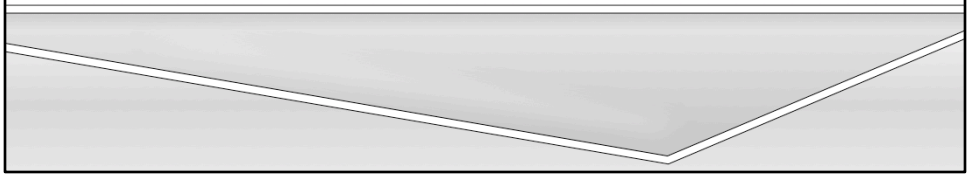
Please check for updated slides at:
https://github.com/desertjim/CustomViewGroups

LEVERAGING CUSTOM VIEWGROUPS

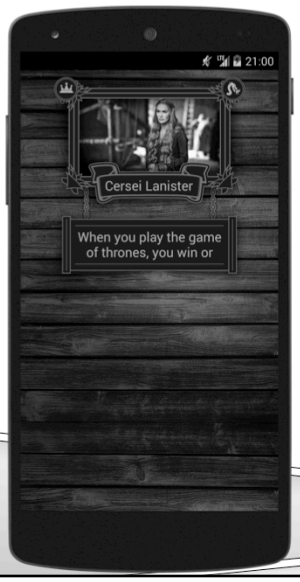
By James Baca

http://jamesbaca.net

Twitter:desertjim



SNEAK PEAK



- Not heavily nested
- Fulfills design requirements

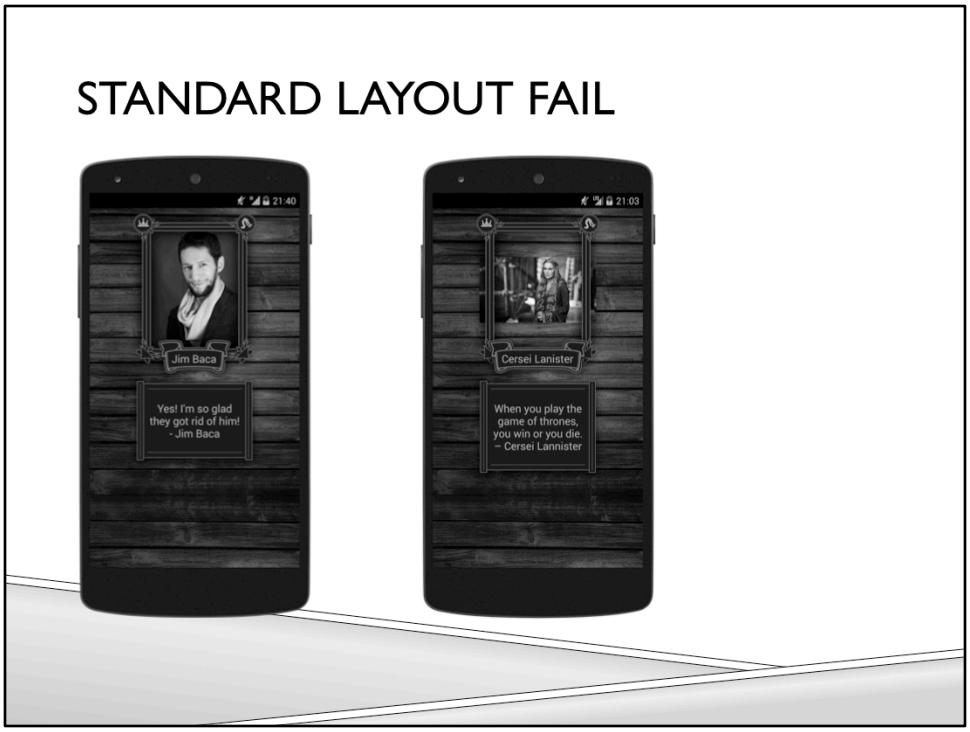
WHAT IS A VIEWGROUP?

- AKA Layout, Container
- LinearLayout
- RelativeLayout
- GridView
- ViewPager
- Absolute(worst)Layout *shudder*
- Anything that extends the ViewGroup

WHY CUSTOM VIEWGROUPS?

- Jank
- Maintenance
- Flexible
- Crazy Mad Designer ;)

STANDARD LAYOUT FAIL



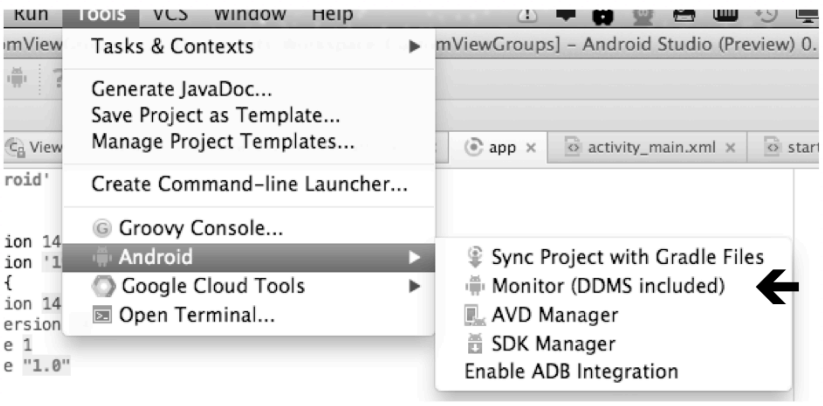
RECOMMENDATIONS

- Android Studio
- Hierarchy Viewer

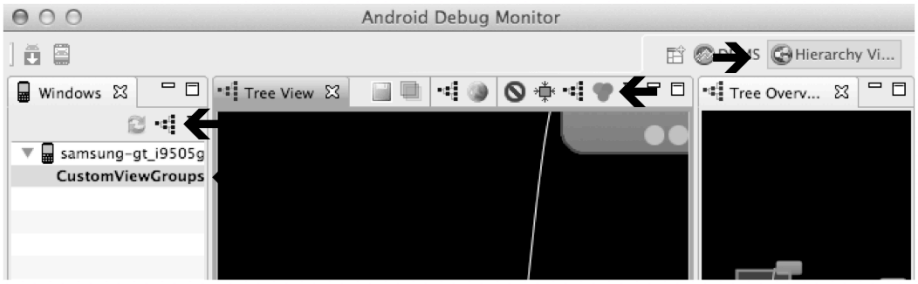
HIERARCHY VIEWER

- Where's my child?!
- Performance Analysis
- Developer Phones Just Work*
- Others: View Server

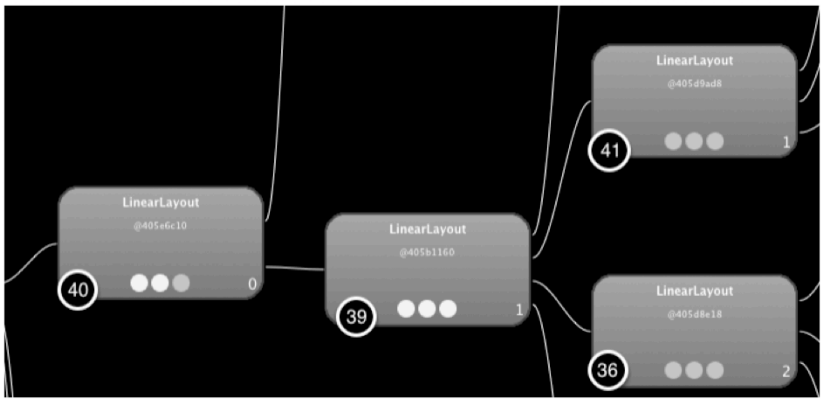
LOAD THE HIERARCHY VIEWER



LOAD THE HIERARCHY VIEWER



HIERARCHY VIEWER



WAY WAY SIMPLIFIED OVERVIEW

- Header Comments in View.java:

Category	Methods
Creation	Constructors
Layout	onMeasure(int, int)
Layout	onLayout(int,int,int,int)
Drawing	dispatchDraw(Canvas)

DECLARING CUSTOM VIEWGROUP

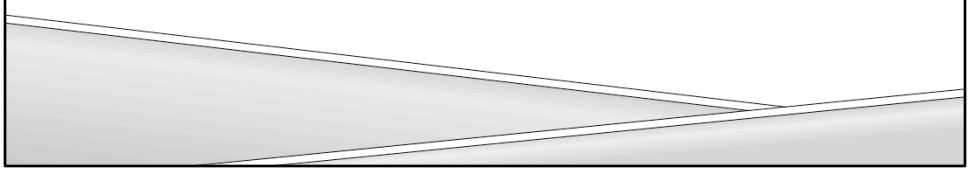
- Name space must match
- E.g. class com.example.ProfileLayout
  - <com.example.ProfileLayout …
- Use xml namespace to use your custom attributes
  - xmlns:profile="http://schemas.android.com/apk/res-auto"

CONSTRUCTORS

- View(Context)
- View(Context, AttributeSet)
- View(Context, AttributeSet, int)

ATTRIBUTE SET?!

- Collection of xml attributes and xml values
- Used for setting custom (layout) parameters

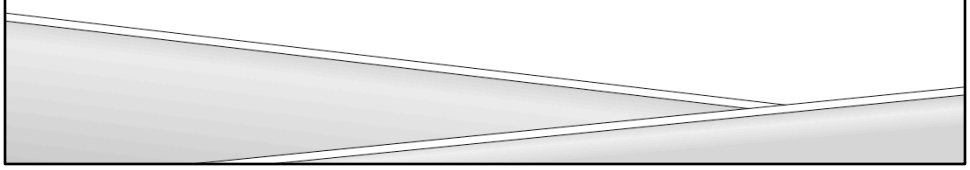


WHAT ARE LAYOUTPARAMS

- Directions
  - Size
  - Placement
- Common examples
  - Layout_width
  - Layout_height
  - In relative layout:
    - layout_centerHorizontal
    - layout_below

WHY CUSTOM LAYOUTPARAMS

- Reusable
- Simplification

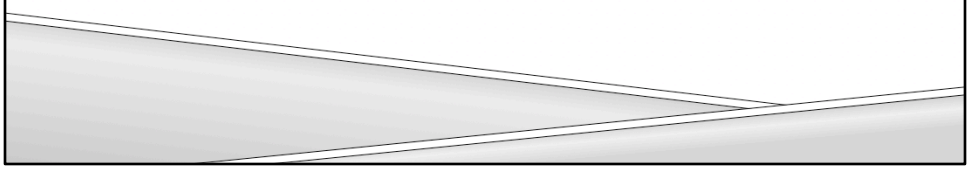


EXAMPLES OF LAYOUTPARAMS

- LinearLayout.LayoutParams
- ViewGroup.MarginLayoutParams
- RelativeLayout.LayoutParams

USING CUSTOM LAYOUT PARAMS

- Extend a Layout Params class
- Generate Extended Layout Params
- Do Something with Layout Params



CUSTOM XML ATTRIBUTES

- Reusable
- Measure Directions
- Layout Directions
- Heck Even Draw Directions

CREATING CUSTOM XML ATTRIBUTES

- res/values/attrs.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="MySuperAwesomeLayout_LayoutParams">
        <attr name="layout_badgeCorner" format="Integer"/>
        <attr name="layout_alignTop" format="boolean"/>
        <attr name="hiddenText" format="string"/>
    </declare-styleable>
</resources>
```

READING ATTRIBUTES

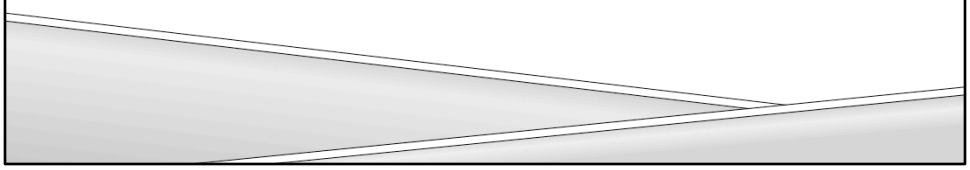
```
public static class MyAwesomeParams extends ViewGroup.MarginLayoutParams {
        int corner = 0;

        public MyAwesomeParams (Context context, AttributeSet attrs) {
            super(context, attrs);
            TypedArray a = context.obtainStyledAttributes(
                    attrs,
                    R.styleable.MyAwesome);
            corner = a.getInt(R.styleable.MyAwesome_layout_badge_corner, 0);
            a.recycle();
        }
}
```

GENERATE CUSTOM LAYOUTPARAMS

```
public ViewGroup.LayoutParams generateLayoutParams(AttributeSet attrs) {
        return new MyAwesomeLayout.LayoutParams(getContext(), attrs);
 }

boolean checkLayoutParams(ViewGroup.LayoutParams)

ViewGroup.LayoutParams generateLayoutParams(ViewGroup.LayoutParams p)

ViewGroup.LayoutParams generateDefaultLayoutParams() {
        return new LayoutParams();
}
```

USING CUSTOM LAYOUT PARAMS

- OnMeasure
- OnLayout
- OnDraw
- Other places

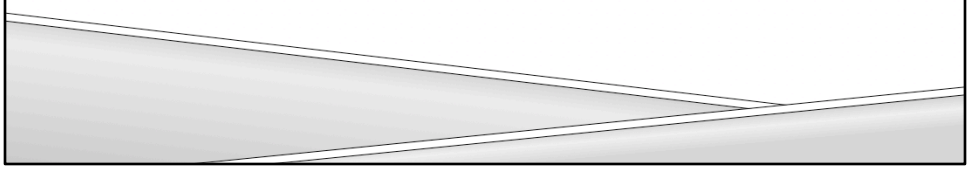


ON MEASURE

- Takes in measure specs(*2x)
- MeasureSpec modes:
  1. Unspecified
  2. AT Most
  3. Exactly
- MeasureSpec size in (raw)pixels
- Responsiblities(children, and self)
- Must call setMeasuredDimension(int, int)

MEASURING CHILDREN

- getChildCount()
- getChildAt(int)
- child.getLayoutParams()
- measureChild*(…) or child.measure(int, int)
- setMeasuredDimension(int, int)!!!
- Multipass but don't go crazy

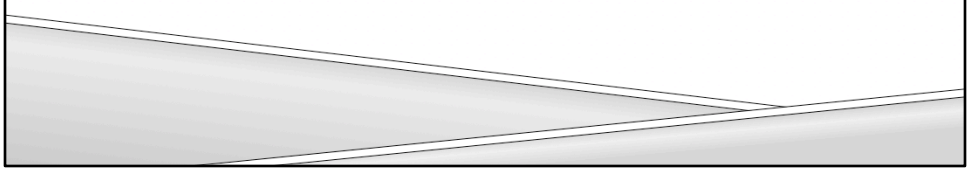


SETMEASUREDDIMENSION

```
protected final void ...setMeasuredDimension(int measuredWidth, int
measuredHeight) {

    mMeasuredWidth = measuredWidth;
    mMeasuredHeight = measuredHeight;
   mPrivateFlags |= MEASURED_DIMENSION_SET;
}
```

MEASURE

- Final method
- Where exception is thrown
- http://goo.gl/wfHQLi

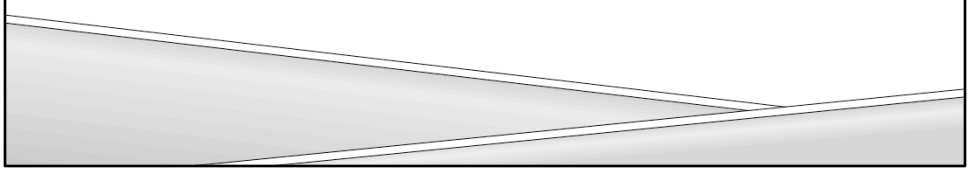


MEASURE CHILD HELPER FUNCTION

```
protected void measureChild(View child, int parentWidthMeasureSpec,
    int parentHeightMeasureSpec) {

    final LayoutParams lp = child.getLayoutParams();

    final int childWidthMeasureSpec = getChildMeasureSpec(parentWidthMeasureSpec,
                        mPaddingLeft + mPaddingRight, lp.width);

    final int childHeightMeasureSpec = getChildMeasureSpec(parentHeightMeasureSpec,
                        mPaddingTop + mPaddingBottom, lp.height);

    child.measure(childWidthMeasureSpec, childHeightMeasureSpec);
}
```

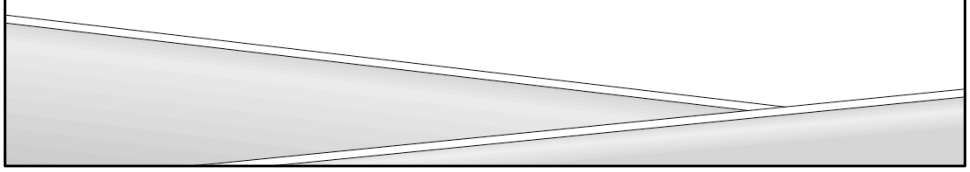
GETCHILDMEASURESPEC

- Hard worker function
- Not always what you want



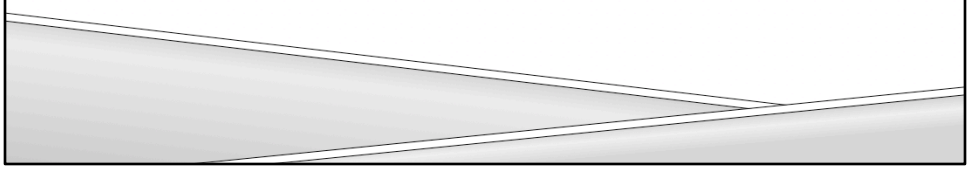
ROLL YOUR OWN MEASURESPEC

- Int = MeasureSpec.getMode(int)
- Int = MeasureSpec.getSize(int)
- Int = MeasureSpec.makeMeasureSpec(int, int)



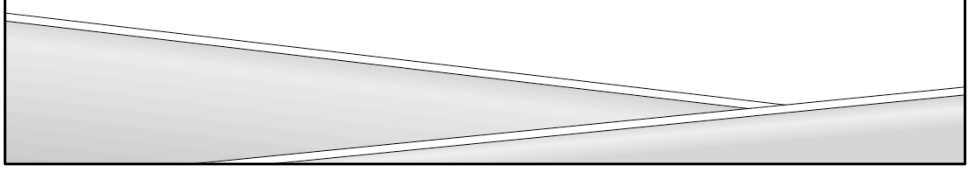
ON LAYOUT

- Where to place each child
- getChildCount
- getChildAt
- child.getLayoutParams()
- child.layout(int, int, int, int)



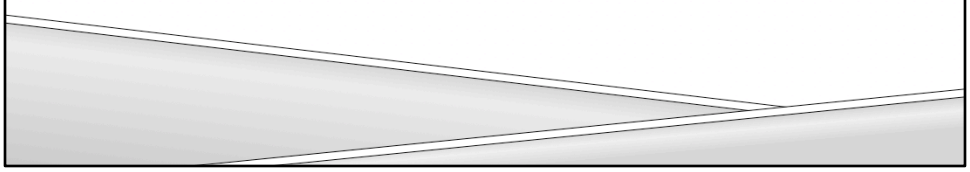
DISPATCH DRAW

- Draw each child



CUSTOM VIEWGROUP STRATEGERY

- Group by proximity
- Order by display



TASK #1

- Create xml views for the following items(with a custom viewgroup as the parent)
  - Crown image
  - Dragon image
  - Portrait
  - Frame
  - Chain 1
  - Chain 2
  - Caption Box
  - Name Scroll

TASK #2

- Create a custom attribute for the portrait
- If the attribute is present on the view center the view horizontally.
- Use measureChild in the onMeasure call for this view

TASK #3

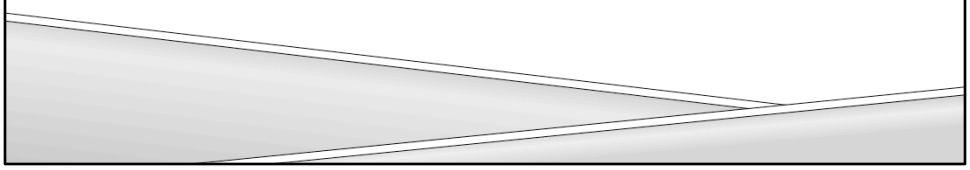
- Create an attribute for the frame
- If the frame attribute is present display it on top of the portrait

TASK #4

- Add custom attributes that define the amount of spacing outside the frames inner bounds area.
- Use these custom attributes to align the picture and the frame so that:
  1. The frame is the perfect size for the portrait
  2. The portrait is displayed in the portrait
- Use measurechild for the frame
  - Why doesn't measure child work? Hint use the debugger to step into ViewGroup.java

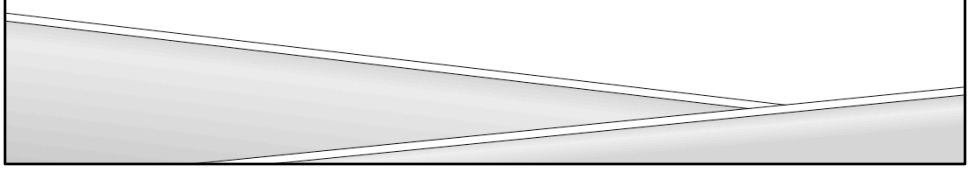
TASK #5

- Create an attribute for identifying the placement of the items that go on top of the frame
- Hint look at the android attrs.xml for some ideas how to accomplish this
- Use this attribute to display the crown in the top left of the frame



TASK #6

- Display one of the symbols on the top right



TASK #7

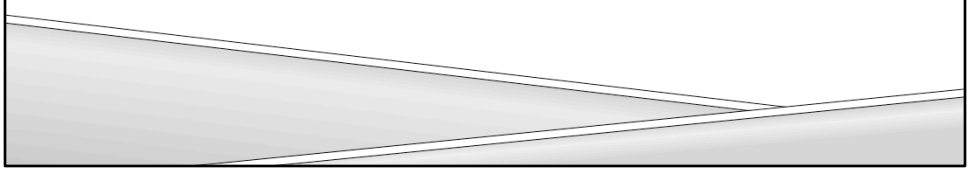
- Display the name scroll using the same custom attribute

TASK #8

- Display the quote TextView under the picture
- Make the TextView sides match the left and right sides
- Make the TextView appropriately sized(hint MeasureSpec.UNSPECIFIED)

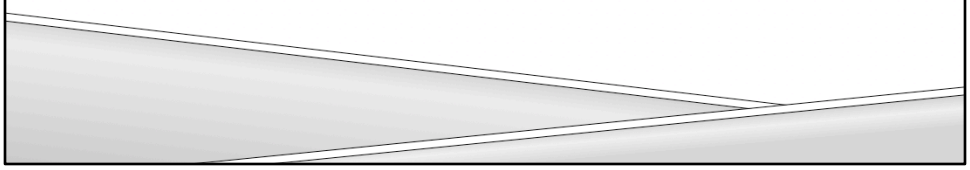
TASK #9

- Display the chains between the picture of the quote TextView
- Make sure that the frame is on top of the chains
- Make sure that the quote box is on top of the chains



TASK #10

- Swap only portrait source image for a landscape one
  - Make sure that it still displays correctly



FINAL THOUGHTS

- Look at the source code for built in layouts
- Hierarchy Viewer is invaluable