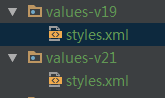
博客地址：http://www.jianshu.com/p/0acc12c29c1b

设置状态栏颜色：

1. 在res目录下创建一个values-v21、values-v19子目录，然后分别在子目录下创建一个styles.xml文件夹



1. 在values-v21的styles.xml文件中定义主题，供需要设置状态栏颜色的Activity引用

<!--方法1.将状态栏和导航栏的颜色设置成透明色 方式一适用于app中没有导航栏，且整体的背景是一张图片的界面-->  
<style name="ImageTranslucentTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <item name="android:windowTranslucentStatus">false</item>  
 <!--Android 5.x开始需要把颜色设置透明，否则导航栏会呈现系统默认的浅灰色-->  
 <item name="android:windowTranslucentNavigation">true</item>

<!--设置状态栏的颜色-->  
 <item name="android:statusBarColor">@color/colorRed</item>  
</style>

3.一定要在values-v19和values文件夹的styles.xml文件中定义同名的主题ImageTranslucentTheme，否则在低版本（<4.4）的机器上运行会报错

Values-v19的styles.xml文件中定义同名主题：

<style name="ImageTranslucentTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <item name="android:windowTranslucentStatus">true</item>  
 <item name="android:windowTranslucentNavigation">true</item>  
</style>

Values的styles.xml文件中定义同名主题：

<style name="ImageTranslucentTheme" parent="AppTheme">  
 <!--在Android 4.4之前的版本上运行，直接跟随系统主题-->  
</style>

4.在Activity中隐藏ActionBar

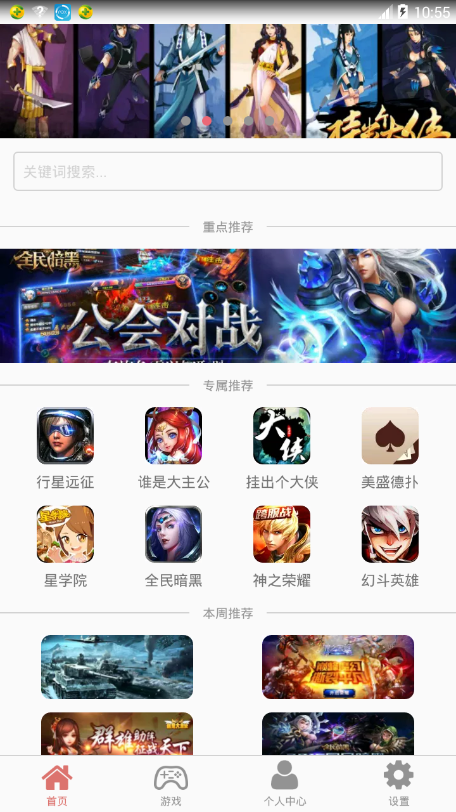
super.onCreate(savedInstanceState);

//隐藏ActionBar  
supportRequestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
setContentView(R.layout.*activity\_main*);

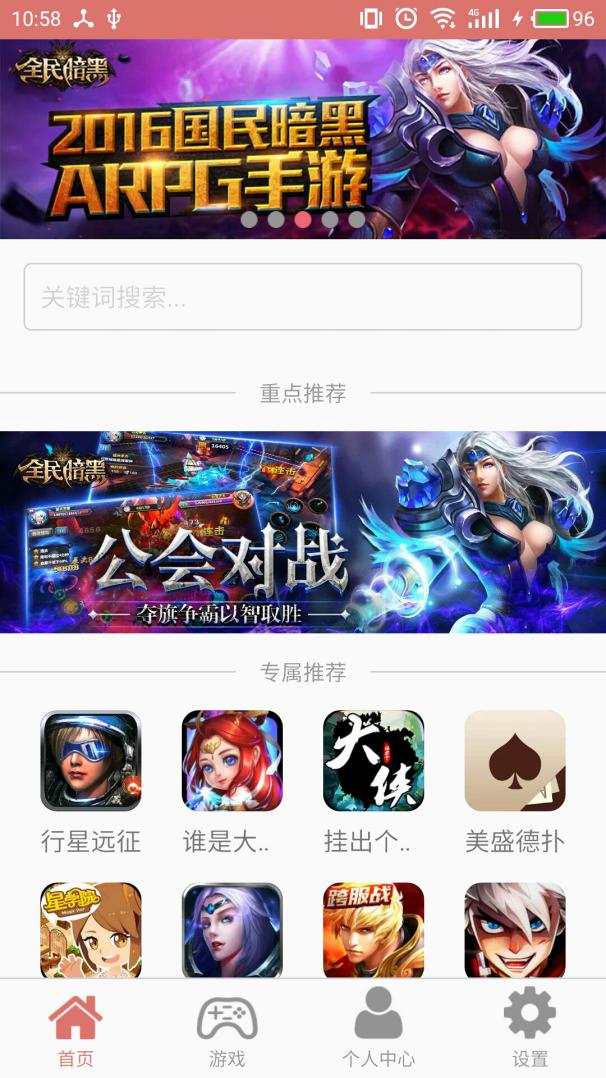
1. 在Activity对应的布局文件（最外层）中设置如下属性

android:fitsSystemWindows="true"

Android4.4设备上运行效果： 黑色的状态栏颜色变成了灰白渐变



Android5.0设备上运行效果：状态栏颜色变成了指定的@color/colorRed



注意：

如果想通过在style中设置parent theme直接去掉ActionBar

<style

name="ImageTranslucentTheme" parent="Theme.AppCompat.Light.DarkActionBar">

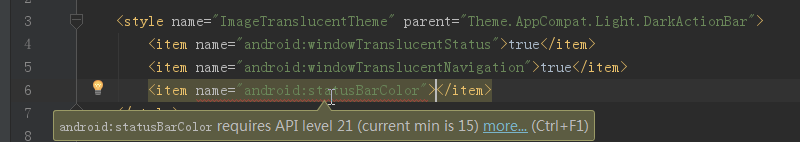
将parent改成

parent="Theme.AppCompat.NoActionBar"

则变成如下效果：



通过statusBarColor设置状态栏颜色只能在API>=21，即5.0的系统上设置



备上运行按照要求，如果需要为指定的Activity设置状态栏颜色，则需要为其对应的布局文件添加fitSystemWindow=“true”属性，有点麻烦，可以通过添加如下代码解决：

在BaseActivity中添加代码：

@Override  
protected void onCreate(@Nullable Bundle savedInstanceState) {  
  
 super.onCreate(savedInstanceState);  
 supportRequestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  
 setContentView(getLayoutId());  
  
 ViewGroup contentFrameLayout = (ViewGroup) findViewById(Window.*ID\_ANDROID\_CONTENT*);  
 View parentView = contentFrameLayout.getChildAt(0);  
 if(parentView != null && Build.VERSION.*SDK\_INT* >= 14){  
 parentView.setFitsSystemWindows(true);  
 }  
}  
  
protected abstract int getLayoutId();

即拿到内容布局，然后通过代码将其fitsSystemWindows属性设置为true

在BaseActivity中定义了一个getLayoutId方法，子类去实现，目的是不同子类Activity设置不同的layout布局文件，

因此，之后如果需要为某个Activity设置状态栏颜色，就不需要在其对应的布局文件中添加fitsSystemWindows属性了！