

# Delighting Details - Animated Vector Drawables

An overview



**Marcos Damasceno**  
**@marcospaulosd**

# MIREGO

<http://life.mirego.com/>



Early Days

# Early Days

## ShapeDrawables

```
<?xml version="1.0" encoding="UTF-8"?>
<shape xmlns:android="http://schemas.android.com/apk/
res/android"
    android:shape="rectangle">
    <solid android:color="#007bb6"/>
    <corners android:bottomRightRadius="4dp"
        android:bottomLeftRadius="4dp"
        android:topLeftRadius="4dp"
        android:topRightRadius="4dp"/>
</shape>
```

# Early Days

ShapeDrawables

StateDrawables

TransitionDrawables



Nowadays



# Nowadays

## VectorDrawables

Introduced with Lollipop

Similar to SVG but a bit limited

Vector drawables automatically scales to the density of the device

# Nowadays

## VectorDrawables

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="48dp"
    android:height="48dp"
    android:viewportHeight="48"
    android:viewportWidth="48">

    <group
        android:name="group"
        android:pivotX="24"
        android:pivotY="24">
        <path android:pathData="M 0 0 L 48 0 L 48 48 L 0 48 Z" />
        <path
            android:name="play"
            android:fillColor="#FFFFFF"
            android:pathData="M 20 15 L 32 24 L 32 24 L 20 33 Z" />
    </group>
</vector>
```

**SVG**



# SVG

XML language, used to draw graphics

It supports gradient, rotations, filter effects, animations and so on.

```
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">  
  <path d="M10 10 H 90 V 90 H 10 L 10 10"/>  
</svg>
```

# SVG

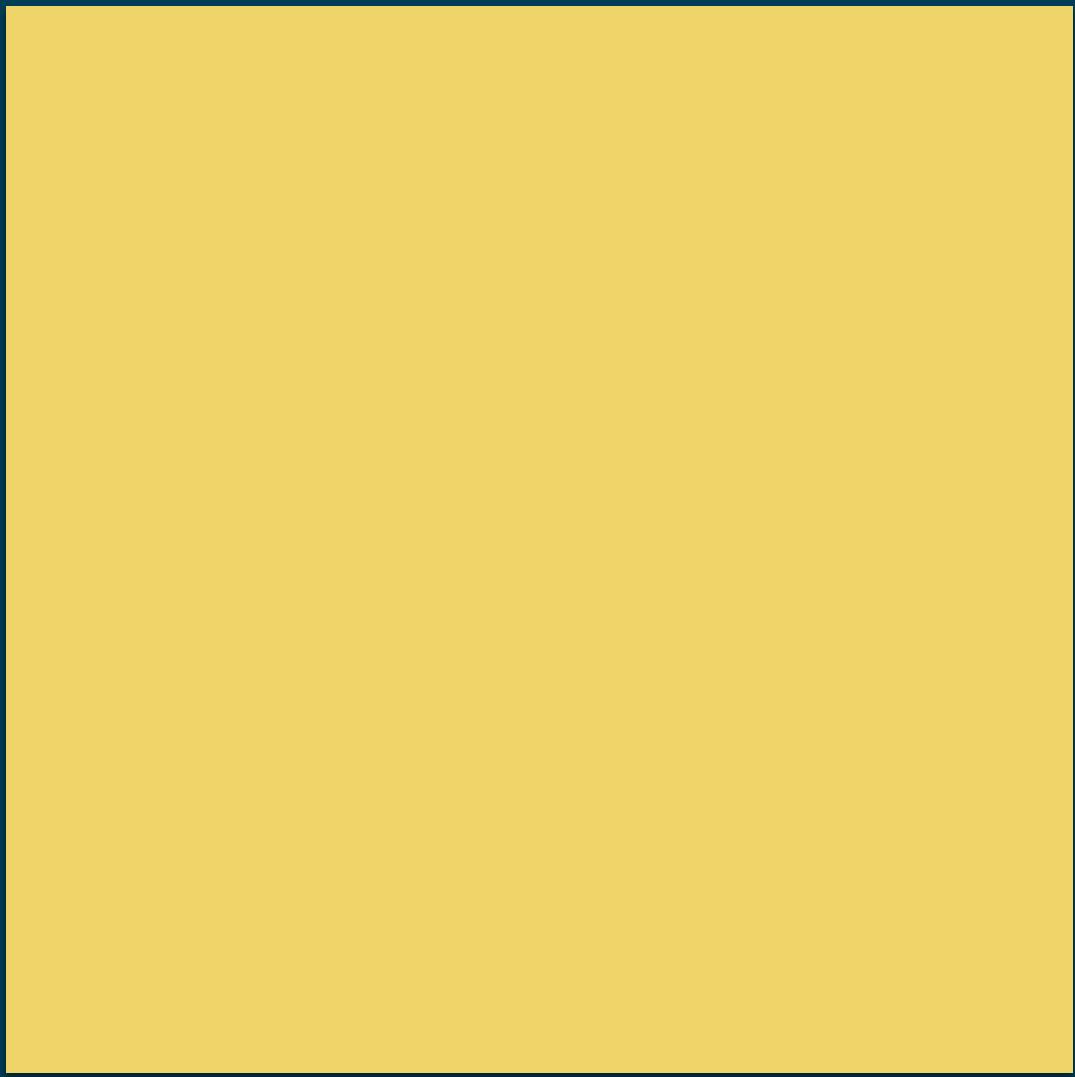
XML language, used to draw graphics

It supports gradient, rotations, filter effects, animations and so on.

```
<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">  
  <path d="M10 10 H 90 V 90 H 10 L 10 10"/>  
</svg>
```

# SVG

```
<path d="M10 10  
H 90 V 90  
H 10 L 10 10"/>
```



# SVG

```
<path d="M10 10  
        H 90 V 90  
        H 10 L 10 10"/>
```

There are 5 line commands for drawing paths.  
Each command draw a straight line between 2 points

# SVG

```
<path d="M10 10  
H 90 V 90  
H 10 L 10 10"/>
```

There are 5 line commands for drawing paths.  
Each command draw a straight line between 2 points

M - Move to

H - Horizontal Line

Z - Close Path

L - Line To

V - Vertical Line

# SVG

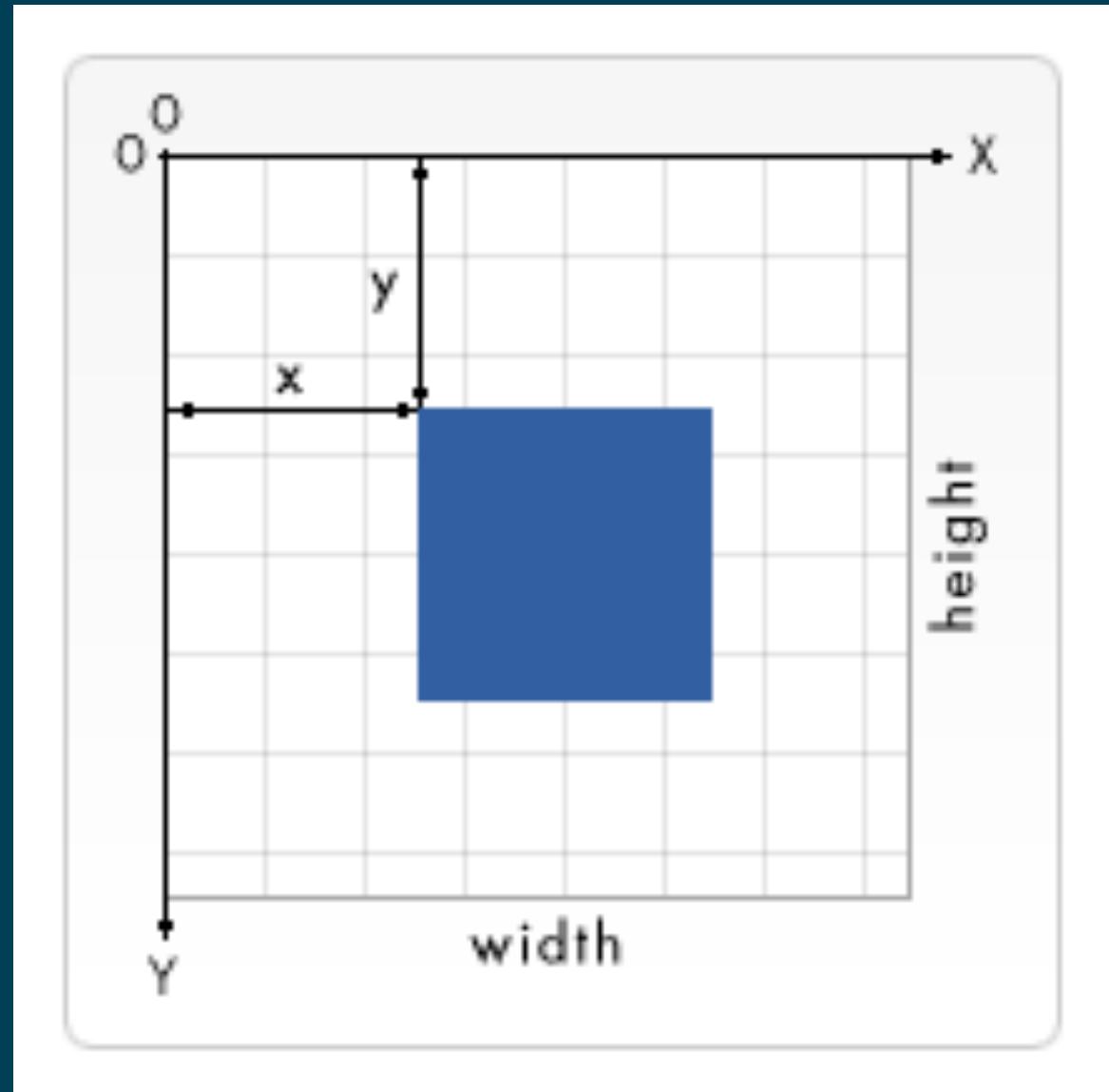
```
<path d="M10 10
```

M is always the first command in a path data, it means “Move To”,

M - Move to

# SVG

```
<path d="M10 10
```



M is always the first command in a path data, it means “Move To”,

M - Move to

# SVG



```
<path d="M10 10
```

M is always the first command in a path data, it means “Move To”,

M - Move to

# SVG

```
<path d="M10 10  
      H 90
```



H draws a horizontal line

M - Move to

H - Horizontal Line

# SVG

```
<path d="M10 10  
      H 90 V 90
```



V draws a vertical line

M - Move to

H - Horizontal Line

V - Vertical Line

# SVG

```
<path d="M10 10  
      H 90 V 90  
      H 10 L 10 10"/>
```



L draws a line from the last point until a coordinate x or y

M - Move to

H - Horizontal Line

V - Vertical Line

L - Line To

# SVG

```
<path d="M10 10  
H 90 V 90  
H 10 L H 10 Z"/>
```



We could even simplify using H 10 and then asking it to close the path

M - Move to

H - Horizontal Line

V - Vertical Line

L - Line To

Z - Close Path

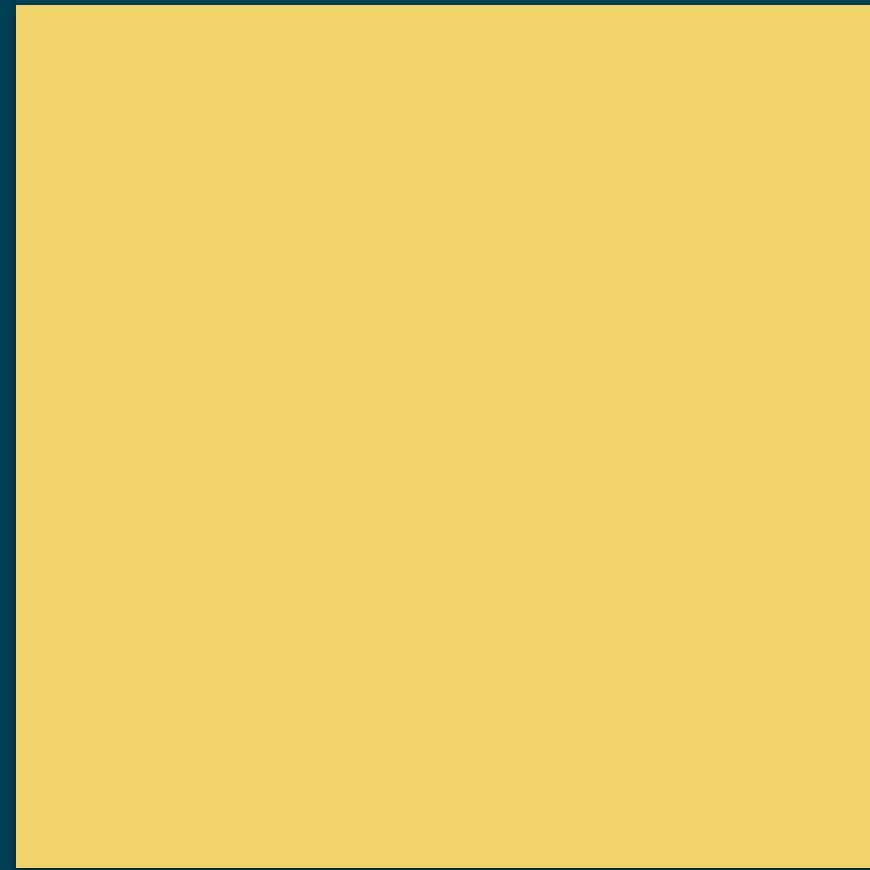
# SVG

```
<path  
d="M10 10 H 90 V 90 H 10 L H 10 Z"  
/>
```



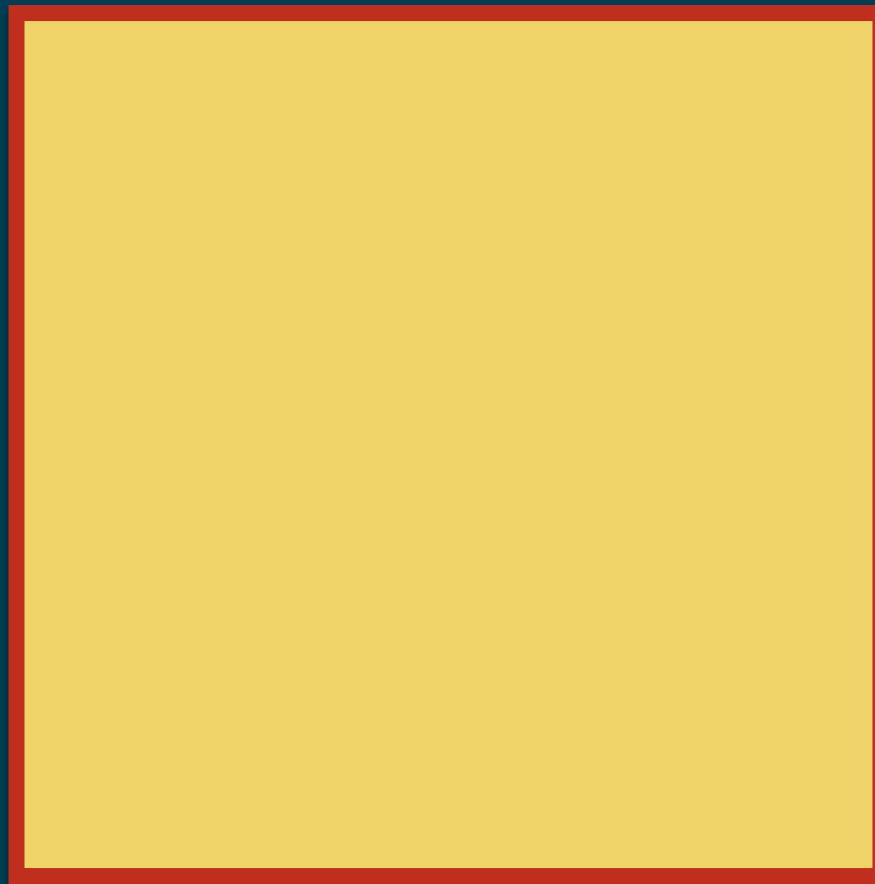
# SVG

```
<path  
d="M10 10 H 90 V 90 H 10 L H 10 Z"  
fill="#EBCD57"  
/>
```



# SVG

```
<path  
d="M10 10 H 90 V 90 H 10 L H 10 Z"  
fill="#EBCD57"  
stroke="#C02F1D"  
/>
```



# SVG

```
<path  
    d="M10 10 H 90 V 90 H 10 L H 10 Z"  
    fill="#EBCD57"  
    stroke="#C02F1D"  
/>
```

# Vector Drawable

```
<path  
    android:strokeColor="#C02F1D"  
    android:fillColor="#EBCD57"  
    android:pathData="M10 10 H 90 V 90 H 10 L H 10 Z" />
```

# Vector Drawable: Properties

<group>

    android:rotation  
    android:pivotX  
    android:pivotY  
    android:scaleX  
    android:scaleY  
    android:translateX  
    android:translateY

<path>

    android:pathData  
    android:fillColor  
    android:strokeColor  
    android:strokeWidth  
    android:strokeAlpha  
    android:fillAlpha  
    android:trimPathOffset  
    android:trimPathStart  
    android:trimPathEnd

# Vector Drawable: Properties

```
<path>  
  
    android:pathData  
    android:trimPathOffset  
    android:trimPathStart  
    android:trimPathEnd
```



Animated Vector Drawables



<https://www.youtube.com/watch?v=jQvSXghFwxw>

# Animated Vector Drawables

It allow us to apply animations for some properties of a Vector Drawable

Animating all properties are natively available on Lollipop+ and available for Kitkat and before via `VectorDrawableCompat`.

The only exception is `PathData` animation which is not available for Kitkat and before

# Animated Vector Drawables

Properties we will cover

TrimPathOffset

TrimPathStart

PathData

TrimPathEnd

# Animated Vector Drawables

**TrimPathStart**

**TrimPathOffset**

**TrimPathEnd**



# Animated Vector Drawables

**TrimPathStart**

The fraction of the path to trim from the start, in the range from 0 to 1.



# Animated Vector Drawables

## TrimPathEnd

The fraction of the path to trim from the end, in the range from 0 to 1.



# Animated Vector Drawables

## TrimPathOffset

Shift trim region (allows showed region to include the start and end), in the range from 0 to 1.



# Animated Vector Drawables

TrimPathStart

TrimPathOffset

TrimPathEnd

LIVE DEMONSTRATION

# Animated Vector Drawables

Working with the vector

LIVE DEMONSTRATION

# Animated Vector Drawables

final vector drawable file



```
<vector
    ...
    <path
        android:name="head"
        android:pathData="M14,5...L14,5 Z"
        android:strokeColor="#FFFFFF"
        android:strokeWidth="2.2"/>
    <path
        android:name="body"
        android:pathData="M5,21 ...18.34 5,21 Z"
        android:strokeColor="#FFFFFF"
        android:strokeWidth="2.2"/>
    <path
        android:name="line"
        android:pathData="M14,13 ... 5,21"
        android:strokeColor="#FFFFFF"
        android:strokeWidth="2.2"/>
</vector>
```

# Animated Vector Drawables

To create an Animated Vector Drawable we  
need to have at least three components

```
<vector
```

```
<animated-vector
```

```
<objectAnimator
```



# Set the vector to the initial position

## ic\_profile\_outlined.xml

```
<vector  
    ...  
    <path  
        android:name="head"  
        android:pathData="M14,5...L14,5 Z"  
        android:strokeColor="#FFFFFF"  
        android:strokeWidth="2.2"  
        android:trimPathEnd="0"  
        android:trimPathOffset="0.5" />  
    </vector>
```



# Set the vector to the initial position

## ic\_profile\_outlined.xml

```
<vector  
    ...  
    <path  
        android:name="head"  
        android:pathData="M14,5...L14,5 Z"  
        android:strokeColor="#FFFFFF"  
        android:strokeWidth="2.2"  
        android:trimPathEnd="0"  
        android:trimPathOffset="0.5" />  
</vector>
```



# Set the vector to the initial position

## ic\_profile\_outlined.xml

```
<vector  
    ...  
    <path  
        android:name="body"  
        android:pathData="M5,21 ...18.34 5,21 Z"  
        android:strokeColor="#FFFFFF"  
        android:strokeWidth="2.2"  
        android:trimPathEnd="0" />  
</vector>
```



# Set the vector to the initial position

## ic\_profile\_outlined.xml

```
<vector  
    ...  
    <path  
        android:name="line"  
        android:pathData="M14,13 ... 5,21"  
        android:strokeColor="#FFFFFF"  
        android:strokeWidth="2.2"  
        android:trimPathEnd="0" />  
</vector>
```



# Set Up Animations

## reveal\_profile\_head\_icon\_animation.xml

```
<set  
    <objectAnimator  
        android:duration="600"  
        android:propertyName="trimPathEnd"  
        android:valueFrom="0"  
        android:interpolator="@android:  
interpolator/accelerate_cubic"  
        android:valueTo="1"/>  
    </set>
```

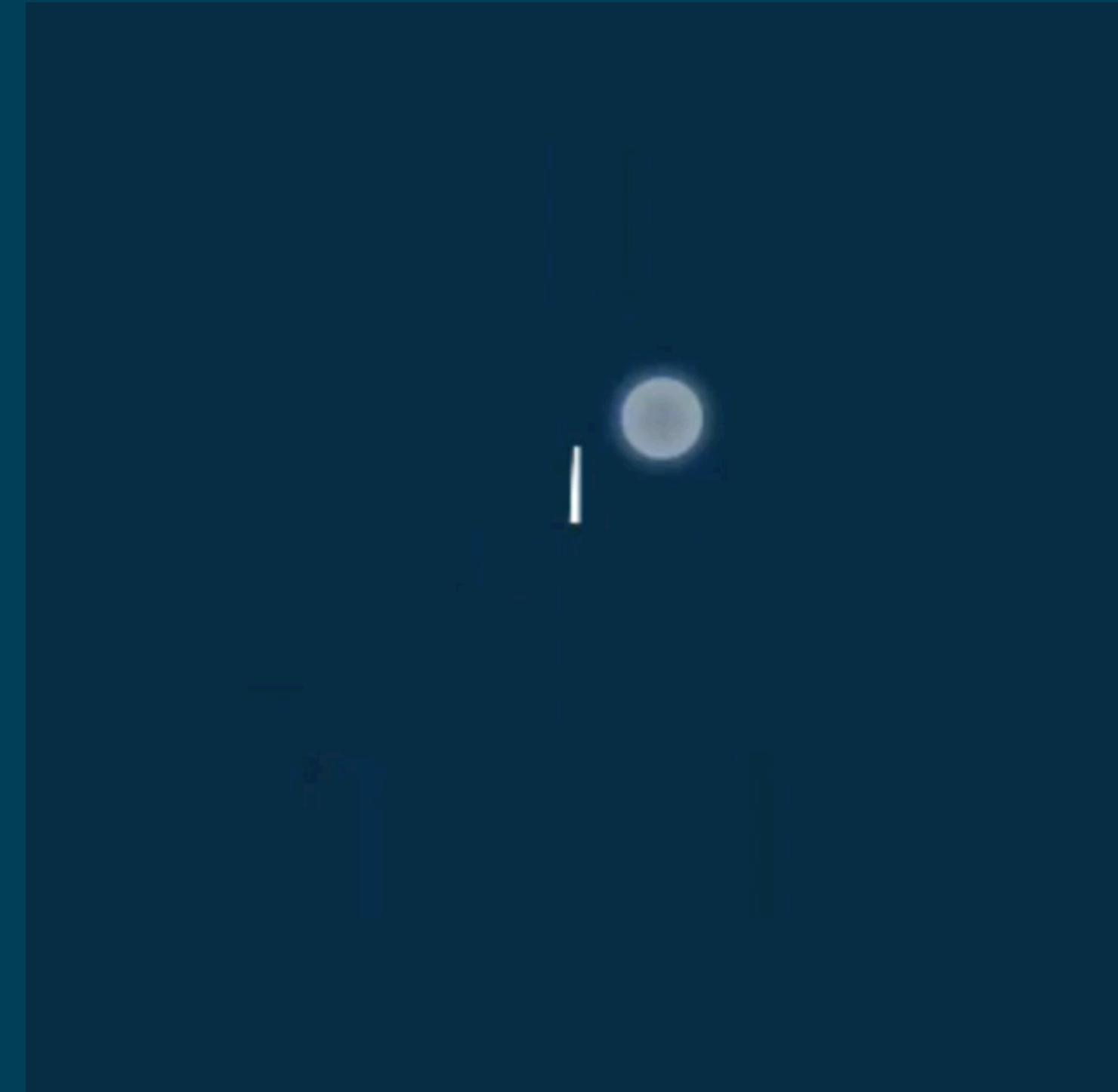


Interpolators are important

With interpolator



No Interpolator



# Set Up Animations

## reveal\_connection.xml

```
<set android:ordering="sequentially">
    <objectAnimator
        android:duration="200"
        android:propertyName="trimPathEnd"
        android:valueFrom="0"
        android:startOffset="500"
        android:interpolator=
            "@android:interpolator/linear"
        android:valueTo="1"/>

    <objectAnimator
        android:duration="300"
        android:propertyName="trimPathStart"
        android:valueFrom="0"
        android:startOffset="300"
        android:interpolator=
            "@android:interpolator/decelerate_cubic"
        android:valueTo="1"/>
</set>
```



# Set Up Animations

## reveal\_connection.xml

```
<set android:ordering="sequentially">
    <objectAnimator
        android:duration="200"
        android:propertyName="trimPathEnd"
        android:valueFrom="0"
        android:startOffset="500"
        android:interpolator=
            "@android:interpolator/linear"
        android:valueTo="1"/>

    <objectAnimator
        android:duration="300"
        android:propertyName="trimPathStart"
        android:valueFrom="0"
        android:startOffset="300"
        android:interpolator=
            "@android:interpolator/decelerate_cubic"
        android:valueTo="1"/>
</set>
```



# Set Up Animations

## reveal\_body.xml

```
<set>
    <objectAnimator
        android:duration="600"
        android:propertyName="trimPathEnd"
        android:valueFrom="0"
        android:startOffset="800"
        android:valueTo="1"/>
</set>
```



# Set Up Animations

## avd\_ic\_profile.xml

```
<animated-vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:drawable="@drawable/ic_player" >  
    <target  
        android:name="body"  
        android:animation="@anim/reveal_player_body_icon_animation" />  
    <target  
        android:name="head"  
        android:animation="@anim/reveal_player_head_icon_animation" />  
    <target  
        android:name="line"  
        android:animation="@anim/reveal_connection" />  
</animated-vector>
```

# Making the animation alive

```
ImageView icProfile = (ImageView) findViewById(R.id.ic_profile);  
  
final AnimatedVectorDrawable avdProfile = (AnimatedVectorDrawable)  
icProfile.getDrawable();  
  
icProfile.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        avdProfile.start();  
    }  
});
```



# Animated Vector Drawables

Properties we will cover

TrimPathOffset

TrimPathStart

PathData

TrimPathEnd

# Animated Vector Drawables

**PathData**



# Animated Vector Drawables

Same concept

<vector

<animated-vector

<objectAnimator



# Animated Vector Drawables

Vectors:

```
ic_play.xml  
ic_stop.xml
```

Animated Vector XML:

```
avd_play_to_stop.xml  
avd_stop_to_play.xml
```



# Animated Vector Drawables

Animator gets tricky.

ic\_play.xml

```
<path
    android:name="play"
    android:fillColor="@android:color/white"
    android:pathData="M 20 15 L 32 24 L 20 33 Z" />
```

ic\_stop.xml

```
<path
    android:name="stop"
    android:fillColor="@android:color/white"
    android:pathData="M 16 16 L 32 16 L 32 32 L 16 32 Z" />
```



# Animated Vector Drawables

Animator gets tricky.

from:

```
android:pathData="M 20 15 L 32 24 L 20 33 Z" />
```

to:

```
android:pathData="M 16 16 L 32 16 L 32 32 L 16 32 Z" />
```

# Animated Vector Drawables

Animator gets tricky.

from:

```
android:pathData="M 20 15 L 32 24          L 20 33 Z" />
```

to:

```
android:pathData="M 16 16 L 32 16 L 32 32 L 16 32 Z" />
```

# Animated Vector Drawables

Animator gets tricky.

from:

```
android:pathData="M 20 15 L 32 24 L 32 24 L 20 33 Z" />
```

to:

```
android:pathData="M 16 16 L 32 16 L 32 32 L 16 32 Z" />
```

# Animated Vector Drawables

anim\_play\_to\_stop\_xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

    <objectAnimator
        android:duration="300"
        android:interpolator="@android:anim/accelerate_decelerate_interpolator"
        android:propertyName="pathData"
        android:valueFrom="M 20 15 L 32 24 L 32 24 L 20 33 Z"
        android:valueTo="M 16 16 L 32 16 L 32 32 L 16 32 Z"
        android:valueType="pathType" />

</set>
```

# Animated Vector Drawables

anim\_stop\_to\_play\_xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

    <objectAnimator
        android:duration="300"
        android:interpolator="@android:anim/accelerate_decelerate_interpolator"
        android:propertyName="pathData"
        android:valueFrom="M 16 16 L 32 16 L 32 32 L 16 32 Z"
        android:valueTo="M 20 15 L 32 24 L 32 24 L 20 33 Z"
        android:valueType="pathType" />

</set>
```

# Animated Vector Drawables

OMG! <animated-vector>, <vector> and a <animator>?

TooMuchFilesException!

**XML bundle format**

**For the rescue**

**Available since Build Tools 24**

**Let you merge the 3 xml files in one only  
using the aapt namespace**

# AnimatedStateListDrawable

Available for  
Marshmallow+

Set up transition for  
changing states easily

```
<animated-selector android:constantSize="true">
    <item
        android:id="@+id/playing"
        android:state_checked="true"
        android:drawable="@drawable/ic_stop" />

    <item
        android:id="@+id/stopped"
        android:drawable="@drawable/ic_play" />

    <transition
        android:fromId="@+id/stopped"
        android:toId="@+id/playing"
        android:drawable="@drawable/avd_play_to_stop" />

    <transition
        android:fromId="@+id/playing"
        android:toId="@+id/stopped"
        android:drawable="@drawable/avd_stop_to_play" />

</animated-selector>
```

Animated Vector Drawables allow us to delight our users  
and to put the attention to the user in the right place

And they're awesome.



**Marcos Damasceno**  
**@marcospaulosd**

**Thank You**