

MotionLayout

Анимации, как у сына
маминой подруги

Что будет?

- Коротко о MotionLayout
- Как работать с MotionScene
- Создаем интерфейсы
- Плюсы и минусы
- Где применить

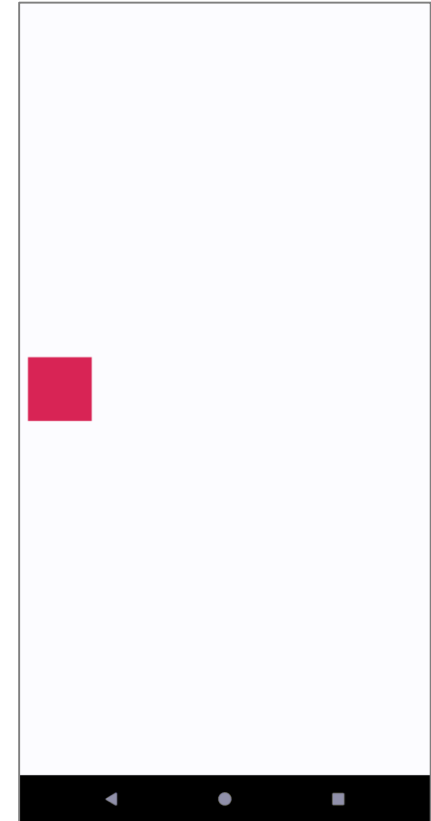
Что такое MotionLayout

MotionLayout

Первая версия в 2018 году

Текущая - beta 4

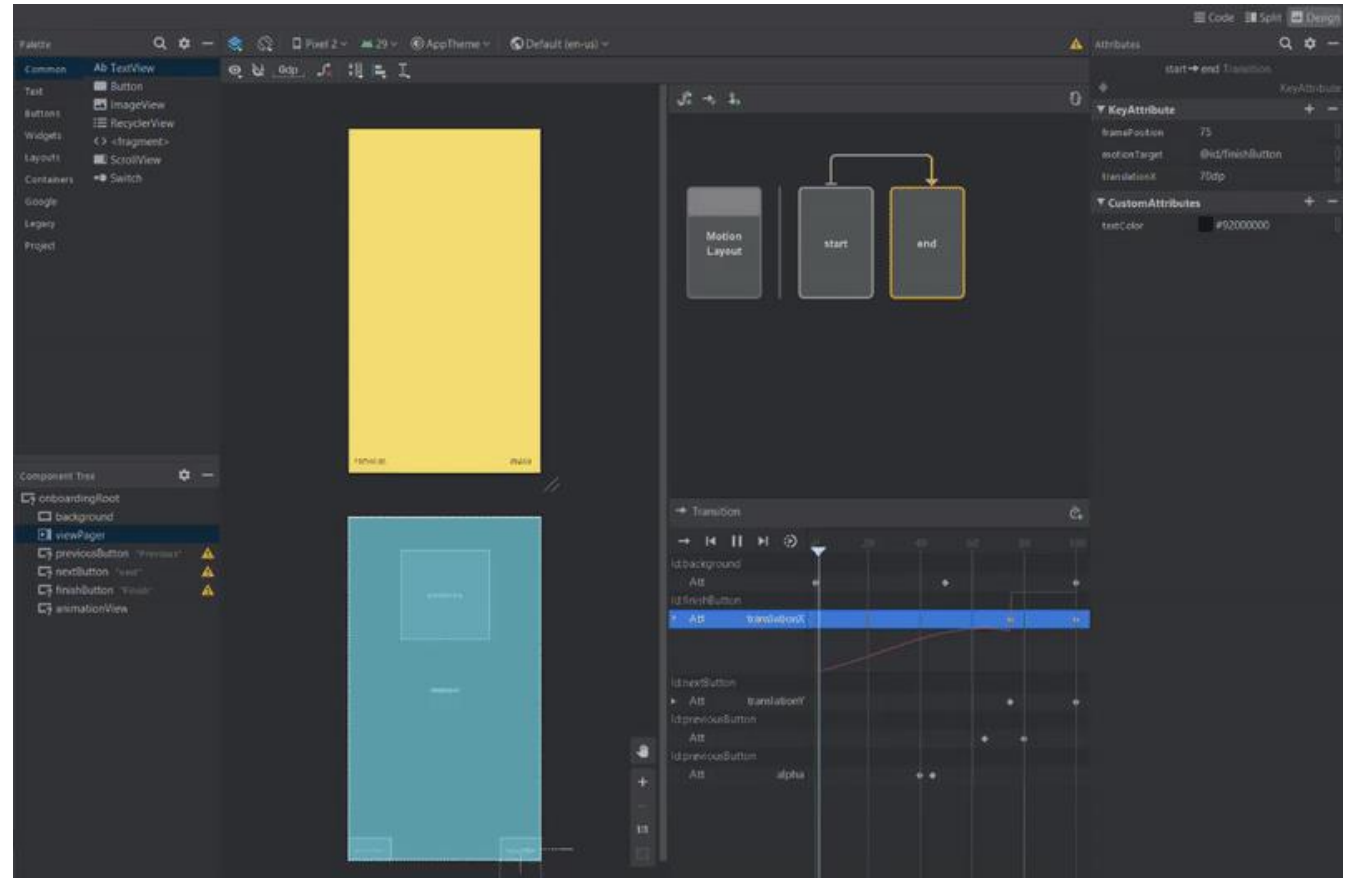
- Без кода
- Функционал LayoutTransitions and TransitionManager, CoordinatorLayout, property animation
- Ключевые кадры
- Управление прогрессом анимации
- Обработка жестами



Motion Editor

AndroidStudio 4.0 CANARY 9

- Базовый функционал работает
- Проблемы с ключевыми кадрами



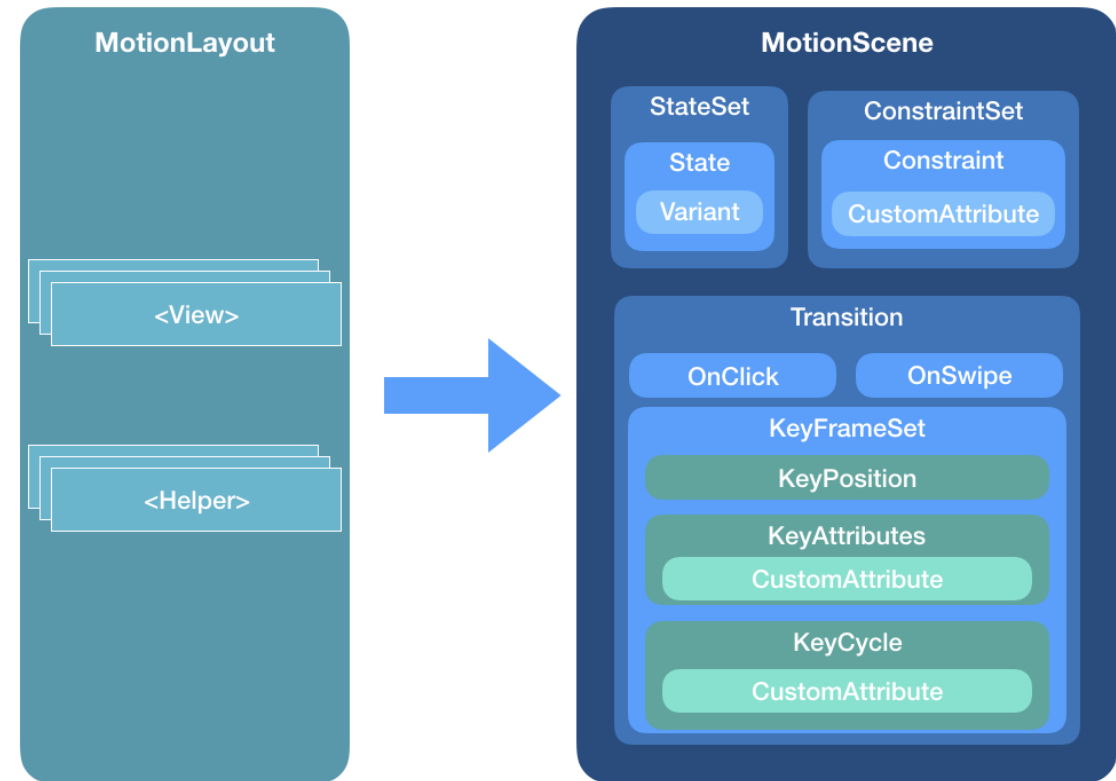
MotionLayout

MotionLayout:

- Views (with id)
- Привязки (опционально)

MotionScene:

- Состояния
- Переходы
- Ключевые кадры
- Жесты



MotionLayout

```
implementation 'androidx.constraintlayout:constraintlayout:2.0.0-beta4'
```

```
<androidx.constraintlayout.motion.widget.MotionLayout
    ...
    app:layoutDescription="@xml/scene">

    <View... />

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

× Как работать с MotionScene

MotionScene (вариант 1)

res/xml/scene.xml

```
<MotionScene>
    <Transition
        motion:constraintSetStart="@+id/start"
        motion:constraintSetEnd="@+id/end" />

    ...
</MotionScene>
```

MotionScene (вариант 1)

res/xml/scene.xml

```
<MotionScene>
  <Transition ... />
  <ConstraintSet android:id="@+id/start">
    <Constraint
      android:id="@+id/view"
      android:layout_width="100dp"
      android:layout_height="100dp"
      motion:layout_constraintBottom_toBottomOf="parent"
      motion:layout_constraintEnd_toEndOf="parent"/>
    ...
  </ConstraintSet>
  ...
</MotionScene>
```

MotionScene (вариант 1)

res/xml/scene.xml

```
<MotionScene>
  <Transition ... />
  <ConstraintSet android:id="@+id/start" ... />

  <ConstraintSet android:id="@+id/end">
    <Constraint
      android:id="@id/view "
      android:layout_width="match_parent"
      android:layout_height="0dp"
      motion:layout_constraintTop_toTopOf="parent"
      motion:layout_constraintBottom_toBottomOf="parent"/>
    ...
  </ConstraintSet>
  ...
</MotionScene>
```

All of ConstraintLayout
+ <CustomAttribute>

MotionScene (вариант 1)



res/xml/scene.xml

```
<MotionScene>
  <Transition ... />
  <ConstraintSet android:id="@+id/start" ... />

  <ConstraintSet android:id="@+id/end">
    <Constraint android:id="@id/view">
      <PropertySet
        android:visibility="visible"/>
    </Constraint>

    ...
  </ConstraintSet>

  ...
</MotionScene>
```

All of **ConstraintLayout**
+ **<CustomAttribute>**

or

Combination of tags: **<Layout>**
<PropertySet> **<Transform>**
<Motion> **<CustomAttribute>**

MotionScene (вариант 1)

res/xml/scene.xml

```
<MotionScene>
  <Transition ... />
  <ConstraintSet android:id="@+id/start" ... />

  <ConstraintSet android:id="@+id/end"
    motion:deriveConstraintsFrom="@id/start">
    <Constraint android:id="@id/view">
      <PropertySet
        android:visibility="visible"/>
    </Constraint>
    ...
  </ConstraintSet>
  ...
</MotionScene>
```

All of ConstraintLayout
+ <CustomAttribute>

or

Combination of tags: <Layout>
<PropertySet> <Transform>
<Motion> <CustomAttribute>

MotionScene (вариант 1)



- + Будет использоваться в MotionEditor
- + CustomAttributes
- Не наглядно без редактора

MotionScene (вариант 2)

res/xml/scene.xml

```
<MotionScene>
  <Transition
    motion:constraintSetStart="@layout/motion_start"
    motion:constraintSetEnd="@layout/motion_end">
  </Transition>

</MotionScene>
```

res/layout/motion_start.xml

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

res/layout/motion_end.xml

```
<androidx.constraintlayout.widget.ConstraintLayout ... >
```

MotionScene (вариант 2)



- + Наглядность и простота реализации
- Больше xml файлов
- Нет CustomAttributes для крайних кадров (решается KeyFrames)

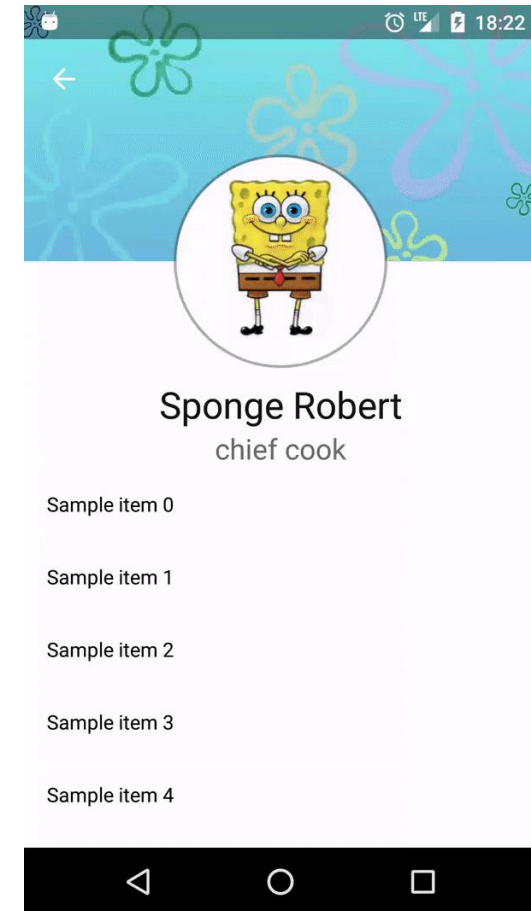
Transition

- Duration
 - Motion Interpolator
 - Auto Transition
 - Staggered
-
- Key Frames
 - OnClick
 - OnSwipe

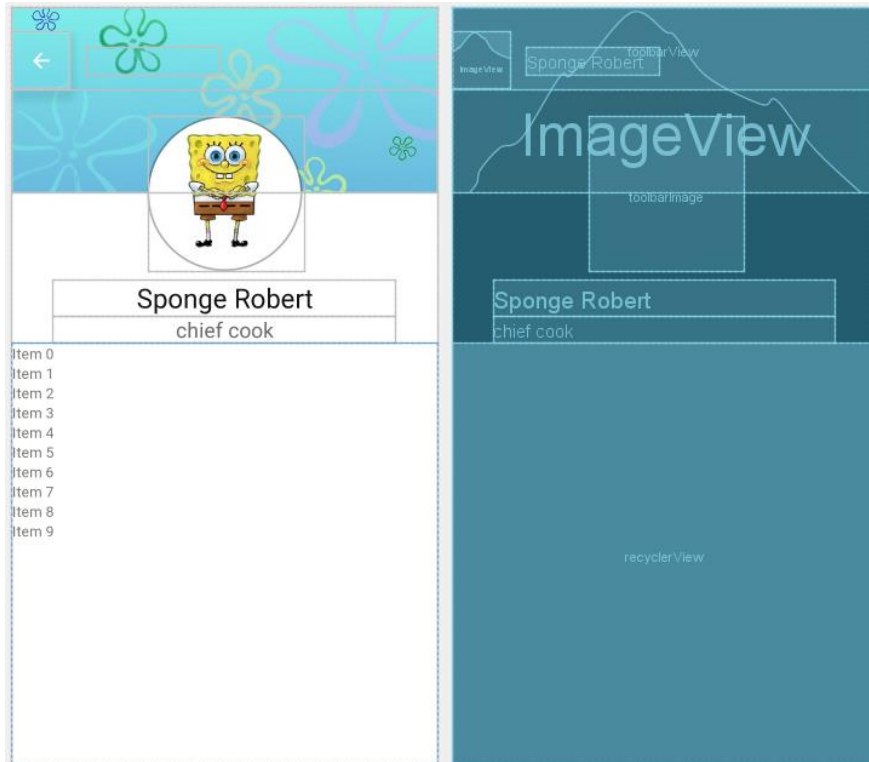
Разработка интерфейсов

Пример #1

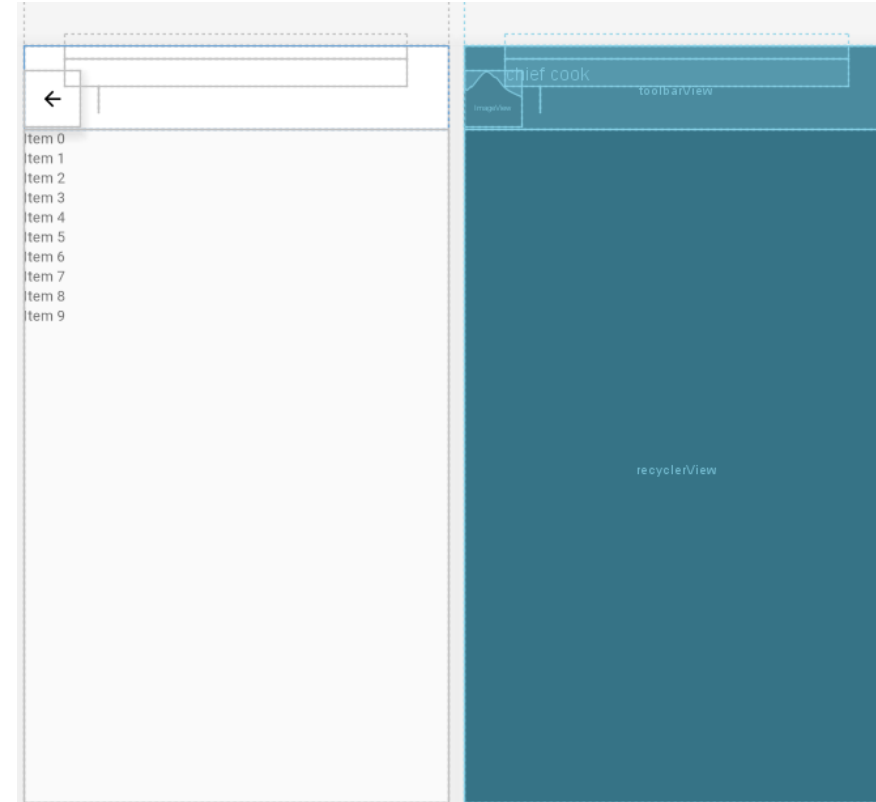
- Начальное и конечное состояния
- Какие переходы
- Что инициирует анимацию



Scene Layouts



start layout



end layout

MotionScene

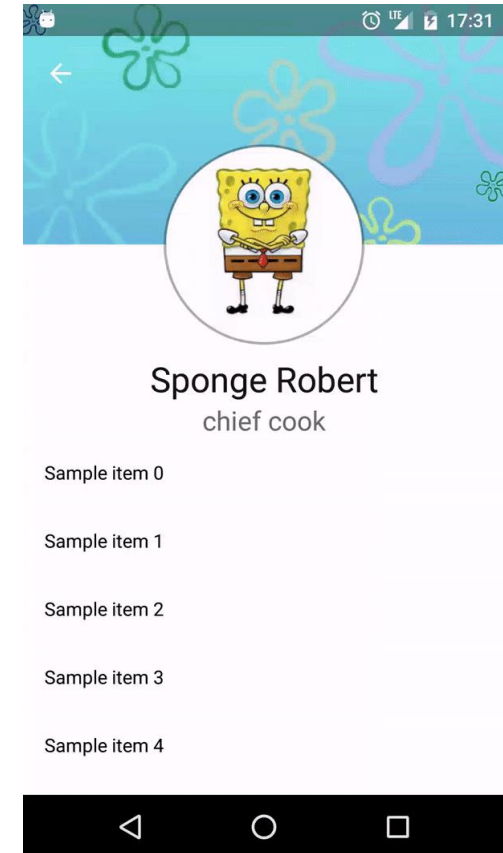
```
<MotionScene>
  <Transition
    app:constraintSetStart="@layout/start"
    app:constraintSetEnd="@layout/end">

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

  </Transition>
</MotionScene>
```

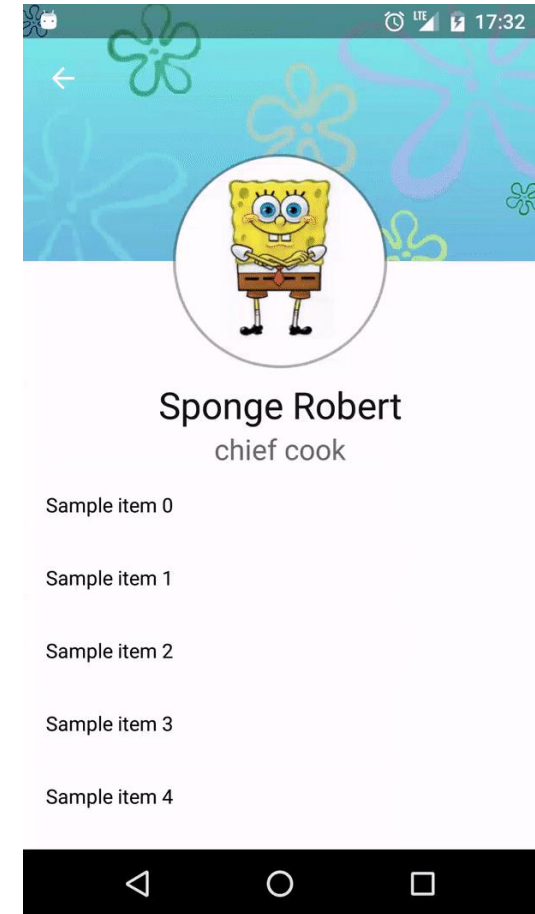
touchAnchorId - Have the drag act as if it is moving the "touchAnchorSide" of this object

touchAnchorSide - The side of the object to move with {top|left|right|bottom}



Auto animations

- Position
- Size
- Constraints
- Alpha
- Visibility: INVISIBLE – Fade
GONE – Fade + Scale to 0



Custom attributes

```
<MotionScene>
  <Transition/>
  ...
  <ConstraintSet android:id="@+id/start">

    <Constraint android:id="@+id/icon" ... >
      <CustomAttribute
        motion:attributeName="colorFilter"
        motion:customColorValue="#D81B60"/>
      <CustomAttribute
        motion:attributeName="backgroundColor"
        motion:customColorValue="#D81B60"/>
      <CustomAttribute
        motion:attributeName="textColor"
        motion:customColorValue="#D81B60"/>
    </Constraint>
  </ConstraintSet>
</MotionScene>
```



Key frames

```
<Transition
```

```
...
```

```
<KeyFrameSet>
```

```
<KeyAttribute
```

```
  motion:framePosition="0"
```

```
  motion:motionTarget="@id/backButton" >
```

```
  <CustomAttribute
```

```
    app:attributeName="colorFilter"
```

```
    app:customColorValue="#fff"/>
```

```
</KeyAttribute>
```

```
<KeyAttribute
```

```
  motion:framePosition="100"
```

```
  motion:motionTarget="@id/backButton" >
```

```
  <CustomAttribute
```

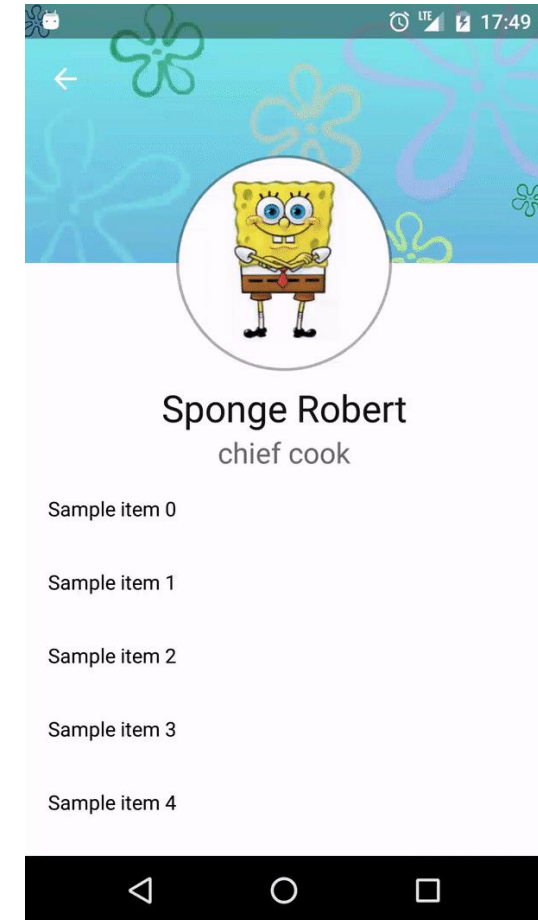
```
    app:attributeName="colorFilter"
```

```
    app:customColorValue="#000"/>
```

```
</KeyAttribute>
```

```
</KeyFrameSet>
```

```
</Transition>
```

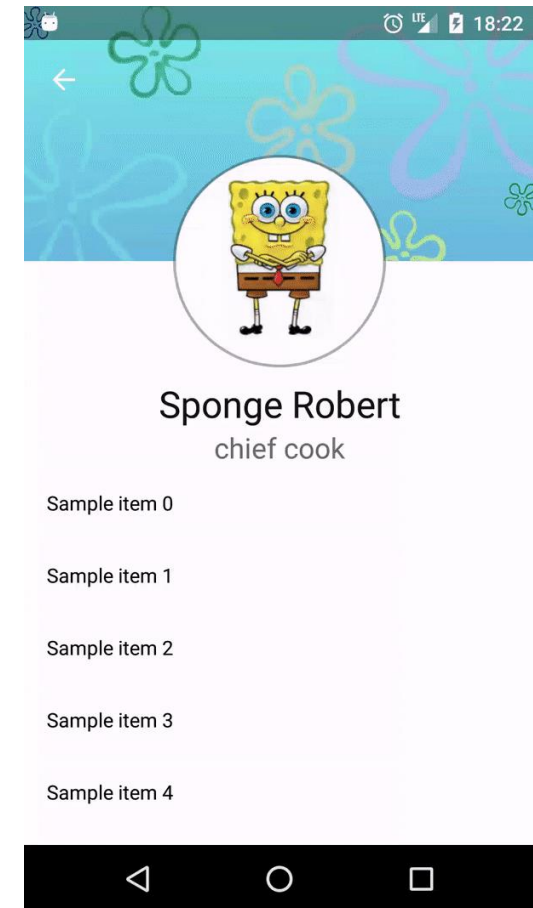


Key frames

```
<KeyFrameSet>
...
  <KeyAttribute
    motion:alpha="1"
    motion:framePosition="30"
    motion:motionTarget="@id/title" />

  <KeyAttribute
    motion:alpha="0"
    motion:framePosition="85"
    motion:motionTarget="@id/title" />

</KeyFrameSet>
```



Key frames

```
<KeyFrameSet>
```

```
...
```

```
<KeyAttribute
```

```
    motion:alpha="1"
```

```
    motion:framePosition="20"
```

```
    motion:scaleX="0.8"
```

```
    motion:scaleY="0.8"
```

```
    motion:motionTarget="@id/toolbarImage" />
```

```
<KeyAttribute
```

```
    motion:alpha="0"
```

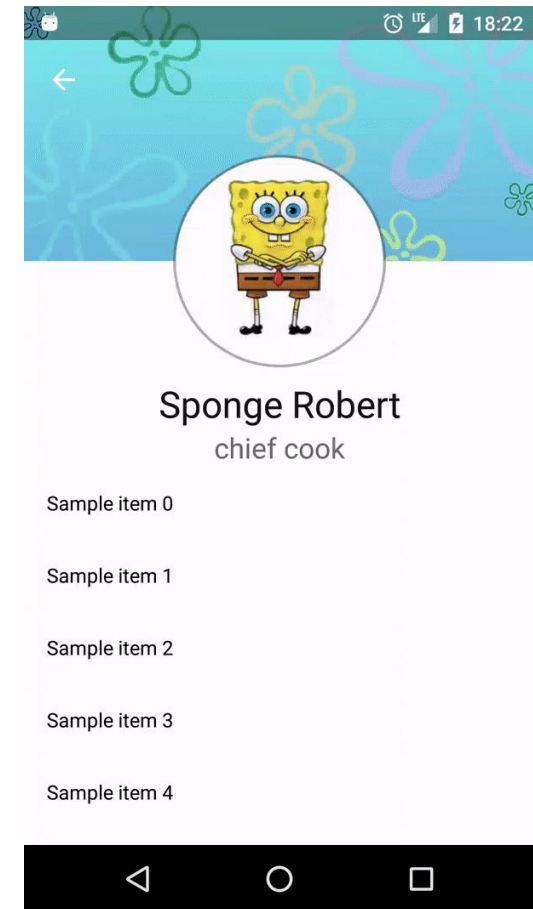
```
    motion:framePosition="60"
```

```
    motion:scaleX="0"
```

```
    motion:scaleY="0"
```

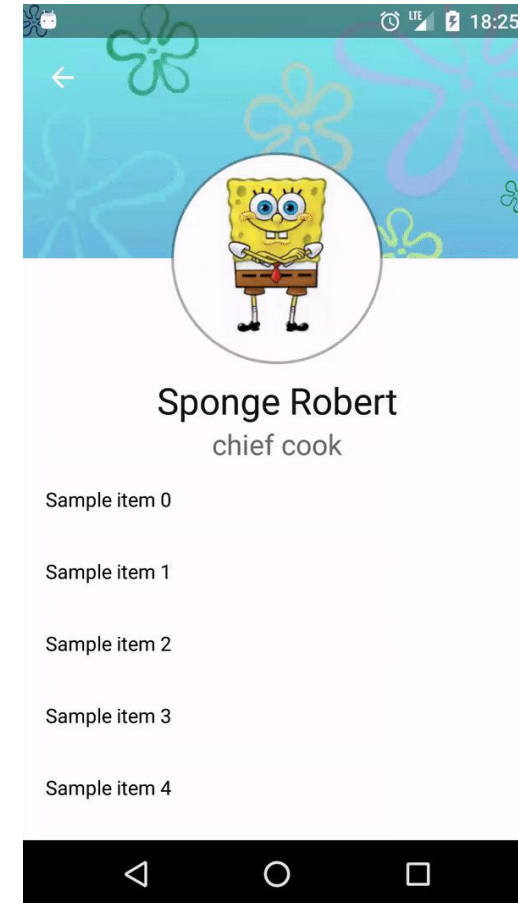
```
    motion:motionTarget="@id/toolbarImage" />
```

```
</KeyFrameSet>
```



“EnterAlways” behavior

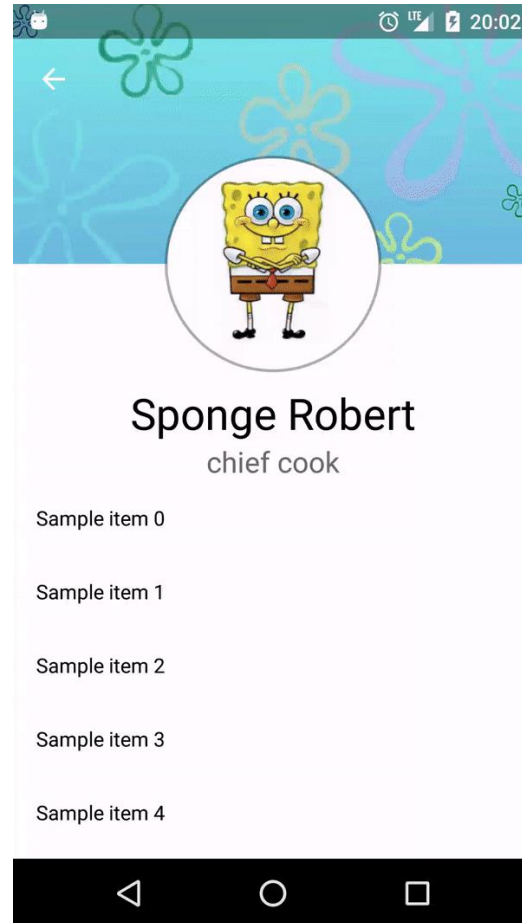
```
<Transition>  
  <OnSwipe  
    ...  
    app:moveWhenScrollAtTop="false"/>  
</Transition>
```



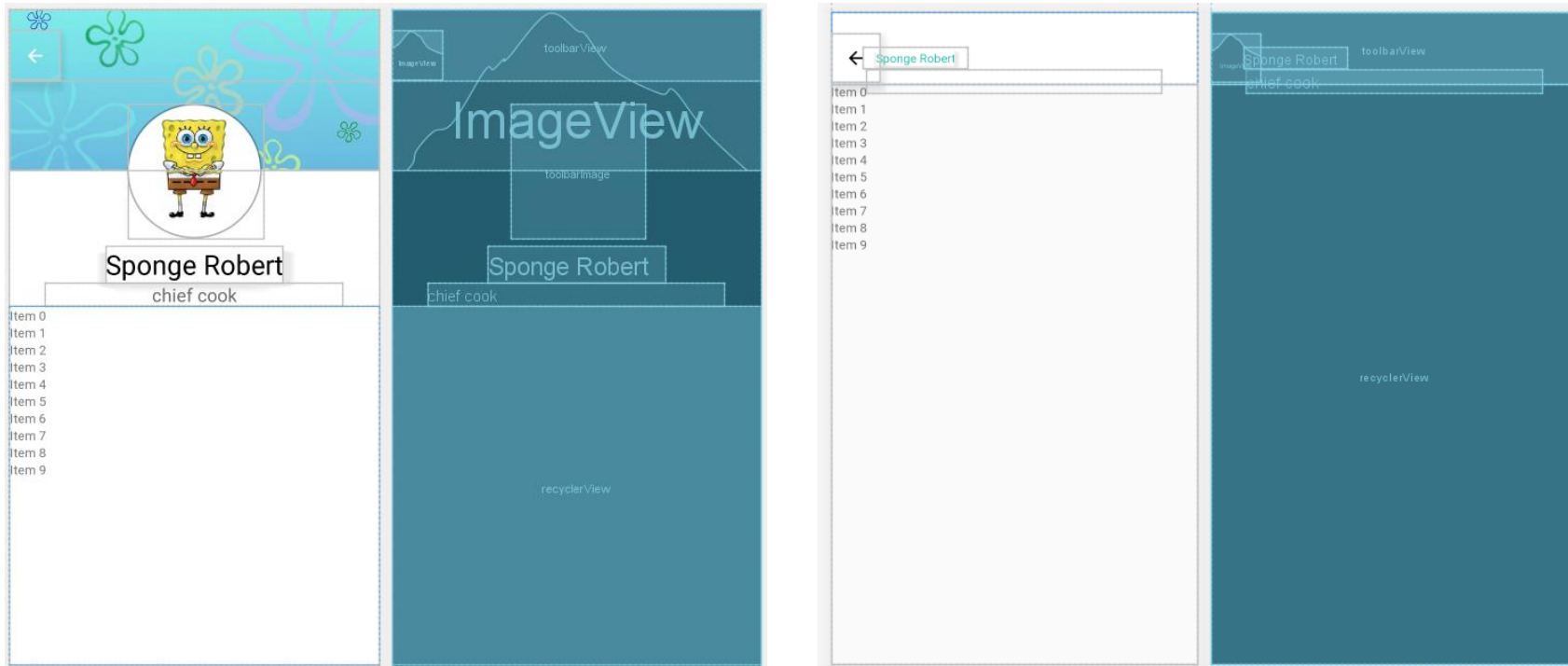
Пример #1

- Анимации из коробки
- Привязка жестов
- CustomAttributes (смена цвета)
- Ключевые кадры
- "Enter always"

Пример #2



Scene Layouts

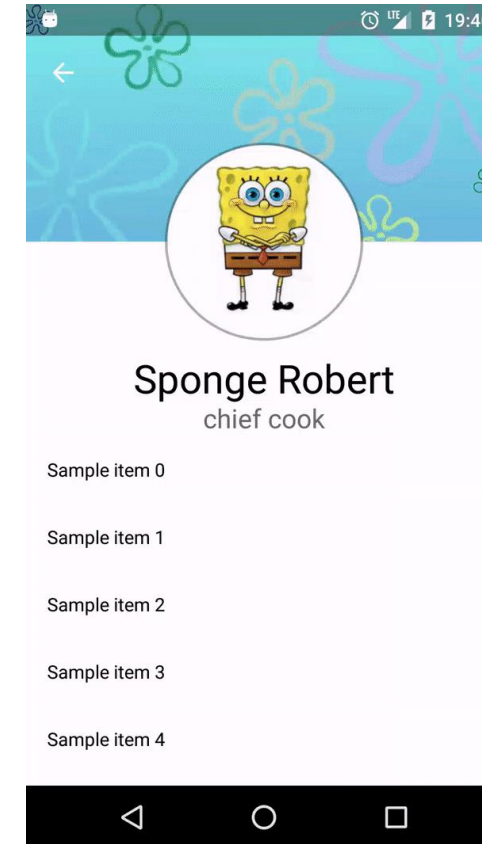


Text Size

```
<KeyFrameSet>
...
<KeyAttribute
  motion:framePosition="0"
  motion:motionTarget="@id/title" >
  <CustomAttribute
    motion:attributeName="textSize"
    motion:customDimension="10sp" />
</KeyAttribute>

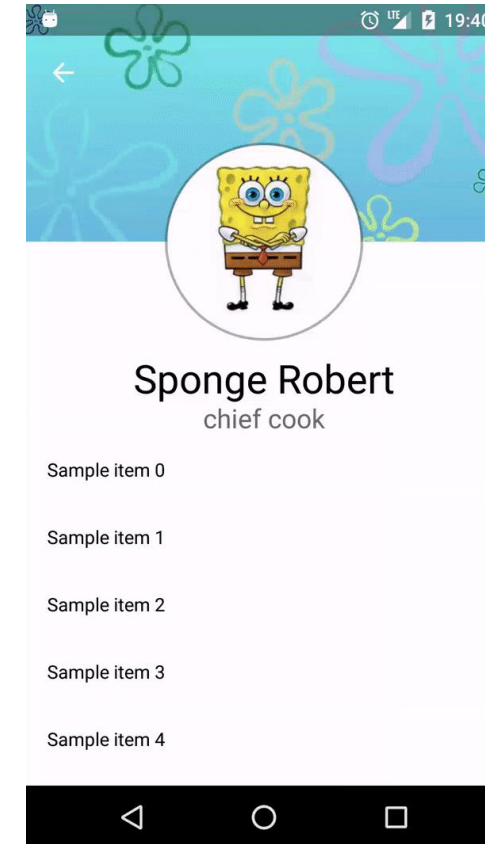
<KeyAttribute
  motion:framePosition="85"
  motion:motionTarget="@id/title" >
  <CustomAttribute
    motion:attributeName="textSize"
    motion:customDimension="7sp" />
</KeyAttribute>

</KeyFrameSet>
```



Text Size

- Мерцания и дерганные движения
- Не расширяет контейнер
- Размер текста больше чем в layout при том же sp



Text Scale

```
<KeyFrameSet>
```

```
...
```

```
<KeyAttribute
```

```
    motion:framePosition="85"
```

```
    motion:motionTarget="@id/title"
```

```
    motion:scaleX="0.75"
```

```
    motion:scaleY="0.75">
```

```
</KeyAttribute>
```

```
</KeyFrameSet>
```

```
in end layout:
```

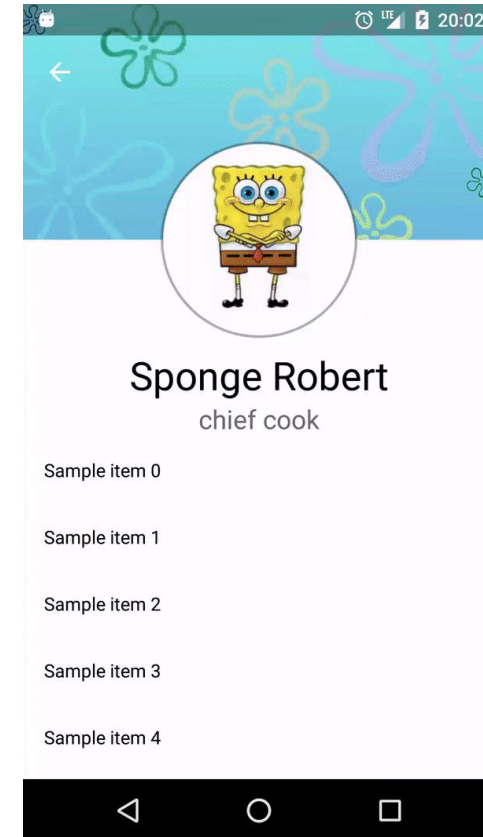
```
<TextView
```

```
    android:id="@+id/title"
```

```
...
```

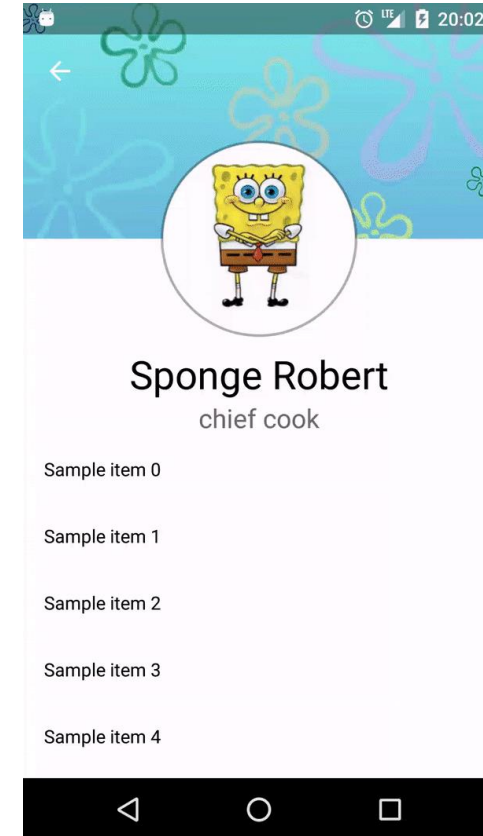
```
    android:scaleX="0.75"
```

```
    android:scaleY="0.75"/>
```

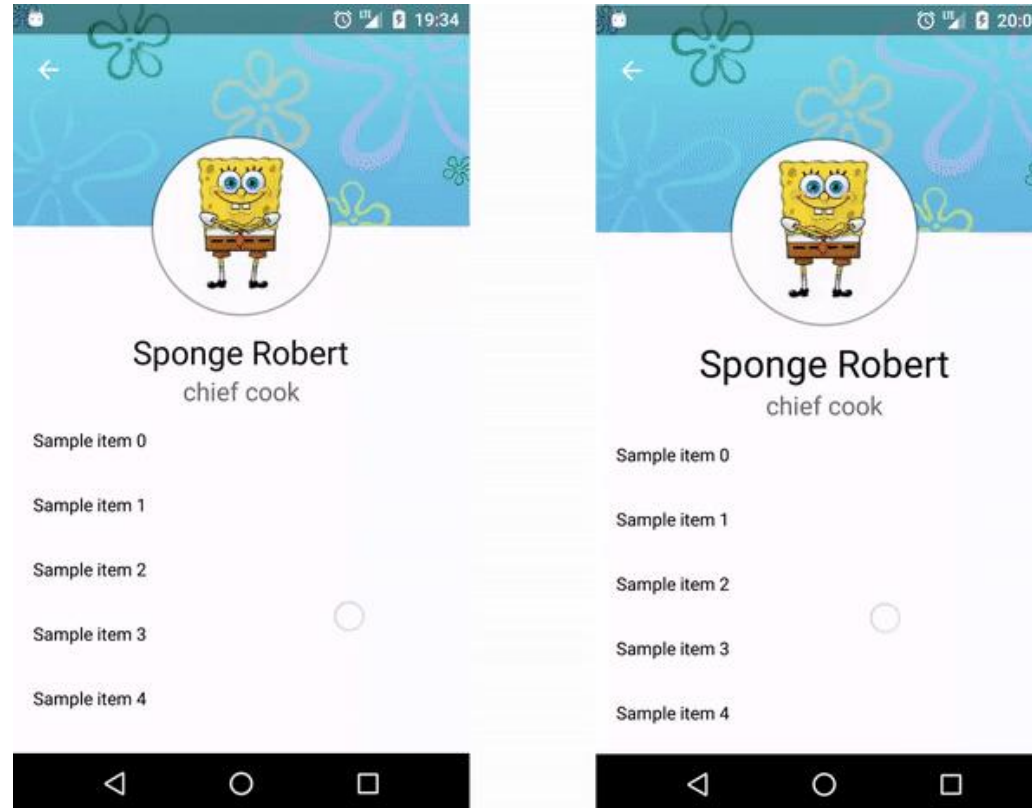


Text Scale

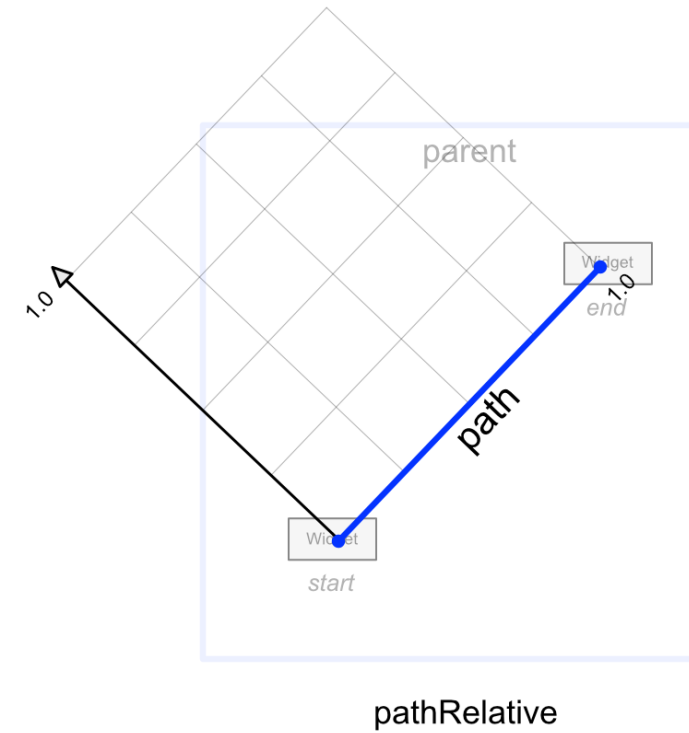
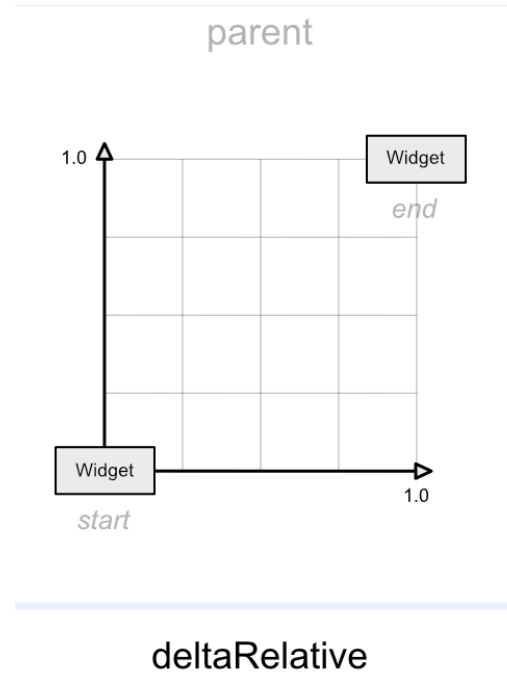
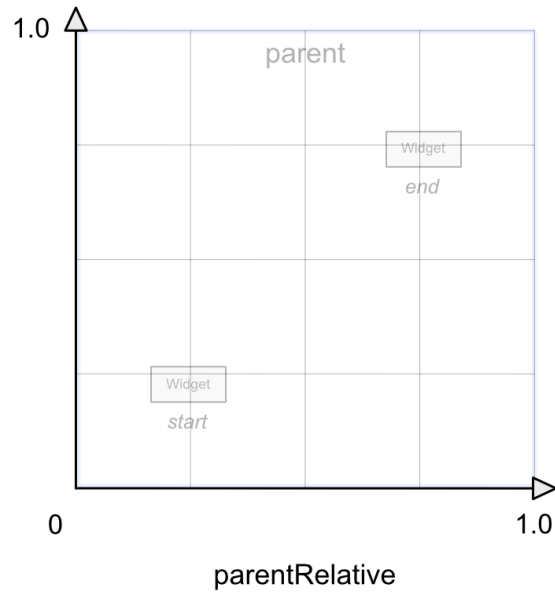
- Размеры контейнера сохраняются
- Масштабирование происходит внутри
- Отступы требуют подгона



Key position



Key position type



Key position

```
<KeyFrameSet>
```

```
...
```

```
<KeyPosition
```

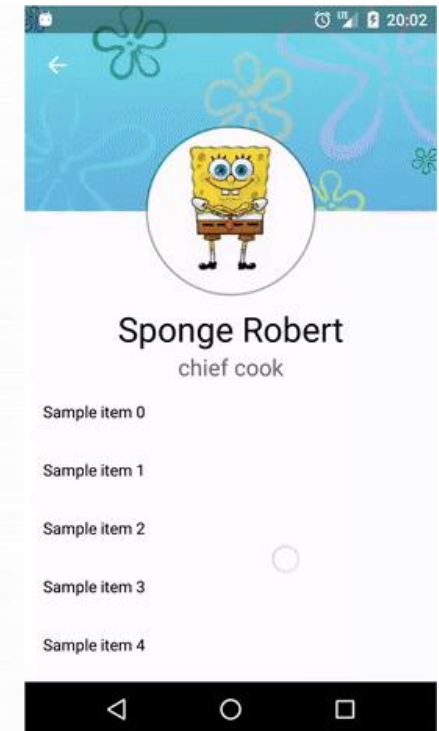
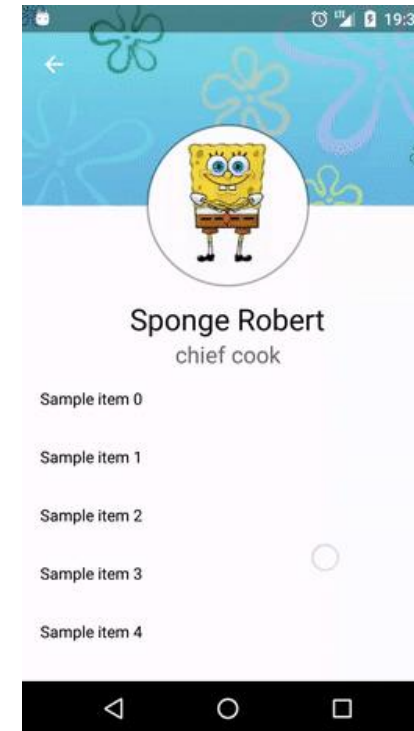
```
  motion:keyPositionType="pathRelative"
```

```
  motion:framePosition="85"
```

```
  motion:motionTarget="@id/title"
```

```
  motion:percentX="1" />
```

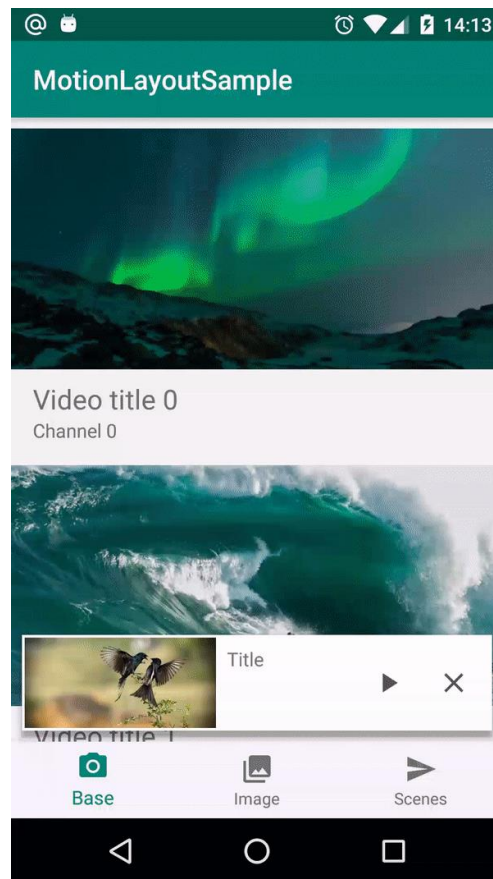
```
</KeyFrameSet>
```



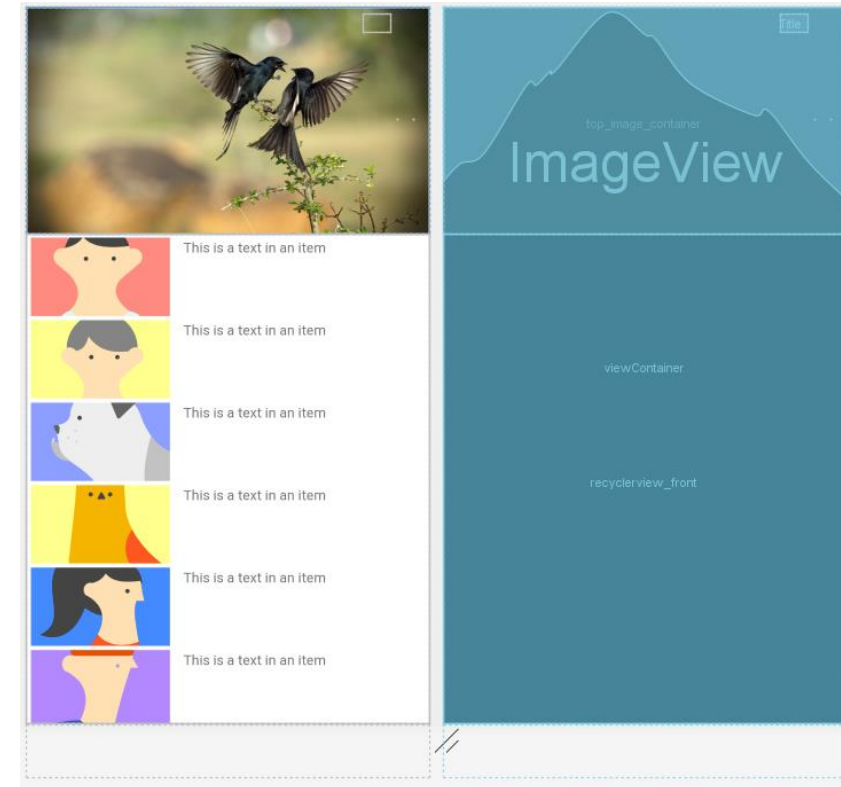
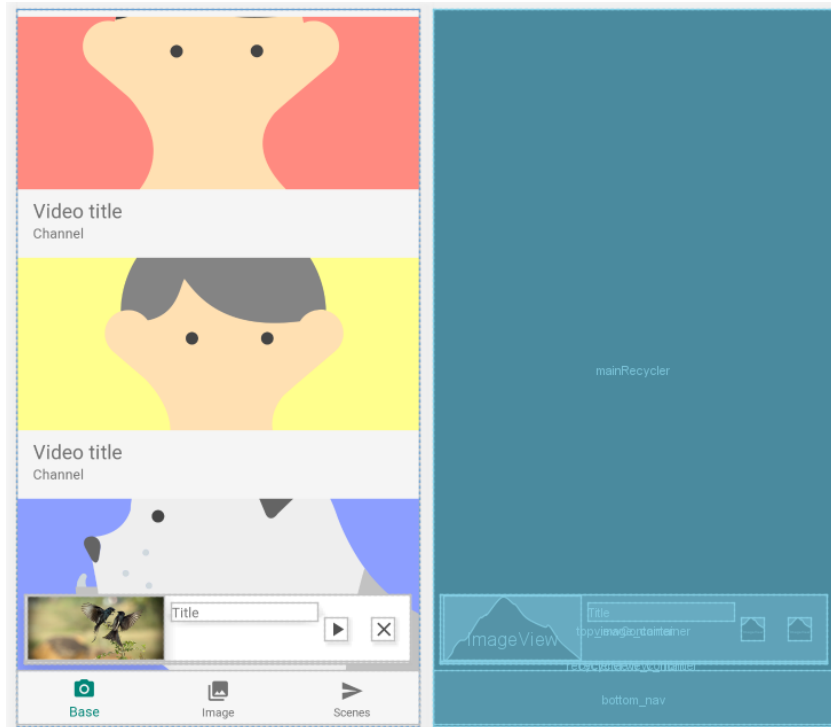
Пример #2

- Размер текста
- Изменение относительного положения объекта
- Системы координат

Пример #3



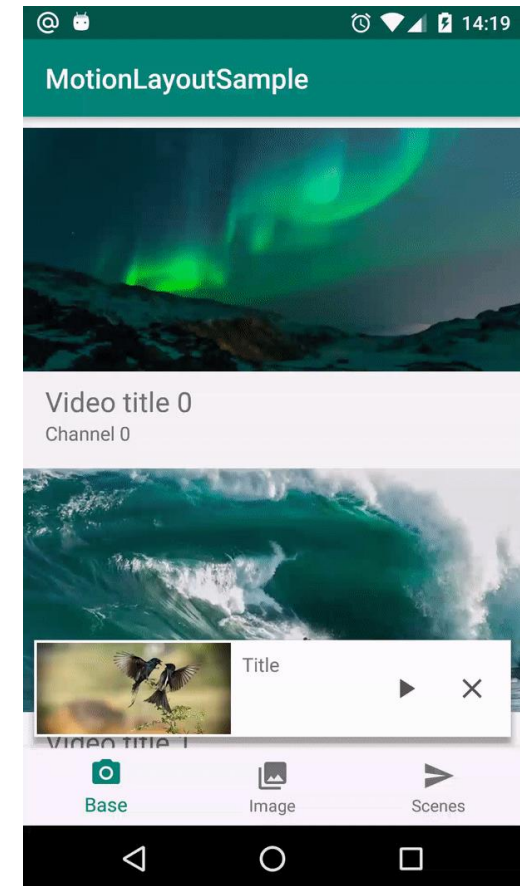
Scene Layouts



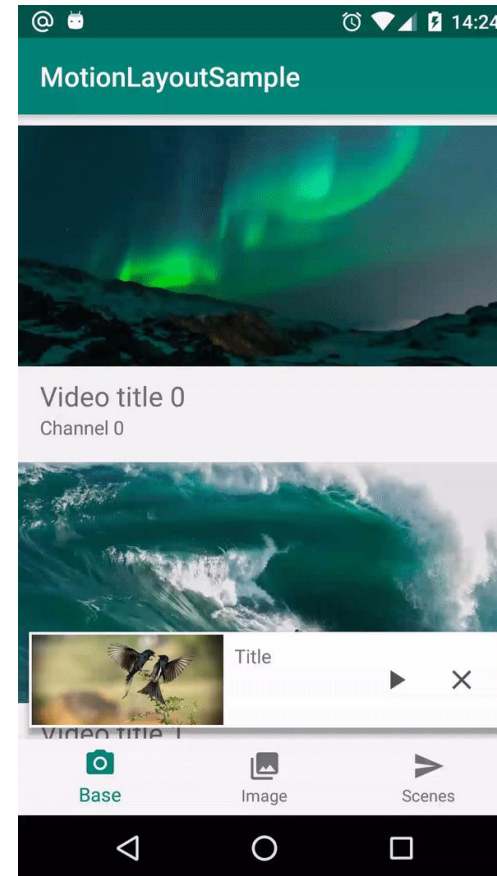
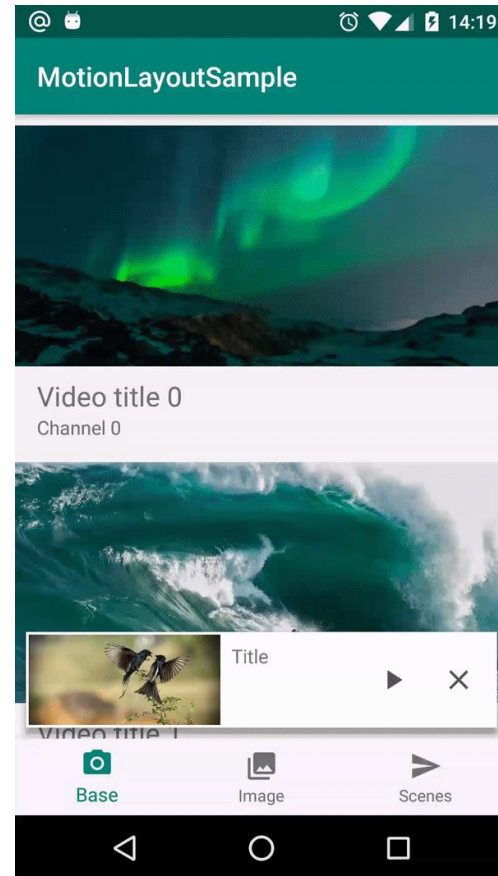
Acceleration

```
<Transition>
  <OnSwipe
    motion:dragDirection="dragUp"
    motion:touchRegionId="@+id/top_image_container"
    motion:maxAcceleration="100" />

</Transition>
```

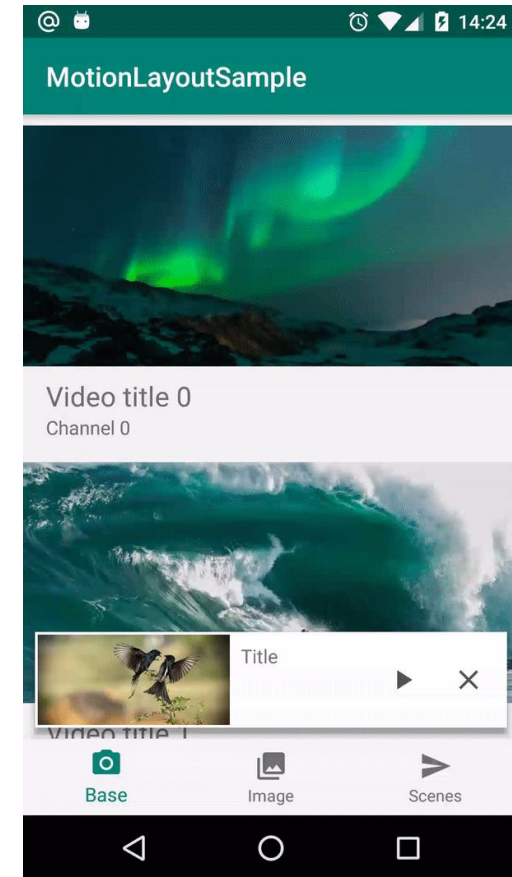


Пример #3



PercentWidth

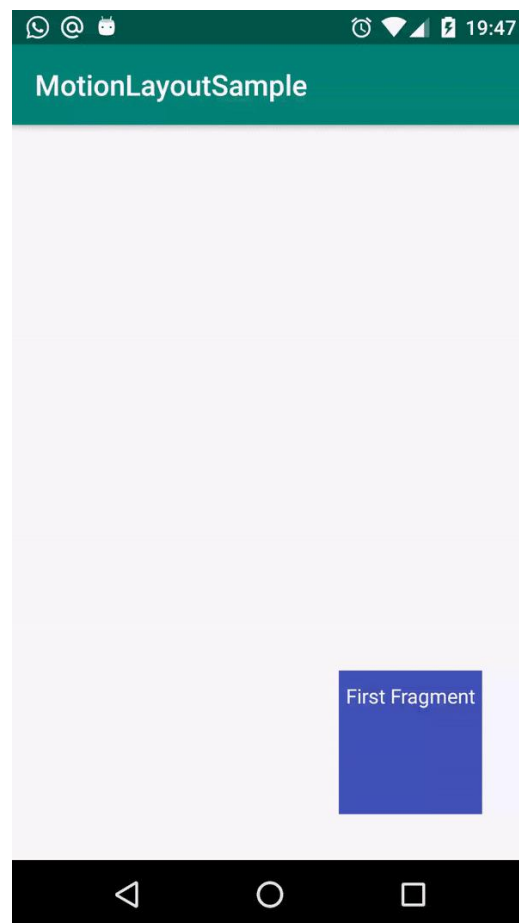
```
<KeyFrameSet>
...
  <KeyPosition
    motion:framePosition="20"
    motion:motionTarget="@id/top_image"
    motion:percentWidth="0.845"
    motion:percentX="0.915" />
</KeyFrameSet>
```



Пример #3

- Скорость завершения анимации
- Изменения ширины объекта

Пример #4



Немного кода



```
motionLayout.progress = 0.7f    [0-1]
```

```
motionLayout.transitionToStart()
```

```
motionLayout.transitionToEnd()
```

```
motionLayout.transitionToState(R.id.someState)
```

Немного кода

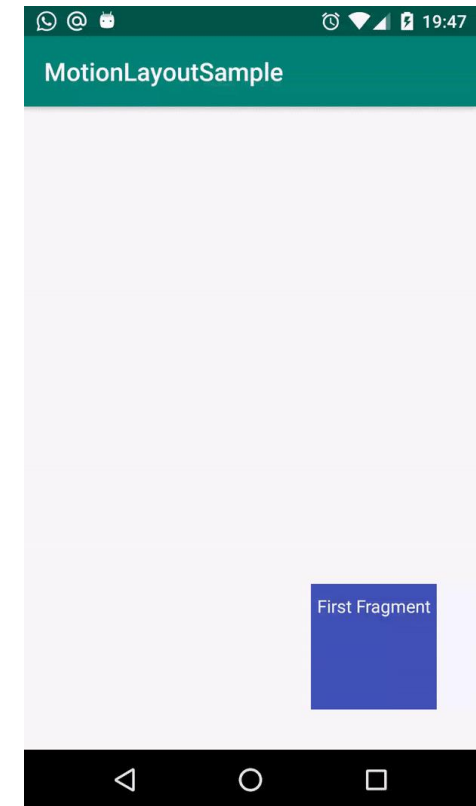
```
motionLayout.setTransitionListener()
```

```
public interface TransitionListener {  
    void onTransitionStarted(motionLayout: MotionLayout, startId: Int, endId: Int);  
  
    void onTransitionChange(motionLayout: MotionLayout, startId: Int, endId: Int,  
                            progress: Float);  
  
    void onTransitionCompleted(motionLayout: MotionLayout, currentId: Int);  
  
    void onTransitionTrigger(motionLayout: MotionLayout, triggerId: Int,  
                            positive: Boolean, progress: Float);  
}
```

or use **TransitionAdapter**

Передача прогресса

```
override fun onTransitionChange(motionLayout: MotionLayout,  
                                startId: Int, endId: Int, progress: Float) {  
    ...  
    var motionLayout = //get motion layout from fragment  
                       //or use set interface  
    motionLayout?.progress = progress  
}
```

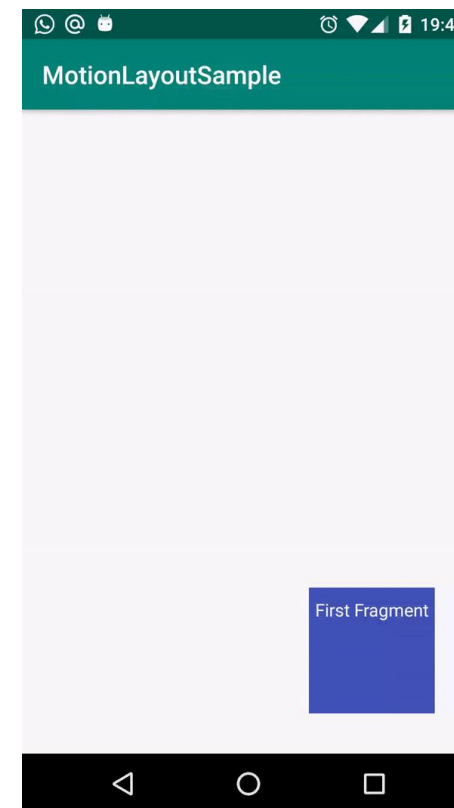


Triggers

```
<KeyFrameSet>
  <KeyTrigger
    motion:triggerId="@+id/trig"
    motion:framePosition="50"
    motion:motionTarget="@+id/container"
    motion:onCross="someFun"
    motion:onPositiveCross="show"
    motion:onNegativeCross="hide"
    motion:triggerSlack="0.1" />
</KeyFrameSet>
```

Смена фрагмента в процессе анимации

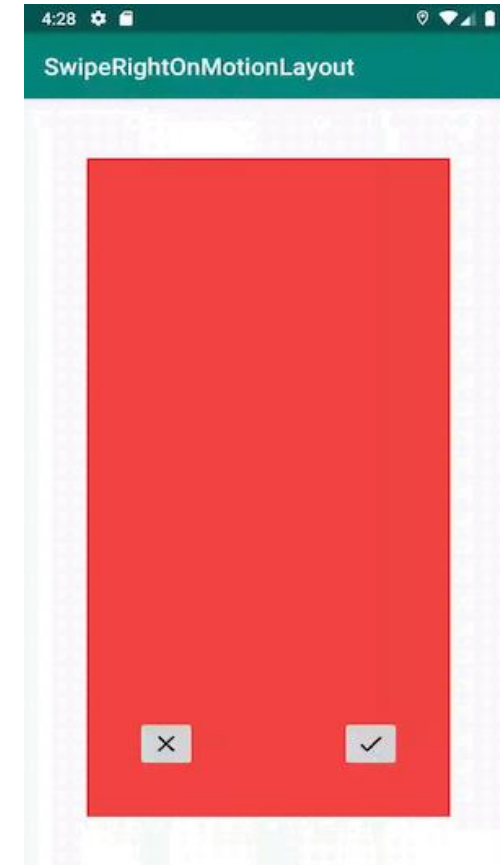
```
override fun onTransitionTrigger(motionLayout: MotionLayout?,  
    triggerId: Int, positive: Boolean,  
    progress: Float) {  
    ...  
    if (positive) {  
        openFragment(SecondFragment())  
    } else {  
        openFragment(FirstFragment())  
    }  
}
```



Несколько переходов

```
<Transition
    motion:constraintSetStart="@id/state_1"
    motion:constraintSetEnd="@id/state_2">
    <OnSwipe
        motion:dragDirection="dragRight"
        motion:touchAnchorId="@id/view" />
</Transition>

<Transition
    motion:constraintSetStart="@id/state_2"
    motion:constraintSetEnd="@id/state_3">
    <OnSwipe
        motion:dragDirection="dragRight"
        motion:touchAnchorId="@id/view" />
</Transition>
```

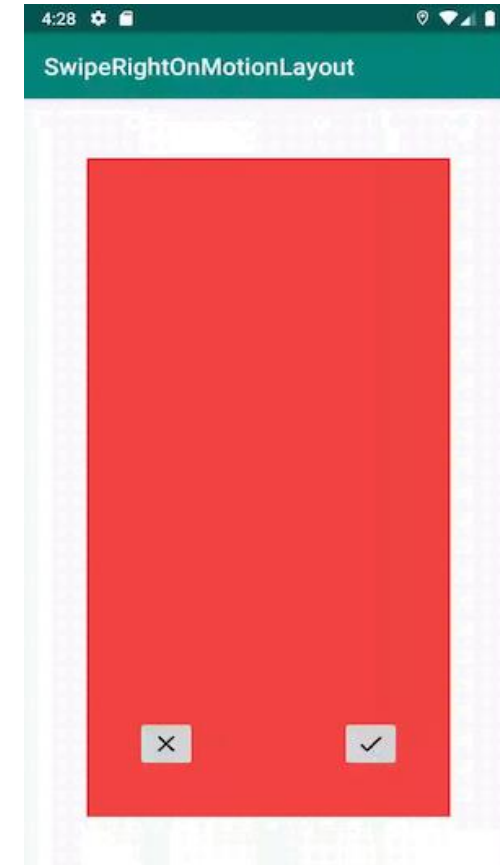


Несколько переходов

```
<Transition
  motion:constraintSetStart="@id/state_1"
  motion:constraintSetEnd="@id/state_2">
  <OnSwipe
    motion:dragDirection="dragRight"
    motion:touchAnchorId="@id/view" />
</Transition>

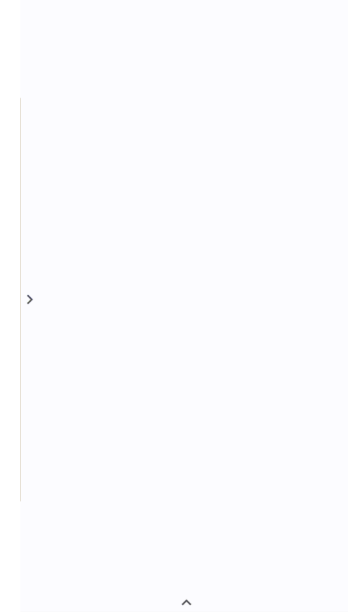
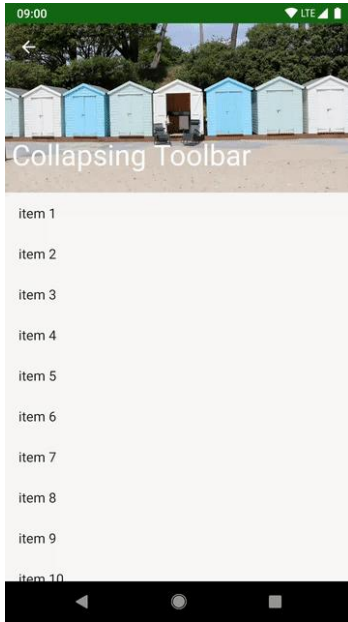
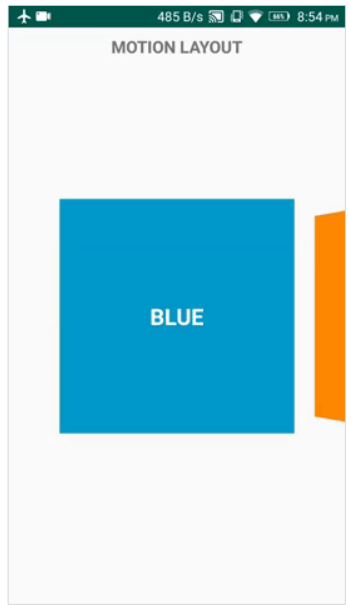
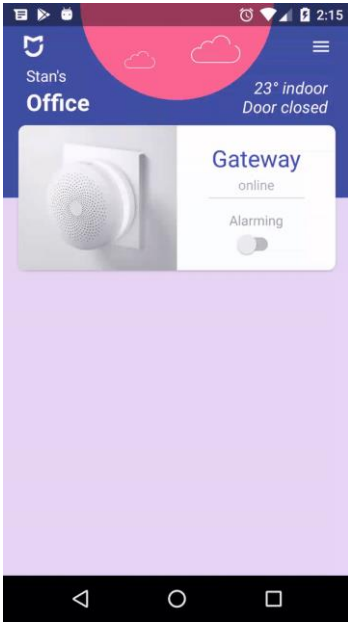
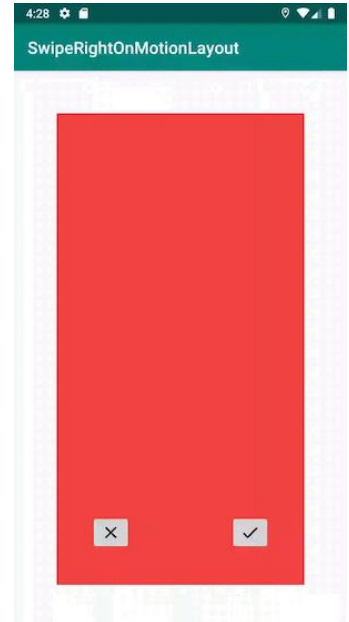
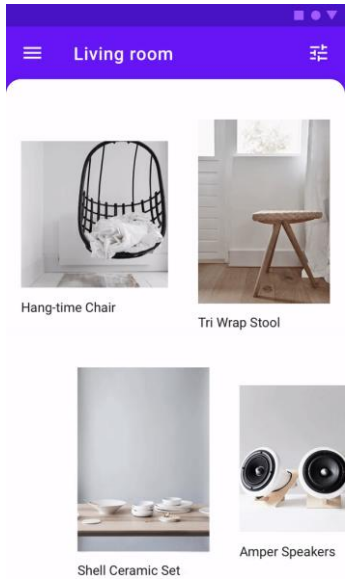
<Transition
  app:autoTransition="animateToEnd"
  motion:constraintSetStart="@id/state_2"
  motion:constraintSetEnd="@id/state_3">
  app:duration="150">

</Transition>
```



Пример #4

- Работа из кода
- `TransitionListener`
- Передача прогресса в другой `MotionLayout`
- Работа с триггерами
- Смена фрагмента по триггеру
- Несколько переходов



Выводы

Плюсы и минусы

- + Понятная простая концепция. Легкий вход
- + Почти без кода
- + Автоматическая анимация между сценами
- + Большие возможности анимации (+ анимация цвета, размера текста...)
- + Работа с жестами
- + Ключевые кадры
- + Set progress
- + Editor в будущем

- Editor пока не в релизе
- Действует только на прямых child view
- Анимацией размера текста не идеальна
- Все еще в бете
- Не все свойства работают правильно или ожидаемо

Не подходит:

- Простые анимации единичных объектов
(когда проще использовать `LayoutTransition` или `TransitionFramework`)
- Enter/Exit transitions и `SharedElement Transitions`

Подходит:

- Переходы между сценами (вместо `TransitionFramework`)
- Swipe/scroll анимация (collapse toolbar, nav drawer, bottom sheet, back drop...)
- Проще, чем кастомизация стандартного решения
- Ключевые кадры, `Set progress` анимации и т.д.
- Смена цвета элементов, размера текста, и др. `Custom attributes`

Sample project:

<https://github.com/andreytumarinson/MotionLayout-sample>

Документация:

<https://developer.android.com/reference/android/support/constraint/motion/MotionLayout>

Уроки от Google:

<https://medium.com/google-developers/introduction-to-motionlayout-part-i-29208674b10d>

Создание tutorial экрана:

<https://www.raywenderlich.com/8883-motionlayout-tutorial-for-android-getting-started>

MotionLayout on Google I/O '18:

<https://www.youtube.com/watch?v=ytZteMo4ETk&feature=youtu.be&t=28m45s>

Google examples:

<https://developer.android.com/training/constraint-layout/motionlayout/examples>

Google examples repo:

<https://github.com/android/views-widgets-samples/tree/master/ConstraintLayoutExamples>

Other examples:

<https://github.com/mtcn/MotionLayoutExamples>

Codelab:

<https://codelabs.developers.google.com/codelabs/motion-layout>

Badoo swipe to like:

<https://habr.com/ru/company/badoo/blog/458854/>

Collapsing toolbar:

<https://blog.stylingandroid.com/motionlayout-collapsing-toolbar-part-1/>

Carousel repo:

<https://github.com/faob-dev/MotionLayoutCarousel>

Контакты

Тумаринсон Андрей Владимирович

 2marinson@mail.ru

 vk.com/caligo_tum

MERA

СПАСИБО