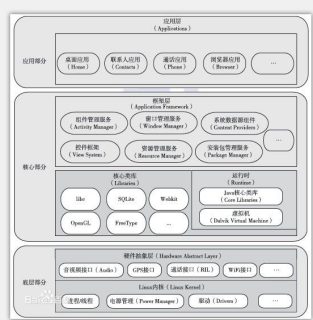
1.Android的系统架构图



1. dalvik与ART 前者是运行时编译，ART是安装时就进行编译。
2. Makefile机制:组合文件，进行自动的可控制的编译
3. ADB命令行：安装apk：  
   adb install 路径  
     
   向手机传输文件：  
   adb push C:/WorkSpace/aa.txt /storage/emulated/0/AmapSdk

手机向电脑传输文件：  
adb pull /storage/emulated/0/AmapSdk C:/WorkSpace/aa.txt  
  
录制视频：  
adb shell screenrecord /storage/emulated/0/AmapSdk/test.mp4  
  
查看系统盘符：  
adb shell df

1. View控件的测量 onMeasure：

三种模式：

EXACTLY,精确值模式，100dp或者match\_parent，默认即为EXACTLY；

AT\_MOST,自适应模式，warp\_content,需要用setMeasureDimension(width,height)设置算好的宽高；

UNSPECIFIED，想要多大即多大，所以可以超过屏幕尺寸，显示不下的在屏幕之外；目前没用过。

1. 自定义View中比较重要的回调方法：
   1. onFinishInflate()：从XML加载组件后回调；
   2. onSizeChanged()：组件大小改变时回调；
   3. onMeasure()：测量view宽高
   4. onLayout()：控件内的子控件的显示位置，自定义ViewGroup时用
   5. onTouchEvent()：触摸事件
2. 触摸事件机制：

触摸方法回调有：默认返回false

disPatchTouchEvent(MotionEvent e)

onInterceptTouchEvent(MotionEvent e)， ViewGroup独有

onTouchEvent(MotionEvent e)

触摸按钮3时的回调机制：

1.disPatch -> 1.intercept ->2.disPatch -> 2.intercept ->3.disPatch ->

3.touch -> 2.touch -> 1.touch

1. 屏幕的坐标原点是左上角，向右是X轴正方向，向下是Y轴正方向
2. 获取坐标值的各种方法

View中提供的获取坐标方法：

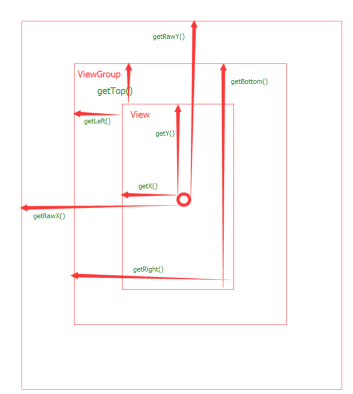
getLeft()与getRight()都是View自身到父布局左边的距离

getTop()与getBottom()都是View自身到父布局顶边的距离

MotionEvent中的方法：

GetX()与getY()是视图坐标，即是触摸点到它自身左边与顶边的距离

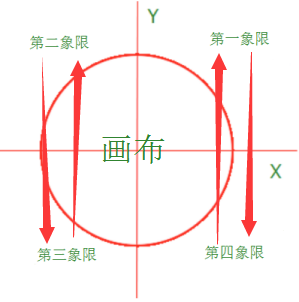
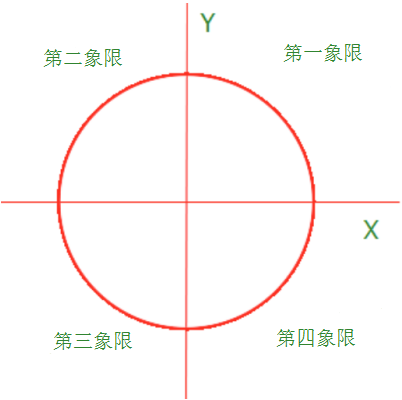
GetRawX()与getRawY()是绝对坐标，即是触摸点到屏幕左边与顶边的距离



1. File文件所在文件夹存在时，写入数据时自动创建；不存在，则需调用mkdis(),当父类的父类也不存在是应调用mkdirs(),创建父类文件夹。
2. MarginLayoutParams是父类
3. scrollBy()与scrollTo，坐标系
4. ViewDragHelper 滑动控制类，子类有DrawerLayout与SlidingPaneLayout
5. Paint的重要属性：TextPaint.getFontMetricsInt()
   1. setAntialias() 抗锯齿，
   2. setColor(),setARGB(),setAlpha(),setTextSize(),setStyle(),setStrokeWidth().

**textHeight**=**paint**.descent()-**paint**.ascent()

* 1. 基准线是baseline
  2. ascent：字体在baseline上方被推荐的距离（一些字体制作商需要参考这个）
  3. descent：字体在是baseline下方被推荐的距离（一些字体制作商需要参考这个）
  4. top：ascent的最大值
  5. bottom：descent的最大值
  6. getTextBounds(String text, int start, int end, Rect bounds):获取文字宽高
  7. measureTest(text)：获取文字宽度
  8. ColorMatrix
  9. PorterDuffXfermode（不好用），
  10. Shader 着色器 渲染器：（TileMode三中模式：CLAMP拉伸；REPEAT重复；MIRROR镜像）
      1. BitmapShader：位图
      2. LinearGradient：线性
      3. RadialGradient：光束
      4. SweepGradient：梯度
      5. ComposeShader：混合
  11. PathEffect 画线用，结合Path的lineTo来使用
      1. CornerPathEffect：拐角圆滑
      2. DiscretePathEffect：相当于画了一条毛茸茸的线
      3. DashPathEffect：画虚线
      4. PathDashPathEffect：升级的画虚线
      5. ComPosePathEffect与SunPathEffect：组合前几者

1. 视图动画：Animation，有TranslateAnimation，ScaleAnimation，RatateAnimation，AlphaAnimation，用AnimationSet可以把四个视图动画组合在一起
2. 属性动画：ValueAnimator ObjectAnimator，AnimatorSet组合，动画属性值：
   1. translationX、translationY，增量平移
   2. rotation、rotationX、rotationY 旋转
   3. pivotX、pivotY 没看出和rotation有什么区别
   4. scaleX、scaleY 缩放
   5. x、y 最终位置
   6. alpha 透明度 默认为1，不透明，0代表完全透明，即不可见
   7. PropertyValuesHolder，ObjectAnimator.ofPropertyValuesHolder(view,pro...)
   8. Value的主要方法：addUpdateListener()
3. AnimatorSet属性 palyTogether() playSequentially() play().with()、befor()、after();来控制动画的播放规则
4. 布局动画：LayoutAnimationController MainAdapter中有使用
   1. LayoutAnimationController.ORDER\_NORMAL --顺序
   2. LayoutAnimationController.ORDER\_RANDOM --随机
   3. LayoutAnimationController.ORDER\_REVERSE --反序
5. SVG:VectorDrawable（vector静态效果）、AnimatedVectorDrawable（nimated-vector）：animated-vector组合objectAnimator与vector形成动态的动画效果
6. Canvas 方法：
   1. translate(x,y)：将原点(0,0)移动(x,y)
   2. rotate(a,x,y)：将画布以(x,y)为中心旋转a度
   3. save(),saveLayer(),saveLayerAlpha():保存画布当前状态，并返回int值，方便移除此状态
   4. restore():返回画布此次保存的初始状态
   5. restoreToCount():移除画布此次保存的状态
7. Math的三角函数：

**var angle** = 30.0*//角度***val radians** = **Math**.*PI* \* **angle** / 180*//转换成弧度***val ix** = 200 \* **Math**.cos(**radians**)  
**val iy** = 200 \* **Math**.sin(**radians**)

0-360 度，Cos取值范围1..0..-1..0..1，即+--+，Sin 取值范围0..1..0..-1..0,即++--

以x=Cos,y=Sin算为++，-+，--，+-，即一二三四象限，以X正轴为起点，逆时针方向画圆，画布上Y轴正负颠倒，所以是顺时针方向画圆

以x=Sin,y=Cos算为++，+-，--，-+，即一四三二象限，以Y正轴为起点，顺时针方向画圆，画布上Y轴正负颠倒，所以是逆时针方向画圆

**val ix** = 200 \* **Math**.sin(**radians**)  
**val iy** = 200 \* **Math**.cos(**radians**)