

Practical Transition

by punchdrunker

自己紹介

- 2010年～ iOS/Androidのアプリを書いてみる
- 2011年～ SNS mixi(ミクシィ)
- 2016年～ 家族アルバム みてね(ミクシィ)
- DroidKaigiとかshibuya.apkを運営

Sample code

- <https://github.com/punchdrunker/hocho>
- [android-transition-examples](#)
 - <https://github.com/google/android-transition-examples>

目次

- SharedElementを使うまでの経緯
- 用語解説
- お手本
- 基本的な使いかた
- 実際のアプリへの適用
- 問題の解決法と諦めた事
- まとめ

経緯

写真一覧画面から写真の詳細画面に遷移した時に...

- Fling to finish(上下スワイプで閉じる)を実装したい
- どうせなら粋なアニメーション演出も出来ると嬉しい
- ドキュメント通りに書いてるのに動かん!!!!1
 - 書いた通りに動いてました(反省) x 100
- 完全に理解した!!
- 挫折

用語の説明(1)

- Transition framework(遷移)
 - 画面遷移でレイアウトにアニメーション効果を付けることもできるフレームワークの総称
 - minSdkVersion 21
- Transition
 - Transition Managerによって管理される、"遷移"そのもの

用語の説明(2)

- **SharedElement(View)**
 - 遷移元と遷移先で繋ぎ合わせたいView。optionとしてstartActivityに渡す。
- **Transition name(String)**
 - 遷移元と遷移先のViewに同じtransitionNameを付ける事でSharedElementとして認識される

お手本にしたもの

- **android-transition-examples**
 - **<https://github.com/google/android-transition-examples>**
- **twitter公式クライアント**

基本的な使い方

Transitionのパターン

- **Fragment to Fragment**
- **Activity to Activity**

Fragment to Fragment

1つのActivity内での遷移はgoogle exampleが好例。
遷移する時のFragmentManagerにaddSharedElement
メソッドで遷移元のImageViewとtransitionNameをペアでセットする。
遷移先でも同様。
(transitionNameの設定はコード側からでもレイアウトxmlからでも可能。)

Fragment to Fragment

実装例

```
ImageView transitioningView = view.findViewById(R.id.card_image);
fragment.getFragmentManager()
    .beginTransaction()
    .setReorderingAllowed(true) // Optimize for shared element transition
    .addSharedElement(
        transitioningView,
        transitioningView.getTransitionName())
    .replace(
        R.id.fragment_container,
        new ImagePagerFragment(),
        ImagePagerFragment.class.getSimpleName())
    .addToBackStack(null)
    .commit();
```

sampleの動き

Activity to Activity

transitionNameをペアで設定するのは同じ。

startActivityのoptionとして、**sharedElement**をペアにした**bundle**を渡す。

複数渡したければ**Pair**型で可変長引数として渡せる。

```
val intent = ToActivity.createIntent(context, position)
val option = ActivityOptions.makeSceneTransitionAnimation(
    activity,
    view,
    getString(R.string.shared_element)).toBundle()
startActivity(intent, option)
```

実際のアプリへの適用

もともとの仕様

- RecyclerViewで写真をグリッド表示
- 遷移した先の詳細画面では左右にスワイプすると写真を移動できる(ViewPager)
- スワイプした後に画面を閉じてリストに戻ると、その位置にスクロールしている

問題とその解決法

ここからはサンプルを見ながら解説します(item1&2&3)

AppBar/フッタかぶる問題

上下に重なるViewがあったらスクロールで回避

```
private fun shouldScrollList(cell: View): Boolean {  
    val cellLocation = IntArray(2)  
    cell.getLocationOnScreen(cellLocation)  
    val cellTop = cellLocation[1]  
    val listLocation = IntArray(2)  
    binding.list.getLocationOnScreen(listLocation)  
    val diffToAppBar = cellTop - listLocation[1]  
  
    val cellBottom = cellTop + binding.list.height  
    val listBottom = listLocation[1] + binding.list.height  
    val diffToBottom = cellBottom - listBottom  
  
    return (diffToAppBar < 0 || diffToBottom > 0)  
}
```

ViewPagerで困ること

- item4で問題が発生する
- 回避策をitem5で対応

スワイプした後に戻り先のImageViewを変更するには
コールバックで動的に変更できる

ViewPagerとの連携

必須コールバック

- `setEnterSharedElementCallback`
- `setExitSharedElementCallback`

FragmentManagerを使った遷移では、Fragmentで利用

startActivityを使った遷移では、Activityで利用

コールバックで何ができるか

紐付く SharedElement を動的に変更できる

```
setExitSharedElementCallback(object : SharedElementCallback() {  
    // names: List<String>, sharedElements: MutableMap<String, View>  
    override fun onMapSharedElements(names: ..., sharedElements: ...) {  
        val viewHolder = fromFragment?.getViewHolder(newPosition)  
        val itemView = viewHolder?.itemView ?: return  
        val photoView = itemView.findViewById<ImageView>(R.id.card_photo)  
        sharedElements[names[0]] = photoView  
    }  
})
```

コールバックのタイミング

- Activity A(一覧)からActivity B(詳細)に遷移
 - AのExitSharedElementCallbackが呼ばれる
 - BのEnterSharedElementCallbackが呼ばれる
- BからAに戻る
 - BのEnterSharedElementCallbackが呼ばれる
 - AのExitSharedElementCallbackが呼ばれる

なぜ?

戻る時のActivity Transitionは来た時のanimationを
reverseする為

(finishAfterTransition あたりを読むと書いてある)

さらにスクロールが必要

諦めた



さらなる問題

Transition Animation が X??????a 系の端末でまともに動かなかった👻

鋼鐵網

ERN