

HenCoder Plus 讲义

过渡动画与 MotionLayout

过渡动画

过渡动画使用

有两种使用方式

- `TransitionManager#beginDelayedTransition`
- `TransitionManager#go()`

其中使用 `go()` 调用，需要重新绑定数据（因为 View 会被重新添加）。

过渡动画原理

过渡动画指的是两个场景之间的过渡，一个「开始场景」一个「结束场景」

1. 我们就要从场景上记录里面控件的各种参数
2. 根据两个场景的各种参数，创建出属性动画，播放属性动画

对应 Transition 上的关键函数

1. 捕获并记录对应属性

`captureStartValues` 和 `captureEndValues`

2. 基于捕获的值创建对应动画，并播放动画

`createAnimator` 和 `playTransition`

MotionLayout

MotionLayout 使用

- ConstraintLayout 改为 MotionLayout
- 创建 MotionScene 文件并使用 `app:layoutDescription` 关联

Constraint 节点

```
<Constraint id="">
  <Motion/>
  <Layout/>
  <Transform/>
  <CustomAttribute/>
  <PropertySet/>
</Constraint>
```

- 运动模型
弧线路径，时间模型等
- 布局相关
注意，`width`、`height` 和 `margin` 的命名空间是 `android:` (beta1 开始)
而约束相关的命名空间是 `app` (或 `motion`)
- 动画变换
做旋转，位移，缩放，海拔等属性
- 自定义属性
 - `attributeName` 会加上 `set/get` 反射找到真正的函数名，比如 `backgroundColor` 就会调用 `setBackgroundColor()` 函数
 - `custom(xxx)Value` 对应属性的数据类型
- 特定的属性

`visibility`、`alpha` 等属性

时间模型

- 每个控件可以单独设置时间模型

在 `<Motion>` 节点中使用 `app:transitionEasing`

- 整个 Transition 也可以设置插值器

在 `<Transition>` 中使用 `app:motionInterpolator`

MotionScene

```
<MotionScene
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto">
  <Transition
    app:constraintSetStart="@+id/start"
    app:constraintSetEnd="@+id/end"/>
    <ConstraintSet android:id="@+id/start"/>
    <ConstraintSet android:id="@+id/end"/>
  </MotionScene>
```

Transition 节点

```
<Transition
  app:constraintSetStart="@+id/start"
  app:constraintSetEnd="@+id/end"
  app:duration="1000">
    <OnSwipe />
    <OnClick />
```

```

<KeyPositionSet >
  <KeyAttribute>
    <CustomAttribute/>
  </KeyAttribute>
  <KeyPostion/>
  <KeyCycle/>
  <KeyTimeCycle/>
</KeyPositionSet>
</Transition>

```

onSwipe

```

<OnSwipe
  app:touchAnchorId="@id/view"
  app:dragDirection="dragDown"/>

```

`<OnSwipe>` 添加在 `<Transtion>` 节点中支持的参数：

- `touchAnchorId` 指的是哪一个控件响应触摸事件。
- `autoComplete` 默认的是 `true`，会根据动画完成的百分比自动到最近的一个状态
- `dragDirection` 拖拽的方向

OnClick

```

<OnClick
  app:clickAction="toggle"
  app:targetId="@id/toggle"/>

```

- `targetId` 指定控件
- `clickAction`
 - `toggle` 反转状态
 - `transitionToEnd/Start` 通过动画到结束/起始状态

- `jumpToEnd/Start` 没有动画直接到结束/起始状态

KeyPositionSet

- `app:motionTarget` 目标对象 ID
- `app:framePosition` 百分比 (0 - 100)

KeyAttribute 属性关键字

```
<KeyAttribute  
    app:framePosition="0"  
    app:motionTarget="@id/image_film_cover"  
    android:elevation="12dp">  
<CustomAttribute  
    app:attributeName="BackgroundColor"  
    app:customColorValue="@color/colorAccent"/>  
</KeyAttribute>
```

可以设置位移，旋转，缩放等属性，同时还可以通过 CustomAttribute 添加自定义属性

KeyPosition 位置关键帧

percentX/Y 在关键帧时，对应路径的对应百分比

percentWidth/Height 在关键帧时，控件大小改变的百分比

curveFit 运动路径的样式（直线，曲线等）

keyPositionType 坐标系 (结合课上的画图理解)

- parentRelative
 - (0, 0) 为父容器左上角
 - (1, 1) 为父容器右下角

- `deltaRelative`
 - (0, 0) 为起始控件中心
 - (1, 1) 为结束控件中心
- `pathRelative`
 - (0, 0) 为起始控件中心
 - (1, 0) 为结束控件中心

KeyCycle 和 KeyTimeCycle

通过 3 个 KeyCycle 定义一个准确的循环关键帧

```
<KeyFrameSet>
  <KeyCycle
    android:rotation="0"
    app:motionTarget="@id/fab_favourite"
    app:wavePeriod="0"
    app:framePosition="0"
    app:waveShape="sin"/>
  <KeyCycle
    android:rotation="180"
    app:motionTarget="@id/fab_favourite"
    app:wavePeriod="3"
    app:framePosition="50"
    app:waveShape="sin"/>
  <KeyCycle
    android:rotation="0"
    app:motionTarget="@id/fab_favourite"
    app:wavePeriod="0"
    app:framePosition="100"
    app:waveShape="sin"/>
</KeyFrameSet>
```

`wavePeriod` 循环次数 (KeyTimeCycle 表示的每秒循环次数)

`waveShape` 数学模型

问题和建议？

课上技术相关的问题，都可以去群里和大家讨论，对于比较通用的、有价值的问题，可以去我们的知识星球提问。

具体技术之外的问题和建议，都可以找丢物线（微信：diuwuxian），丢丢会为你解答技术以外的一切。



觉得好？

如果你觉得课程很棒，欢迎给我们好评呀！<https://ke.qq.com/comment/index.html?cid=381952>

一定要是你真的觉得好，再给我们好评。不要仅仅因为对丢物线的支持而好评（报名课程已经是你最大的支持了，再不够的话 B 站多来点三连我也很开心），另外我们也坚决不做好评返现等任何的交易。我们只希望，在课程对你有帮助的前提下，可以看到你温暖的评价。

更多内容：

- 网站：<https://hencoder.com>；<https://kaixue.io>
- 各大搜索引擎、微信公众号、微博、知乎、掘金、哔哩哔哩、YouTube、西瓜视频、抖音、快手、微视：统一账号「扔物线」，我会持续输出优质的技术内容，欢迎大家关注。
- 哔哩哔哩快捷传送门：<https://space.bilibili.com/27559447>

大家如果喜欢我们的课程，还请去扔物线的哔哩哔哩，帮我素质三连，感谢大家！