Material Animations

and other related stories

Agenda

- How Android draws
- Why 60 fps?
- Shadows
- Animations
- Activity Transitions
- Performance tools #perfmatters

David González

Technical Product Owner at Novoda

Google Developer Expert for Android

@dggonzalez

+DavidGonzalezMalmstein

malmstein

novoda

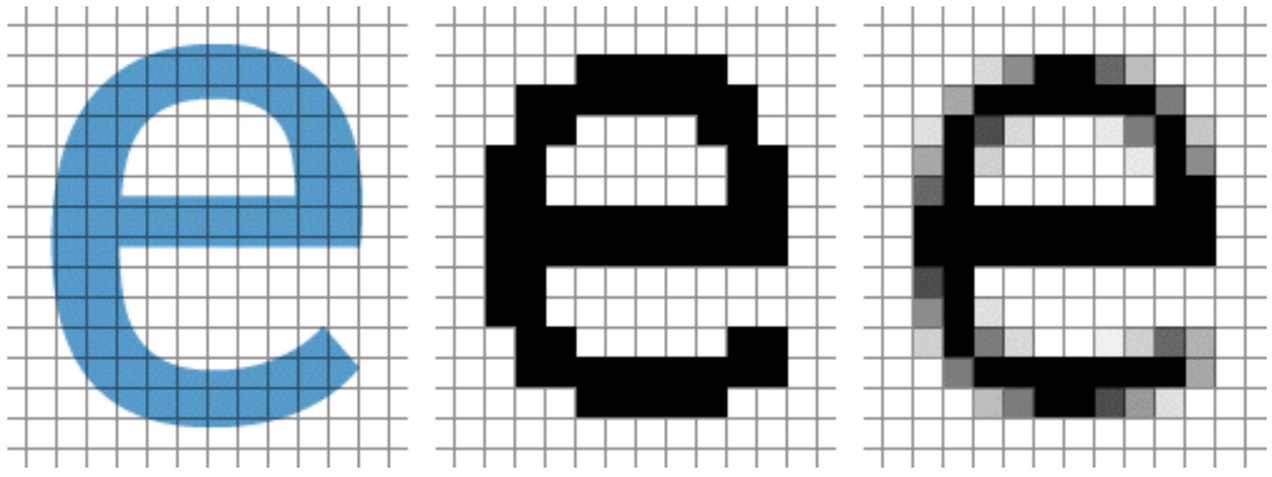
Est. 2008

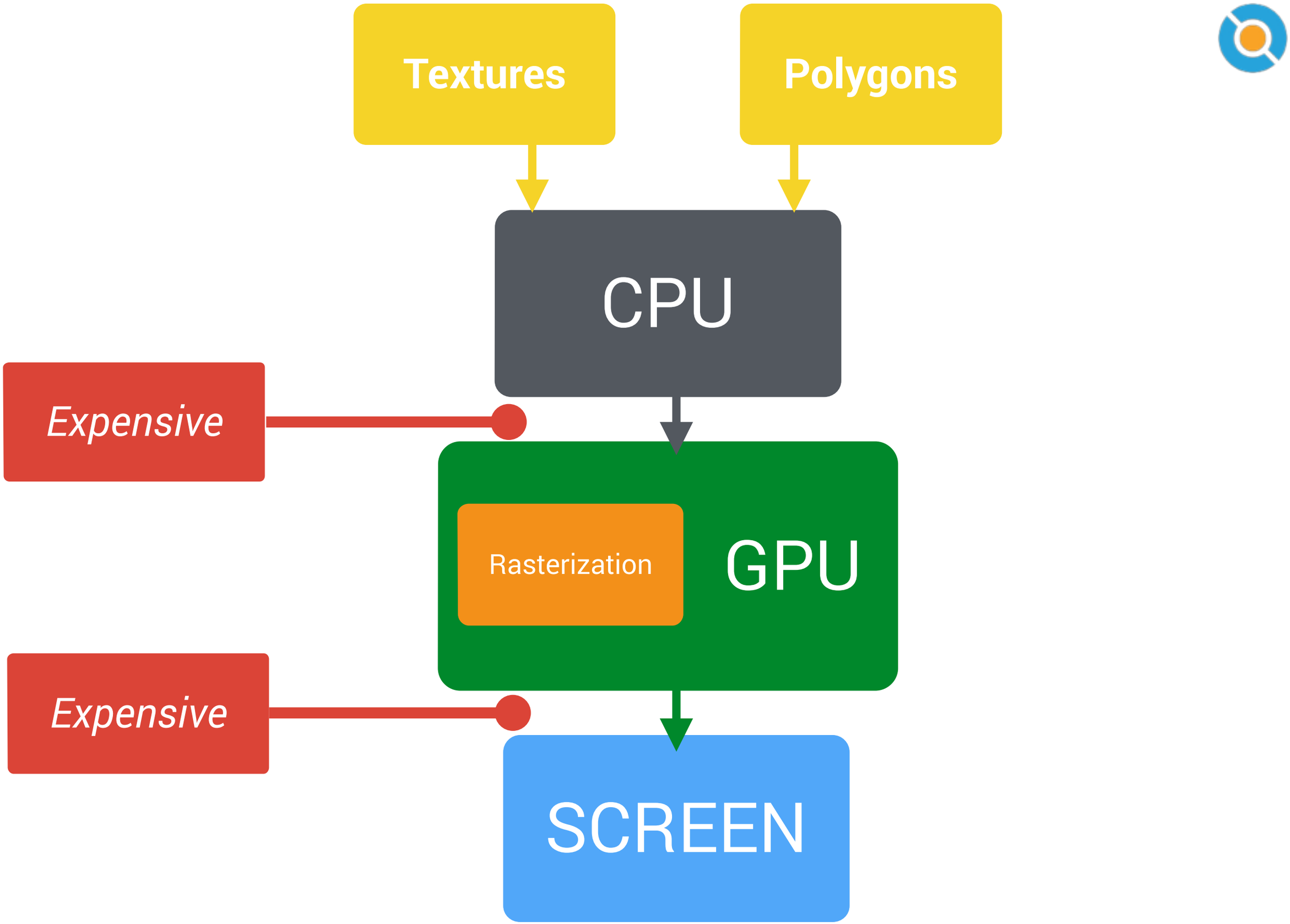
Google
eveloper Expert
Android
2014

Scrum
Allian

How Android draws

Rasterization




Textures
Polygons
CPU
Expensive
Rasterization
GPU
Expensive
SCREEN

Why 60 fps?

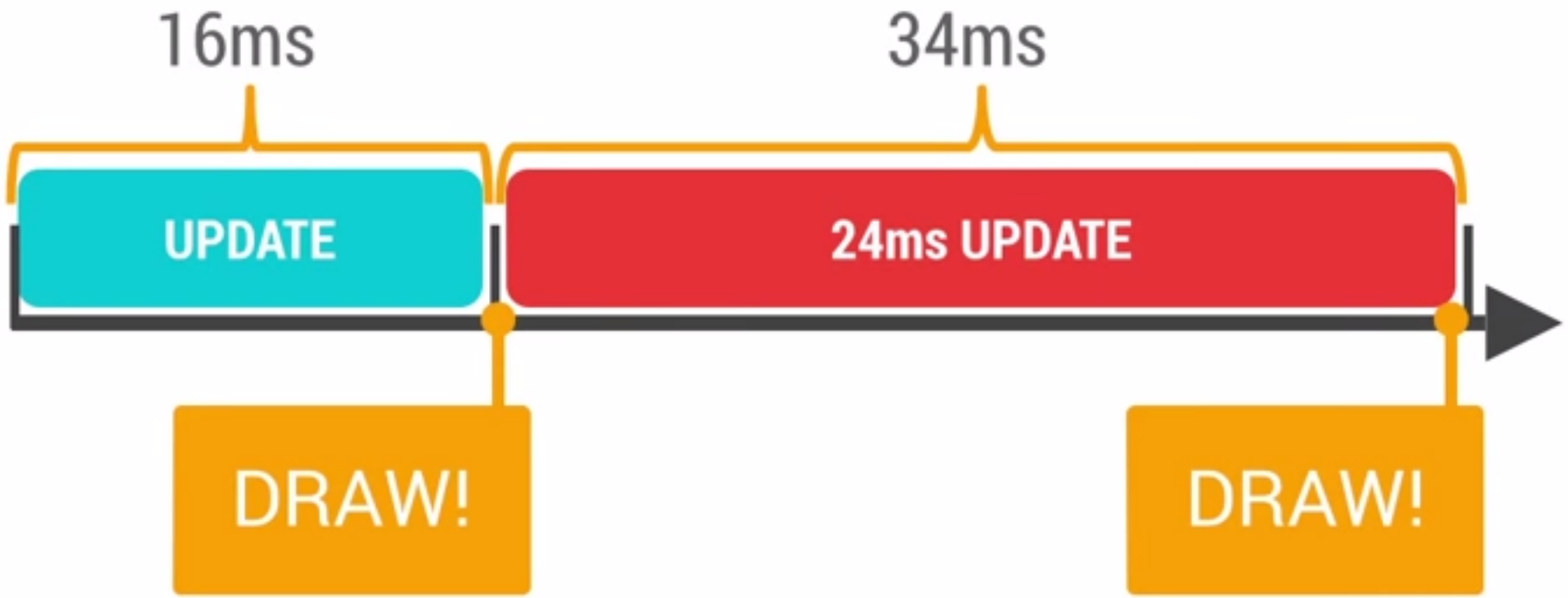
Normal speed

30fps

60fps

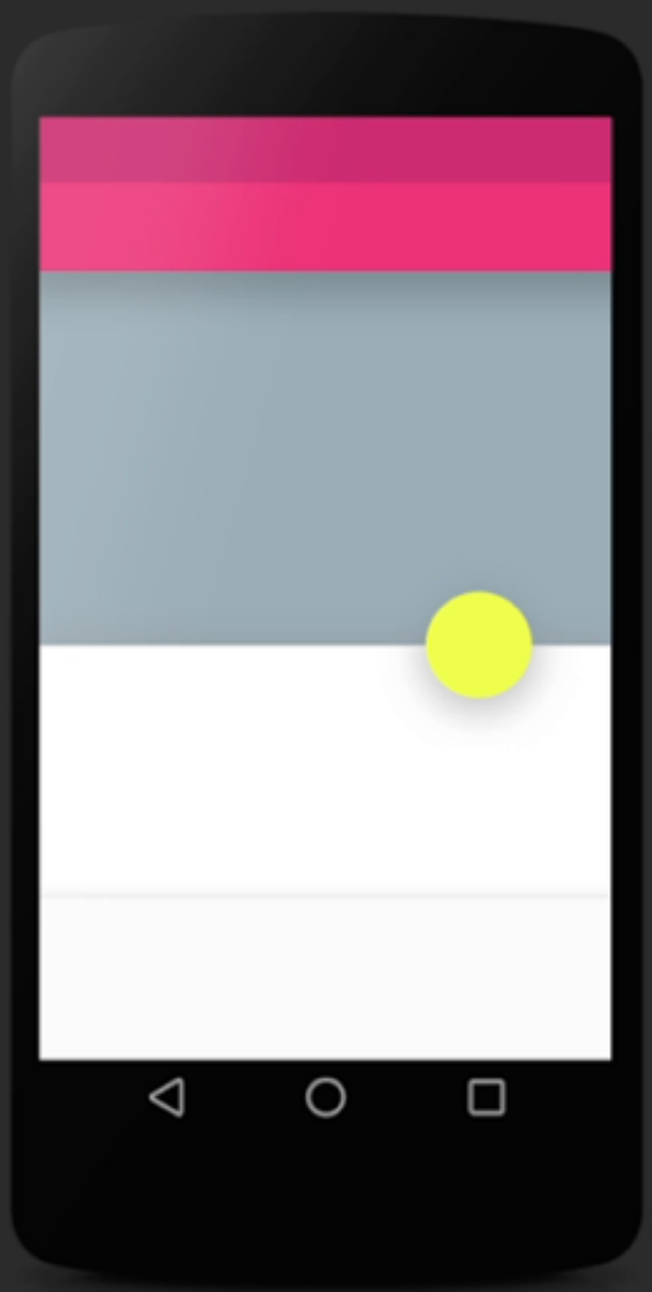
http://www.techverticals.com/wp-content/uploads/2012/04/24fps-vs-30fps-vs-60fps.jpg

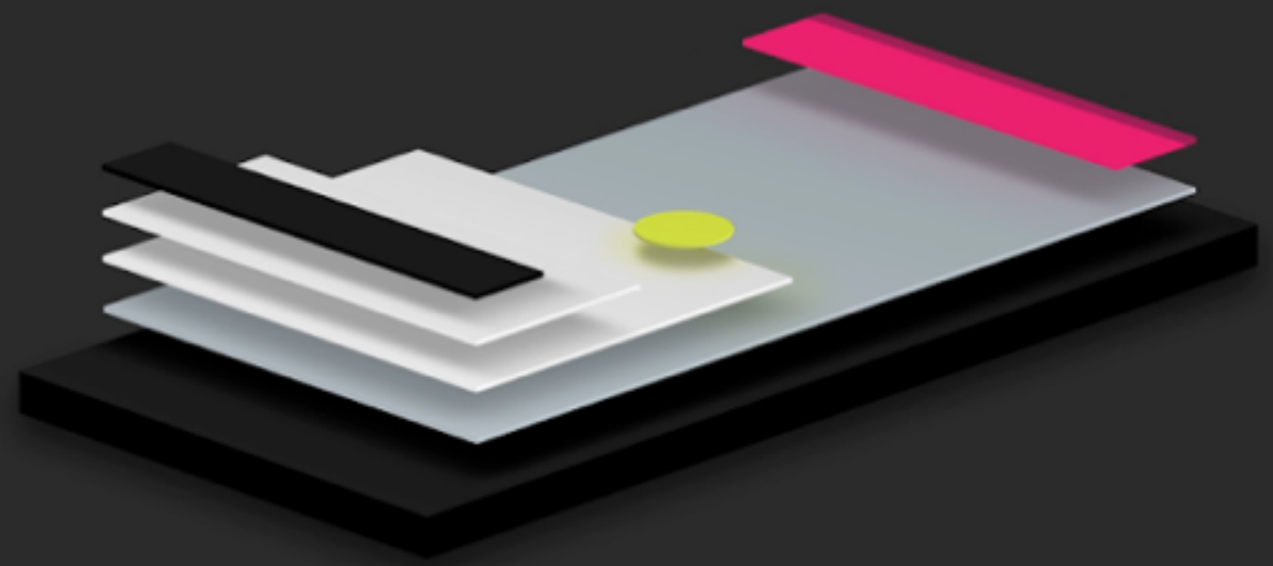
Design vs Performance



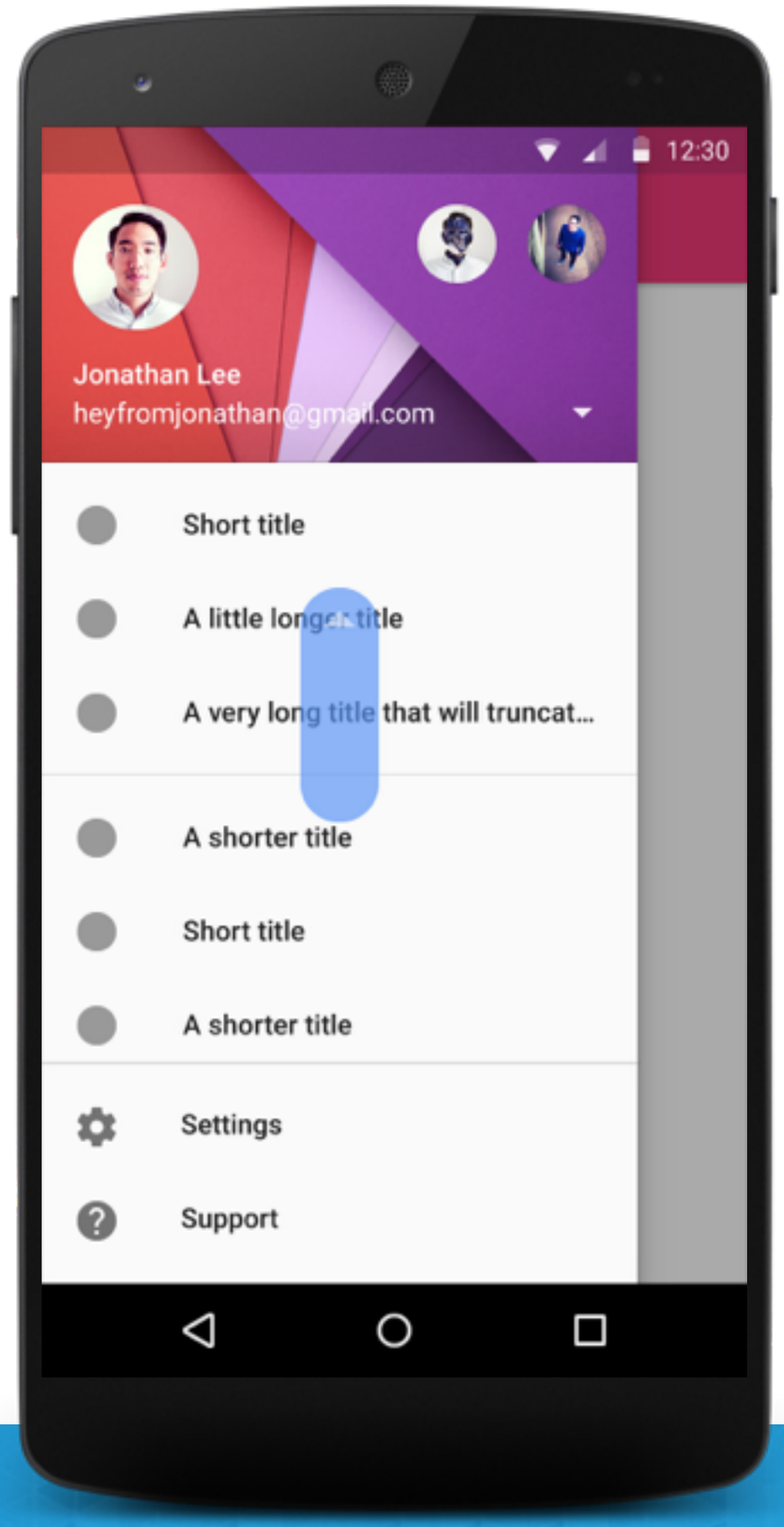
1000ms / 60 frames = 16.666 ms / frame

Overdraw



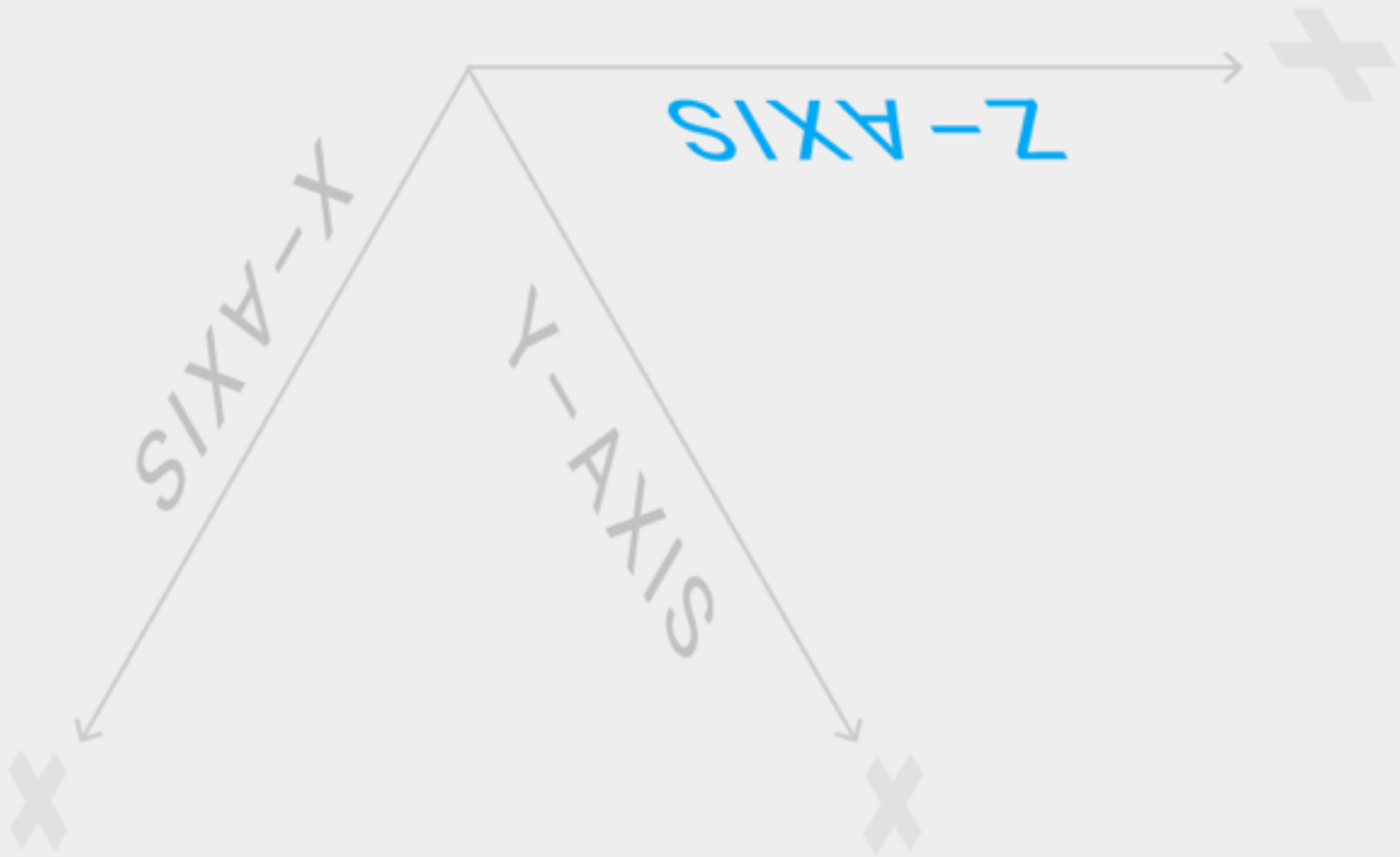


Overdraw


12:30
Jonathan Lee
heyfromjonathan@gmail.com
Short title
A little longer title
A very long title that will truncat...
A shorter title
Short title
A shorter title
Settings
Support

Shadows

- Views with elevation casts showdown onto background
- Orthographic projection
- Animated like other View properties

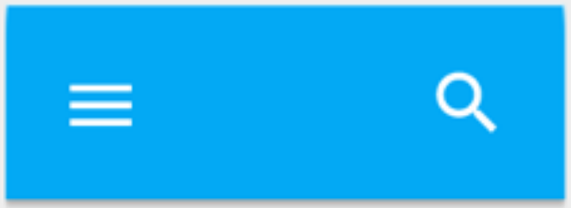

Z- AXIS
X-AXIS
Y-AXIS

Shadows

Z = Elevation + TranslationZ

Elevation

App bar



4dp

Floating action button

8dp

Card

2dp

Outline

res/layout/fragment_sample.xml

```
<ImageView
   …
   android:background=“@drawable/myrect“/>
```

res/drawable/myrect.xml

```
<shape android:shape=”rectangle”>
  <solid android:color=”#42000000” />
  <corners android:radius=”5dp” />
</shape>
```

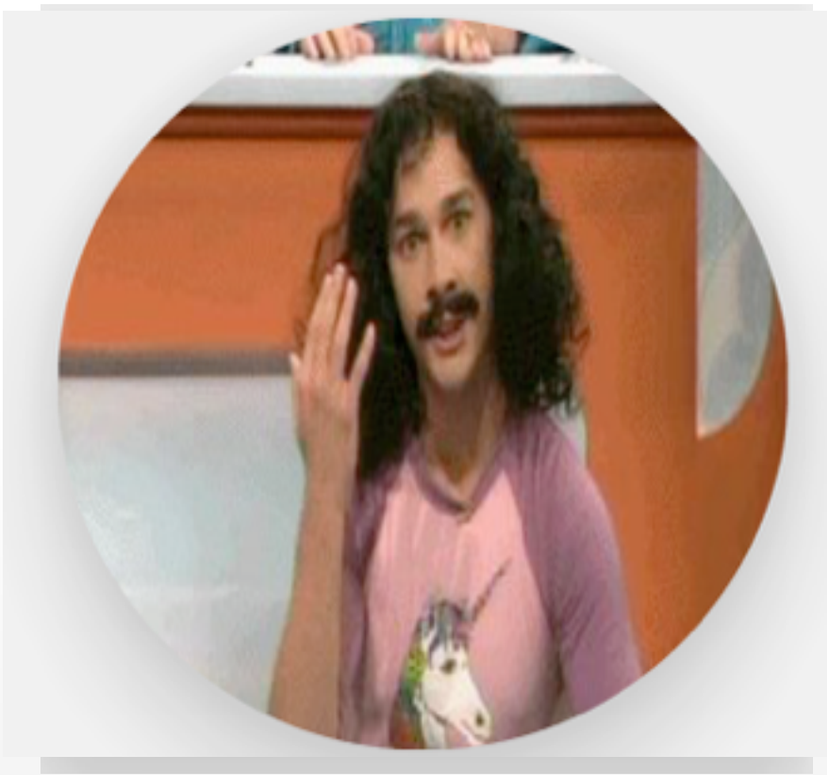
OvalOutlineProvider.java

```
static class OvalOutlineProvider
        extends ViewOutlineProvider{

    @Override
    public void getOutline(View view, Outline outline) {
        outline.setOval(0, 0,
          view.getWidth(),
          view.getHeight(),
          10);
    }
}
```

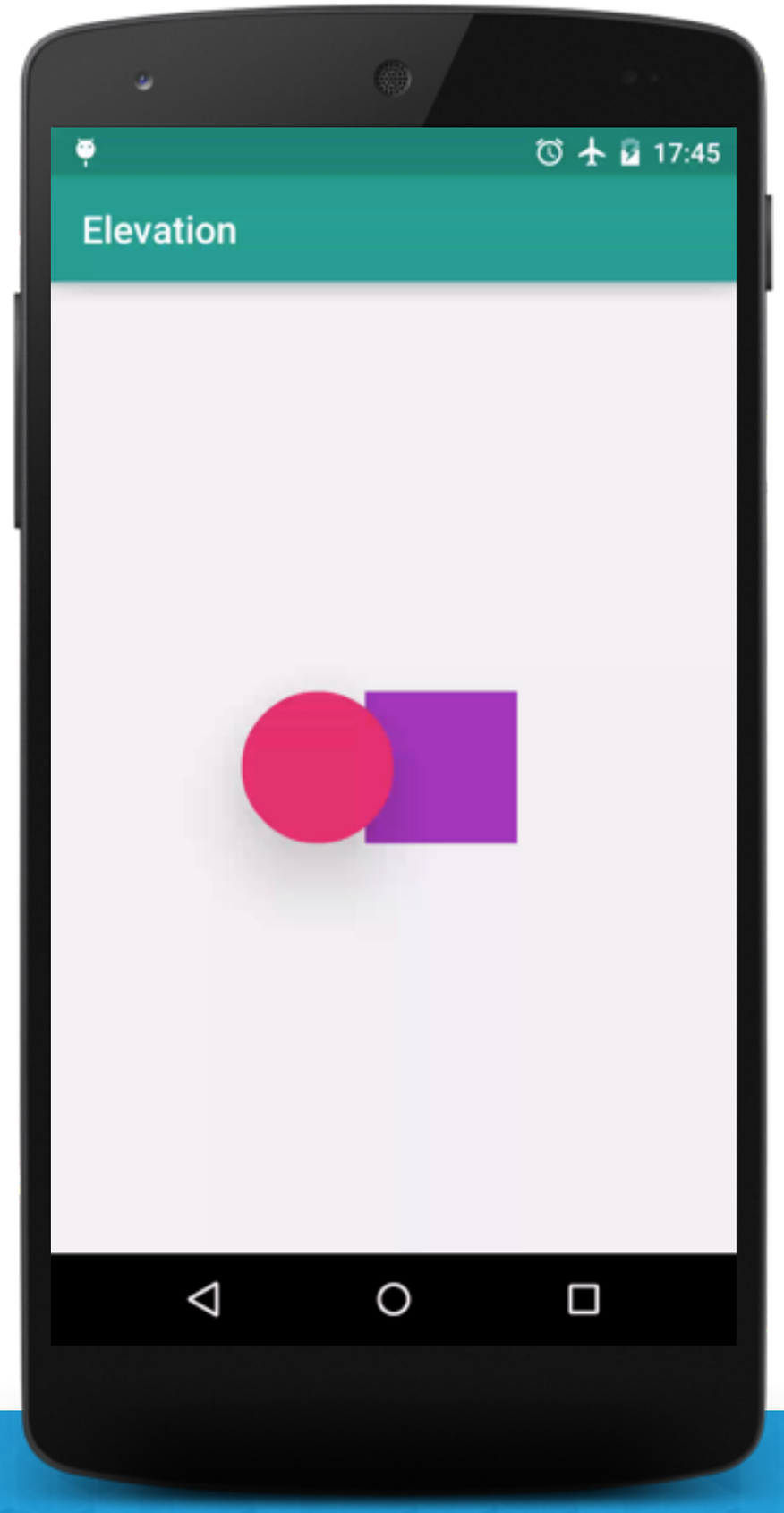
Define your own one

MyFragment.java

```
view.setClipToOutline(true);

view.setOutlineProvider(new
RoundRectOutlineProvider());
```



Basic elevation



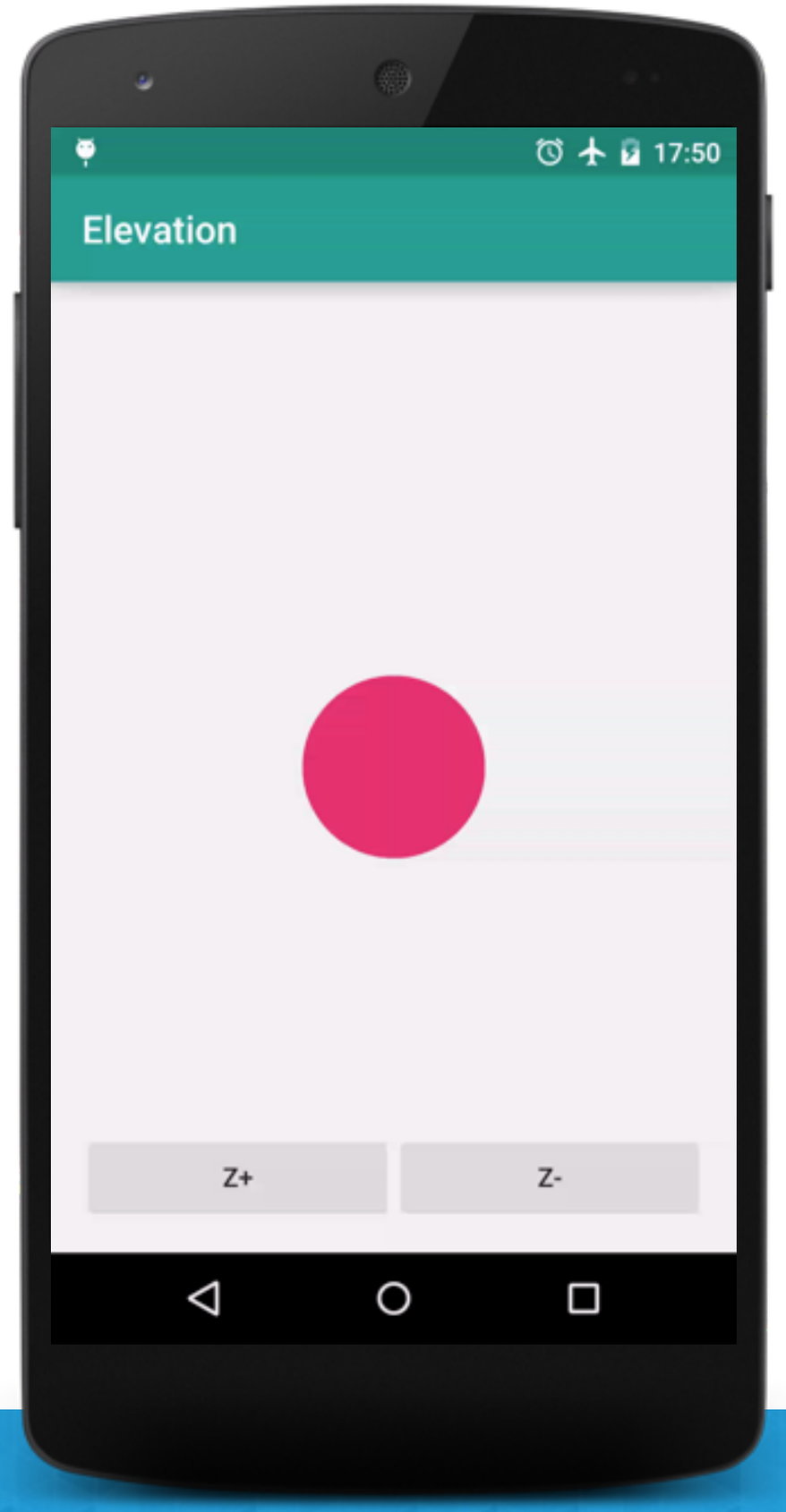
res/layout/elevation_basic.xml

```
<FrameLayout
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
  <View
    android:id="@+id/floating_shape"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginRight="40dp"
    android:background="@drawable/shape"
    android:elevation="30dp"
    android:layout_gravity="center"/>
  <View
    android:id="@+id/floating_shape_2"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="25dp"
    android:background="@drawable/shape2"
    android:layout_gravity="center"/>
</FrameLayout>
```

ElevationBasicFragment.java

```
shape2.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        int action = motionEvent.getActionMasked();
        switch (action) {
            case MotionEvent.ACTION_DOWN:
                view.animate().setDuration(100)
                  .scaleX(1.2f).scaleY(1.3f).translationZ(120);
                break;
            case MotionEvent.ACTION_UP:
                view.animate().setDuration(100)
                  .scaleX(1).scaleY(1).translationZ(0);
                break;
            default:
                return false;
        }
        return true;
    }
});
```

Drag elevation



res/layout/elevation_drag.xml

```
<FrameLayout
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
  <View
    android:id="@+id/floating_shape"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginRight="40dp"
    android:background="@drawable/shape"
    android:elevation="30dp"
    android:layout_gravity="center"/>
  <View
    android:id="@+id/floating_shape_2"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_marginLeft="25dp"
    android:background="@drawable/shape2"
    android:layout_gravity="center"/>
</FrameLayout>
```

ElevationDragFragment.java

```
/* Raise the circle in z when the "z+" button is clicked. */
rootView.findViewById(R.id.raise_bt).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mElevation += mElevationStep;
        floatingShape.setElevation(mElevation);
    }
});
```

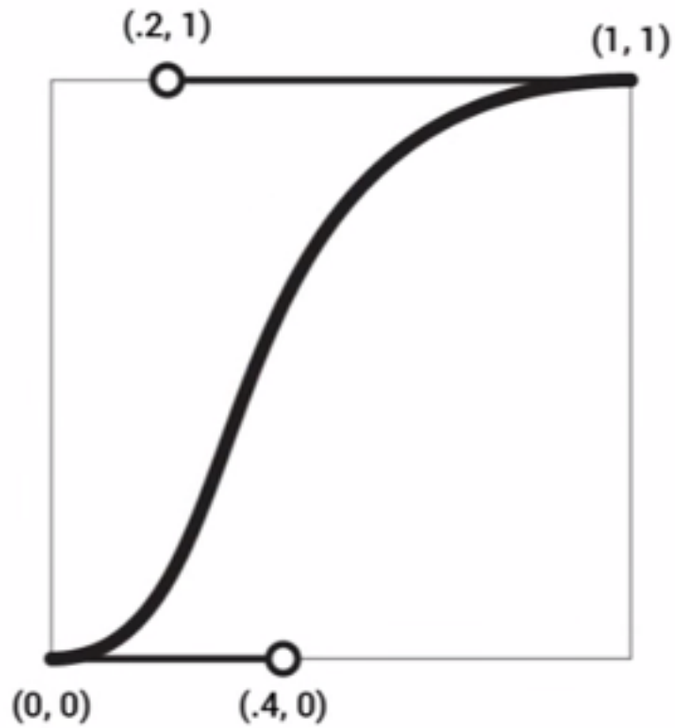
```
/* Lower the circle in z when the "z-" button is clicked. */
rootView.findViewById(R.id.lower_bt).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mElevation -= mElevationStep;
        // Don't allow for negative values of Z.
        if (mElevation < 0) {
            mElevation = 0;
        }
    }
});
```

ElevationDragFragment.java

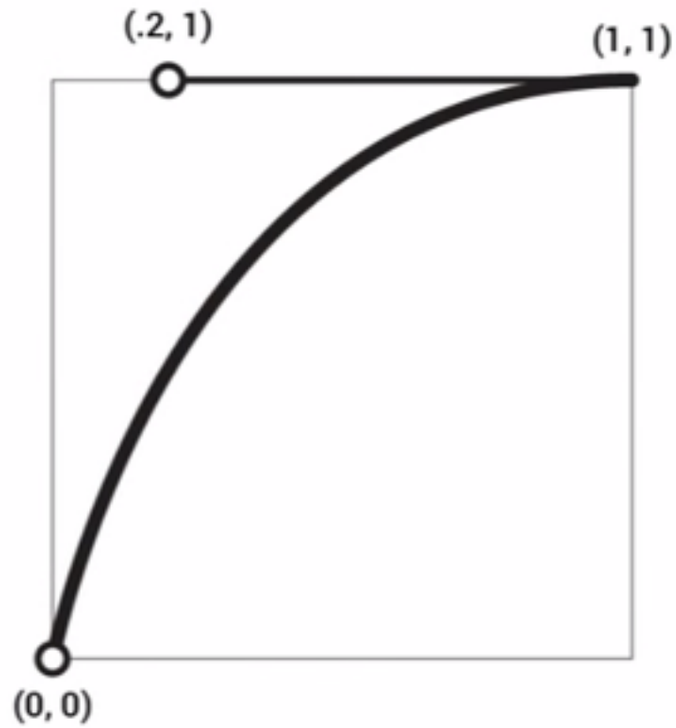
```
dragLayout.setDragFrameController(new
    DragFrameLayout.DragFrameLayoutController() {

    @Override
    public void onDragDrop(boolean captured) {
         /* Animate the translation of the {@link View}.
           Note that the translation
         is being modified, not the elevation. */
         floatingShape.animate()
                 .translationZ(captured ? 50 : 0)
                 .scaleX(1.2f)
                 .scaleY(1.3f)
                 .setDuration(100);
         Log.d(TAG, captured ? "Drag" : "Drop");
    }
});
```

Animations

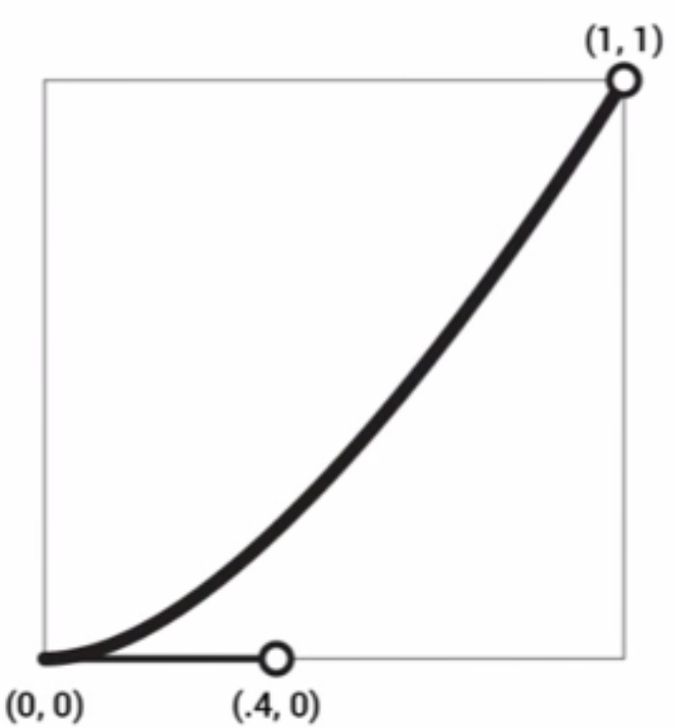
Animation curves



@interpolator/fast_out_slow_in

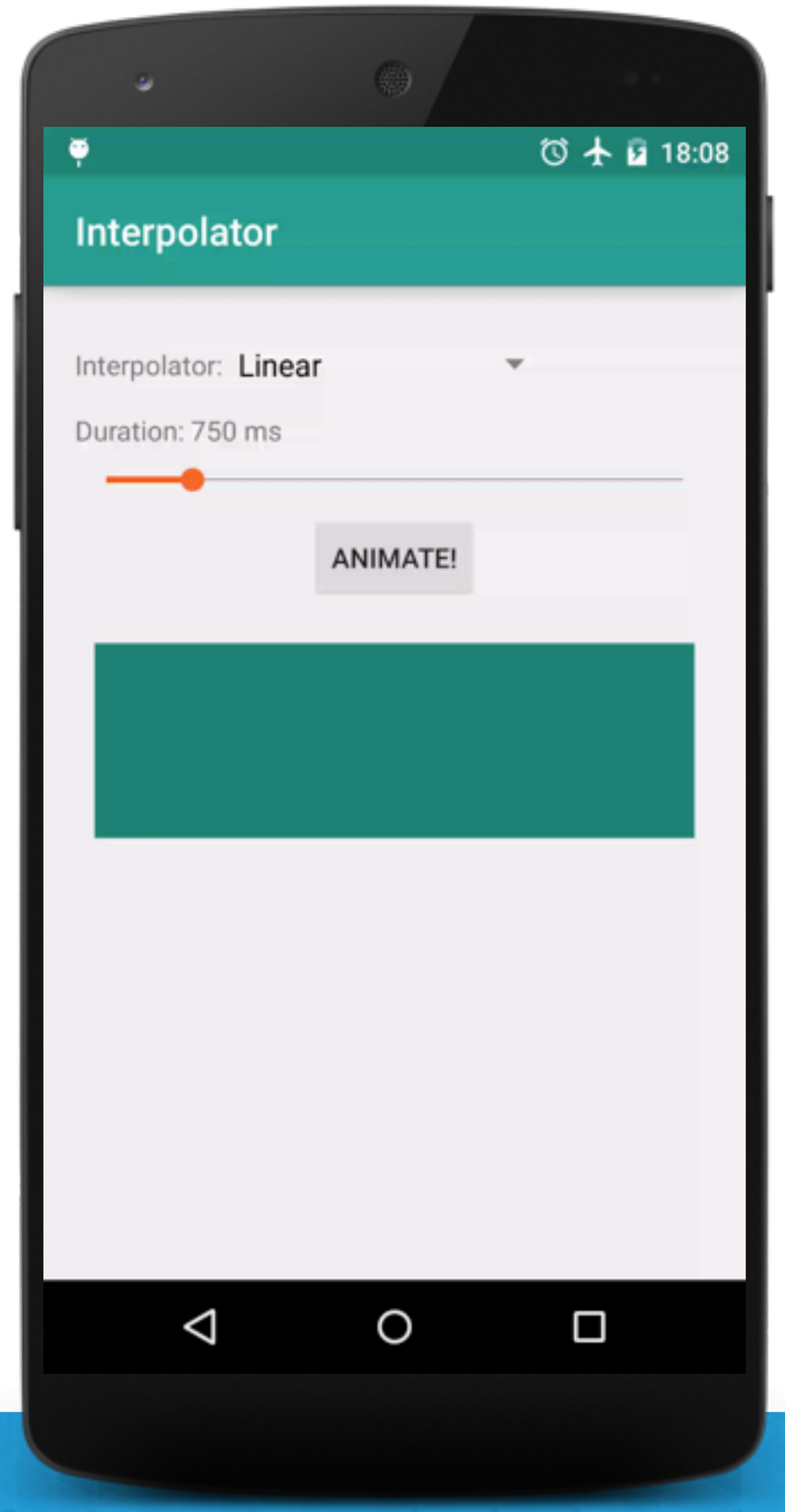


@interpolator/linear_out_slow_in



@interpolator/fast_out_linear_in

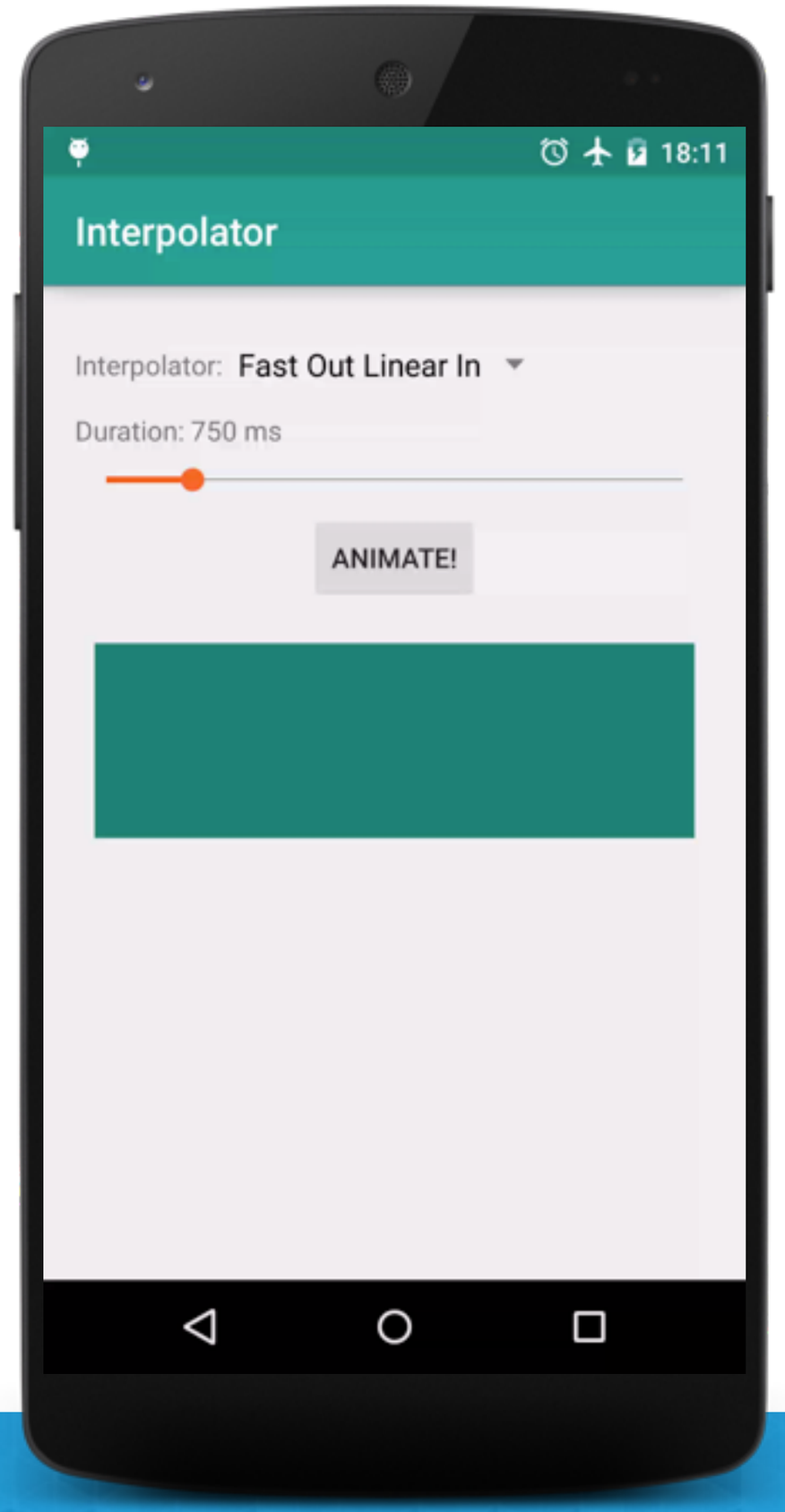
Path Interpolator



res/interpolator/linear.xml

```
<linearInterpolator />
```

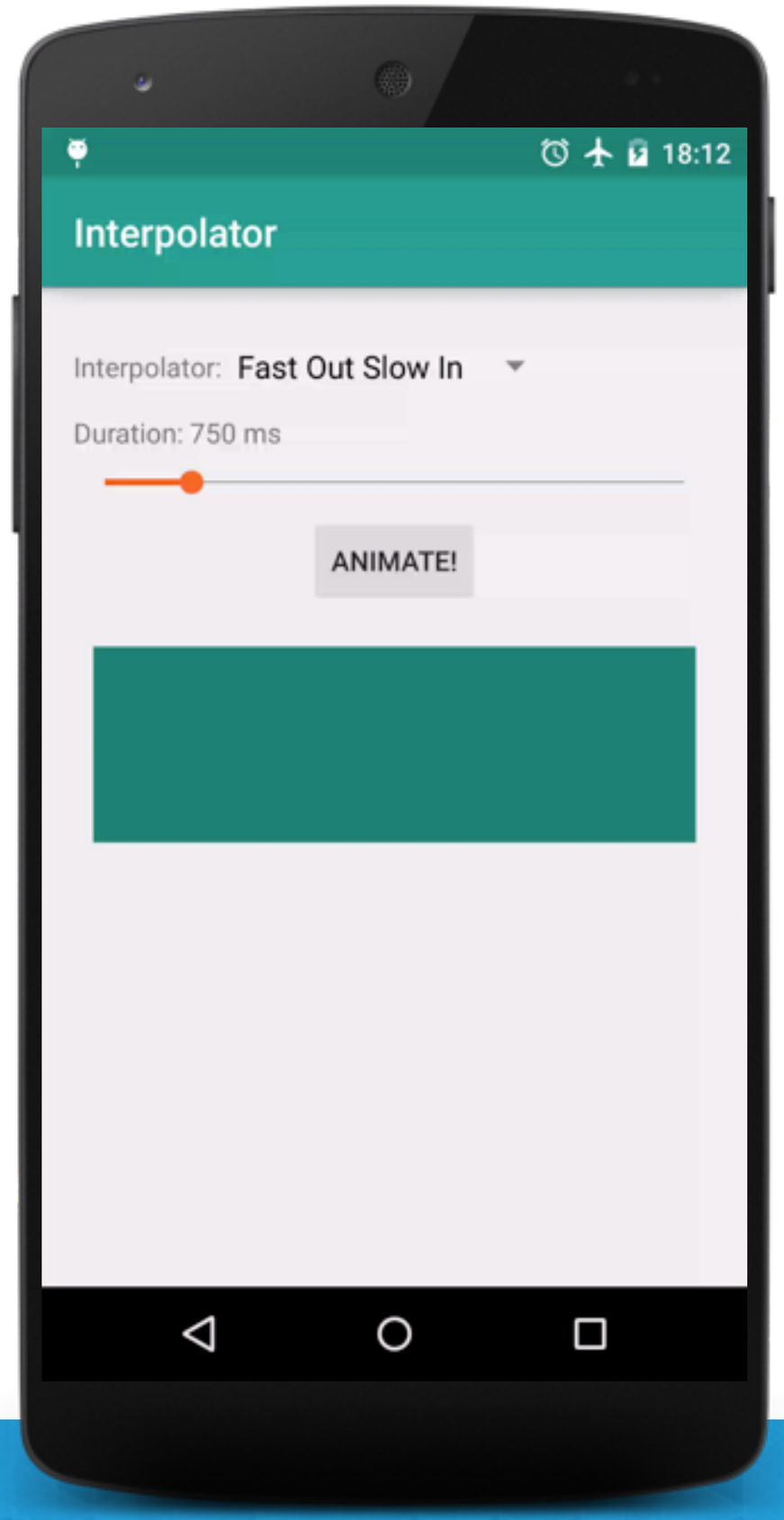
Path Interpolator



res/interpolator/fast_out_linear_in.xml

```
<pathInterpolator
    android:controlX1="0.4"
    android:controlY1="0"
    android:controlX2="1"
    android:controlY2="1"/>
```

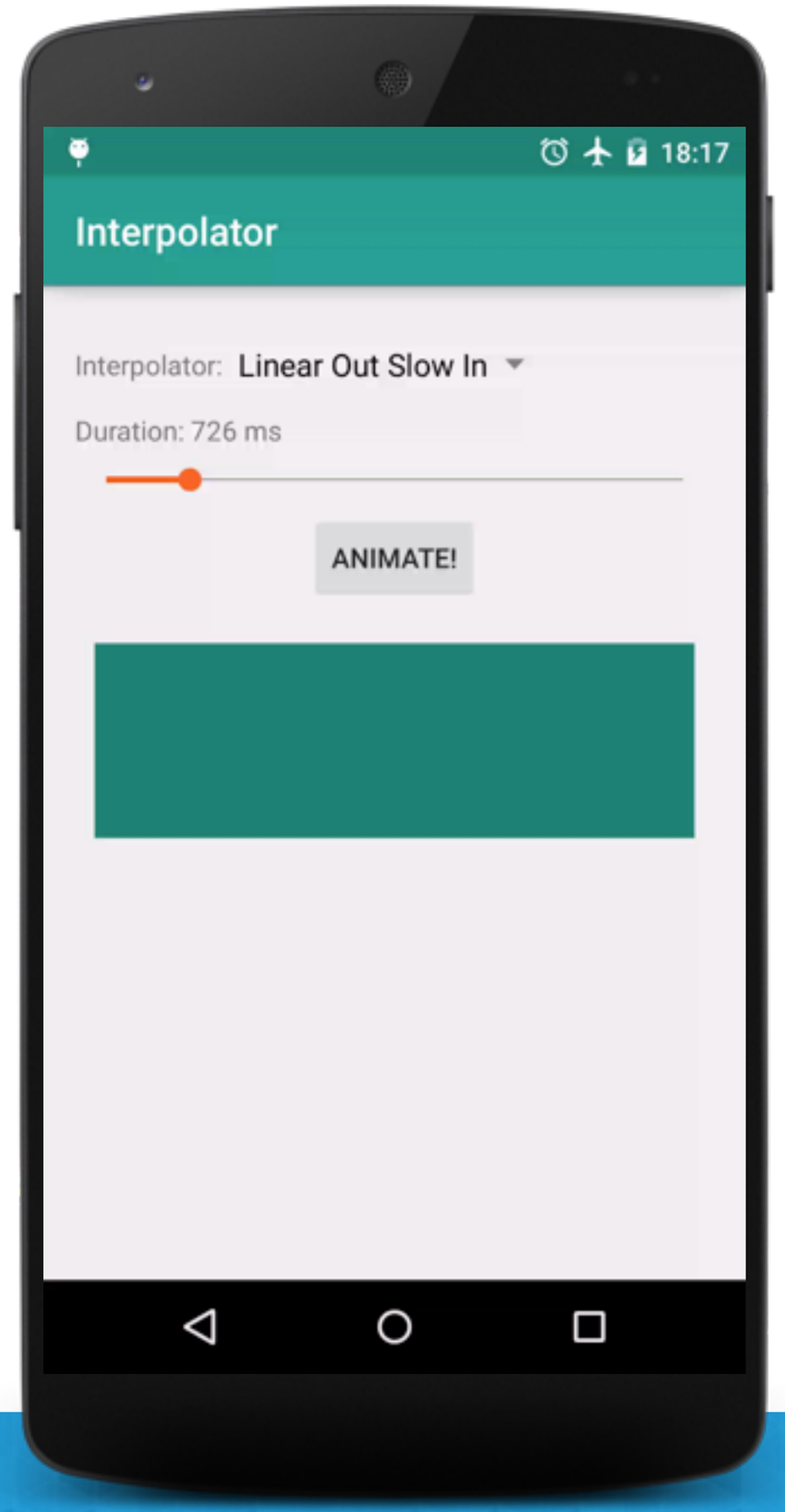
Path Interpolator



res/interpolator/fast_out_slow_in.xml

```
<pathInterpolator
    android:controlX1="0"
    android:controlY1="0"
    android:controlX2="0.2"
    android:controlY2="1"/>
```

Path Interpolator



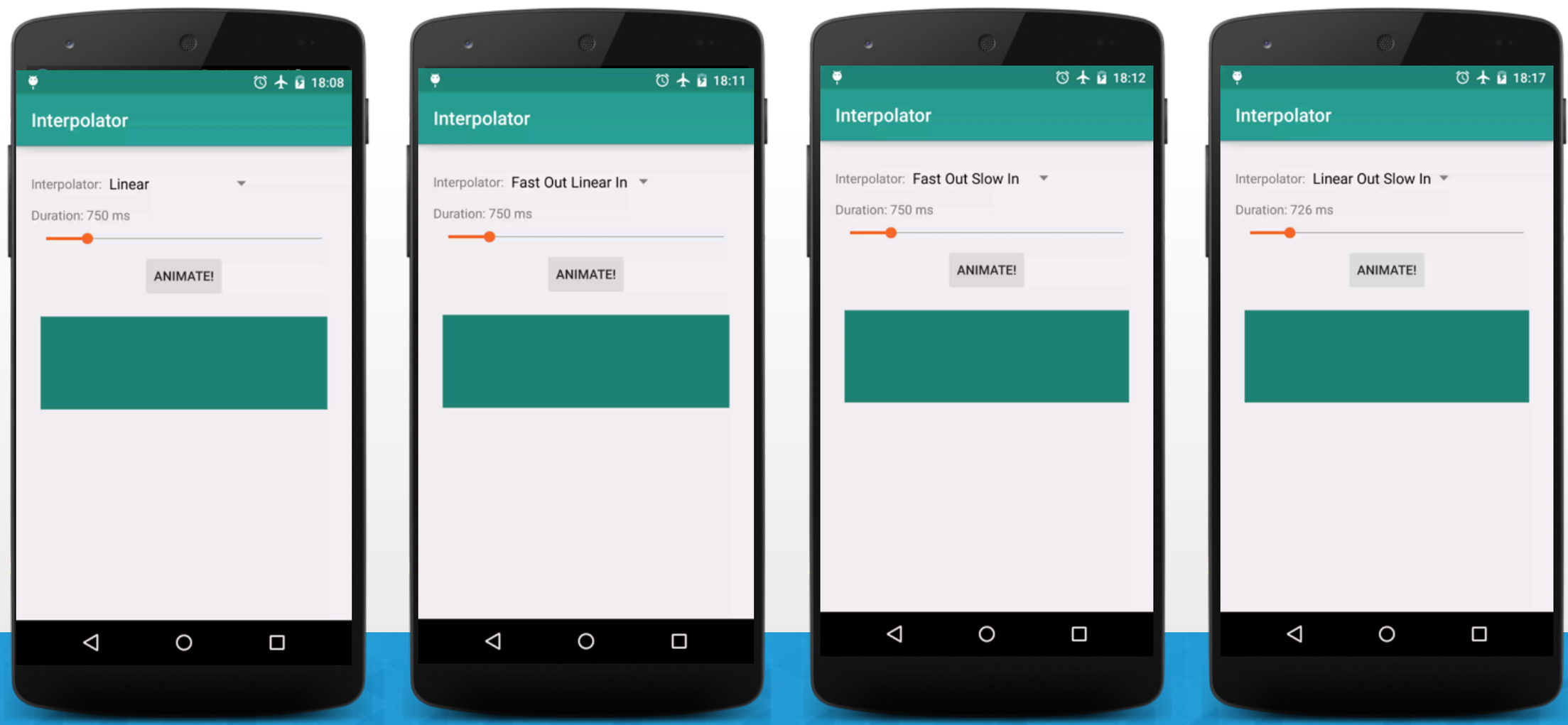
res/interpolator/linear_out_slow_in.xml

```
<pathInterpolator
    android:controlX1="0"
    android:controlY1="0"
    android:controlX2="0.2"
    android:controlY2="1"/>
```

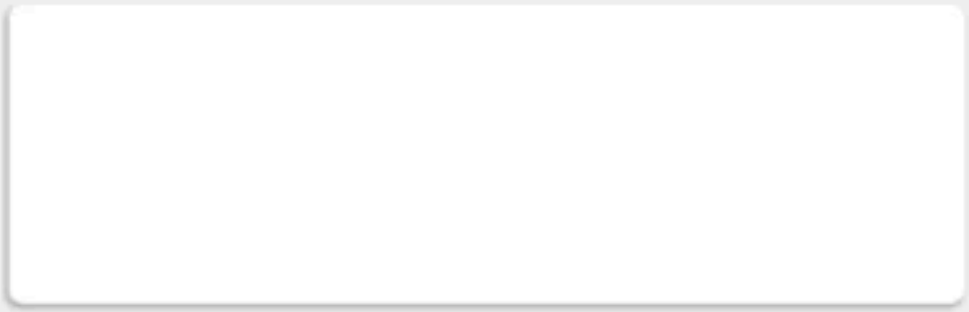
InterpolatorFragment.java

```
// Path for 'in' animation: growing from 20% to 100%
mPathIn = new Path();
mPathIn.moveTo(0.2f, 0.2f);
mPathIn.lineTo(1f, 1f);

// Path for 'out' animation: shrinking from 100% to 20%
mPathOut = new Path();
mPathOut.moveTo(1f, 1f);
mPathOut.lineTo(0.2f, 0.2f);
ObjectAnimator animator =
    ObjectAnimator.ofFloat(mView, View.SCALE_X, View.SCALE_Y, path);



// Set the duration and interpolator for this animation
animator.setDuration(duration);
animator.setInterpolator(interpolator);

animator.start();
```

The Interpolator party!



Touch feedback

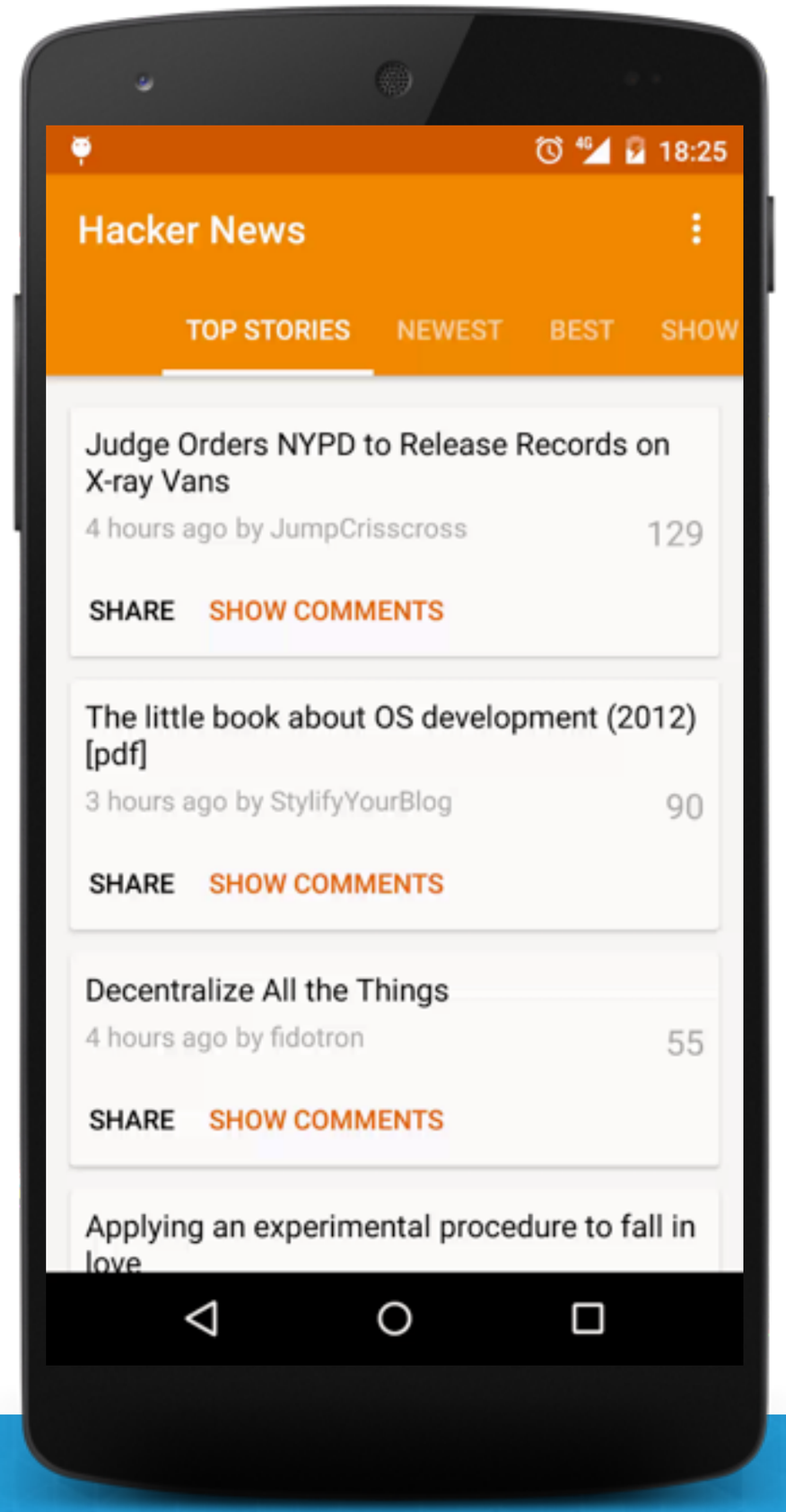


res/drawable/mystatedrawable.xml

```
<selector xmlns:android=“…”/>

   <item android:state_pressed="true">
    <objectAnimator
      android:propertyName="translationZ"
      android:duration="100"
      android:valueTo="5dp" />
   </item>

   <item android:state_pressed="false">
    <objectAnimator
      android:propertyName="translationZ"
      android:duration="100"
      android:valueTo="0" />
   </item>

</selector>
```

Touch feedback

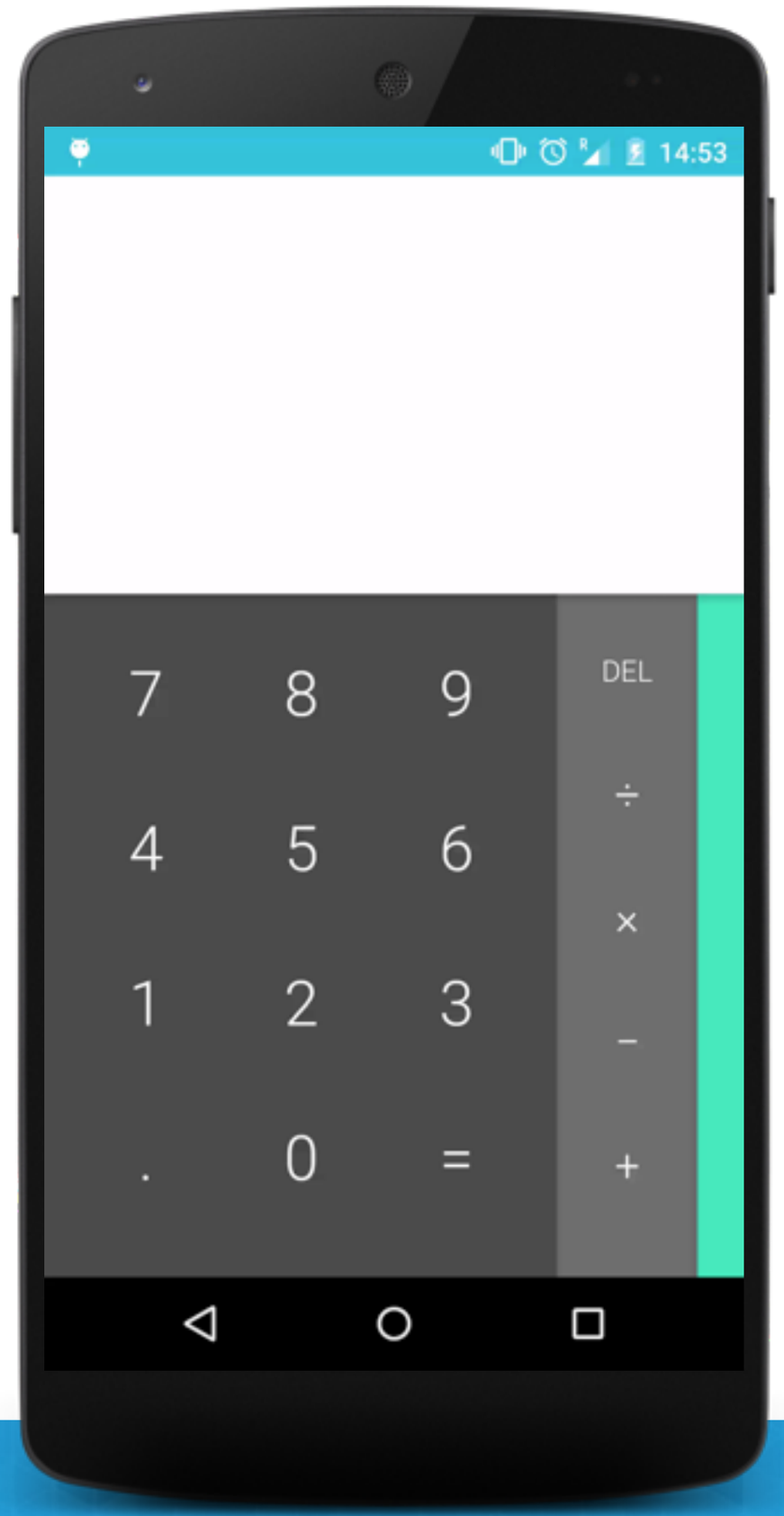
Render Thread



Reveal effect

MyActivity.java

```
Animator anim =
ViewAnimationUtils.
    createCircularReveal(
        myView,
        centerX,
        centerY,
        startRadius,
        finalRadius);

anim.start();
```



MyActivity.java

```
final View myView = findViewById(R.id.my_view);

Animator anim =
    ViewAnimationUtils.createCircularReveal(myView,
    cx,
    cy,
    initialRadius,
    0);

anim.addListener(new AnimatorListenerAdapter() {
    @Override
    public void onAnimationEnd(Animator animation) {
        super.onAnimationEnd(animation);
        myView.setVisibility(View.INVISIBLE);
    }
});

anim.start();
```

Activity transitions

- Window Transitions animate windows
- Activity transitions animate window components
- Animate when launching one activity from another
- Shared elements are transferred via ActivityOptions
- Based on the Transitions API released with KitKat

Enable transitions

MainActivity.java

```
getWindow().
   requestFeature(Window.FEATURE_CONTENT_TRANSITIONS);
```

res/values/theme.xml

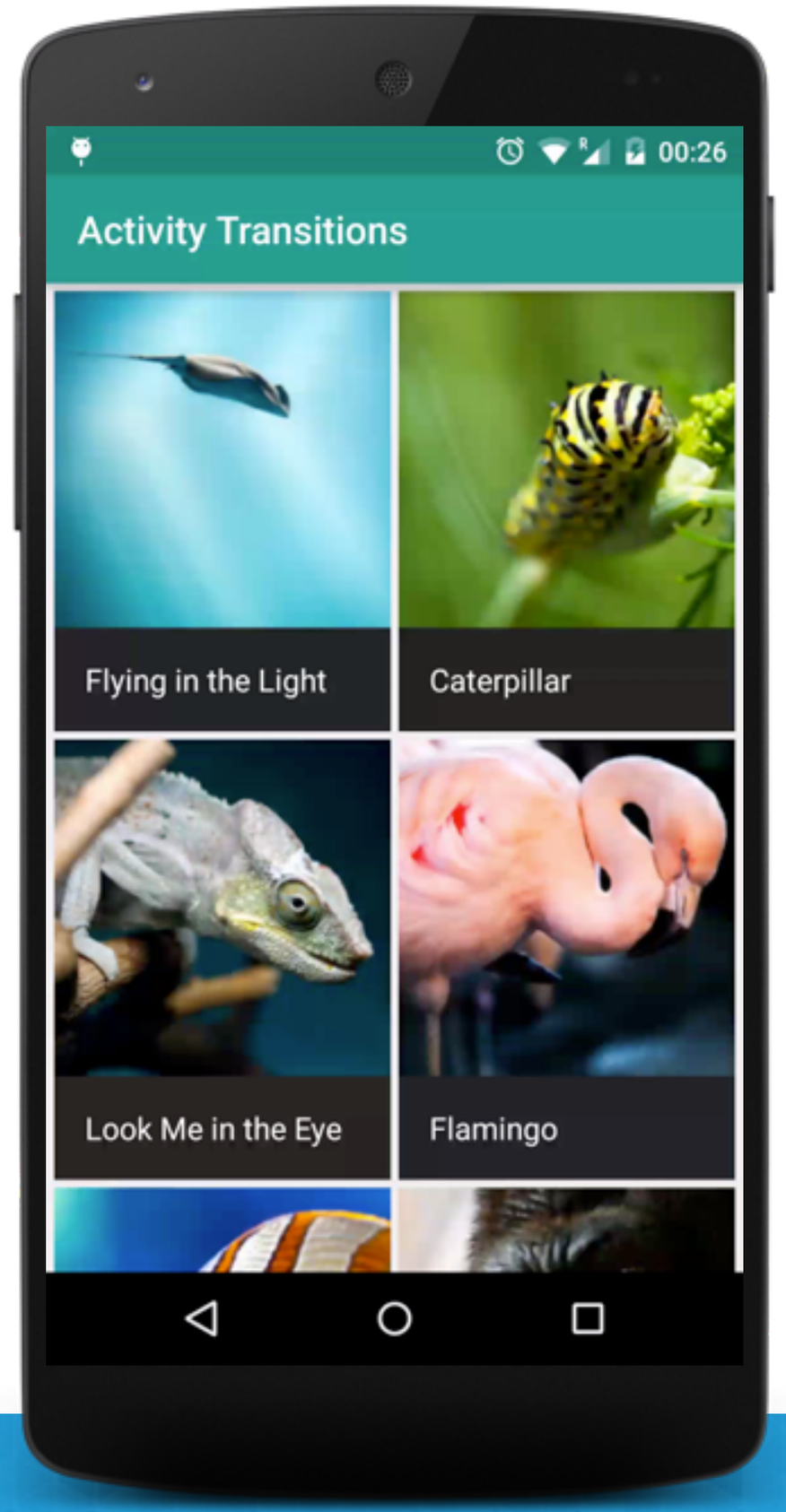
```
<style name="BaseAppTheme" parent="android:Theme.Material">
 <!-- enable window content transitions -->
 <item name="android:windowContentTransitions">true</item>
</style>
```

Activity transition

res/values/theme.xml

```
<style name="BaseAppTheme" parent="android:Theme.Material">

  <!-- specify enter and exit transitions -->
  <item name="android:windowEnterTransition">
          @transition/explode</item>
  <item name="android:windowExitTransition">
          @transition/explode</item>
</style>
```

Example


00:26
Activity Transitions
Flying in the Light
Caterpillar
Look Me in the Eye
Flamingo

Activity A

res/layout/grid_item.xml

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:orientation="vertical">

  <ImageView
    android:id="@+id/imageview_item"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

  <TextView
    android:id="@+id/textview_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?android:attr/colorPrimary"/>

</LinearLayout>
```

Activity B

res/layout/detail.xml

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:orientation="vertical">

  <ImageView
    android:id="@+id/imageview_header"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

  <TextView
    android:id="@+id/textview_title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@android:style/Theme.Material"/>

</LinearLayout>
```

TransitionActivity.java

```
ActivityOptionsCompat activityOptions =
   ActivityOptionsCompat.makeSceneTransitionAnimation(
        this,
           new Pair<View,
            String>(view.findViewById(R.id.imageview_item),
              DetailActivity.VIEW_NAME_HEADER_IMAGE),
           new Pair<View,
            String>(view.findViewById(R.id.textview_name),
              DetailActivity.VIEW_NAME_HEADER_TITLE));

// Now we can start the Activity, providing the
activity options as a bundle
ActivityCompat.startActivity(this, intent,
activityOptions.toBundle());
```

DetailActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_transition_details);

    /**
     * Set the name of the view's which will be transition
to, using the static values above.
     * This could be done in the layout XML, but exposing it
via static variables allows easy
     * querying from other Activities
     */
    ViewCompat.setTransitionName(mHeaderImageView,
        VIEW_NAME_HEADER_IMAGE);
    ViewCompat.setTransitionName(mHeaderTitle,
        VIEW_NAME_HEADER_TITLE);

}
```

Shared element transition

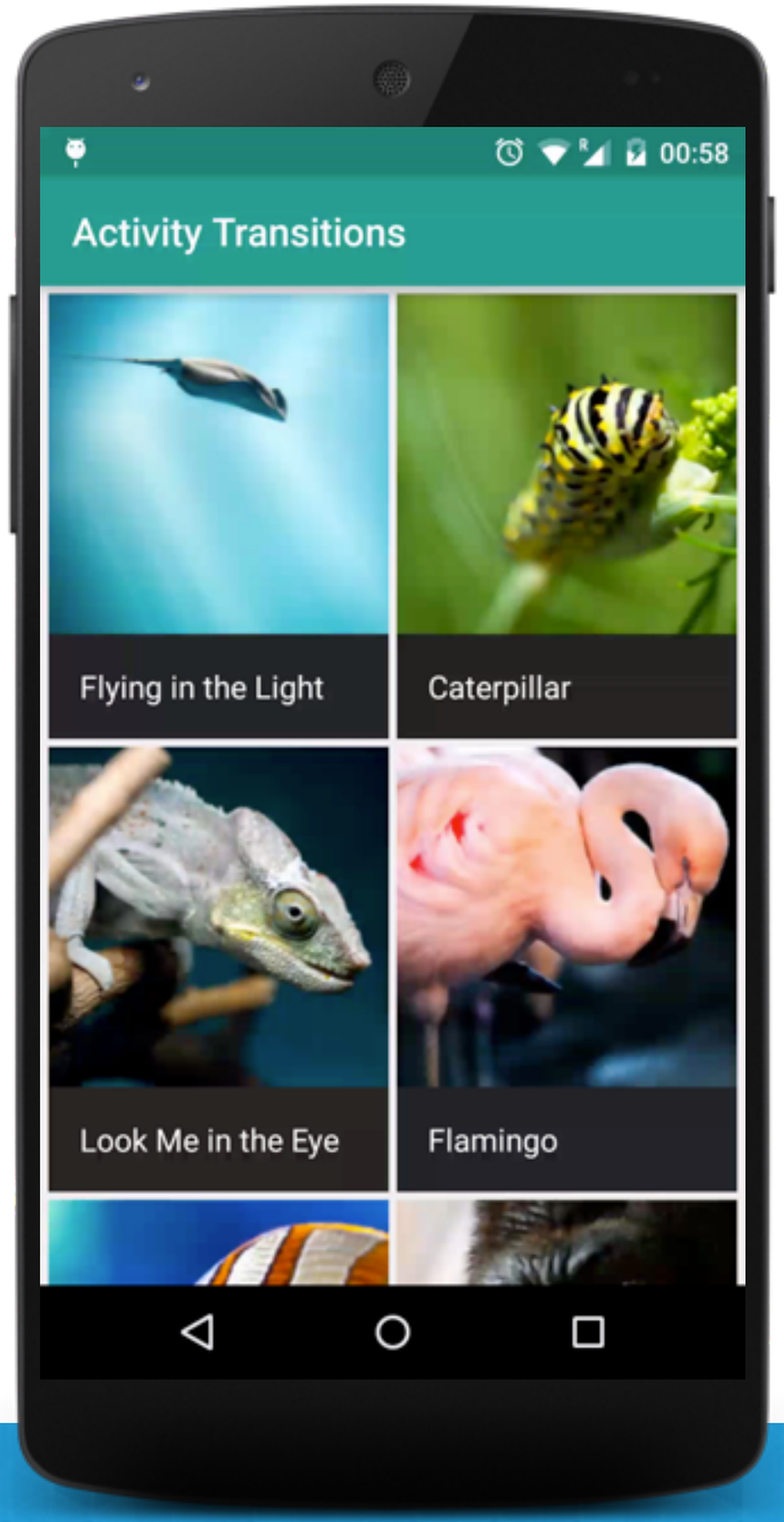
res/layout/fragment_sample.xml

```
<ImageView
   android:layout_width="200dp"
   android:layout_height="200dp"
   android:id="@+id/robotoView"
   android:layout_centerVertical="true"
   android:layout_centerHorizontal="true"
   android:background="@drawable/magic"
   android:transitionName="@transition/my_transition"/>
```

res/transition/grid_detail_transition.xml

```
<transitionSet xmlns:android="…">
  <changeBounds/>
  <changeImageTransform/>
</transitionSet>
```

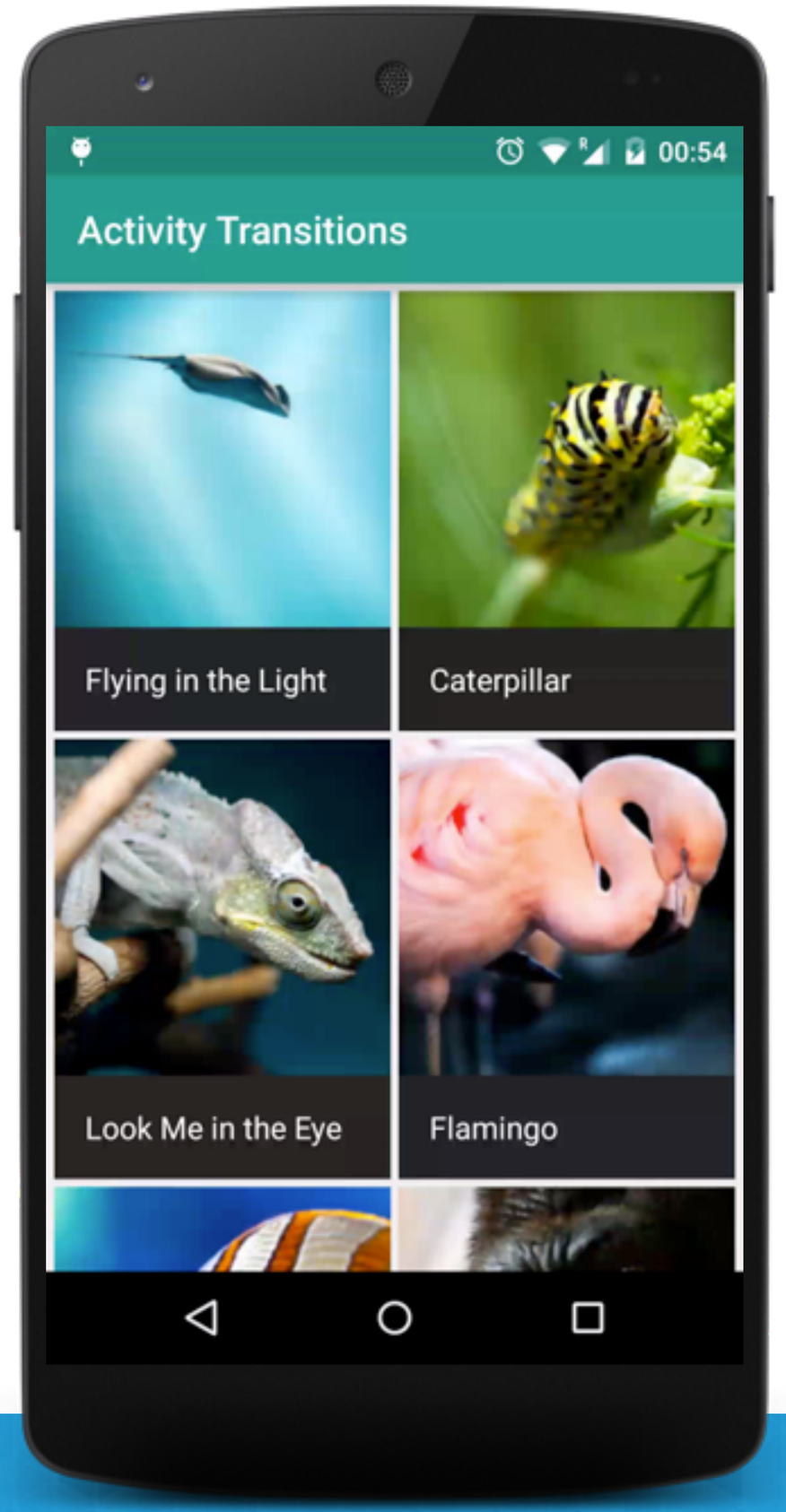
Sync Transitions and animations



DetailActivity.java

```
getWindow().getEnterTransition().addListener(new
    Transition.TransitionListener() {
     @Override
    public void onTransitionEnd(Transition transition) {
                 mFab.animate()
                         .translationY(0)
                         .setInterpolator(new
                             OvershootInterpolator(1.f))
                         .setStartDelay(300)
                         .setDuration(400)
                         .start();
        }
});
```

Animate before transition



Animate before transition

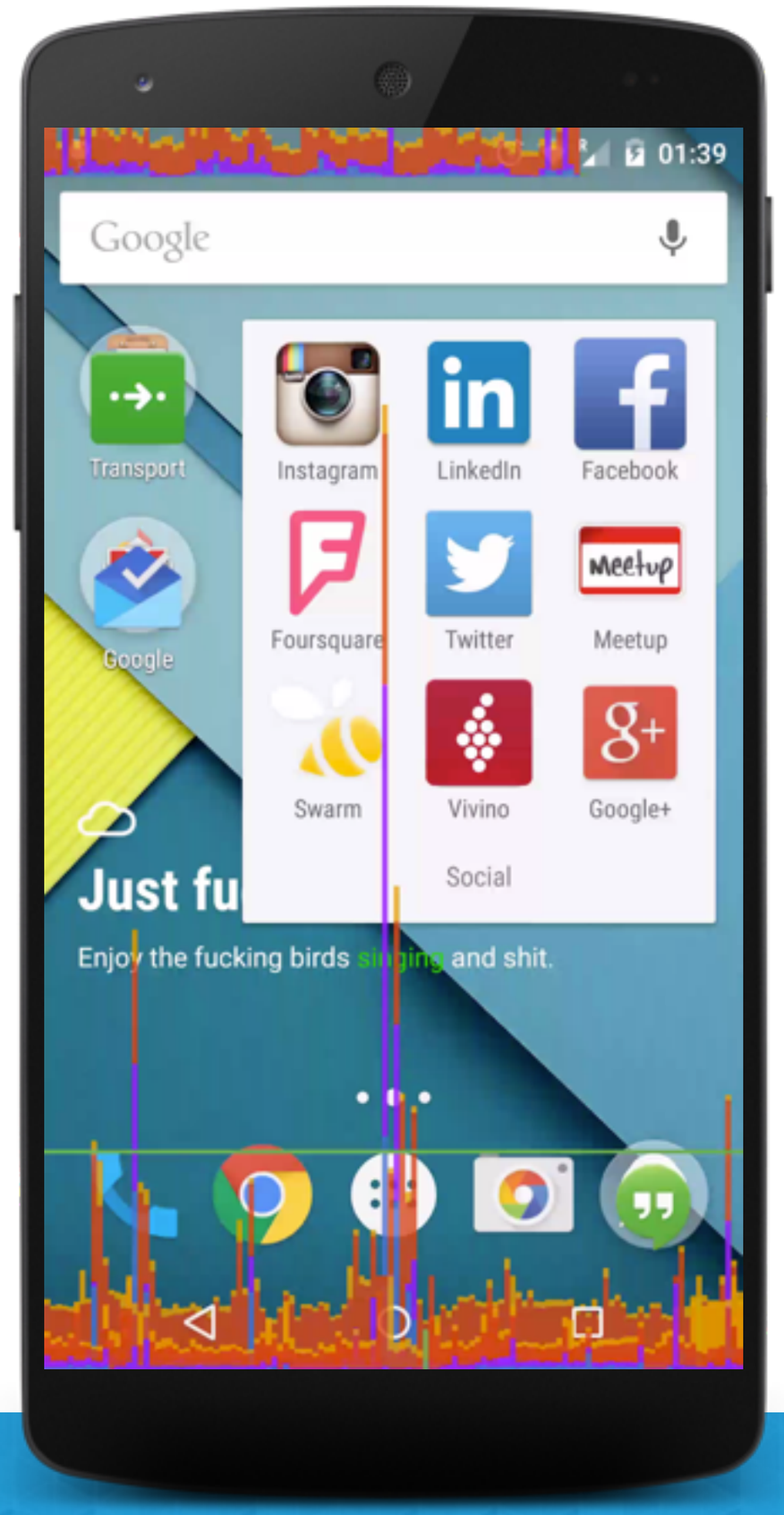
DetailActivity.java

```
@Override
public void onBackPressed() {
    mFab.animate()
            .translationYBy(2 * 56)
            .setInterpolator(new
                OvershootInterpolator(1.f))
            .setDuration(400)
            .withEndAction(new Runnable() {
                @Override
                public void run() {
                    finishAfterTransition();
                }
            });
}
```

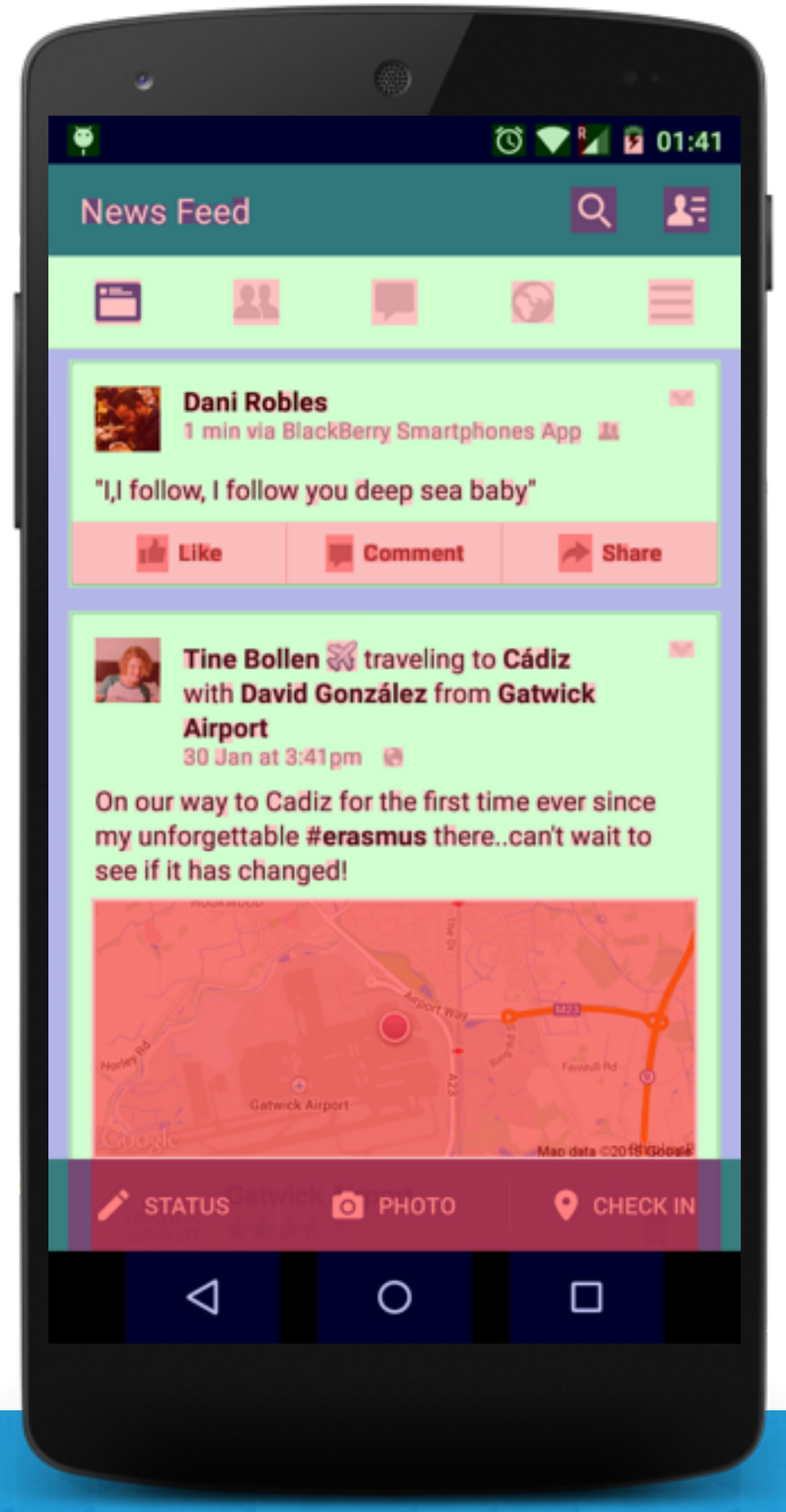
Performance tools

Join the **Android Performance Patterns G+ Community** to continue the discussion
goo.gl/g7mxml

Profile GPU Rendering



Show GPU Overdraw

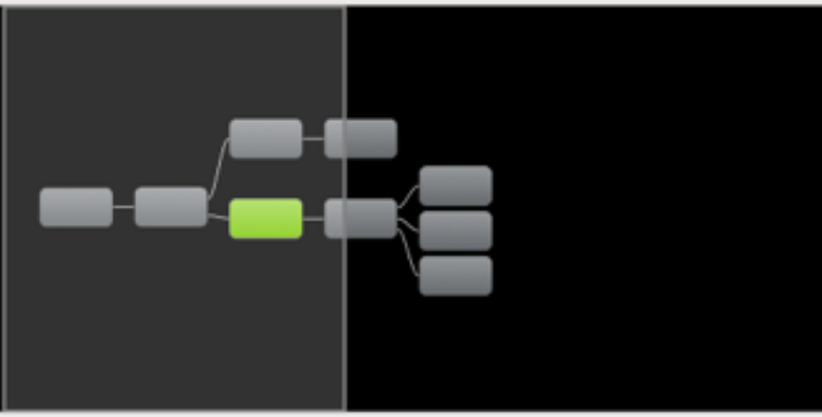


Hierarchy Viewer

le Tree View Help

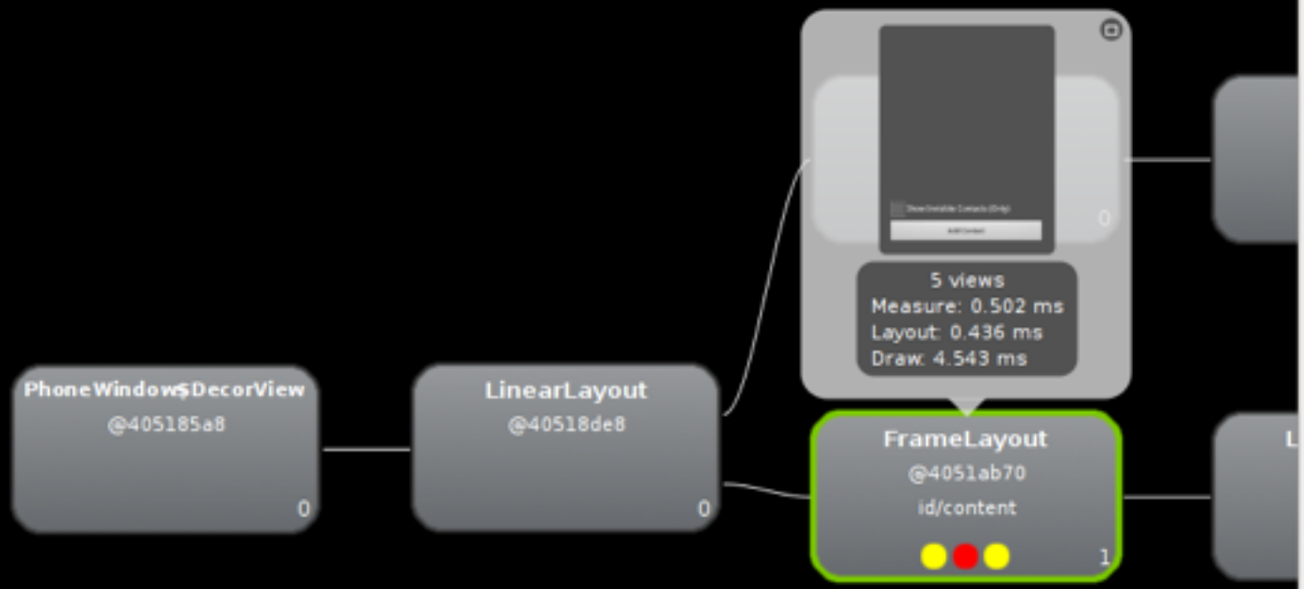
Save as PNG | Capture Layers | Load View Hierarchy | Display | Request Layout

Tree Overview pane

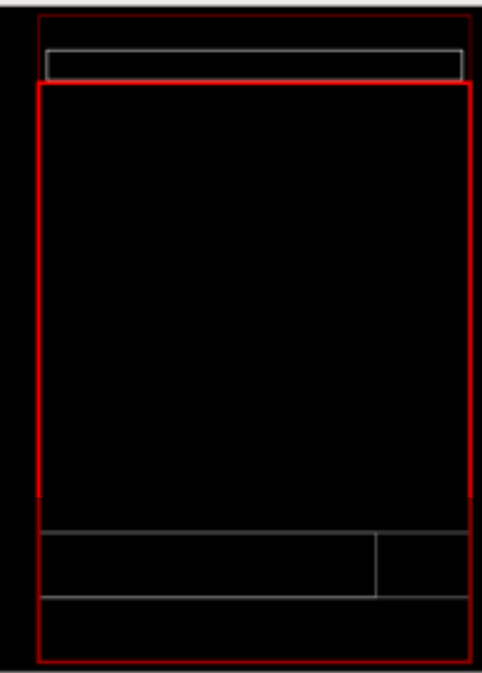


Property	Value
Drawing	
Focus	
Layout	
Measurement	
Miscellaneous	
Padding	
Scrolling	

Property View pane



Show Extras | Load All Views



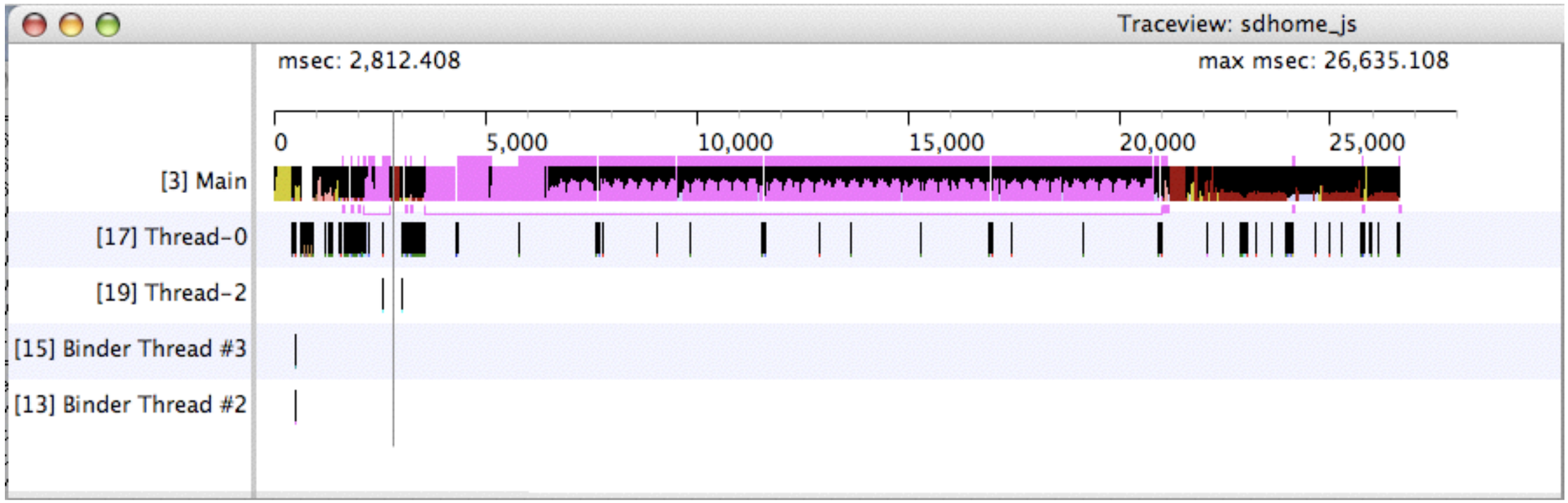
Layout View pane

Tree View pane

Filter by class or id: | 20% | 200%

Hierarchy Viewer

Traceview


Traceview: sdhome_js
msec: 2,812.408
max msec: 26,635.108
0
5,000
10,000
15,000
20,000
25,000
[3] Main
[17] Thread-0
[19] Thread-2
[15] Binder Thread #3
[13] Binder Thread #2

Fork me!

MaterialAnimations

Examples of animations and transitions

http://github.com/malmstein/MaterialAnimations

Thank you!
Questions?

@dggonzalez

+DavidGonzalezMalmstein

malmstein