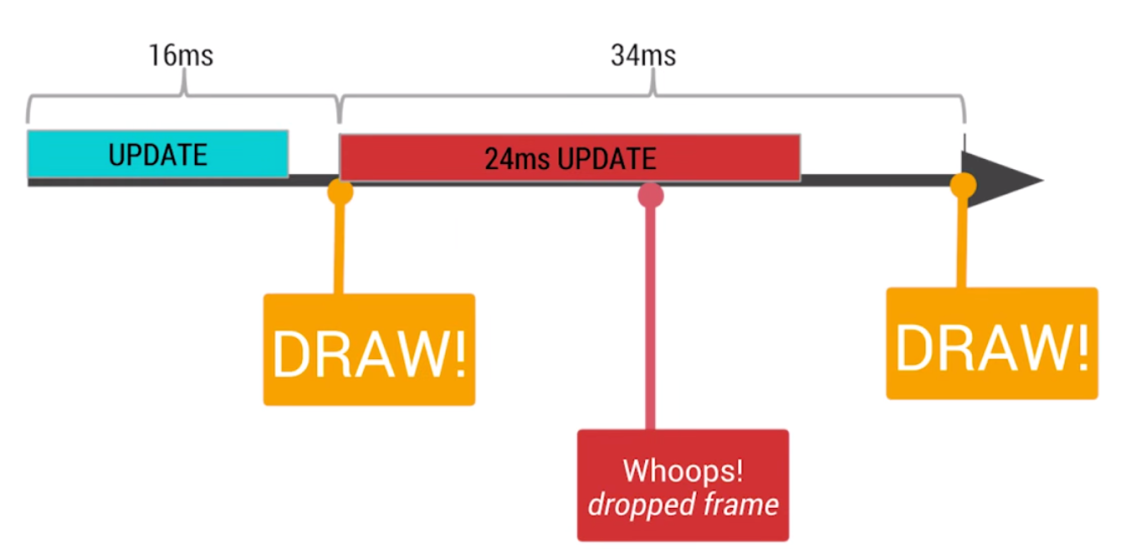
Android渲染机制

Android系统每隔16ms发出VSYNC信号，触发对UI进行渲染，那么整个过程如果保证在16ms以内就能达到一个流畅的画面。那么如果操作超过了16ms就会发生下面的情况：



如果系统发生的VSYNC信号，而此时无法进行渲染，还在做别的操作，那么就会导致丢帧的现象，（大家在察觉到APP卡顿的时候，可以看看logcat控制太，会有drop frames类似的警告）。这样的话，绘制就会在下一个16ms的时候才进行绘制，即使只丢一帧，用户也会发现卡顿

卡顿的原因就在于没有办法在16ms内完成该完成的操作，而主要因素是在于没有必要的layouts、invalidations、Overdraw

通过Hierarchy Viewer去检测渲染效率，去除不必要的嵌套

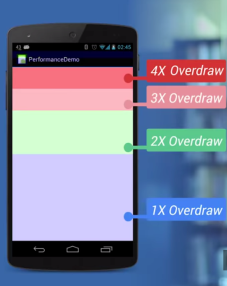
通过Show GPU Overdraw去检测Overdraw，最终可以通过移除不必要的背景以及使用canvas.clipRect解决大多数问题。

Overdraw的检测

对于性能优化，那么首先肯定是去发现问题，那么对么overdraw这个问题，还是比较容易发现的。

按照以下步骤打开Show GPU Overrdraw的选项：

设置 -> 开发者选项 -> 调试GPU过度绘制 -> 显示GPU过度绘制



Overdraw 的处理方案一：移除不必要的background

Overdraw 的处理方案二：clipRect的妙用

View已经有了不透明的背景，完全可以移除Window的背景了，即在Activity中，添加getWindow().setBackgroundDrawable(null);