内存

1. 申请内存需要控制大小，尤其是高分辨率bitmap和大图resources

缓存要有严格的上限控制。

确实需要额外内存，native申请，绕过虚拟机

2. 将activity作为context，被静态类静态引用后无法释放

3. 线程未退出，对相关对象的引用不能释放

4. 学会用mat看java的内存泄漏，mat也可以用来查询native（C,C++）的内存泄漏问题

界面

1. 尽量减少view层级，使用merge减少view层次

```
<merge xmlns:android="http://schemas.android.com/apk/res/android">
<ImageView
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:scaleType="center"
android:src="@drawable/golden_gate" />
</merge>
```

2. ViewStub，暂时不用的view，可以用viewstub占位，方便且高效

3. view以及各png的id最好唯一，以前android有bug，循环溢出，容易解析失败

界面2

人眼很挑剔，想要流畅必须50fps，4ms画界面，16ms处理任务

1. ondraw函数千万别做耗时函数，例如在ondraw里给textview setbackground，ondraw函数里也不许有业务逻辑，我见过用ondraw来计算位置，和动画时间的

2. 切记：主线程不能执行长时间操作，要放到一个非主线程处理，同时也防止anr，写代码时一定要注意上下文

3.单核手机上，即使非主线程处理，也要谨慎看是否能避开cpu的高峰，否则界面一样卡，多核手机上cpu占用上60%也会卡

界面3

1. listveiw，adapter getview一定要用viewholder，基本概念请掌握
2. 刷新界面必须在主线程，或者postinvalidate，opengl必须在render线程，否则会报错或者各种诡异问题
3. inflate是个耗时操作，尽量避免，relativelayout比linearlayout快

其它

1. 线程同步：Synchronized用法不当导致ANR，wait方法使用不当导致ANR
2. 数据库db操作，不能多线程写
3. 数据库db操作，transaction的使用
4. 有Exception抛出时，catch 后一定要有log输出，否则后续有异常很难查找，outofmemory的父类不是Exception，所以抓Exception是涵盖不了OOM的

深度优化，提高技术水平

- 1. 了解unix/linux编程，熟习posix，利用native优化，比如java访问文件慢，用c实现
- 2. arm neon simd并行指令集
- 3. 了解GPU，利用GPU协助运算（OPENGL 2.0）

深度优化，提高技术水平二

1. 了解dalvik虚拟机，知道各种GC的触发条件

   GC_FOR_MALLOC: Occurs when the heap is too full to allocate memory, and memory must be reclaimed before the allocation can proceed
   GC_CONCURRENT: Occurs when a (possibly partial) collection kicks in, usually as there are enough objects to reclaim

2. Hierarchy Viewer和TraceVIew工具的使用

3. 学习《android应用性能优化》