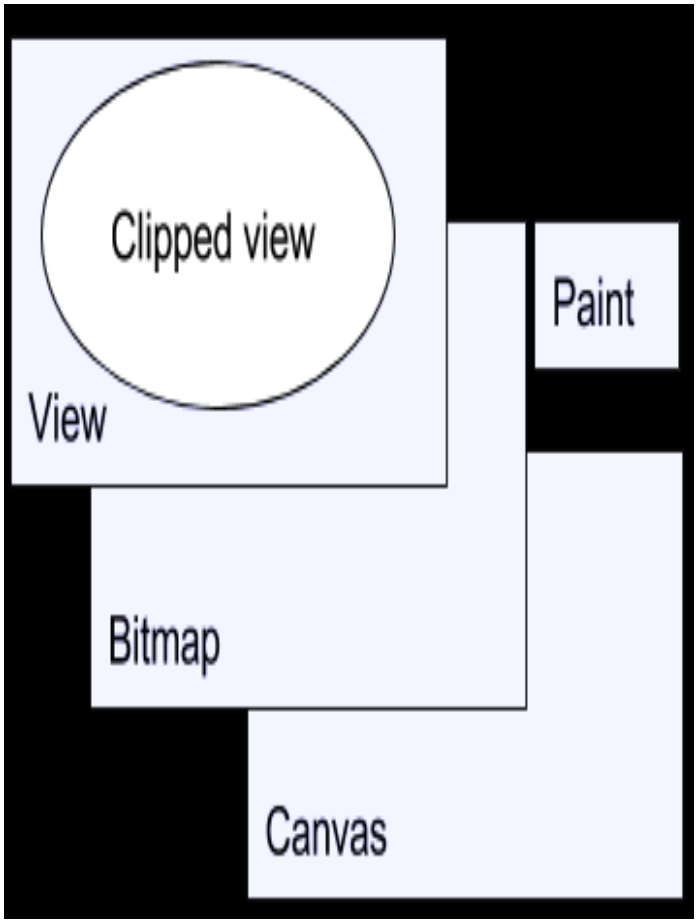


- Framework of android provides **2D APIs** which we can render on to screen.
- For **2D graphics** we usually opt for any of the two following options:

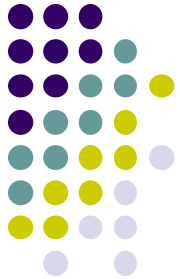
1. **Graphics** or **animation** object is drawn into View object from layout.
2. We can draw graphics **directly** onto the **canvas**.

Android Canvas class

- **Android Canvas class** encapsulates the **bitmaps** used as **surface**. It exposes the **draw methods** which can be used for designing.
- **Let us first clear the following terms:**
  - **Bitmap**: The **surface** being drawn on.
  - **Paint: It** lets us specify how to draw the **primitives** on bitmap. It is also referred to as "**Brush**".
  - **Canvas**: It supplies the **draw methods** used to **draw primitives** on underlying **bitmap**.

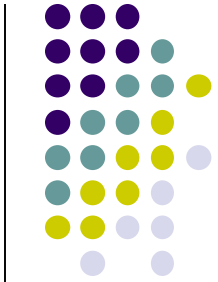


The Paint class



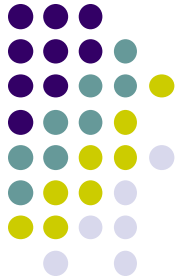
- For drawing on the **Canvas** object you use an object of type **Paint**.
- The **Paint class** allows to specify the **color**, **font** and certain effects for the drawing operation.
- The **setStyle() method** allows to specify how it should be drawn. Option are to paint:
  - Only the **outline** (**Paint.Style.STROKE**)
  - the **filled part** (**Paint.Style.FILL**)
  - **both** the outline and the filled part (**Paint.Style.STROKE_AND_FILL**)
- You can set the alpha channel of the **Paint** via the **setAlpha()** method.
- Via ***Shaders*** you can define that the Paint is filled with more than one color.

drawing objects



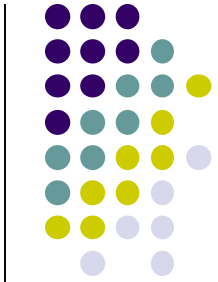
1. **drawArc**: draws an arc between the two angles bounded by an area of rectangle.
2. **drawBitmap:** It draws an bitmap on canvas.
3. **drawRGB/drawARGB/drawColor**: This fills the canvas with a single color.
4. **drawBitmapMesh:** It draws a bitmap using a mesh.
5. **drawCircle:** draws a circle on a specified radius centered on a given point.
6. **drawLine(s):**it draws a line (or series of lines) between points.

drawing objects



7. **drawOval:** it draws an oval which is bounded by the area of rectangle.
8. **drawPaint**: It fills the entire canvas with a specific paint.
9. **drawPath**: It draws a path as per specification.
10. **drawPicture:** It draws a picture specified on a rectangular area.
11. **drawPosText:** it draws a text string specifying the offset of each character.
12. **drawRect:** It draws a rectangle.
13. **drawRoundRect:** it draws a rectangle with round edges.
14. **drawText:** It draws a text string on canvas.

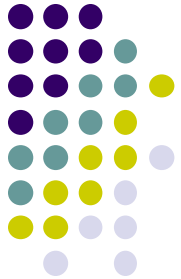
how to draw on Canvas in Android



- There are **2 ways** to draw things on a canvas in Android

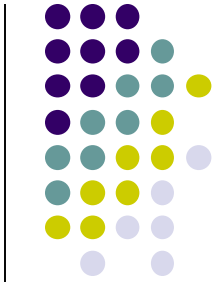
1. draw on a **View**,
2. draw on a **SurfaceView**.

Drawing on a View



- **more suitable** when your application does not need to update the **screen really fast**.
- It can be used for a game of **chess** or a similar slow-paced application.
- The Android system provides a **Canvas object** that will display our graphics in a **View** which lets you take those **Drawables** and pass them onto the **Canvas** to be drawn there.

draw using the SurfaceView class



- This **subclass** of **View** lets us create a dedicated drawing surface inside a View,
- This surface is offered a separate thread where all the drawing happens
- you need to create a **custom class**. extends **SurfaceView**
- It should also **implement SurfaceHolder.Callback** class. used to inform you when the underlying Surface dispatches an event, **for example**, when it is **created** or **changed**.

References