

百度一下

您查询的关键词是：**circular reveal** 以下是该网页在北京时间 2016年08月27日 08:52:45 的快照：
如果打开速度慢，可以尝试[快速版](#)；如果想更新或删除快照，可以[投诉快照](#)。
百度和网页 http://blog.csdn.net/caroline_wendy/article/details/50756717 的作者无关，不对其内容负责。百度快照谨为网络故障时之索引，不代表被搜索网站的即时页面。

Mystra

知及之，仁不能守之，虽得之，必失之。

☰ 目录视图

⋮

[【公告】博客系统优化升级](#) [【知识库发布】国外敏捷转型案例-敏捷软件开发知识体系](#) [博乐招募开始啦](#) [不得不学的编程语言](#)

使用CircularReveal动画效果切换页面

标签：[Mystra](#) [Android](#) [Circular](#) [Reveal](#) [页面切换](#)

2016-02-27 20:17

1769人阅读

评论

☰ 分类：[Android开发 \(64 \)](#) ▼

❗ 版权声明：本文为博主原创文章，未经博主允许不得转载。

本文地址: http://blog.csdn.net/caroline_wendy/article/details/50756717

欢迎Follow我的[GitHub](#), 关注我的[CSDN](#).

Android的Material Design设计理念, 带来很多绚丽的动画效果. 在页面切换中, 最常用的就是**SharedElementTransition**, 通过进入时把控件变换为页面, 在退出时, 把页面变换为控件, 同时, 可以设置控件移动的轨迹. 这样的控件, 可以应用于**消息通知**, 或者户体验. 那么是如何实现的呢?



随着厂商的版本迭代, 超过三分之一的手机都是5.0以上的操作系统, 随着更多便宜的低端手机普及5.0+ 系统(如红米系列), 将会大大增加应用留存率.

个人资料



u012515223

[关注](#) [发私信](#)



访问：1328918次
积分：22671
等级：[BLOG > ?](#)
排名：第213名

原创：694篇 转载：1篇
译文：0篇 评论：324条

自我简介

本人代码娴熟, 涉猎广泛.
欢迎Follow我的[GitHub](#).
欢迎访问我的[个人博客](#):
<http://www.wangchenlong.org/>

文章搜索

文章分类

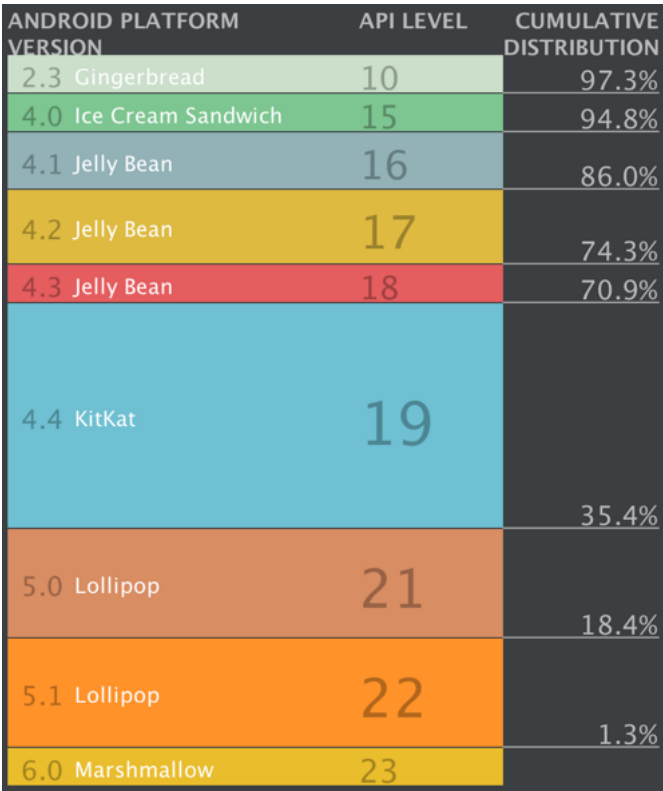
[Android开发](#) (65)
[RN开发](#) (19)
[Work随笔](#) (9)
[Android](#) (151)
[IOS](#) (33)
[Python](#) (42)
[Java](#) (35)

- C++ (102)
- Algorithm (92)
- Design Pattern (27)
- Swift (0)

- 文章存档
- 2016年06月 (5)
- 2016年05月 (9)
- 2016年04月 (10)
- 2016年03月 (4)
- 2016年02月 (11)
- 2016年01月 (18)
- 2015年12月 (11)
- 2015年11月 (17)
- 2015年10月 (11)
- 2015年09月 (6)
- 2015年08月 (15)
- 2015年07月 (55)
- 2015年05月 (1)
- 2015年04月 (1)
- 2015年02月 (13)
- 2015年01月 (13)
- 2014年12月 (18)
- 2014年11月 (53)
- 2014年10月 (24)
- 2014年09月 (28)
- 2014年07月 (61)
- 2014年06月 (51)
- 2014年05月 (21)
- 2014年04月 (67)
- 2014年03月 (56)
- 2014年02月 (5)
- 2014年01月 (3)
- 2013年12月 (38)
- 2013年11月 (51)
- 2013年10月 (21)

- 阅读排行
- Microsoft Visual Studio 201... (22589)
- Android - Android Studio修... (18884)
- C++ - "emplace_back" 和 "p... (12915)
- OpenCV - cv::Mat 和 IplImag... (12046)
- Android - Android Studio 自... (11014)
- 模式识别 - 局部二值模式(Loca... (10666)
- Android - Android Studio 安... (10561)
- Python - 字符串(string) 详解... (10162)
- Eclipse - "Eclipse CDT" (Eclip... (8709)
- Python - 格式化(format())输... (8439)

- 评论排行
- Android 6.0: 动态权限管理的... (14)
- 使用BatteryHistorian分析和优... (11)
- 使用RecyclerView实现滚动控... (11)
- 炫丽的朋友圈视频滚动播放功能 (10)
- 笔试 - 高德软件有限公司pyth... (10)
- 应用插件化实践--DroidPlugin... (10)
- 动画效果--漫天飞雪 (9)
- 防止密码输入错误, 密码明文显... (7)
- 属性动画的概念解析--实现星... (7)
- ReactNative的Atom编辑器 (7)

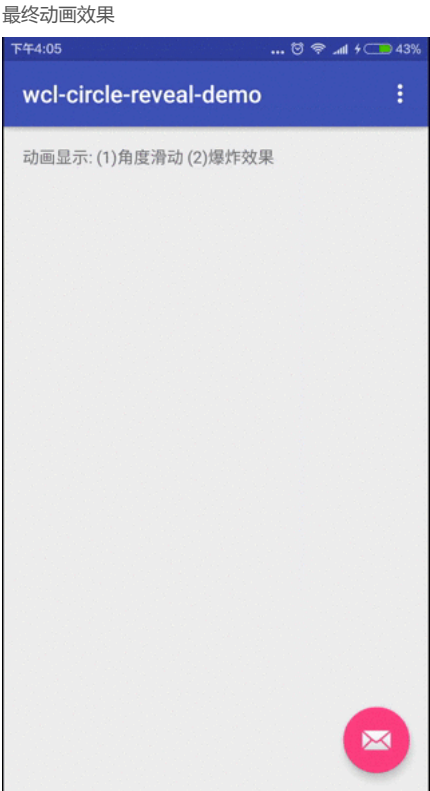


本文主要内容:

(1) 修改分享元素的滑动轨迹, 以90°圆弧方式出现和返回, 即transition动画.

(2) 生成和销毁页面的爆炸和凝聚效果, 即CircularReveal的使用方式.

本文源码的GitHub[下载地址](#).



1. 首页

新建一个含义Fab按钮的HelloWorld工程. 添加ButterKnife和Lambda库.

推荐文章

- * 郭神带你真正理解沉浸式模式
- * 优秀代码的格式准则
- * Hadoop的数据仓库实践——OLAP与数据可视化（二）
- * Android 视图篇——恼人的分割线留白解决之道
- * 移动端开发者眼中的前端开发流程变迁与前后端分离

最新评论

Android - Gradle "xxx" project refresh ...
Cheng123456789000 : asdasdasd

在应用中更新App版本
a591326912 : 感谢分享。没看懂楼主用的哪个框架。。。能提点下不

OpenCV - cv::Mat 和 IplImage 的转换
qingfenglu : Mat 装Ipl应该是IplImage ipl = matimg;

C语言 - 结构体(struct)的位字段() 详解
qqyuanhao163 : 可能是IDE的问题吧，我用VS2008。第一，越界会报错的。第二，sizeof输出的结果为8。

热爱编程
u011240877 : 编程之路，就是不断磨练、提升思维的过程

使用 RxAndroid 处理异步任务
u010632913 : 这样就that's all了吗

使用CircularReveal动画效果切换页面
u013503465 : 4.4的如何兼容呢

使用DialogFragment实现底部弹窗布局
Backkom1209 : 动画呢？咋弄

Android - Android Studio 的 Preview窗...
qq_26783345 : 我的View -> Tool Windows 里面没有Preview怎么办

重写MPAndroidChart显示标记
u014234287 : 楼主求救。我的x坐标太长了。我想换行。怎么办啊。。。我快疯了。。。救命啊

设置Fab按钮的跳转事件.

```
// Fab的跳转事件
public void startOtherActivity(View view) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        ActivityOptions options =
            ActivityOptions.makeSceneTransitionAnimation(this, mFab, mFab.getTransitionName());
        startActivity(new Intent(this, OtherActivity.class), options.toBundle());
    } else {
        startActivity(new Intent(this, OtherActivity.class));
    }
}
```

确保版本号大于5.0, 支持Material Design的动画效果.

设置Fab的TransitionName, 即变化名称.

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="@dimen/fab_margin"
    android:clickable="true"
    android:onClick="startOtherActivity"
    android:src="@android:drawable/ic_dialog_email"
    android:transitionName="@string/other_transition_name"
    tools:targetApi="lollipop"/>
```

注意android:transitionName属性, 表示变换名称.

2. 跳转页

页面由背景, 变化控件, 关闭控件, 这三部分组成.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/other_rl_container"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/transparent">

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/other_fab_circle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:transitionName="@string/other_transition_name"
        app:backgroundTint="@color/colorAccent"
        app:elevation="0dp"
        app:fabSize="normal"
        app:pressedTranslationZ="8dp"
        tools:targetApi="21"/>

    <TextView
        android:id="@+id/other_tv_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/colorAccent"
        android:gravity="center"
        android:text="@string/other_text"
        android:textColor="@android:color/white"
        android:textSize="40sp"
        android:textStyle="bold"
        android:visibility="invisible"
        tools:visibility="visible"/>
```

```
<ImageView
    android:id="@+id/other_iv_close"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:clickable="true"
    android:contentDescription="@null"
    android:onClick="backActivity"
    android:src="@drawable/ic_close_white"
    android:visibility="invisible"/>

</RelativeLayout>
```

注意, 变换控件FloatingActionButton与主页的变换控件有相同的transitionName.

显示逻辑, 设置入场和退场动画, 爆炸的动画效果.

```
@Override protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_other);
    ButterKnife.bind(this);

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        setupEnterAnimation(); // 入场动画
        setupExitAnimation(); // 退场动画
    } else {
        initView();
    }
}
```

退出逻辑, 退出动画, 凝聚的动画效果.

```
// 退出按钮
public void backActivity(View view) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        onBackPressed();
    } else {
        defaultBackPressed();
    }
}
```

下面仔细分析显示和退场动画.

3. 显示动画

分享元素切换使用弧度规矩, 即arc_motion, 90°旋转过去和回来. 在动画结束后, 页面显示使用爆炸效果.

```
// 入场动画
@TargetApi(Build.VERSION_CODES.LOLLIPOP)
private void setupEnterAnimation() {
    Transition transition = TransitionInflater.from(this)
        .inflateTransition(R.transition.arc_motion);
    getWindow().setSharedElementEnterTransition(transition);
    transition.addListener(new Transition.TransitionListener() {
        @Override public void onTransitionStart(Transition transition) {

        }

        @Override public void onTransitionEnd(Transition transition) {
            transition.removeListener(this);
            animateRevealShow();
        }

        @Override public void onTransitionCancel(Transition transition) {

        }

        @Override public void onTransitionPause(Transition transition) {

        }

        @Override public void onTransitionResume(Transition transition) {
```

```
    }
    });
}

// 动画展示
@TargetApi (Build.VERSION_CODES.LOLLIPOP)
private void animateRevealShow() {
    GuiUtils.animateRevealShow(
        this, mRlContainer,
        mFabCircle.getWidth() / 2, R.color.colorAccent,
        new GuiUtils.OnRevealAnimationListener() {
            @Override public void onRevealHide() {

            }

            @Override public void onRevealShow() {
                initView();
            }
        });
}
```

arc_motion角度变换

```
<transitionSet
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="500"
    android:interpolator="@android:interpolator/linear_out_slow_in">

    <changeBounds>
        <!--suppress AndroidElementNotAllowed -->
        <arcMotion
            android:maximumAngle="90"
            android:minimumHorizontalAngle="90"
            android:minimumVerticalAngle="0"/>
    </changeBounds>
</transitionSet>
```

爆炸的动画效果

```
// 圆圈爆炸效果显示
@TargetApi (Build.VERSION_CODES.LOLLIPOP)
public static void animateRevealShow(
    final Context context, final View view,
    final int startRadius, @ColorRes int color,
    OnRevealAnimationListener listener) {
    int cx = (view.getLeft() + view.getRight()) / 2;
    int cy = (view.getTop() + view.getBottom()) / 2;

    float finalRadius = (float) Math.hypot(view.getWidth(), view.getHeight());

    // 设置圆形显示动画
    Animator anim = ViewAnimationUtils.createCircularReveal(view, cx, cy, startRadius, finalRadius);
    anim.setDuration(300);
    anim.setInterpolator(new AccelerateDecelerateInterpolator());
    anim.addListener(new AnimatorListenerAdapter() {
        @Override public void onAnimationEnd(Animator animation) {
            super.onAnimationEnd(animation);
            view.setVisibility(View.VISIBLE);
            listener.onRevealShow();
        }

        @Override public void onAnimationStart(Animator animation) {
            super.onAnimationStart(animation);
            view.setBackgroundColor(ContextCompat.getColor(context, color));
        }
    });

    anim.start();
}
```

使用CircularReveal, 即圆形显示的动画效果。
第一个参数是显示的视图, 第二个和第三个是变换的中心位置, 第三个和第四个是变换的起始半径和结束半径。

4. 退出动画

退出动画是凝聚效果, 同样使用的是CircularReveal.

```
// 退出事件
@Override public void onBackPressed() {
    GuiUtils.animateRevealHide(
        this, mRlContainer,
        mFabCircle.getWidth() / 2, R.color.colorAccent,
        new GuiUtils.OnRevealAnimationListener() {
            @Override
            public void onRevealHide() {
                defaultBackPressed();
            }

            @Override
            public void onRevealShow() {

            }
        });
}
```

退出和显示的区别就是起始和终止的半径不同.

```
// 圆圈凝聚效果
@TargetApi(Build.VERSION_CODES.LOLLIPOP)
public static void animateRevealHide(
    final Context context, final View view,
    final int finalRadius, @ColorRes int color,
    OnRevealAnimationListener listener
) {
    int cx = (view.getLeft() + view.getRight()) / 2;
    int cy = (view.getTop() + view.getBottom()) / 2;
    int initialRadius = view.getWidth();
    // 与入场动画的区别就是圆圈起始和终止的半径相反
    Animator anim = ViewAnimationUtils.createCircularReveal(view, cx, cy, initialRadius, finalRadius);
    anim.setDuration(300);
    anim.setInterpolator(new AccelerateDecelerateInterpolator());
    anim.addListener(new AnimatorListenerAdapter() {
        @Override public void onAnimationStart(Animator animation) {
            super.onAnimationStart(animation);
            view.setBackgroundColor(ContextCompat.getColor(context, color));
        }

        @Override public void onAnimationEnd(Animator animation) {
            super.onAnimationEnd(animation);
            listener.onRevealHide();
            view.setVisibility(View.INVISIBLE);
        }
    });
    anim.start();
}
```

最后说一些有关用户体验的事情, 对于一款应用而言, 好的性能固然重要, 但优秀的设计也非常关键, 优异的产品需要优异的展现方

OK, that' s all! Enjoy it!

顶 踩

4 1

- [上一篇](#) 实现卡片翻转的动画效果
- [下一篇](#) 使用BatteryHistorian分析和优化应用电量

猜你在找

查看评论

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场