

2D Graphics with Canvas on SurfaceView

2D Canvas on SurfaceView

- SurfaceView is a special subclass of View that offers a dedicated drawing surface within the View hierarchy
 - Offers this drawing surface to an application's secondary thread, so that the application isn't required to wait until the system's View hierarchy is ready to draw
 - Instead, a secondary thread that has reference to a SurfaceView can draw to its own Canvas at its own pace.

2D Canvas on SurfaceView

- In your code extend the SurfaceView class
- Need to implement the SurfaceHolder.Callback
- Also the interface requires three additional methods to be implemented: surfaceCreated(), surfaceChanged(), and surfaceDestroyed()
 - Important so that you know when you can start drawing, whether you need to make adjustments based on new surface properties, and when to stop drawing and potentially kill some tasks.

2D Canvas on SurfaceView

- The surface object is handled via a SurfaceHolder which you get when you initialize the SurfaceView, by calling `getHolder()`
- Inform the SurfaceHolder that you will handle the callbacks by calling `holder.addCallback(this)`. Then override the Callback methods within your SurfaceView: `surfaceCreated()`, `surfaceChanged()`, and `surfaceDestroyed()`
- Create a secondary thread that will control all the drawings to the surface, and pass it the SurfaceHolder
 - Within the thread get hold of the canvas: `surfaceHolder.lockCanvas()`
 - Then make modifications to the canvas
 - Finally, `surfaceHolder.unlockCanvasAndPost(canvas)`

Shooting Game with Music Exercise

- Off to our next exercise
 - Extending the Shooting Game with background music
 - Using the MediaPlayer Class
 - Initializing the player
 - Controlling playback using MediaPlayer methods
 - Menu items in the toolbar/actionbar