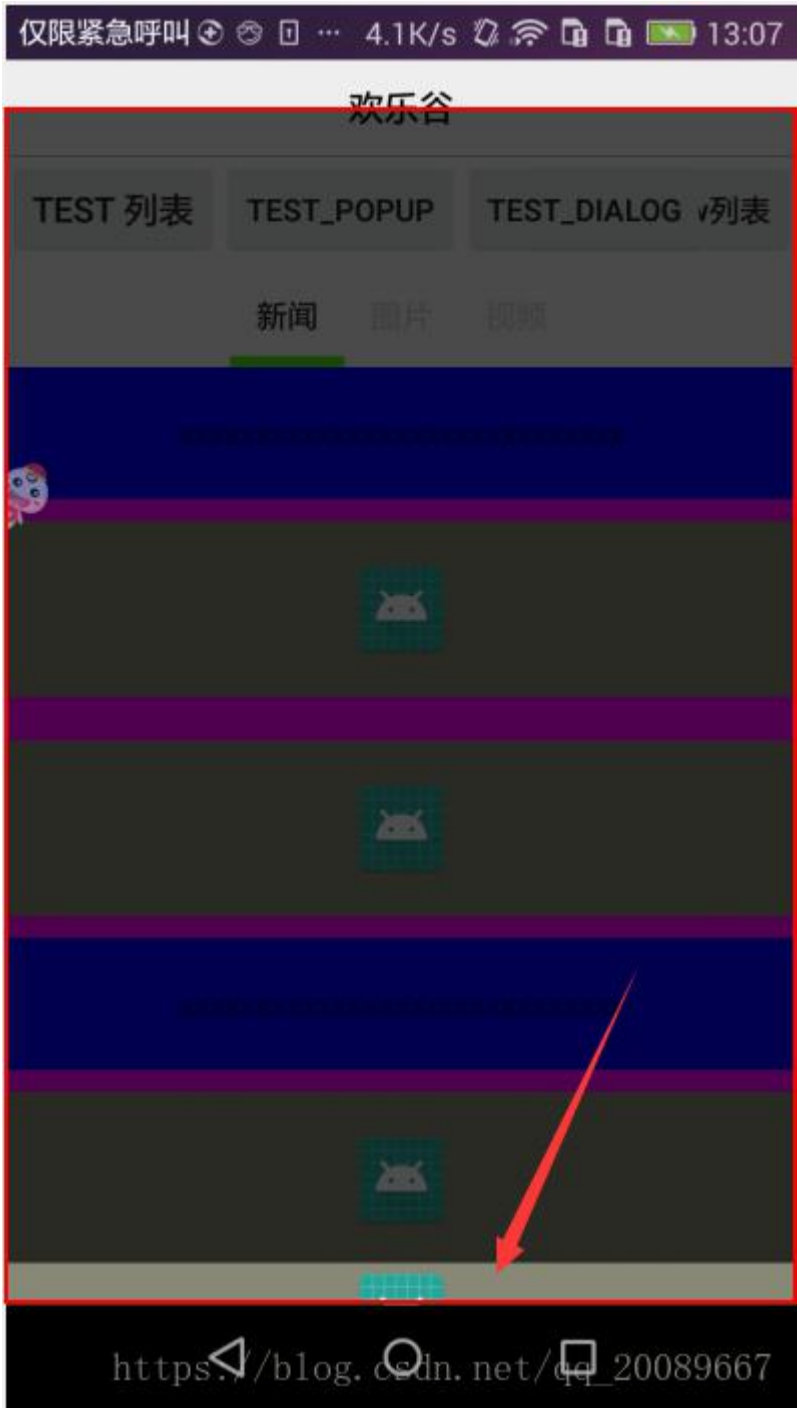
关于 popupWindow 底部与导航栏（ navigation bar）重叠，显示不全的问题分析

最近在做项目遇到个问题：自定义 popupWindow,调用

```
public void showAtBottom(View parent) {
       View view = mPopupLayout.findViewById(getContentViewId());
       setAnimation(view);//自定义动画
       showAtLocation(parent, Gravity.BOTTOM |
Gravity.CENTER_HORIZONTAL, 0, 0);
    }
```

结果：



这个是测试 demo 里的效果。

看表面现象应该是系统的 navigation bar 遮挡住了 popupWindow 布局，怀着这个心思，自然想到 popupWindow 的 setContentView 设置布局方法。

以下是源码部分：

```
public void setContentView(View contentView) {
       if (isShowing()) {
          return;
```

```
        }

        mContentView = contentView;

        if (mContext == null && mContentView != null) {
            mContext = mContentView.getContext();
        }

        if (mWindowManager == null && mContentView != null) {
            mWindowManager = (WindowManager)
mContext.getSystemService(Context.WINDOW_SERVICE);
        }

        // Setting the default for attachedInDecor based on SDK version
here
        // instead of in the constructor since we might not have the context
        // object in the constructor. We only want to set default here if
the
        // app hasn't already set the attachedInDecor.
        if (mContext != null && !mAttachedInDecorSet) {//这个是会走的，源
码就不分析了
            // Attach popup window in decor frame of parent window by default
for
            // {@link Build.VERSION_CODES.LOLLIPOP_MR1} or greater. Keep
current
            // behavior of not attaching to decor frame for older SDKs.

setAttachedInDecor(mContext.getApplicationInfo().targetSdkVersion
                    >= Build.VERSION_CODES.LOLLIPOP_MR1);
        }

    }
```

定位到最后一行：

```
setAttachedInDecor(mContext.getApplicationInfo().targetSdkVersion
                    >= Build.VERSION_CODES.LOLLIPOP_MR1);//api22
```

sdk 做了下 api 适配：

```
/**
     * <p>This will attach the popup window to the decor frame of the parent
window to avoid
```

```
    * overlaping with screen decorations like the navigation bar.
Overrides the default behavior of
    * the flag {@link
WindowManager.LayoutParams#FLAG_LAYOUT_ATTACHED_IN_DECOR}.
    *
    * <p>By default the flag is set on SDK version {@link
Build.VERSION_CODES#LOLLIPOP_MR1} or
    * greater and cleared on lesser SDK versions.
    *
    * @param enabled true if the popup should be attached to the decor
frame of its parent window.
    *
    * @see WindowManager.LayoutParams#FLAG_LAYOUT_ATTACHED_IN_DECOR
    */
   public void setAttachedInDecor(boolean enabled) {
      mAttachedInDecor = enabled;
      mAttachedInDecorSet = true;
   }
```

注释的大概意思是：这将把弹出窗口附加到其父窗体的装饰框（个人理解应该是类似于窗体的根布局 decorview 概念） 目的是避免 popupWindow 布局与屏幕装饰物重叠，比如导航栏。重写了默认的 windowManager.LayoutParams 为 FLAG_LAYOUT_ATTACHED_IN_DECOR

看下参数的含义：如果 enabled 为 true 那么就会避免与导航栏重叠

FLAG_LAYOUT_ATTACHED_IN_DECOR 又是什么鬼？

```
/**
       * Window flag: When requesting layout with an attached window,
the attached window may
       * overlap with the screen decorations of the parent window such
as the navigation bar. By
       * including this flag, the window manager will layout the attached
window within the decor
       * frame of the parent window such that it doesn't overlap with
screen decorations.
       */
      public static final int FLAG_LAYOUT_ATTACHED_IN_DECOR =
0x40000000;
```

大体意思和上面的翻译吻合。原来如此，现在已经明确了

原来系统通过

```
setAttachedInDecor
```

方法设置 windowManager.Layoutparams,如果 api>=22 就设置为 true，反之 false

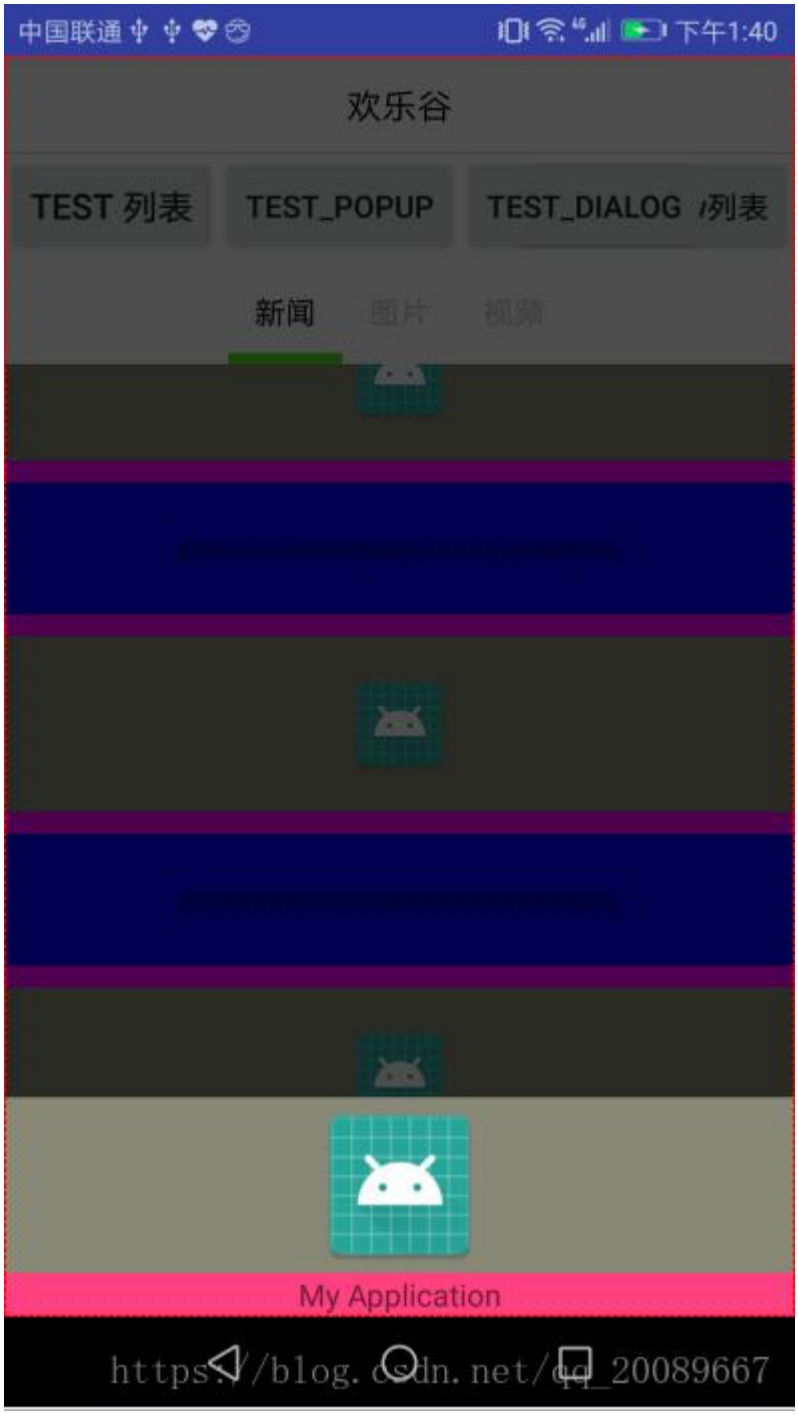
那 setAttachedInDecor 里的 mAttachedInDecor 字段在哪里生效的呢？继续看源码：

在 PopupWindow 里的 computeFlags 方法里：

```
if (mAttachedInDecor) {
          curFlags |=
WindowManager.LayoutParams.FLAG_LAYOUT_ATTACHED_IN_DECOR;
       }
```

与上面的翻译注释吻合。

下面是 api=24，华为荣耀 8 的结果（显示正常）：



现在的问题是我测试机 api21，所以方法设置为 false 了。

更令人可恶的是 setAttachedInDecor(boolean)方法只能 api22 以上调用

怎么办呢？

解决方案 1：

1.将 api<21 的设备，popupWindow 的布局上移导航栏的高度

代码：

```
@Override
    protected void setAnimation(final View view) {
        final int final_location;
        if (view.getContext().getApplicationInfo().targetSdkVersion
                < Build.VERSION_CODES.LOLLIPOP_MR1) {//api<22,这个地方就好
比给 sdk 里打了个补丁
            final_location =
-ScreenUtils.getNavBarHeight(view.getContext());
        } else {
            final_location = 0;
        }
        view.post(new Runnable() {
            @Override
            public void run() {
                int height = view.getHeight();
                ValueAnimator animator = ObjectAnimator
                        .ofFloat(view, "translationY",
height,final_location );//如果不需要动画，可以直接将布局上移这个高度
                animator.setDuration(200).start();
            }
        });
    }
```

获取导航栏高度：

```
public static int getNavBarHeight(Context context) {
        Resources resources = context.getResources();
        int resourceId = resources.getIdentifier("navigation_bar_height",
"dimen", "android");
        if (resourceId > 0) {
            return resources.getDimensionPixelSize(resourceId);
        }
        return 0;
    }
```

目前这种解决方案在真机都没问题，但是夜神模拟器比较特殊，夜神底下没有导航栏，但是获取的导航栏的高度却不是 0，难道他获取的导航栏在右侧？？

解决方案 2：

在 popupWindow 的布局里添加个高度>=0 的可滚动的列表（比如 ListView,RecyclerView,ScrollView），如果不需要此控件，高度可设置为 0

```
<?xml version="1.0" encoding="utf-8"?><RelativeLayout
    android:id="@+id/root"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    >

   <LinearLayout
       android:id="@+id/content"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:layout_alignParentBottom="true"
       android:background="#887"
       android:orientation="vertical"
       >

      <ImageView
           android:id="@+id/iv"
           android:layout_width="match_parent"
           android:layout_height="80dp"
           android:src="@mipmap/ic_launcher"
           />
      <!--解决 popupWindow 与导航栏重叠问题-->
      <ListView
           android:layout_width="match_parent"
           android:layout_height="0dp"
           />
      <TextView
           android:layout_width="match_parent"
           android:layout_height="20dp"
           android:gravity="center"
           android:background="@color/colorAccent"
           android:text="@string/app_name"/>
   </LinearLayout></RelativeLayout>
```

顺便贴下 basePopupWindow 的代码：

```
public abstract class BasePopupWindow<T> extends PopupWindow {
    private View mPopupLayout;
    private View.OnClickListener mListener;
```

```
    protected List<T> mBeans;

    public BasePopupWindow(Context context, View.OnClickListener
listener, List<T> beans) {
        super(context);
        initView(context);
        mBeans = beans;
        mListener = listener;
    }

    private void initView(Context context) {
        mPopupLayout = ((LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE)).inflate(ge
tLayoutId(), null);
        setContentView(mPopupLayout);
        setWidth(ViewGroup.LayoutParams.MATCH_PARENT);
        setHeight(ViewGroup.LayoutParams.MATCH_PARENT);
        setFocusable(true);
        ColorDrawable dw = new ColorDrawable(0xb0000000);
        setBackgroundDrawable(dw);
        mPopupLayout.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View v, final MotionEvent event) {
                mPopupLayout.findViewById(getContentViewId()).post(new
Runnable() {
                    @Override
                    public void run() {
                        int height =
mPopupLayout.findViewById(getContentViewId()).getTop();
                        int y = (int) event.getY();
                        if (event.getAction() == MotionEvent.ACTION_UP) {
                            if (y < height) {
                                dismiss();
                            }
                        }
                    }
                });

                return true;
            }
        });
        setData();
    }
```

```
    private View getView(int viewId) {
        return mPopupLayout.findViewById(viewId);
    }

    /**
     * 如果有必要-子类根据 model 返回的数据写逻辑
     *
     * @return
     */
    protected BasePopupWindow setData() {
        return this;
    }

    public BasePopupWindow setListener(int viewId) {
        getView(viewId).setOnClickListener(mListener);
        return this;
    }

    /**
     * 位置-底部
     *
     * @param parent
     */
    public void showAtBottom(View parent) {
        View view = mPopupLayout.findViewById(getContentViewId());
        setAnimation(view);
        showAtLocation(parent, Gravity.BOTTOM |
Gravity.CENTER_HORIZONTAL, 0, 0);
    }

    protected void setAnimation(View target) {

    }

    /**
     * 点击不消失的 content view
     *
     * @return
     */
    protected abstract int getContentViewId();

    protected abstract int getLayoutId();
```

动画写在继承他的子类里。

我的 csdn 地址：https://blog.csdn.net/qq_20089667/article/details/80192884