

Android 转场动画

课程目标

- ✓ 掌握定义转场动画的方法

学习内容

➤ 作用

➤ 时机

➤ 方法

作用

➤ 视觉连续性

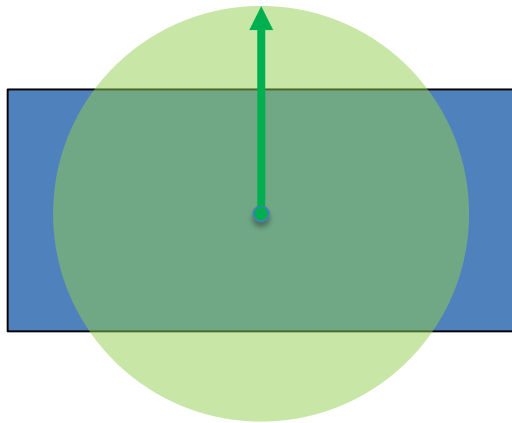
时机

➤ 视觉状态改变

- 单个视图
- 布局
- Activity跳转

揭露效果

➤ View状态改变



android.transition

➤ **Scene**

➤ **Transition**

➤ **TransitionManager**

Scene

```
Scene.getSceneForLayout(  
    sceneRoot,  
    R.layout.scene_overview,  
    this);
```

Transition

➤ **Fade**



➤ **ChangeBounds**

➤ **AutoTransition**



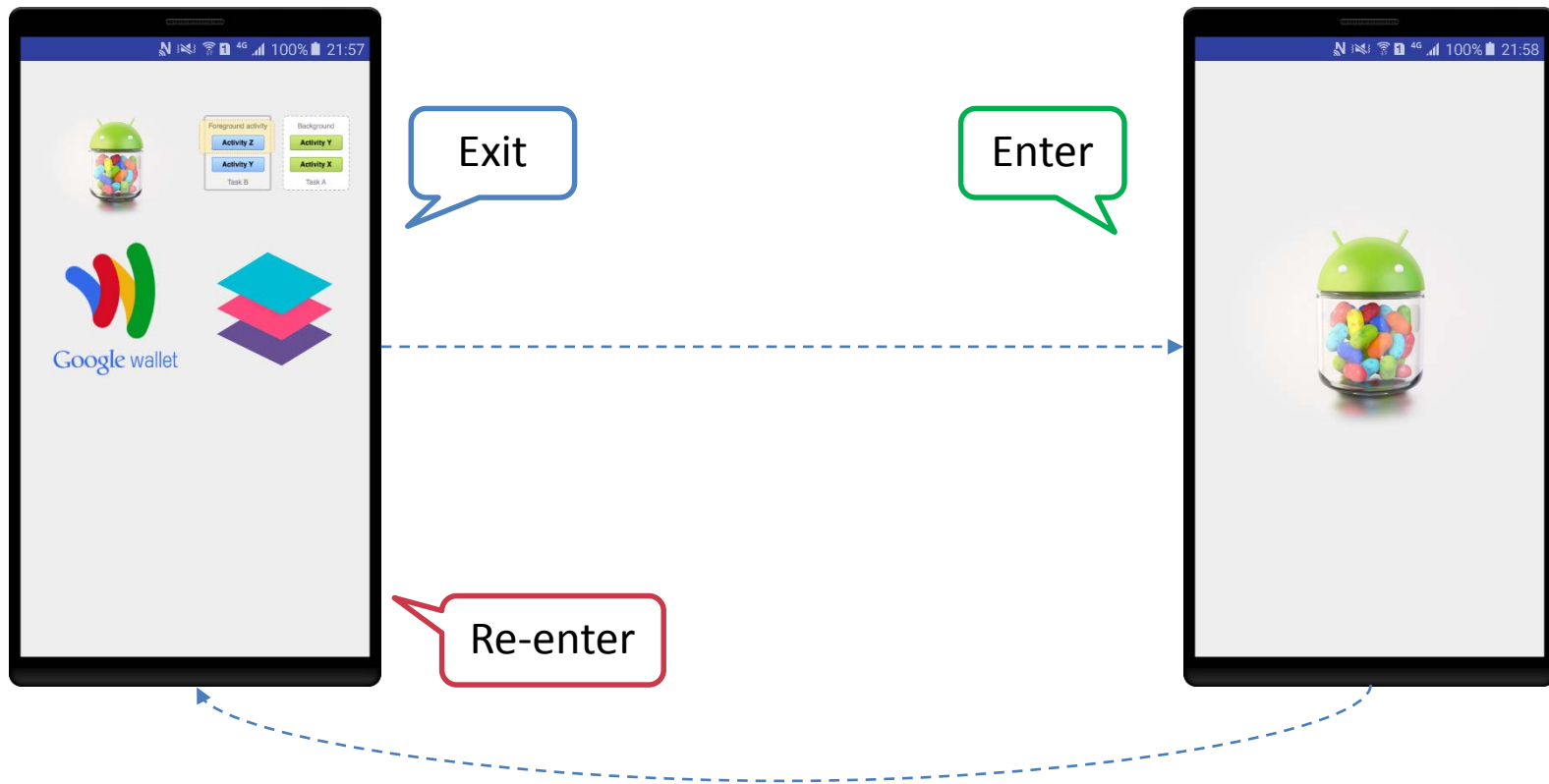
Transition

```
TransitionInflater.from(getBaseContext())  
    .inflateTransition(R.transition.transition);
```

TransitionManager

```
TransitionManager.go(mTargetScene, transition);
```

Activity跳转

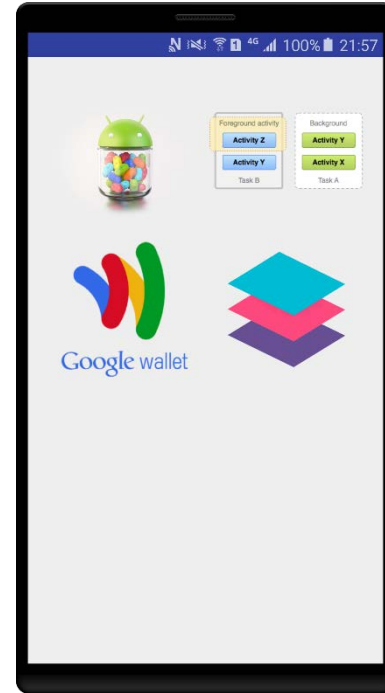


ActivityOptions

```
ActivityOptions options  
    = ActivityOptions.makeSceneTransitionAnimation(this);  
startActivity(intent, options.toBundle());
```

Transition

- **Fade**
- **Slide**
- **Explode**



启动转场

```
<style name="BaseAppTheme" parent="android:Theme.Material">
  <!-- 启动窗口内容转场 -->
  <item name="android:windowContentTransitions">true</item>

  <!-- 窗口进入/离开时演示的动画 -->
  <item name="android:windowEnterTransition">@transition/explode</item>
  <item name="android:windowExitTransition">@transition/explode</item>

  <!-- 共享元素进入/离开时演示的动画 -->
  <item name="android:windowSharedElementEnterTransition">
    @transition/change_image_transform</item>
  <item name="android:windowSharedElementExitTransition">
    @transition/change_image_transform</item>
</style>
```

小结

- 转场
- 视觉连续