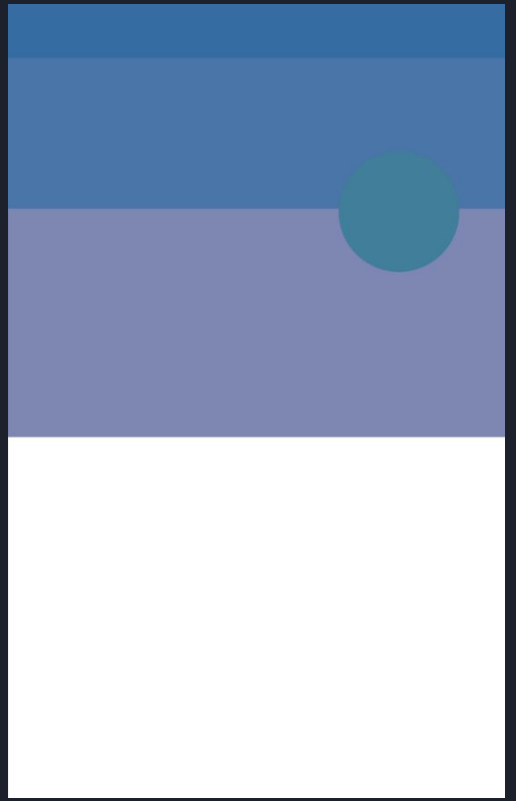
View Rendering and Animation



UI Rendering

- generating a frame & drawing on the screen
- 16 ms to achieve 60 frames per second
- what if frame gets dropped?

Reusing Frames

- refresh rate and frame rate
- vsync maintains two buffers:
  - back buffer
  - frame buffer
- frame is rendered to back buffer first

Challenges

- too much work for ui thread
- laggy animations

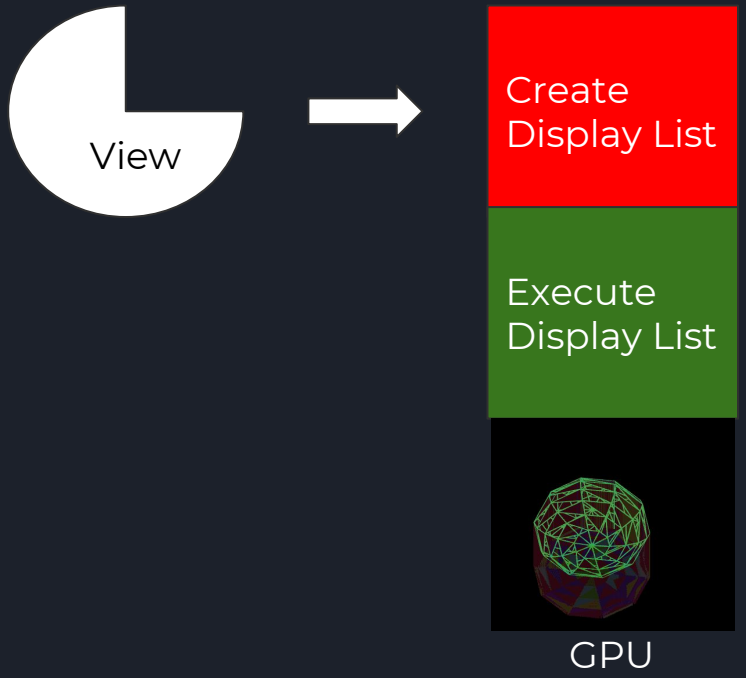
Drawing Modes

- software based drawing mode
  - only cpu
- hardware based (introduced in API 11)
  - both cpu and gpu

Solution?

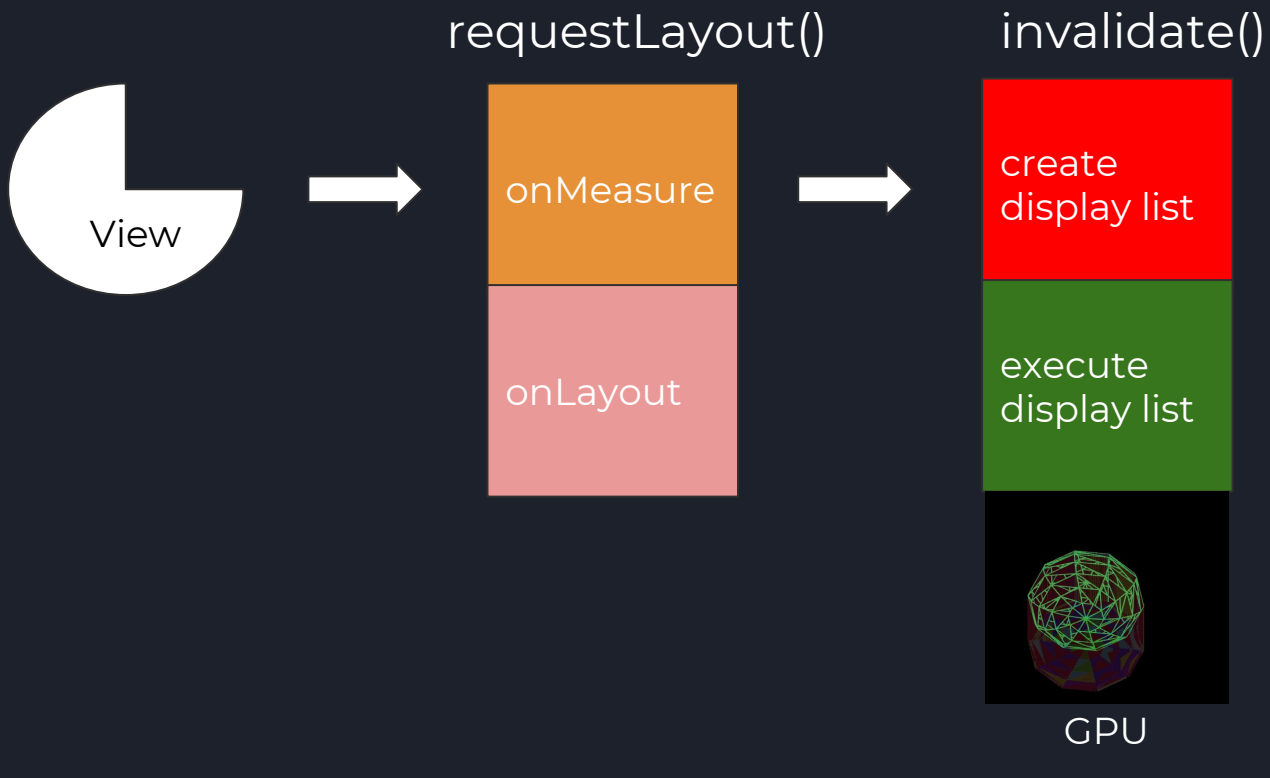
- ui and render thread (API 21)
- ui thread: record view (#draw)
  - onMeasure
  - onLayout
  - onDraw
  - creating set of drawing operations
- render thread: frame drawing
  - executing drawing operations

View Creation



Display List = Set of drawing operations

View Creation

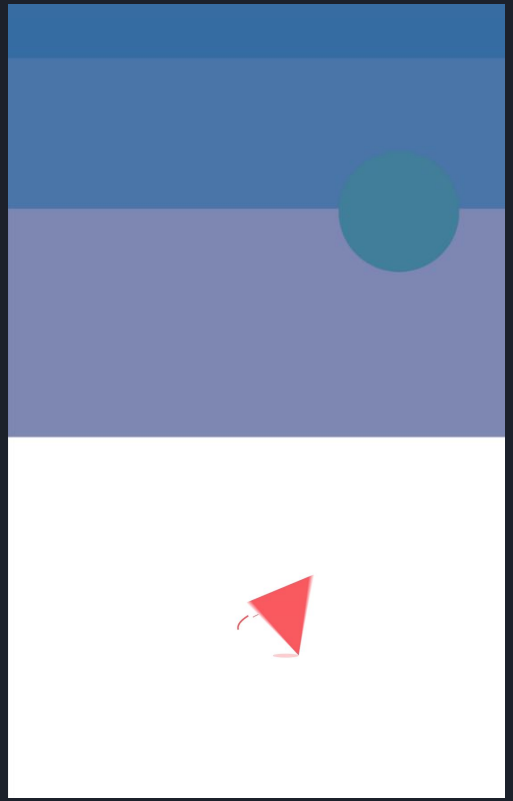


What else about Views?

- by default view bounds are rectangle
- views can have different shapes

View Shapes

- xml drawables
- custom layouts



Android Outline API

- defines a simple shape, used for bounding graphical regions.

- can be computed for a View
  - Provided by drawable
  - providing a custom outline

activity_layout.xml

```
<FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/bg_outline_sample"/>
```

bg_outline_sample.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="url"
         android:shape="rectangle">
        <size android:width="90dp" android:height="90dp"/>
        <solid android:color="#ffebeb" />
</shape>
```

Outline Clipping

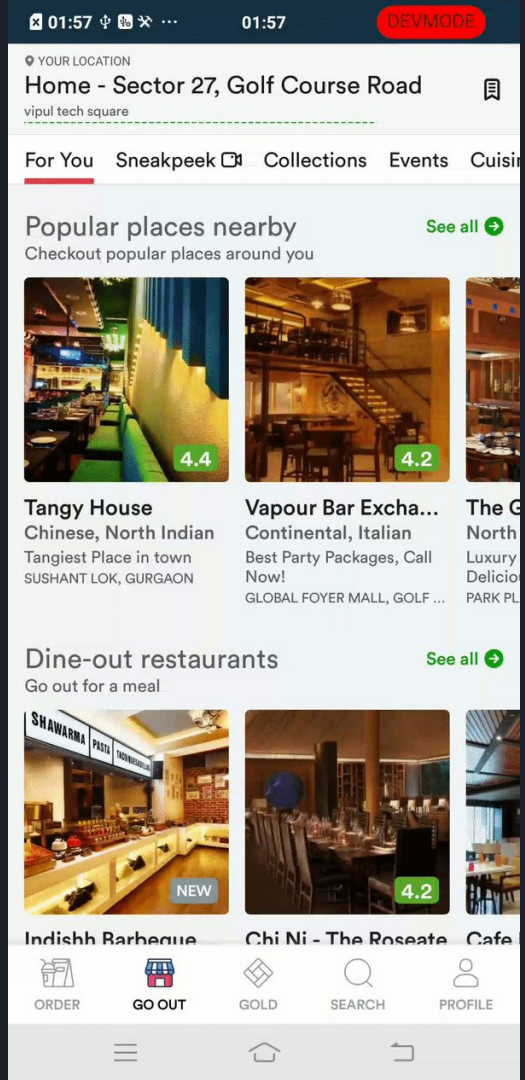
- native way of clipping views
- produces antialiased outlines
- setRect(), setRoundRect(), setOval()

ViewOutlineProvider

```
class MyCustomViewOutlineProvider : ViewOutlineProvider {

        @Override
        fun getOutline(view View, outline Outline) {
                outline.setRect(0, 0, view.width, view.height);
        }
}
```

set view.clipToOutline = true and you are done!

Use Case

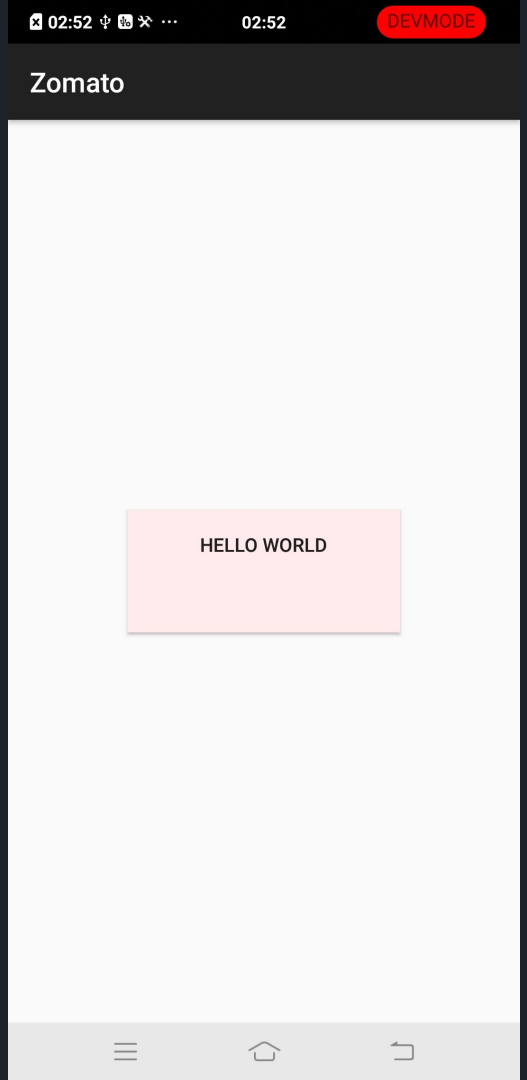

01:57
01:57
DEVMODE
YOUR LOCATION
Home - Sector 27, Golf Course Road
vipul tech square
For You
Sneakpeek
Collections
Events
Popular places nearby
See all
Checkout popular places around you
4.4
Tangy House
Chinese, North Indian
Tangiest Place in town
SUSHANT LOK, GURGAON
4.2
Vapour Bar Excha...
Continental, Italian
Best Party Packages, Call Now!
GLOBAL FOYER MALL, GOLF ...
Dine-out restaurants
See all
Go out for a meal
SHAWARMA
PASTA
NEW
4.2
ORDER
GO OUT
GOLD
SEARCH
PROFILE

Performance

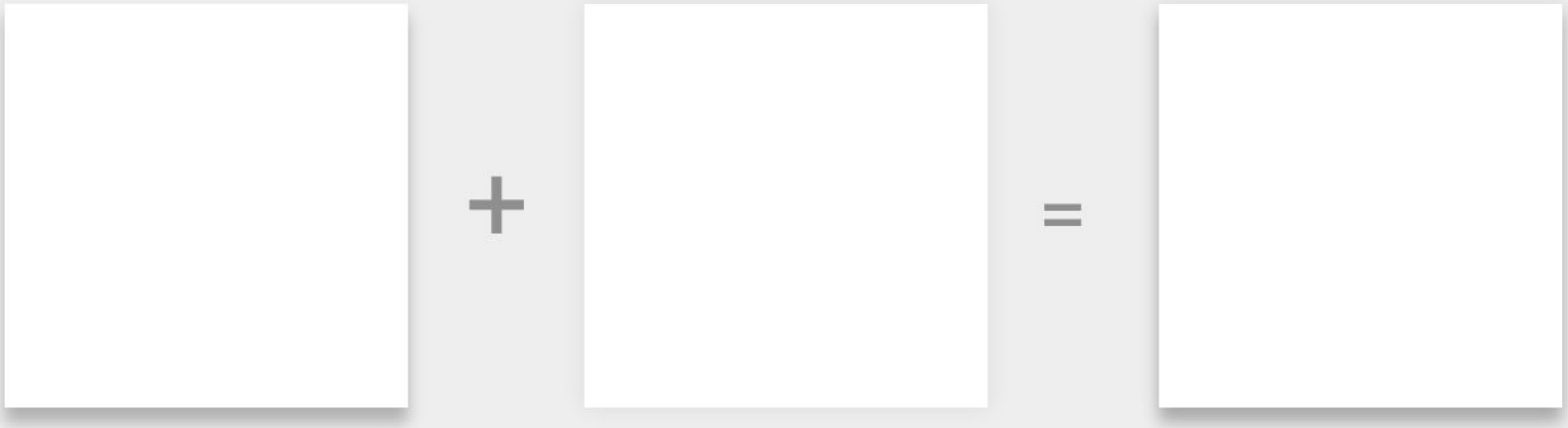
- 11% cpu performance improvement
- 17% memory efficient

Outline Shadows

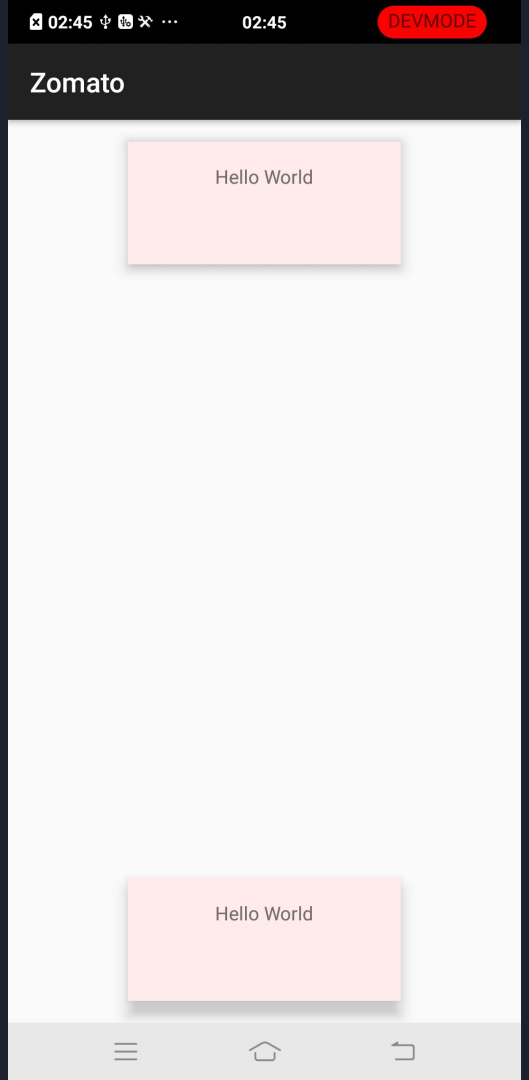
```
<Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:elevation="8dp"
        android:gravity="center_horizontal"
        android:background="@drawable/bg_outline_sample"
        android:padding="@dimen/activity_horizontal_margin"
        android:layout_gravity="center_horizontal"
        />
```

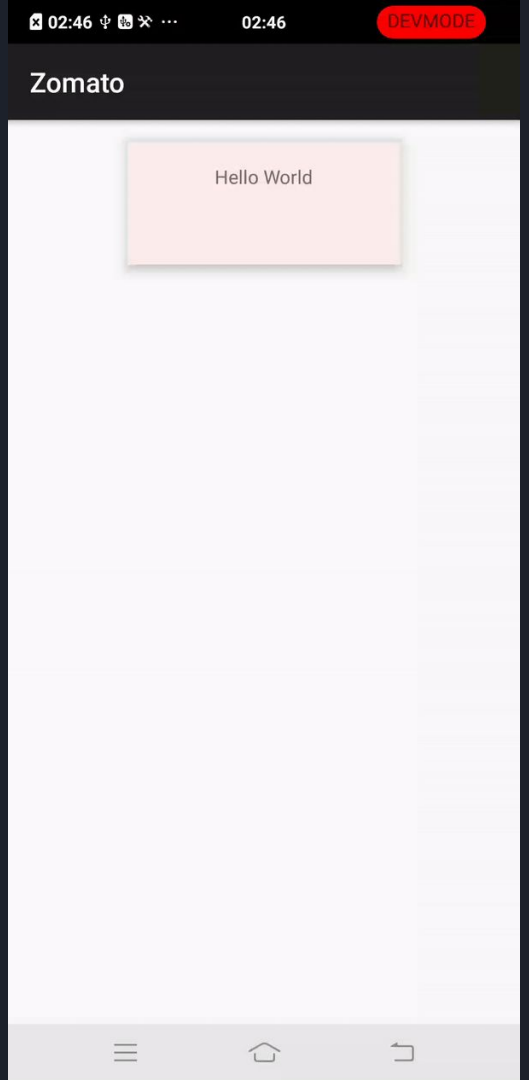


Android Shadow



key light ambient light result


02:45
02:45
DEVMODE
Zomato
Hello World
Hello World


02:46
02:46
DEVMODE
Zomato
Hello World

Motion Layout

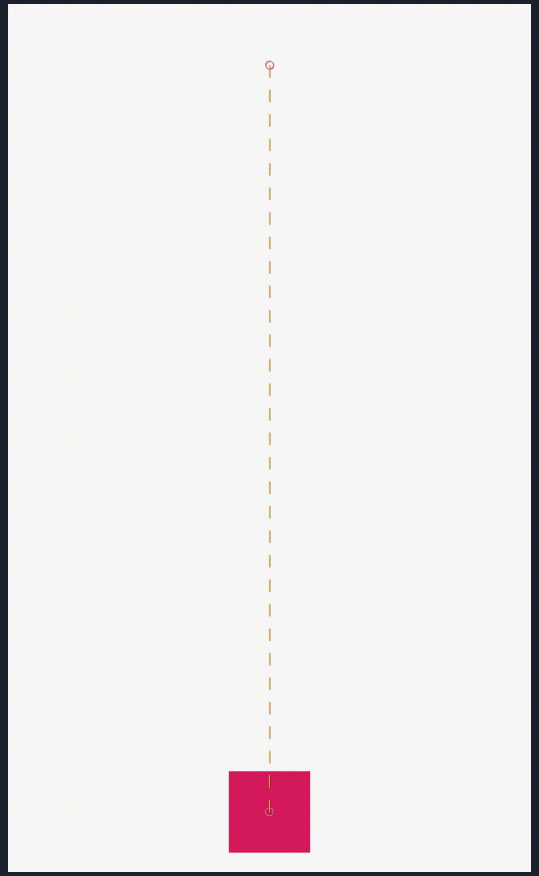
Topics to cover

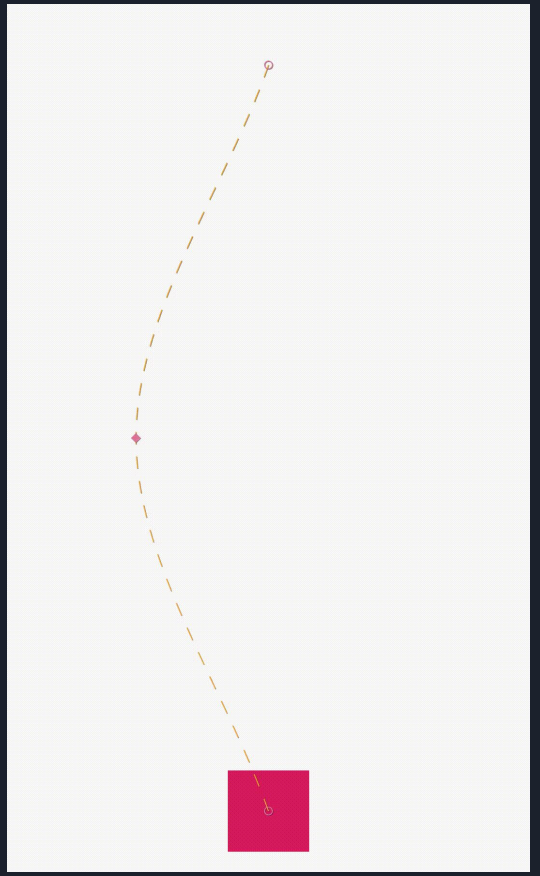
1. What is Motion Layout?
2. Motion Scene
3. Key Frames
4. Can we use Motion Layout on an our existing project?

What is Motion Layout?

- A mix of animation property framework, transition manager and coordinator layout.
- Extends Constraint Layout.
- Should be used if there is a ‘transition intent’ from the user.
- Effective only on direct children.
- Fully declarative.

Why use Motion Layout when you have Transition Manager?





What’s so different about Motion Layout from Constraint Layout?

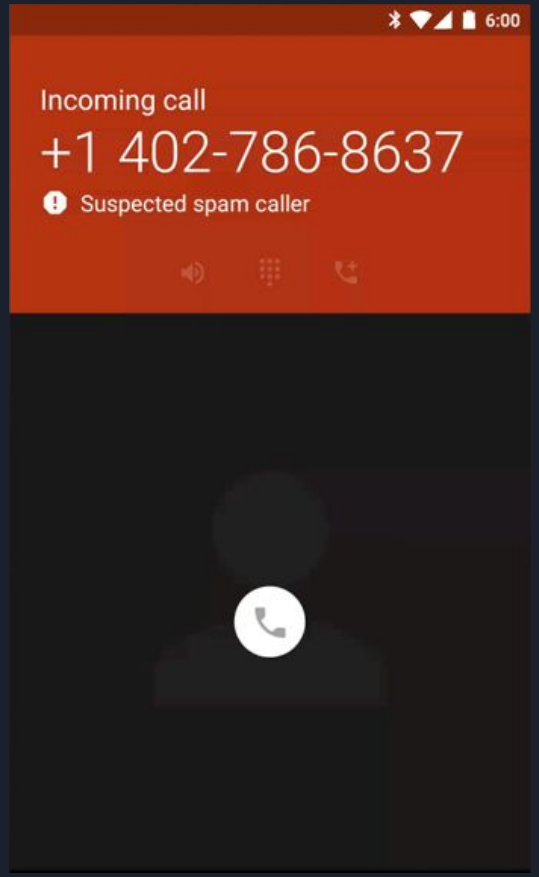
Layout XML

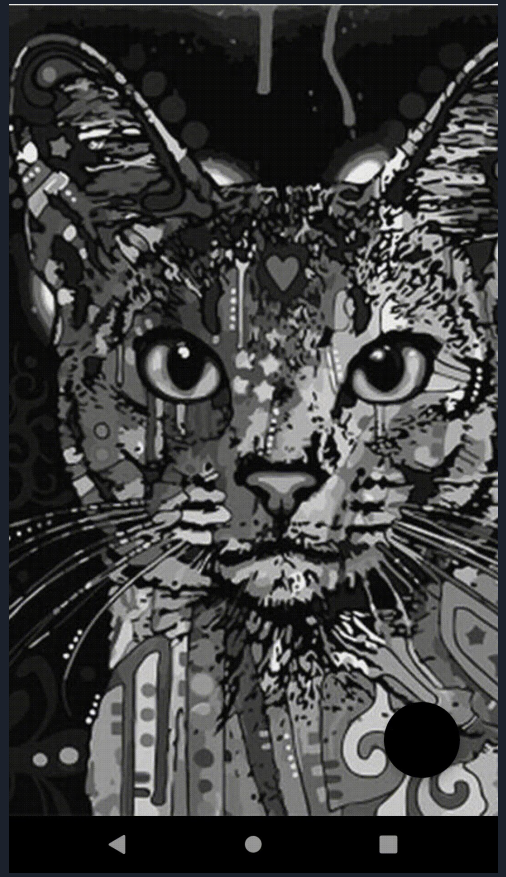
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.motion.widget.MotionLayout
        xmlns:android="http://schemas.android.com/apk/res/
        xmlns:tools="http://schemas.android.com/tools"
        xmlns:app="http://schemas.android.com/apk/res-auto
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:showPaths="true"
        android:id="@+id/ml_box"
        app:layoutDescription="@xml/motion_scene_two"
        tools:context=".views.activities.BoxActivity">

    <View
            android:id="@+id/box_view"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:background="@color/colorAccent"/>

</androidx.constraintlayout.motion.widget.MotionLayout>
```

Motion Scene XML

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


    <!--Box Start-->
    <ConstraintSet
            android:id="@+id/start">

        <Constraint
                android:id="@id/box_view"
                android:layout_width="64dp"
                android:layout_height="64dp"
                android:layout_marginTop="16dp"
                motion:layout_constraintTop_toTopOf="parent"
                motion:layout_constraintLeft_toLeftOf="parent"
                motion:layout_constraintRight_toRightOf="parent">

            <CustomAttribute
                    motion:attributeName="backgroundColor"
                    motion:customColorValue="@color/colorAccent"/>
        </Constraint>
    </ConstraintSet>



    <!--Box End-->
    <ConstraintSet
            android:id="@+id/end">
        <Constraint
                android:id="@id/box_view"
                android:layout_width="64dp"
                android:layout_height="64dp"
                android:layout_marginBottom="16dp"
                motion:layout_constraintBottom_toBottomOf="parent"
                motion:layout_constraintLeft_toLeftOf="parent"
                motion:layout_constraintRight_toRightOf="parent">

            <CustomAttribute
                    motion:attributeName="backGroundColor"
                    motion:customColorValue="@color/colorPrimary"/>
        </Constraint>
    </ConstraintSet>
```

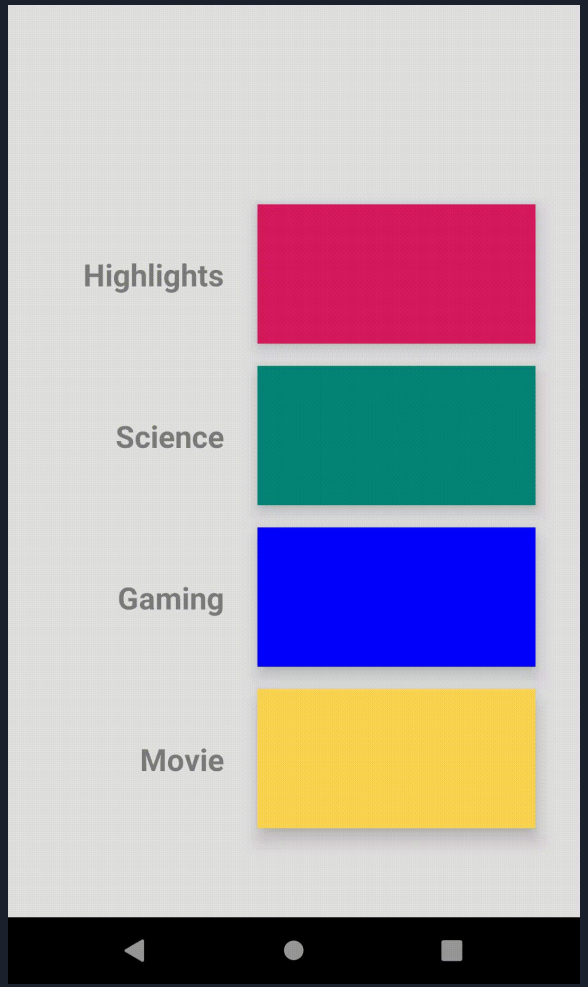
Should we use Motion Layout everywhere?


6:00
Incoming call
+1 402-786-8637
Suspected spam caller



Motion Scene

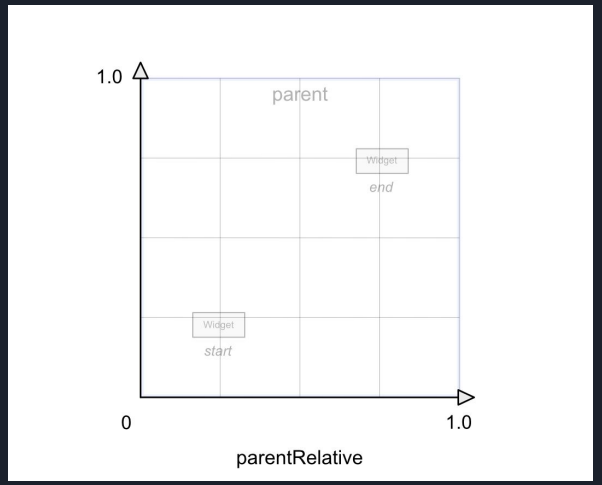
- Constraint Sets will replace all constraints defined in the layout XML.
- Custom Attributes use the view's properties to apply custom values.
- Start and layout constraints are finally defined inside the Transition tag.
- OnAction tags accept the view's ID, direction/mode

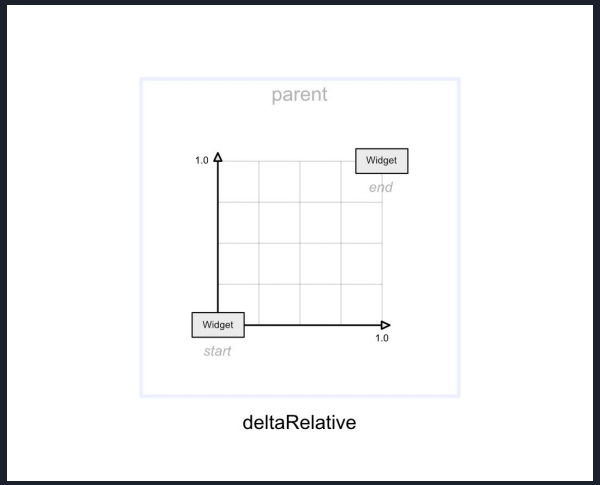

Highlights
Science
Gaming
Movie

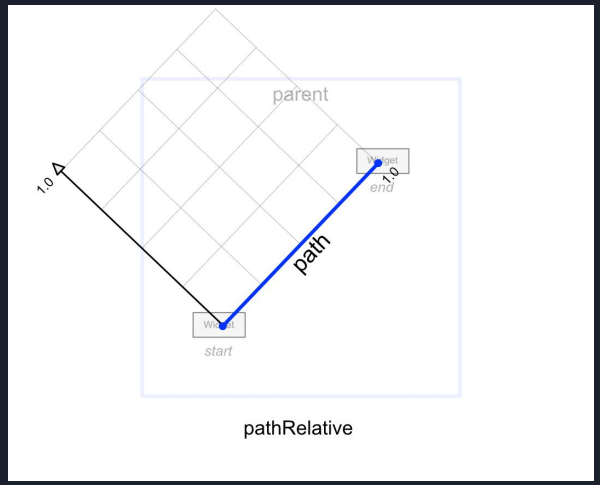
Key Frames

- Control your animation on each frame.
- Translate your animating view with Key Position.
- Scale, Rotate, Skew your view with Key Attribute.
- Make crazy oscillating animations with Key Cycle.
- Define Arc paths in your XML

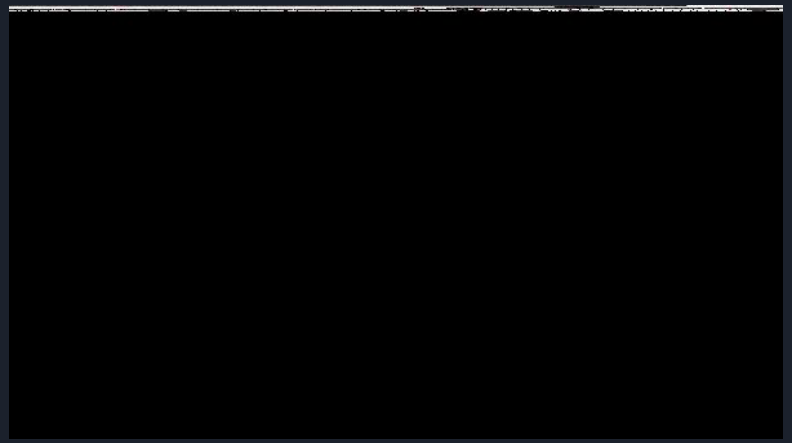
Understanding Key Position Types

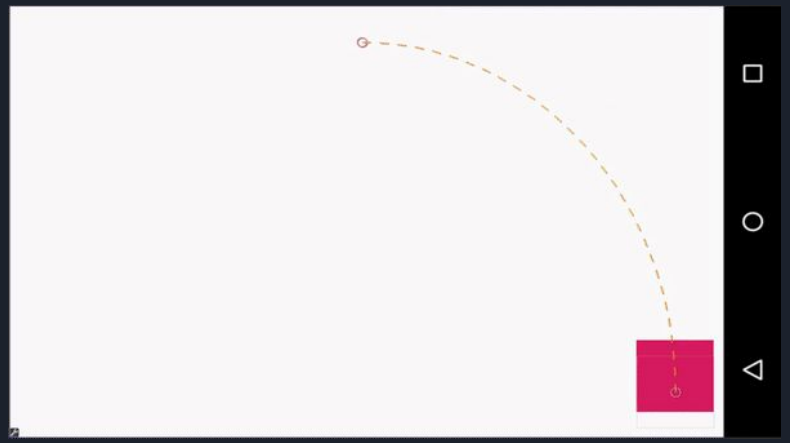






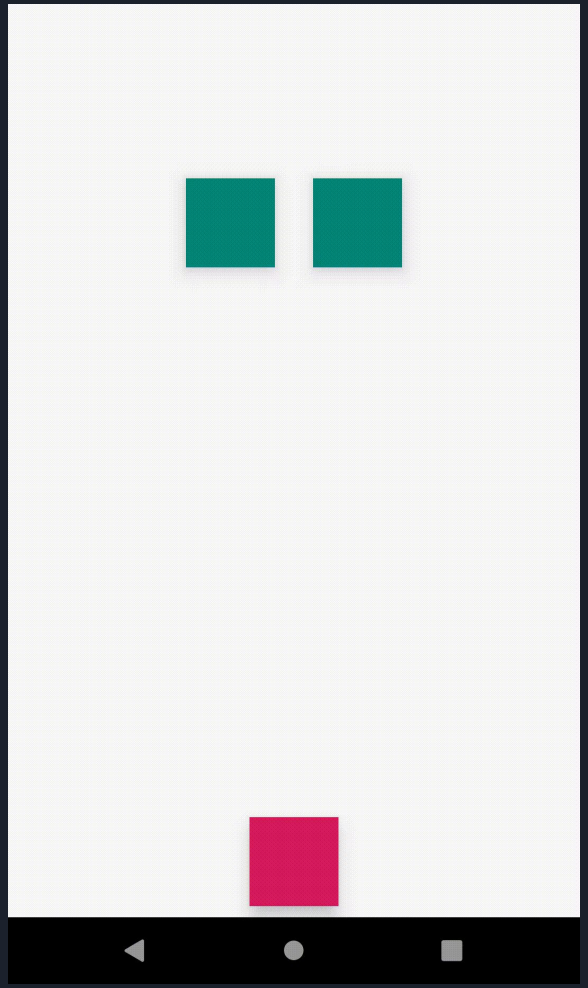
Defining Arc Paths





`motion:pathMotionArc="startHorizontal"`

`motion:pathMotionArc="startVertical"`



Start Constraint

```
<ConstraintSet
        android:id="@+id/start_box">

    <!--Box Up-->
    <Constraint
            android:id="@id/box_up"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_marginBottom="8dp"
            android:layout_marginEnd="8dp"
            android:layout_marginStart="8dp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent">
        <CustomAttribute
                app:attributeName="backgroundColor"
                app:customColorValue="@color/colorAccent"/>
    </Constraint>

    <!--Box Left-->
    <Constraint
            android:id="@id/box_left"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_marginTop="128dp"
            android:layout_marginStart="128dp"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">
        <CustomAttribute
                app:attributeName="backgroundColor"
                app:customColorValue="@color/colorPrimary"/>
    </Constraint>

    <!--Box Right-->
    <Constraint
            android:id="@id/box_right"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_marginEnd="128dp"
            android:layout_marginTop="128dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintTop_toTopOf="parent">
        <CustomAttribute
                app:attributeName="backgroundColor"
                app:customColorValue="@color/colorPrimary"/>
    </Constraint>

</ConstraintSet>
```

End Constraint

```
<ConstraintSet
        android:id="@+id/end_box">

    <!--Box Up-->
    <Constraint
            android:id="@id/box_up"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_marginTop="128dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent">
        <CustomAttribute
                app:attributeName="backgroundColor"
                app:customColorValue="@color/colorPrimaryDark"/>

    </Constraint>

    <!--Box Left-->
    <Constraint
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:id="@id/box_left"
            android:layout_marginTop="16dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            android:layout_marginStart="16dp">

        <CustomAttribute
                app:attributeName="backgroundColor"
                app:customColorValue="@color/colorPrimaryDark"/>

    </Constraint>

    <!--Box Right-->
    <Constraint
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:id="@id/box_right"
            android:layout_marginTop="16dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            android:layout_marginEnd="16dp">

        <CustomAttribute
                app:attributeName="backgroundColor"
                app:customColorValue="@color/colorPrimaryDark"/>

    </Constraint>

</ConstraintSet>
```

Transition

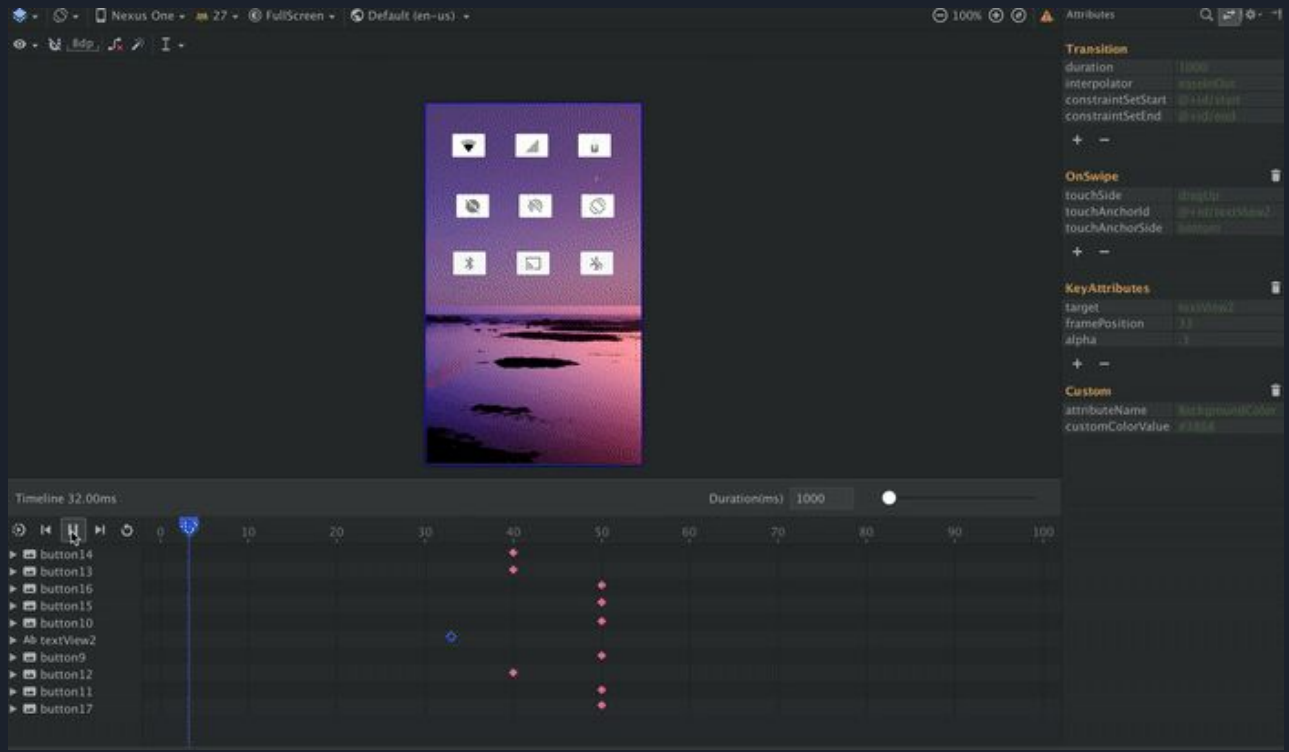
```
<Transition
        app:constraintSetStart="@id/start_box"
        app:constraintSetEnd="@id/end_box">

    <OnSwipe
            app:touchAnchorId="@id/box_up"
            app:dragDirection="dragUp"/>

    <KeyFrameSet>
        <KeyAttribute
                android:rotation="360"
                app:framePosition="100"
                app:target="@id/box_up"/>

        <KeyAttribute
            android:rotation="90"
            app:framePosition="100"
            app:target="@id/box_left"/>

        <KeyAttribute
            android:rotation="-90"
            app:framePosition="100"
            app:target="@id/box_right"/>
    </KeyFrameSet>

</Transition>
```

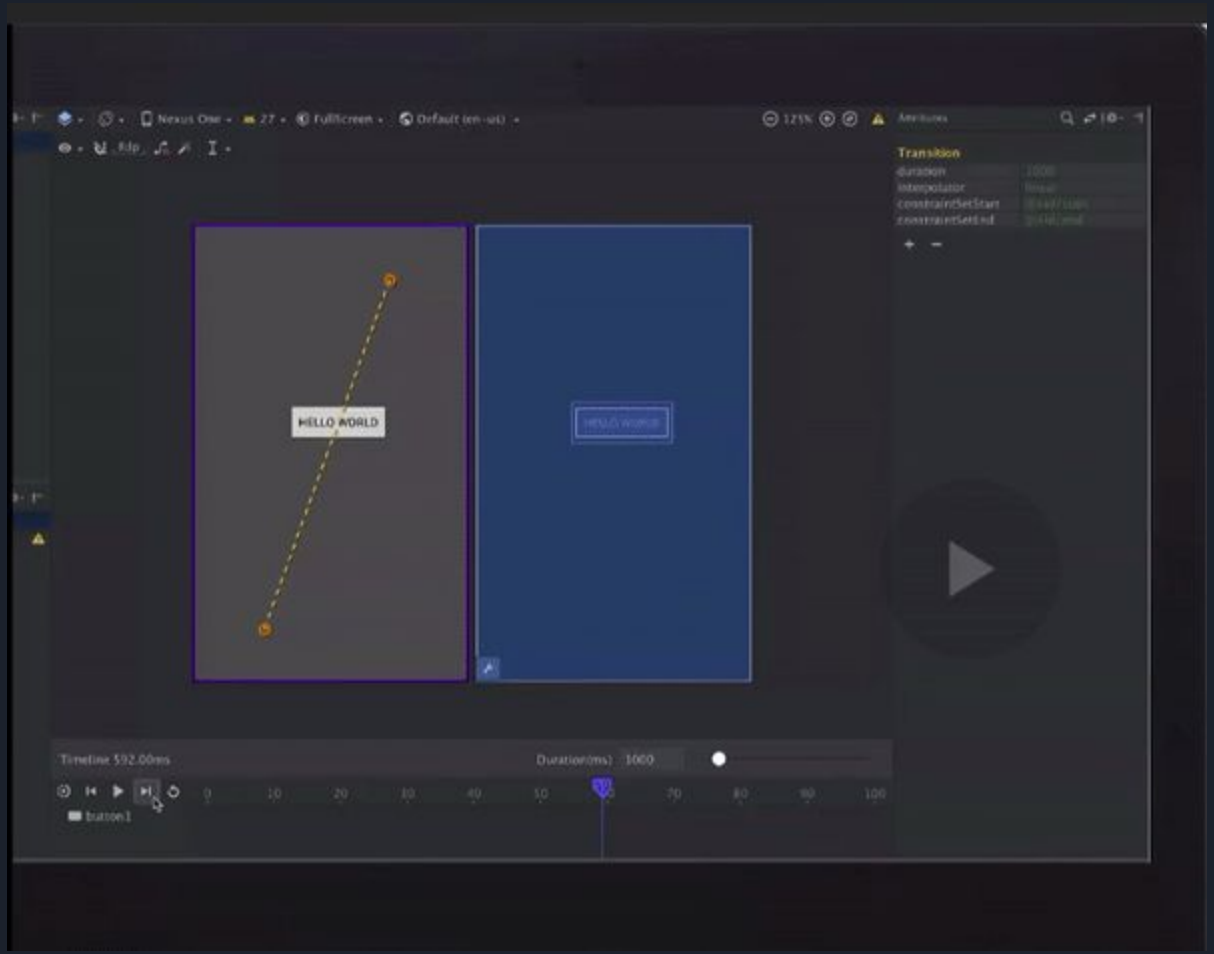
How can we incorporate Motion Layout on our existing project?

- More than one constraint is needed for Motion Layout to work.
- Motion Layout can be applied over DrawerLayout, ViewPager, RecyclerView, AppBarLayout etc.
- We only need to assign the offset value to the motion layout ‘progress’.
- <OnSwipe> and <OnClick> tags can be omitted in the MotionScene

```kotlin
class MotionDrawerLayout @JvmOverloads constructor(
    context: Context,
    attrs: AttributeSet? = null,
    defStyleAttr: Int = 0) : MotionLayout(context, attrs, defStyleAttr), DrawerLayout.DrawerListener {

    override fun onDrawerStateChanged(newState: Int) {

    }

    override fun onDrawerSlide(drawerView: View, slideOffset: Float) {
        progress = slideOffset
    }

    override fun onDrawerClosed(drawerView: View) {

    }

    override fun onDrawerOpened(drawerView: View) {

    }

    override fun onAttachedToWindow() {
        super.onAttachedToWindow()
        (parent as? DrawerLayout)?.addDrawerListener( listener: this)
    }
}
```

Future for Motion Layout




Nexus One
27
Fullscreen
Default (en-us)
125%
Transition
duration
interpolator
HELLO WORLD
Timeline 592.00ms
Duration(ms) 1000
button1

Thank You