|  |  |  |
| --- | --- | --- |
| **修改内容** | **修改人** | **修改时间** |
| 添加变色不重启application但重启activity的接口  参考 六 | 刘帅 | 2015-1-1 |
| 添加变色回调接口（当接收到变色广播时，会回调应用注册的监听）  参考 七 | 刘帅 | 2015-1-30 |
| 添加带参数的变色回调接口（参数为变色的类型） | 赵玉龙 | 2015-5-25 |
| 添加注册不变色页面的接口（红色部分标出） | 魏栋 | 2016-8-19 |

## 变色龙接口文档

1. 变色龙整体架构



1. 省电模式广播定义

//省电模式的广播action: "amigo.intent.action.chameleon.POWER\_SAVING\_MODE"

Intent mPowerSavingModeIntent = **new** Intent("amigo.intent.action.chameleon.POWER\_SAVING\_MODE");

//省电模式开关：isChecked为true,通知变色龙进入省电模式；isChecked为false,关闭省电模式

mPowerSavingModeIntent.putExtra("is\_power\_saving\_mode", isChecked);

//发送打开/关闭省电模式广播

sendBroadcast(mPowerSavingModeIntent);

1. 查询是否是省电模式

通过以下代码查询：true为省电模式；false为非省电模式

ChameleonColorManager.getInstance().isPowerSavingMode();

1. ChameleonColorManager类的接口定义
2. **public static** **synchronized** ChameleonColorManager getInstance()

获取ChameleonColorManager的单例对象，用于注册和反注册

1. **public** **void** register(Context context)

注册：应用调用此接口后，ChameleonColorManager会做以下几件事情

（1）初始化ChameleonColorManager里的颜色信息， （2）注册变色龙的BroadcastReceiver，接收到变色龙的广播后，会重启应用

1. **public** **void** register(Context context , **boolean** restart )

当restart为true时： 功能跟**public** **void** register(Context context)相同，接收到变色广播后会重启应用

当restart为false时，不会重启应用，但会重启注册的Activity：

应用调用此接口后，ChameleonColorManager会做以下几件事情

（1）初始化ChameleonColorManager里的颜色信息，

（2）注册变色龙的BroadcastReceiver，接收到变色龙的广播后，会重新读取所有的颜色配置信息，并且重启调用了以下方法的Activity。

方法如下：

ChameleonColorManager.getInstance().onCreate(this);

1. **public** **void** onCreate(Actitity activity )

将Activity放入ChameleonColorManager的队列，当接收到变色广播后，会重启队列里所有的Activity

1. **public** **void** onDestroy (Actitity activity )

将Activity从ChameleonColorManager的队列里移除

1. **public** **void** addOnChangeColorListener (OnChangeColorListener onChangeColorListener)

添加变色回调，当接收到变色广播后，会回调onChangeColorListener的onChangeColor()接口.

注意：当不需要使用此回调时需removeOnChangeColorListener。

1. **public void** addOnChangeColorListenerWithParams(OnChangeColorListenerWithParams onchangeColorListenerWithParams)

添加变色回调，当接收到变色广播后，会回调OnChangeColorListenerWithParams的onChangeColor(final int changeColorType)接口.

注意：

changeColorType的取值只有2种：

//默认的变色类别

public static final int TYPE\_DEFAULT\_CHANGE\_COLOR = 0;

//切换主题的变色类别

public static final int TYPE\_THEME\_CHANGE\_COLOR = 1;

当不需要使用此回调时需removeOnChangeColorListenerWithParams。

1. **public void** removeOnChangeColorListener(OnChangeColorListener onChangeColorListener)

删除变色回调

1. **public void** removeOnChangeColorListenerWithParams(OnChangeColorListenerWithParams onChangeColorListenerWithParams)

删除带参数的变色回调

1. **public** **void** unregister()

反注册：应用调用此接口后，会反注册变色龙的BroadcastReceiver

1. **public** **static** **boolean** isNeedChangeColor()

应用据此判断，是否要执行变色逻辑

1. **public** **static** **int** getAppbarColor\_A1()

获取顶部ActionBar背景颜色

1. **public** **static** **int** getBackgroudColor\_B1()

获取应用的主区域背景颜色

1. **public** **static** **int** getPopupBackgroudColor\_B2()

获取popup的背景色

1. **public** **static** **int** getEditTextBackgroudColor\_B3()

获取EditText背景颜色

1. **public** **static** **int** getButtonBackgroudColor\_B4()

获取Button背景颜色

1. **public** **static** **int** getStatusbarBackgroudColor\_S1()

获取Statusbar背景颜色

1. **public** **static** **int** getSystemNavigationBackgroudColor\_S2()

获取系统虚拟按键背景颜色

1. **public** **static** **int** getAccentColor\_G1()

获取强调色（多用于小控件Enable状态颜色）

1. **public** **static** **int** getContentColorPrimaryOnAppbar\_T1()

获取顶部ActionBar前景颜色（如：文字、图标等）

1. **public** **static** **int** getContentColorSecondaryOnAppbar\_T2()

获取顶部ActionBar前景颜色（如：文字、图标等）二档

1. **public** **static** **int** getContentColorThirdlyOnAppbar\_T3()

获取顶部ActionBar前景颜色（如：文字、图标等）三档

1. **public** **static** **int** getContentColorForthlyOnAppbar\_T4()

获取顶部ActionBar前景颜色（如：文字、图标等）四档

1. **public** **static** **int** getContentColorPrimaryOnBackgroud\_C1()

获取应用的主区域上前景颜色（如：文字、图标等）

1. **public** **static** **int** getContentColorSecondaryOnBackgroud\_C2()

获取应用的主区域上前景颜色（如：文字、图标等）二档

1. **public** **static** **int** getContentColorThirdlyOnBackgroud\_C3()

获取应用的主区域上前景颜色（如：文字、图标等）三档

1. **public** **static** **int** getContentColorOnStatusbar\_S3()

获取状态栏上的前景颜色

1. **public** **static** **int** getThemeType()

获取主题类型（明主题、暗主题）1：明主题；2：暗主题

1. **public** **static** **boolean** isPowerSavingMode()

是否是省电模式， true:是省电模式；false:不是省电模式

30．**public void** registerNoChangeColor(Activity activity)

将Activity放入ChameleonColorManager的不变色队列，当应用变色时，放到此队列中的应用页面不会进行变色

31．**public void** unregisterNoChangeColor (Activity activity)

将Activity从ChameleonColorManager的不变色队列中移除

1. 变色功能使用（会重启应用）：
2. 应用需要在Application启动时调用如下方法：

ChameleonColorManager.getInstance().register(**this**);

1. 需在AndroidManifest.xml里添加

<uses-permission **android:name=**"android.permission.KILL\_BACKGROUND\_PROCESSES"/>

应用接收到变色广播后会重启。

1. 使用amigo公共控件的，公共控件会去做变色，应用不用关心
2. 未使用公共控件的地方，应用需要使用ChameleonColorManager类提供的颜色去设置。

参考代码如下：

if(ChameleonColorManager . isNeedChangeColor ()){

view.setBackgroundColor (ChameleonColorManager . getAppbarColor\_A1 ());

}

1. 变色功能使用（不会重启应用，会重新加载Activity）:

1. 应用需要在Application启动时调用如下方法：

ChameleonColorManager.getInstance().register(**this, false**);

1. 需在所有Activity里添加

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

ChameleonColorManager.getInstance().onCreate(this);

}

@Override

protected void onDestroy() {

// TODO Auto-generated method stub

super.onDestroy();

ChameleonColorManager.getInstance().onDestroy(this);

}

1. 使用amigo公共控件的，公共控件会去做变色，应用不用关心
2. 未使用公共控件的地方，应用需要使用ChameleonColorManager类提供的颜色去设置。

参考代码如下：

if(ChameleonColorManager . isNeedChangeColor ()){

view.setBackgroundColor (ChameleonColorManager . getAppbarColor\_A1 ());

}

1. 变色功能使用（不会重启应用，也不会重启activity，当接收到变色广播时会回调应用设置的接口）:

1. 应用需要在Application启动时调用如下方法：

ChameleonColorManager.getInstance().register(**this, false**);

2.

mOnChangeColorListener = new OnChangeColorListener() {

@Override

public void onChangeColor() {

// TODO Auto-generated method stub

}

};

mOnChangeColorListenerWithParams = new OnChangeColorListenerWithParams () {

@Override

public void onChangeColor(final int changeColorType) {

// TODO Auto-generated method stub

}

}; ChameleonColorManager.getInstance().addOnChangeColorListener(mOnChangeColorListener);

ChameleonColorManager.getInstance().addOnChangeColorListener(mOnChangeColorListenerWithParams);

注意：当不需要使用此回调时需removeOnChangeColorListener;

当不需要使用此回调时需removeOnChangeColorListenerWithParams。

1. 使用变色时控制应用内的页面不进行变色

需在不变色的Activity里添加

@Override

public void onCreate(Bundle savedInstanceState) {

ChameleonColorManager.getInstance().registerNoChangeColor(this); super.onCreate(savedInstanceState);

}

@Override

protected void onDestroy() {

// TODO Auto-generated method stub

super.onDestroy();

ChameleonColorManager.getInstance().unregisterNoChangeColor(this);

}

注意：registerNoChangeColor此方法必须在super.onCreate之前调用，并且此接口不能和ChameleonColorManager.getInstance().onCreate(this);同时用于同一个activity.

1. 公共控件的使用

1. AmigoTextView

当应用在xml里使用特定的颜色资源去设置textcolor时，AmigoTextView会根据这些特定的资源跟C1,C2,C3,T1,T2,T3,T4进行对应，进而实现变色。

例如：

|  |  |  |
| --- | --- | --- |
| 资源名称 | 色值 | 对应变色龙资源 |
| android:textColor="@amigo:color/amigo\_content\_color\_primary\_on\_backgroud\_c1" | #CC000000 | ContentColorPrimaryOnBackgroud\_C1 |
| android:textColor="@amigo:color/amigo\_content\_color\_secondary\_on\_backgroud\_c2" | #66000000 | ContentColorSecondaryOnBackgroud\_C2 |
| android:textColor="@amigo:color/amigo\_content\_color\_thirdly\_on\_backgroud\_c3" | #33000000 | ContentColorThirdlyOnBackgroud\_C3 |
| android:textColor="@amigo:color/amigo\_content\_color\_primary\_on\_appbar\_t1" | #ffffffff | ContentColorPrimaryOnAppbar\_T1 |
| android:textColor="@amigo:color/amigo\_content\_color\_secondary\_on\_appbar\_t2" | #ccffffff | ContentColorSecondaryOnAppbar\_T2 |
| android:textColor="@amigo:color/amigo\_content\_color\_thirdly\_on\_appbar\_t3" | #50ffffff | ContentColorThirdlyOnAppbar\_T3 |
| android:textColor="@amigo:color/amigo\_content\_olor\_forthly\_on\_appbar\_t4" | #20ffffff | ContentColorForthlyOnAppbar\_T4 |
| android:textColor="@amigo:color/ amigo\_accent\_color\_g1" | #cc6600 | AccentColor\_G1 |

2. AmigoEditText

当应用在xml里使用特定的颜色资源去设置textcolor时，AmigoEditText会根据这些特定的资源跟指定的资源进行对应，进而实现变色。

对应关系如下：

|  |  |  |
| --- | --- | --- |
| 资源名称 | 色值 | 对应变色龙的资源 |
| android:textColor="@amigo:color/content\_color\_primary\_on\_backgroud\_c1" | #CC000000 | TextColor: ContentColorPrimaryOnBackgroud\_C1  HintTextColor: ContentColorSecondaryOnBackgroud\_C2 |
| android:textColor="@amigo:color/content\_color\_primary\_on\_appbar\_t1" | #ffffffff | TextColor: ContentColorPrimaryOnAppbar\_T1  HintTextColor: ContentColorSecondaryOnAppbar\_T2 |

1. 已封装的公共控件如下，这些控件的变色已由公共控件处理，应用不用关心。

具体的应用场景请参考公共控件demo。

svn地址：<http://192.168.110.97/svn/Amigo_WidgetDemo/trunk/>

AmigoActionBar, (具体场景请参考公共控件demo)

AmigoAlertDialog,

AmigoListView(包含快速滑动条、滑动条、按下涟漪效果),

AmigoActivity(窗口背景及状态栏变色)，

AmigoButton,

ContextMenu,

OptionsMenu,

AmigoPreference,

AmigoSwitchPreference,

AmigoListPreference,

AmigoEditText,

AmigoTextView,

AmigoCheckBox,

AmigoRadioButton,

AmigoSwitch,

AmigoProgressBar,

AmigoExpandableListView,

AmigoSeekBar,

AmigoSpiner,

AmigoEditModeView，

AmigoSearchView,