

CAN301 Group Project 项目过程记录（部分）

- Error 1 - 11月24日晚 23:45

位置：src/main/java/com/example/androidapplicationtest/settingActivity.java

原因：NameNotFoundException

状态：☑ 于11月26日凌晨1:45

任务是尝试设计一个简单的设置界面以控制系统屏幕亮度和音量，音量和自动亮度的控制控件成功实现了，但是在尝试实现亮度调节的时候出现了问题。尝试对 Settings 页面的 亮度控制滑块 应用 SeekBar.setOnSeekBarChangeListener 时，主页面虽然成功显示，但是在成功安装后报错：

```
E/InterceptorFactory: NameNotFoundException
android.content.pm.PackageManager$NameNotFoundException:
com.miui.contentcatcher
    at
    android.app.ApplicationPackageManager.getPackageInfoAsUser(ApplicationPackag
eManager.java:174)
    at
    android.app.ApplicationPackageManager.getPackageInfo(ApplicationPackageManag
er.java:145)
    at
    miui.contentcatcher.InterceptorFactory.getInterceptorPackageInfo(Interceptor
Factory.java:80)
    at
    miui.contentcatcher.InterceptorFactory.initInterceptorClass(InterceptorFacto
ry.java:99)
    at
    miui.contentcatcher.InterceptorFactory.createInterceptor(InterceptorFactory.
java:41)
    at
    miui.contentcatcher.InterceptorProxy$H.handleMessage(InterceptorProxy.java:3
13)
        at android.os.Handler.dispatchMessage(Handler.java:106)
        at android.os.Looper.loop(Looper.java:164)
        at android.os.HandlerThread.run(HandlerThread.java:65)
```

并且在每次尝试进入 Settings 页面时，程序崩溃并显示如下错误：

```
E/AndroidRuntime: FATAL EXCEPTION: main
Process: com.example.androidapplicationtest, PID: 11744
java.lang.RuntimeException: Unable to start activity
ComponentInfo{com.example.androidapplicationtest/com.example.androidapplicat
iontest.settingActivity}: java.lang.NullPointerException: Attempt to invoke
virtual method 'void
android.widget.SeekBar.setOnSeekBarChangeListener(android.widget.SeekBar$OnS
eekBarChangeListener)' on a null object reference
    at
    android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2816)
```

```

    at
    android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2894)
    at android.app.ActivityThread.-wrap11(Unknown Source:0)
    at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1615)
    at android.os.Handler.dispatchMessage(Handler.java:106)
    at android.os.Looper.loop(Looper.java:164)
    at android.app.ActivityThread.main(ActivityThread.java:6637)
    at java.lang.reflect.Method.invoke(Native Method)
    at
    com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java
    :518)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:822)
    Caused by: java.lang.NullPointerException: Attempt to invoke virtual method
    'void
    android.widget.SeekBar.setOnSeekBarChangeListener(android.widget.SeekBar$OnS
    eekBarChangeListener)' on a null object reference
    at
    com.example.androidapplicationtest.settingActivity.globalBrightSeekBarManage
    r(settingActivity.java:69)
    at
    com.example.androidapplicationtest.settingActivity.onCreate(settingActivity.
    java:38)
    at android.app.Activity.performCreate(Activity.java:7094)
    at android.app.Activity.performCreate(Activity.java:7085)
    at
    android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1215)
    at
    android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2769)
    at
    android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2894)
    at android.app.ActivityThread.-wrap11(Unknown Source:0)
    at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1615)
    at android.os.Handler.dispatchMessage(Handler.java:106)
    at android.os.Looper.loop(Looper.java:164)
    at android.app.ActivityThread.main(ActivityThread.java:6637)
    at java.lang.reflect.Method.invoke(Native Method)
    at
    com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java
    :518)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:822)

```

观察“com.miui.contentcatcher”错误信息，猜测可能和小米平板系统屏幕亮度控制、或者和系统设置 API 的应用不正确有关：

```

Settings.System.getInt(getContentResolver(),
Settings.System.SCREEN_BRIGHTNESS);

```

但观察“on a null object reference”错误信息，结合

<https://www.cnblogs.com/edensyd/p/8854965.html>，猜测可能是在使用 findViewById 引用 View 时出现错误，

```
private void globalBrightSeekBarManager() {
    globalBrightSeekBar = (SeekBar) findViewById(R.id.);
    globalBrightSeekBar.setMax(255);
    int currentScreenBrightness = 0;
    global_volume_Seekbar ( = 1000103)
    global_bright_Seekbar ( = 1000088)
```

11月26日凌晨 1:45 - 后来发现其实只是因为我在复制代码块的时候忘记改变量名了而已.....
草!!!

```
// Error 1 - 11月24日晚 23:45
globalVolumeSeekBar.setOnSeek
```

(此处应该是 globalBrightSeekBar)

- Suggestion 1 - 11月24日晚

关于上文中和屏幕亮度调节有关的问题里，有部分资料提到说，可以选择性控制本窗口以及系统全局亮度的选项。但是本项目并不打算对此进行进一步的处理。

- Propose 1 - 11月25日

内容：主页面取消掉 ActionBar，这鬼玩意儿太丑了

状态： ☒

就是安卓系统默认设计界面顶部的那个 bar。

- Propose 2 - 11月25日

内容：ActionBar / 主题颜色更改

状态： ☒ 于12月4日19:18

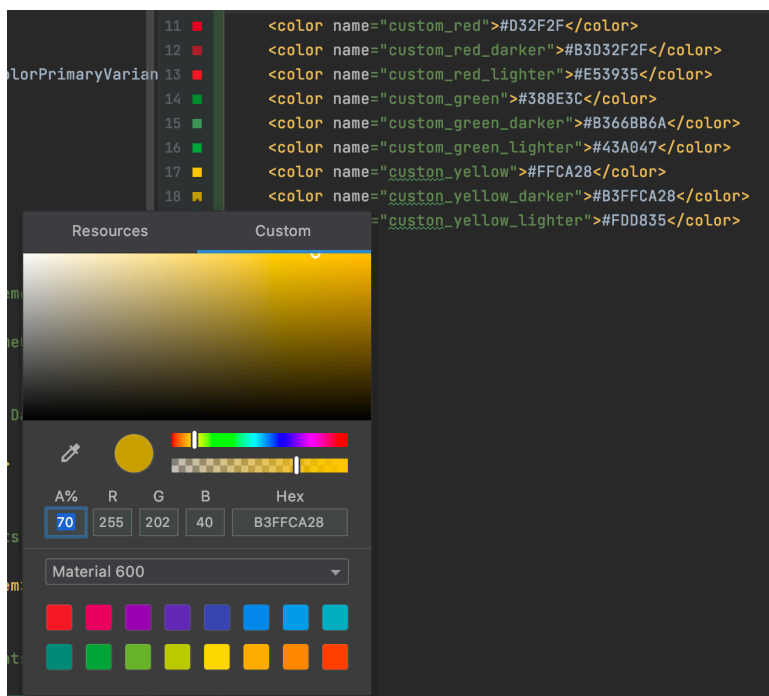
之前看 fireship.io 看到的例子，说是能不能更改 app 顶部的那个 bar 的颜色。

其实整个 app 的各个主要部件的配色是可以在 src/res/values/colors.xml (定义配色名称和对应的 HTML FF颜色值) 和 src/res/values/themes.xml 里改的。

(其实还有很多其他 Issues 还没搞但是我却很想做这玩意儿.....)

1. 自定义主题内容

xxxcolor_darker 其实就是 A% 降低 30%，xxxcolor_lighter 就是将颜色集从 Material 700 改为 600。



在 themes.xml 中添加以下自定义 attributes:

```

<!-- Customized Color Theme -->
<style name="CustomColorTheme_red"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <item name="colorPrimary">@color/custom_red</item>
    <item name="colorPrimaryVariant">@color/custom_red_darker</item>
    <item name="colorOnPrimary">@color/white</item>
</style>
<style name="CustomColorTheme_green"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <item name="colorPrimary">@color/custom_green</item>
    <item name="colorPrimaryVariant">@color/custom_green_darker</item>
    <item name="colorOnPrimary">@color/white</item>
</style>
<style name="CustomColorTheme_yellow"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <item name="colorPrimary">@color/custon_yellow</item>
    <item name="colorPrimaryVariant">@color/custon_yellow_darker</item>
    <item name="colorOnPrimary">@color/white</item>
</style>

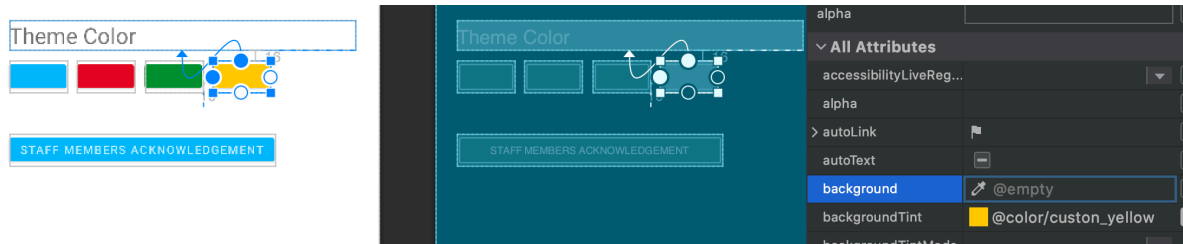
```

2. Button 组件只能更死当前设定的主题颜色?

这里有个问题是, 对于 Button 以及其他安卓提供的组件来说, 其似乎只能跟死当前设定的主题颜色, 不能给每个按钮单独设定颜色。网上也有遇到类似想要单独给按钮设置颜色但是失败只能去改 themes.xml 的情况, 但是我这个例子中就是为了要能够提供其他 themes 的颜色选择的。

不过倒是找到了一个可能不是正确解决方案的方案, 就是使用 backgroundTint 属性。查其

他文章说这个“tint”应该是类似于对于某个控件的再上色功能，猜的没错的话其概念和画画软件里的颜色覆盖模式差不多。



网上提供的解决方案是：将 backgroundTint 设置为 @null，然后设置 background 为其他颜色。我后来也这么做了。

3. 如何切换主题颜色

网上提供的解决方案都是：

```
// 在 onCreate 中
setTheme -> 如需更改主题，则 recreate 强制重绘，强制 onCreate 重来一遍
-> setTheme -> setContentView
```

虽然说这个很简单的 app 并不会出现网上说的“强制重绘会导致闪屏”的情况，但是问题在于这几个方法的应用顺序是有要求的，也就是说，我需要一个外部变量用于保存当前所需要选择使用的主题是哪个，然后再在 recreate 之后 setTheme。

我尝试用一个 private int 来保存当前所选择使用的主题，然后意识到说 private int 不可改，于是改为 public int，然后尝试用一个 switch-case 来识别：

```
public int themeColorFlag;
// Default 1 = Blue, 2 = Red, 3 = Green, 4 = Yellow
```

然后虽然按下那四个主题颜色选择按钮的时候的确都会重绘当前页面（因为我临时写的一个用于 debug 的每次新进入 Settings Activity 都会跳出来的 Toast 会跳出来），但是主题颜色并没见变化，而且那个 themeColorFlag 也没见变。

思考说这个变量毕竟是被存在这个 settingActivity 类里，于是我忽然想到了之前老师提过但是我以为这个应用不会用到的 SharedPreferences。

4. 尝试使用 SharedPreferences 记录当前 Activity 使用的 theme

SharedPreferences 是 Android 系统提供的一个通用的轻量级数据持久化框架，用于在内部存储中存储和读取 key-value 类型的原始基本数据对。目前仅支持 boolean、float、int、long 和 string 等基本类型的存储，对于自定义的复合数据类型，是无法使用 SharedPreferences 进行存储的。

哦草。

遂在研究了接近半小时的 SharedPreferences 并且把那个最左边的蓝色按钮也单独设置了颜色（免得在更改主题色之后颜色也跟着变了）之后，成功倒是成功了，但是……

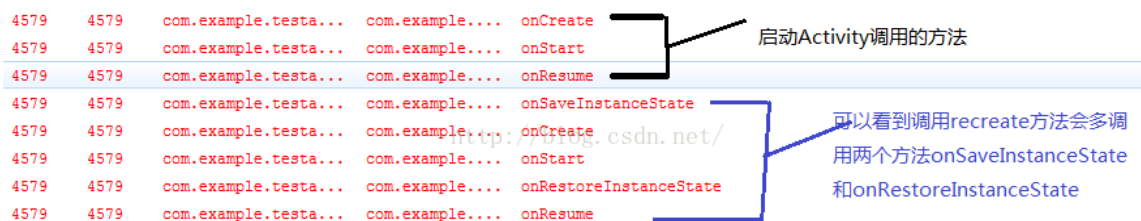
5. 如何使得前一个 Activity 也应用 theme 的变化

我从 settingActivity 退出去之后发现丫我改的其实还是只是当前这个 Activity 的 theme 而已.....因此我似乎还是得在 MainActivity 里加一个判断当前使用的 Theme 是什么的部分。但是在12月4日下午尝试实现此功能时又发现了问题：因为 MainActivity 活动页面栈是在 settingActivity 之前的，在 settingActivity 设置了 theme 之后无法把变化应用到前面的 MainActivity 中。而且虽然 MainActivity 在创建页面的时候已经读过一遍存在 SharedPreferences 里的主题样式了，但是它读的是 settingActivity 修改过的那个。刚开始我不是很想解决这个问题，于是我直接打算选择让用户在选择完 Theme 之后直接 reload required，代价就是我没法向用户显示非常炫酷的点一下即刷新颜色主题的功能了。但是我仍旧不甘心，于是去搜了一下“该如何刷新前一个 Activity”，结果，就和本 Propose 的第四条过程一样，我又回到了老师在课上所讲过的知识：

6. 尝试使用 startActivityForResult & onActivityResult

想法是使用 startActivityForResult 在 MainActivity 开启 settingActivity，然后 settingActivity 返回的 resultCode 如果是1，则 MainActivity 需要 recreate 以使用其他指定的 theme 并重绘。

这里有一个个人顾虑：因为在 settingActivity 中我使用了 recreate 方法以刷新当前 Activity，那么 recreate 后我尝试从 settingActivity 传给 MainActivity 的信息还在吗？后来参考到这篇文章 https://blog.csdn.net/zhuhai_yizhi/article/details/47395357 说是 recreate 方法似乎是会在再次重绘此 Activity 的时候 call 一系列用于保存本 Activity 状态的方法，如 onSaveInstanceState 等，那这样的话应该就没问题了。



4579	4579	com.example.testa...	com.example....	onCreate
4579	4579	com.example.testa...	com.example....	onStart
4579	4579	com.example.testa...	com.example....	onResume
4579	4579	com.example.testa...	com.example....	onSaveInstanceState
4579	4579	com.example.testa...	com.example....	onCreate
4579	4579	com.example.testa...	com.example....	onStart
4579	4579	com.example.testa...	com.example....	onRestoreInstanceState
4579	4579	com.example.testa...	com.example....	onResume

启动Activity调用的方法

可以看到调用recreate方法会多调用两个方法onSaveInstanceState和onRestoreInstanceState

但是在尝试应用 startActivityForResult 的时候发现，老师以及网上给的例子都说是，在负责返回信息给前一个 Activity 的那个 Activity (“OtherActivity”) 里，设置好 setData 之后就直接 finish 了：

```
Intent i = new Intent(settingActivity.this, MainActivity.class);
setResult(1, i);
finish();
```

似乎是 onActivityResult 只会在 otherActivity 被 finish 之后才会获得从 otherActivity 返回的参数。

也有尝试在从 MainActivity 跳到 settingActivity 之后 finish 掉 MainActivity，寻思这样会不会在从 settingActivity（以系统返回键的方式）回到 MainActivity 之后后者会被重绘，但是实操后发现并不会，这么干之后在 settingActivity 应用返回键就直接跳回桌面了。

后来查到说可以在 settingActivity 中实现 onBackPressed 方法，在这个方法里加入 finish 方法以实现在使用系统返回键回到 MainActivity 的时候结束掉 settingActivity 的目的，然后也成功了。

但是紧接着又出现了新的问题：

7. 似乎从 MainActivity 回到 settingActivity 的时候，在 settingActivity 设置的 SharedPreferences 没被 MainActivity 得到.....

你妈的为什么!!!

不过在后来重新看了一下 SharedPreferences 的笔记之后发现丫的我在之前第四点的时候用的是另一个只在当前 Activity 保存的 SharedPreferences:

```
Activity.getPreferences(int mode)
    Retrieve a SharedPreferences object for accessing preferences that are
    private to this Activity.

// 应该用的是这个:
Context.getSharedPreferences(String name, int mode)
    Retrieve a SharedPreferences object for accessing preferences of whole
    system.
```

直接在 MainActivity 和 settingActivity 改了一下之后就没事了，主页会跳出来的弹窗的字体也变成了之前选择的 theme 的颜色。yay。

8. 继续修缮: 关于 NoActionBar

以及顺便注意一下关于 theme.xml 中定义主题的时候，因为计划在 settingActivity 里带 ActionBar 而主页面没有，因此还基于昨天写的 theme 样式各复制了一份带 ActionBar 的版本并选择在 settingActivity 里应用带 ActionBar 的 theme。

9. 继续修缮: 自定义的 theme 中对于状态栏颜色的设置

忘记加这一行了，现在加上:

```
<!-- Status bar color. -->
<!-- <item name="android:statusBarColor" tools:targetApi="l">?
attr/colorPrimaryVariant</item> -->
<!-- 说实话不知道这个 ?attr/ 是干什么的不过反正我直接改用 @color 了 -->
<item name="android:statusBarColor">@color/color_primary_blue_darker</item>
```

10. 删掉 MainActivity 的 debug message，彻底完成这个 Propose

```
// Debug Msg (Delete in final product)
String debugToastMsg = " * Debug Msg *" +
    "\nthemeColorMode: " +
    themeColorSharedPreferencesManager.getInt("currentThemeColorMode", 1) +
    "\nonActivityResult received resultCode: " + resultCode;
Toast t = Toast.makeText(MainActivity.this, debugToastMsg,
    Toast.LENGTH_SHORT);
t.setGravity(Gravity.FILL_HORIZONTAL, 0, 0);
t.show();
```


11. Evaluation: 仍旧没有“持续性地更改主题”

虽然做到了从 `settingActivity` 跳回 `mainActivity` 时可以识别并切换主题，但是从 `mainActivity` 进入 `selectGameActivity` 的时候仍旧是默认主题，思考说应该 `setTheme` 并没有持久地更改系统主题（因为如果真的持续性更改了的话我觉得也不会有上文这么多bug了），目前只能选择在栈之后的所有 `Activity` 里都加入几行和 `mainActivity` 一样的识别 `theme` 的方法了。

不过我这次就不打算继续研究下去具体的解决方案了。

- Propose 3 - 11月25日

内容: Staff List

状态: ☒

12月3日周五出来开会的时候合了张影。

- Suggestion 2 - 12月3日晚

我觉得用印象笔记来做这样的项目状况跟踪还是没有上学年学的 DevOps 来的方便，果然姜还是老的辣。

不过我暂时不打算改了，而且反正也只是给我自己看的而已。

- Issue 1 - 12月2日下午 16:50

位置: GitHub

原因: GitHub's "master" issue

状态: ☒ 于12月2日下午 17:00

在昨（前）天完成了 INT305 的 Assessment 2 之后转头继续开搞 CAN301 了。

因为要研究一下该如何将其他组员写的 `Activity` 合并过来，想着说能不能顺便研究一下 Git，然后还成功地创建了人生中第一个 `branch`。

但是在 GitHub 的主页面我一直刷新却一直都没看到我刚才 `commit` 的内容，后发现主页默认显示的 `branch` 的名字叫“`main`”而不是“`master`”，后来去查了一下.....

而最近 GitHub 官方发布信息称，从 2020 年 10 月 1 日起，在 Github 平台上创建的所有新的源代码仓库将默认命名为 "main"，不再是原先的 "master"。

On October 1, 2020: newly-created repositories will default to main

On October 1, 2020, any new repositories you create will use `main` as the default branch, instead of `master`. This change **does not impact any of your existing repositories**: existing repositories will continue to have the same default branch they have now.

You can **opt out of this change** at any time. To opt out, set the default branch for new repositories for your user, organization, or enterprise:

1. For **users**, on the <https://github.com/settings/repositories> page
2. For **organization owners**, on the <https://github.com/organizations/YOUR-ORGANIZATION/settings/repository-defaults> page
3. For **enterprise administrators**, on the https://github.com/enterprises/YOUR-ENTERPRISE/settings/member_privileges page

但有一点，是需要注意的，现有的存储库不会受到此更改的影响。

就目前来说，这是一个比较折中的方法了，对旧项目的影响也没那么大。

那为什么突然会把 "master" 将更改为 "main" 呢？

在 6 月份，受美国大规模的 "Black Lives Matter" 运动影响，很多人认为 "master" 这个词，有不好的隐喻，有种族歧视色彩。

"master" 这个词在科技圈运用的还是挺多的，但为了平息舆论，很多科技巨头都宣布了调整。

比如：

MySQL 宣布删除 master、黑名单白名单等术语。

Twitter、微软、OpenSSL、Android 移动操作系统、Go 编程语言等都宣布要对此类术语进行删除或更改。

在 6 月 8 日，知名的技术博主 Scott Hanselman 也发文呼吁将 Git 的默认分支从

反正后来也没管太多，继续我的 "master" 去了。

我只想知道各位腐狗玩家会怎么想。

- Issue 2 - 12月2日下午 17:23

位置：src/main/java/com/example/androidapplicationtest/settingActivity.java

原因：merge issues

状态：null

尝试合并 guessGame 老哥发来的代码.....果然还是要面临 merge hell 的。

首先问题是关于代码合并的问题，结合安卓 app 的代码结构，大概有以下内容需要合并：

1. AndroidManifest.xml 中定义的 Activities

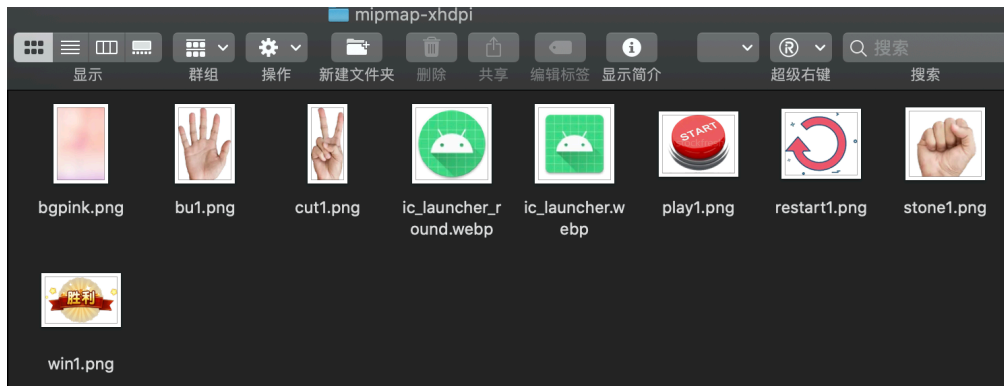
Manifest.xml 注释胆敢给我用的 double slash，没有 private 任何 onCreate 之外的变量，多变量定义的时候甚至给我直接一整行不带空格的地域，气得我当场在宿舍摔了眼镜结果把鼻梁架给摔断一个。

2. 各 Activity Class 的名字, 以及各 Activity Class 中选择显示的页面

```
setContentView(R.layout.activity_main);  
// 各自负责写的 Activity
```

3. res/drawable

如果是我的话我大概会选择用一些更好看的图片.....



4. 非代码要求

他妈的不要给我发 Doc 文档啊操!

我算是非常能理解为什么上次看 GitHub 的那个《提问的艺术》说是不要用 docx 了, 我都不强求用 Markdown 了你好歹发个 txt 或者直接复制粘贴到微信里发给我都可以啊.....

- Error 2 - 12月3日下午 17:19

位置: src/main/java/com/example/androidapplicationtest/settingActivity.java

原因: 动态权限请求失败

状态: ☒ 于12月4日20:10

原本一切正常的设置在刚才和组员见面尝试演示的时候突然失败。滚动亮度滑块又一次闪退。IDE 报错:

```
E/AndroidRuntime: FATAL EXCEPTION: main  
    Process: com.example.androidapplicationtest, PID: 5152  
    java.lang.SecurityException: com.example.androidapplicationtest was not  
    granted this permission: android.permission.WRITE_SETTINGS.  
        at android.os.Parcel.readException(Parcel.java:2005)  
        at  
    android.database.DatabaseUtils.readExceptionFromParcel(DatabaseUtils.java:18  
    3)  
        at  
    android.database.DatabaseUtils.readExceptionFromParcel(DatabaseUtils.java:13  
    5)  
        at
```

```

android.content.ContentProviderProxy.call(ContentProviderNative.java:651)
    at
android.provider.Settings$NameValueCache.putStringForUser(Settings.java:1857)
    at
android.provider.Settings$System.putStringForUser(Settings.java:2308)
    at
android.provider.Settings$System.putIntForUser(Settings.java:2413)
    at android.provider.Settings$System.putInt(Settings.java:2407)
    at
com.example.androidapplicationtest.settingActivity$1.onProgressChanged(settingActivity.java:84)

```

看情况又是和权限有关的问题，打开 mi Pad 设置页面查看发现果然如此。但是明明之前已经在 Activity 里应用过了相关的权限请求的。

不过后来想来说应该是之前在测试的时候已经手动去设置页面里设置了权限。而且引用之前在网上看到的：

在 Android 6.0（平台23）及以后，WRITE_SETTINGS 权限的保护等级已经由原来的 *dangerous* 升级为 *signature*，这意味着我们的 APP 需要用系统签名或者成为系统预装软件才能够申请此权限.....

也就是说这个 android.permission.WRITE_SETTINGS 必须得用户手动跳到设置里改才行了。

之前的方案如下：

```

// mainAction 中:
String[] permissionStrings = new String[]{
    // Manifest.permission.WRITE_EXTERNAL_STORAGE,
    Manifest.permission.WRITE_SETTINGS,
    // Manifest.permission.ACCESS_NOTIFICATION_POLICY
};
for (int i = 0; i < 3; i++) {
    if (ContextCompat.checkSelfPermission(this, permissionStrings[i]) !=
        PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]{
            permissionStrings[i]
        }, 1);
    }
}

// ActivityCompat.requestPermissions(this, permissionStrings, 1);

```

现在加了个弹窗机制，当判断到 app 没有相关权限的话，跳出弹窗要求去手动赋予权限：

```

/**
 * Grant Permissions WRITE_SETTINGS using a Alert Window
 * this project will only run on SDK > 23 (Android 8+)
 * so implementation for eariler Android build is ignored :P
 */
private void grantingPermissions() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);

```

```

        builder.setTitle("System Permission Required!");
        builder.setMessage("This app may require system permissions to change screen brightness and system volumes." +
            "\n\nClick \"Go to Settings\" to enable related permissions.");
        builder.setPositiveButton("Go to Settings", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Intent intent = new Intent(Settings.ACTION_MANAGE_WRITE_SETTINGS);
                intent.setData(Uri.parse("package:" + getPackageName()));
                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivityForResult(intent, 0);
            }
        });
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                finish();
            }
        });

        builder.setCancelable(true);
        AlertDialog dialog = builder.create();
        dialog.show();
    }
}

```

System Permission Required!

This app may require system permissions to change screen brightness and system volumes.

Click "Go to Settings" to enable related permissions.

CANCEL GO TO SETTINGS

我自己都觉得我自己比较狗的一点是，在设置这个弹窗的时候我直接说如果你不给我权限的话我就直接 finish 掉当前 Activity，也就是直接不给你用.....

但是之后本以为已经结束了，但是发现好像每次重新进入一次 app 都会跳出这个弹窗，即使说在设置里已经把相关选项打开了。看来还是在“判定是否已经获取此权限”上出了问题。后来发现之前写成：

```

if (ContextCompat.checkSelfPermission(this,
    Manifest.permission.WRITE_SETTINGS) == PackageManager.PERMISSION_DENIED)

```

其实应该是：

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
    // IDE 自动要求一定要加判断版本的部分  
    if(!Settings.System.canWrite(this))  
        grantingPermissions();  
}
```

- Porpose 4 - 12月3日周五下午17:20会议

内容：局域网内多设备联机打游戏。

状态：Deprecated

可以考虑基于 Java socket 进行信息交流，但是因为之前在 CAN301 上学到的是 Python 的 Socket Programming，而且估计这几天的时间已经不太够做这个了，因此还是算了。

- Propose 5 - 12月3日周五下午17:20会议

内容：一个类似于“小决定” iOS app 内某个功能的多点触控游戏

状态：Deprecated

虽然说老师有过一节和这个有关的 lab，但是我偷懒没做.....似乎是得从头开始研究。就其实我个人不是很想做这个东西.....因为我还是很不幸地发现说自己的工作其实就已经是这个组里最多了的，讲实话这个看似简单的 mainMenu 以及 setttingActivity 着实废了我不少脑细胞（虽然写起来也很好玩），但是原本就分配任务说四个人中有三个人每个人各做一个游戏然后我负责做主菜单，结果其中一位觉得说多个这个“稍微更复杂一点点的游戏有利于加分”，于是就很不客气地把这个新游戏的任务丢给了我。后来又说暂缓，因此就先暂缓吧。

- Error 3 - 12月3日周五下午17:20

位置：src/main/java/com/example/androidapplicationtest/guessGameActivity.java

内容：guessGame 那位老哥在自己 Activity 里放的 *IntentService* 播放背景音乐即使在退出应用之后仍旧会继续响。

状态：NotMyIssue

我暂时不打算搞这个问题，我觉得这个问题应该他自己来搞。

- Error 4 - 12月3日周五下午17:20

位置：src/main/java/com/example/androidapplicationtest/guessGameActivity.java

内容：尝试将 guessGame 老哥写的代码和我这边的 mainActivity 合并之后发现游戏界面的元素根本没法点击。

状态：NotMyIssue

同上。

- Issue 3 - 12月4日凌晨0:25

位置：Git

内容：GitHub 新手常见失误

状态：☑

我后来花了接近一小时挨个排查了一遍昨天下午和晚上改过的所有代码并确认和最后一次在 pad 上跑的情况一样之后才算松了口气。

(不过其实是可以全部 redo 的)

```
git checkout -- <filename>
```

但是我仍旧没能搞懂到底发生了什么，以及我需要抽时间重新学习一下 git 的用法。

- Issue 4 - 12月6日晚22:00

位置：主要在 splashActivity

内容：改变 Splash 的背景颜色

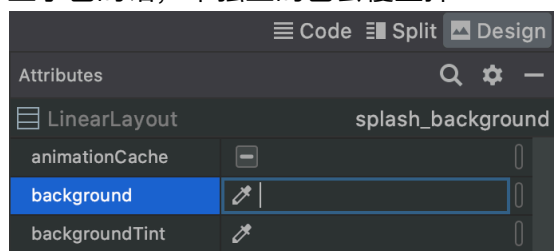
状态：☑ 于12月7日凌晨0:10

splashActivity 不会跟着改主题颜色，因此尝试对 *splashActivity* 如法炮制之前判断主题的方法，这次是尝试更改 *splashActivity* 的 *LinearLayout* 的背景颜色：

```
LinearLayout background = (LinearLayout)
findViewById(R.id.splash_background);
// background.setBackgroundColor(getResources().getColor(R.color.RED));
background.setBackgroundColor(Color.parseColor("#D32F2F"));
```

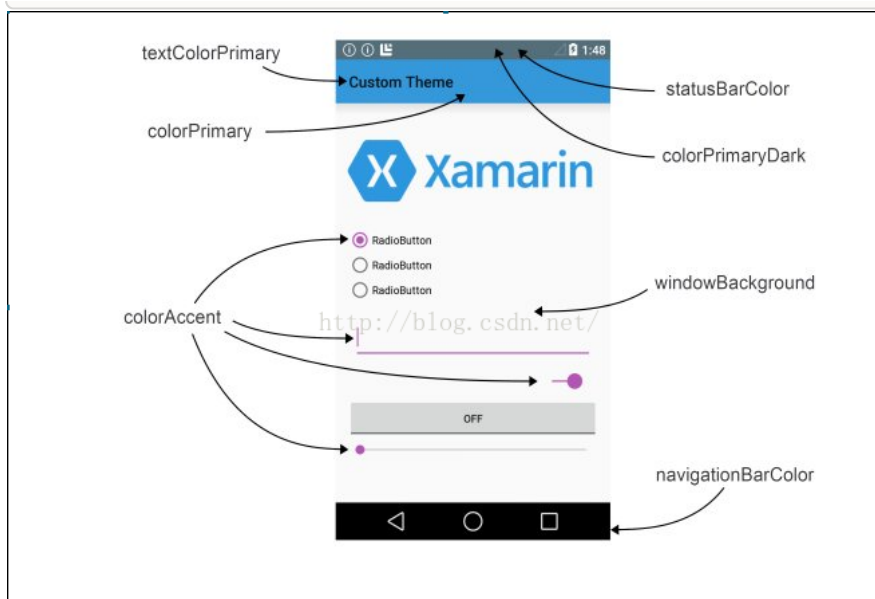
但是后来我发现我把这个判断 theme 的方法放在 *splashActivity* 的 *setContentView* 前面了，自然此时 *background.setBackgroundColor* 会找不到。

后来在回头查了一下笔记发现其实 theme 里是可以设置 *windowBackground* 的，于是直接去 theme.xml 里多加了 attributes（但是如果在 Design 页面里单独给 *windowBackground* 上了色的话，单独上的色会覆盖掉 theme 提供的 attributes）。



顺便注意到说好像 *ActionBar* 的字体颜色也需要另外的 attribute 来设置，否则如果没有这个另外的 attribute 的话会默认显示黑色，因此也加了相关的 attributes。

```
<!-- ActionBar text color. -->
<item name="android:textColorPrimary">@color/white</item>
<!-- Window background color.-->
<item name="android:windowBackground">@color/custom_green_lighter</item>
```



后来注意到说，在每个单独的 Action 的 View 的设计页面的每个单独组件是可以设置主题的，而当设置其 style 为某个自定义主题时：

```
<TextView
...
style="@style/Theme.primaryTheme"/>
```

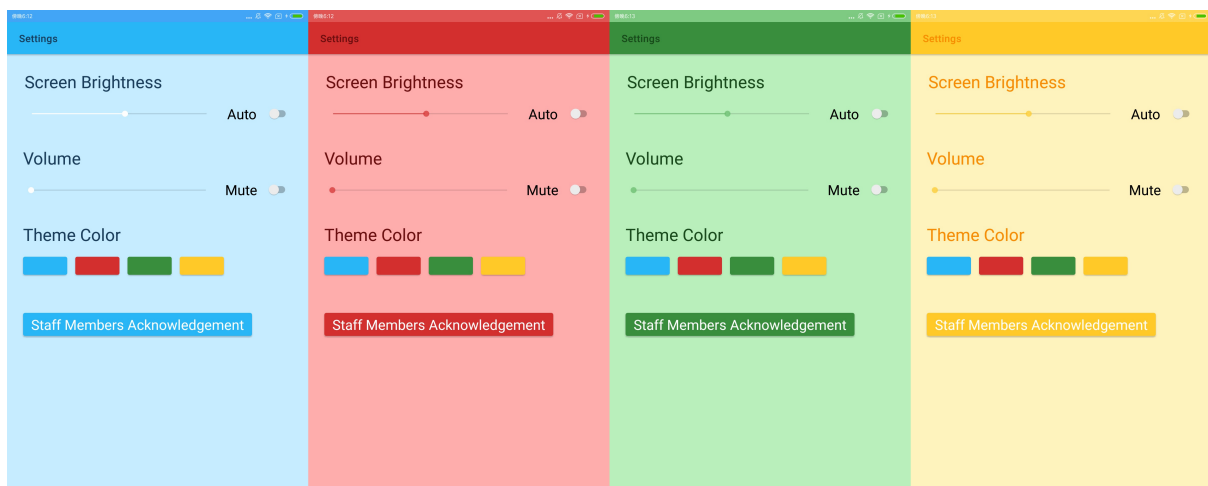
这个组件的字体颜色会跟着这个主题的那一系列和颜色有关的 attributes 走，对应关系大概如下：

如果被设置 style 的组件是 TextView，则这个 TextView 的颜色会跟着 style 的 textColorPrimary 属性走。

	Button
colorOnPrimary	SeekBar
colorSecondary	Switch
[unknown]	
more to follow...	

（之后可能需要去这篇文章 <https://blog.csdn.net/dong161114/article/details/88823305> 做进一步参考）

而且当我对这个 Activity 应用 setTheme 的时候，因为整个 Activity 的主题都变了，因此组件的主题也会跟着变，因此我给这个 app 里所有需要跟着主题一起变色的所有组件都加上了 style attributes，并且给这四个自定义颜色都给另外创建了一堆的衍生颜色（darker、darkest、lighter）搞得 theme.xml 文件行数几乎翻了一倍后得到了大概这样的成果：



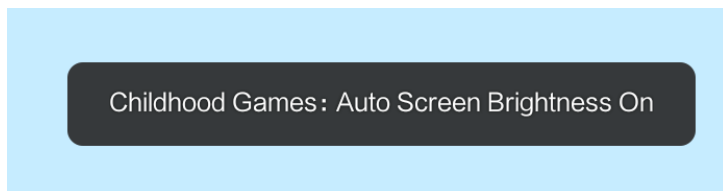
- Issue 5 - 12月6日晚 23:57

位置: settingActivity.java

内容: Toast Msg 在 MIUI 上的 bug

状态: ☒ 于12月7日凌晨 0:05

部分厂商系统的 Toast 由于某种原因会在 Toast 名字前面出现本 Activity 的名字的情况:



应用名字的代码啊,

不对, 还有一个就是我build.gradle里的各种SDK版本太高问题?, 试试看, 一路从低到高测试了个遍, 也不是这个问题。最后只剩下手机了, 先来试试模拟器, 咦, 居然正常的, 在试试不是小米手机会怎么样, 咦, 也是正常的, 想起来了, 小米手机前段时间才更新到了最新系统MIUI9.7.12.28,所以确定是小米手机新系统问题, 这时候去小米官方论坛看看, 发现了一堆....

Toast弹出时带应用名, 看起来很不舒服

1 70

2017-12-31 11:36 - kanggh - BUG反馈

toast前缀应用名称, 给开发人员造成困...

3 162

查了一些文章给出的解决方案是相较于之前一行解决而言显得非常不优雅的: 通过 Toast.makeText 生成一个 Toast 对象、调用这个 Toast 对象的 setText 方法为其重新设置值。

```
// 正常情况
Toast.makeText(getApplicationContext(), "Muted", Toast.LENGTH_SHORT).show();
```

```
// 现在
Toast t = Toast.makeText(getApplicationContext(), "", Toast.LENGTH_SHORT);
t.setText("Message");
t.show();
```

- Issue 6 - 12月6日下午

位置：res/drawable

内容：为 App 添加程序图标

状态： ☒

根据网上的文章教程后来又去查了安卓官方文档之后发现自己用 Procreate 画的那鬼玩意儿一直没能出现在桌面上，后来发现我似乎好像忘记改 manifest 了，遂改之，顺便把原先提供的默认图标给随手删了，于是成功了。



其实还是有点那种 iOS App Store 以及 Google Play Store 那种骗广告费的免费傻逼小游戏的感觉的（哈）。

- Error 5 - 12月7日下午17:50

位置：项目根目录的 build 文件夹

内容：莫名其妙多出了几个多余的 class 文件，还被 Finder 自动在后面加了个序号2

状态： ☒

后来想到说这个 build 文件夹顾名思义应该是用于编译的，我直接把这一整个文件夹删了重新编译一遍不就得了.....

- Propose 6 - 反正是12月7日前的某天

位置：每个独立的 gameActivity

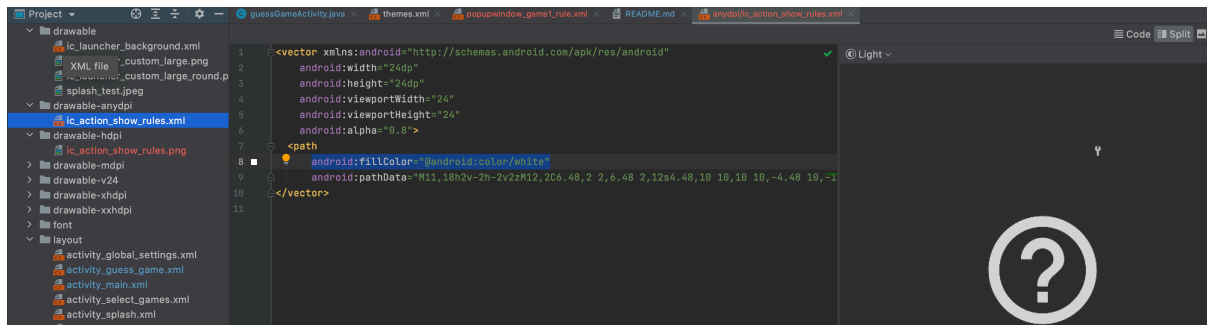
内容：为每个 game 的 Activity View 的 ActionBar 设置“规则”功能

状态： ☒ 于12月8日下午18:38

基本思想就是在 ActionBar 上添加一个按钮，按下此按钮可以跳出一个浮动弹窗 PopupWindow。

1. 用 Image Assistant 找到安卓自带图标库，并设置一个问号图标作为显示游戏规则的按钮

因为显示游戏规则的问号 icon 按钮用的是 Image Asset 创建的，我之前以为改变它的颜色可以直接像前面自动识别并改变 theme 那样，但是实际上是要去 `drawable-anydpi/ic_action_show_rules.xml` 里设置：

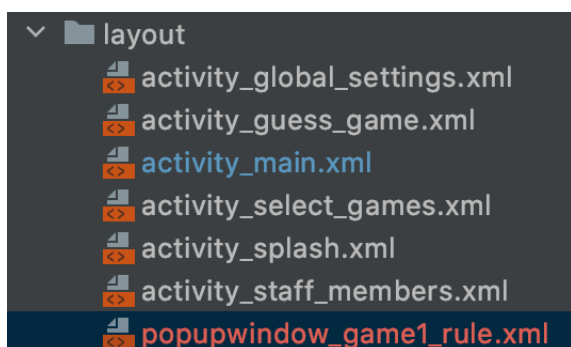


但是我已经不是很想再单独让这玩意儿的颜色跟着 theme 走了，想想都觉得麻烦.....直接一个黑色并且设置透明度得了，这样的话不透明度叠加使得颜色也许正好可以达到之前 theme 里类似的深色效果呢。

2. 创建给 ActionBar 用的布局文件



3. 在 guessGame Activity 里 @Override `onCreateOptionsMenu` 和 `onOptionsItemSelected` 方法
4. 在 `onOptionsItemSelected` 方法中，给这个 showRule 按钮创建一个动作以产生一个 PopupWindow
5. 创建给 PopupWindow 用的布局文件



6. 由于游戏规则有多行文字，因此在此 PopupWindow 的布局文件中需要另外对文字样式进行指定

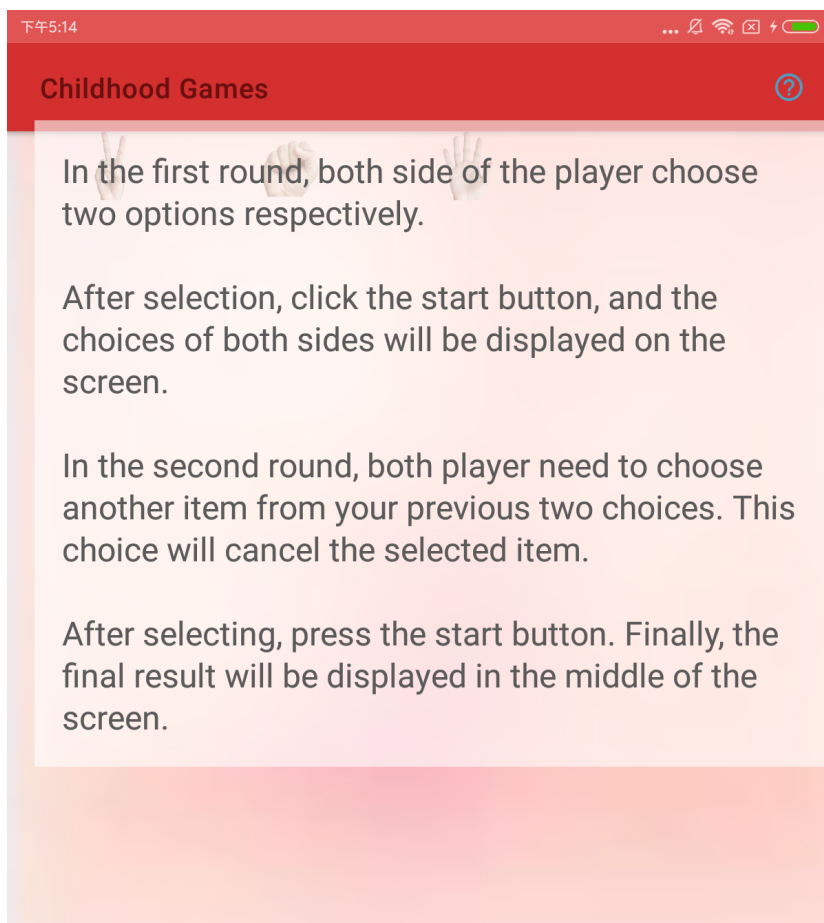
```
// string.xml
<!-- 我尝试使用 <b />, 但教程说应该是用的 \n -->
<string name="game_1_rule">In the first round, both side of the player
choose two options respectively.
    \nAfter selection, click the start button, and the choices of both sides
will be displayed on the screen.</string>

// pop....xml
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:lineSpacingExtra="2dp"
    android:scrollbars="vertical"
    android:singleLine="false"
    android:text="@string/game_1_rule"
/>
```

关于自定义的 PopupWindow 使用的布局（用的是 ConstrainLayout）和在 guessGameActivity 中跳出此自定义 PopupWindow 使用的布局：

```
popupWindow.setWidth(ViewGroup.LayoutParams.WRAP_CONTENT);
popupWindow.setHeight(ViewGroup.LayoutParams.WRAP_CONTENT);
```

这两者似乎是产生了矛盾或者其他什么很奇怪的反应，使得显示出来的 PopupWindow 并不能达到预期的自定义效果：



我后来也没彻底搞明白该如何高度自定义化 PopupWindow 的大小，反正现在这个成果看起来还算不错，因此就打算这样了。

7. 创建 PopupWindow 的实现方法

又是一轮艰苦的查找 PopupWindow 有关资料文章的过程。又不得不感叹说找的一堆关于 PopupWindow 的文章都讲的好杂，代码风格也很杂（有的是在设置 PopupWindow 的属性时候直接用参数最齐全的 PopupWindow 构造器直接一行流，有的是分了很多步骤）：

```
PopupWindow popupWindow = new PopupWindow();
popupWindow.setWidth(ViewGroup.LayoutParams.WRAP_CONTENT);
popupWindow.setHeight(ViewGroup.LayoutParams.WRAP_CONTENT);
popupWindow.setContent(View.inflate(this, R.layout.layout_popup, null));
/**
 * or
 * View v = LayoutInflater
 *         .from(MainActivity.this)
 *         .inflate(
 *             R.layout.layout_pop,
 *             [root:]null,
 *             [attachToRoot:]false
 *         );
 * popupWindow.setContent(v);
 */
// 以上代码可以简化为：
PopupWindow p = new PopupWindow(
    View.inflate(this, R.layout.layout_popup, null),
    ViewGroup.LayoutParams.WRAP_CONTENT,
    ViewGroup.LayoutParams.WRAP_CONTENT,
    [focusable:]false
);
```

而且很多技术细节，因为毕竟这一整个安卓项目就只是基于“遇到一个问题就去查解决方案并

copy + modify code”的方法完成的，因此有很多基础的技术细节到头来并不知道具体是怎么实现的，而且这种情况在我尝试转向官方手册的时候变得更为恶化了起来。比如我尝试改变 PopupWindow 的 `showAsDropDown` 为 `showAtLocation`，但是查官方手册的时候看到：

void	<code>showAsDropDown(View anchor, int xoff, int yoff, int gravity)</code> Displays the content view in a popup window anchored to the corner of another view.
void	<code>showAsDropDown(View anchor, int xoff, int yoff)</code> Display the content view in a popup window anchored to the bottom-left corner of the anchor view offset by the specified x and y coordinates.
void	<code>showAtLocation(View parent, int gravity, int x, int y)</code> Display the content view in a popup window at the specified location.

噢他妈的这个 anchor View 该怎么获取呢？于是就又：



结果搜了一大堆不知所然的东西，于是放弃了，还是换回了原来的 `showAsDropDown`。暂时没有实现“再按一下 ActionBar 上的那个问号按钮以取消显示 PopupWindow”的功能。

8. 修缮：修改各个游戏的 ActionBar 显示的文字

需要在 AndroidManifest 里的每个 Game Activities 里增加一个 Attributes：

```
<!-- Guess Game Activity -->
<activity
    android:name=".guessGameActivity"
    android:label="Guess What Other Is?"/>
```

9. 修缮：PopupWindow 内显示的内容并不能滚动，而且显示效果很恶心

后来尝试给一整个 PopupWindow 加了 ScrollView。

因为到写下这段话的期间我仍旧没能完全搞懂 WRAP_CONTENT 和 MATCH_CONTENT 的区别，因此也只能估摸着给 PopupWindow 和里面的 TextView 分别加了一些 margin 使得规则页面不会直接占满一整个屏幕，然后根据实际显示效果调整 margin 的大小。

说实话这个 Propose 花的时间比我预计的要长，大半个下午泡在图书馆就完成了这一个功能给我的感觉还是不太好的。

- Issue 7 - 12月8日下午18:39

位置：activity_splash.xml & activity_main.xml

内容：修改原先用来占位的沙雕开屏页面，并且在主页面加入一个好看的图片

状态： ☒ 12月8日晚19:45

早知道说用 Image Asset 来自动设置 logo 的时候不需要严格对应分辨率，我当时画 logo 的时候就不应该开这么小个画布，害得我现在为了给主页做个图还得把之前画好的小图用 waifu2x 放大.....

我直接将这两个 activity 的页面全部改成了 RelativeLayout。

比如说，以下这两个属性里，第二个才是“与xx View的底部对齐”的效果：

```
android:layout_alignBottom=xxx
android:layout_below=xxx
```

顺便既然都把 Splash 和 Main 的两个布局设计的差不多一致了的话，那就顺便可以考虑优化一下从 splashActivity 过渡到 mainActivity 的动画了。

- Propose 7 - 12月8日晚19:50

位置：主要在各个 Activity View 之间

内容：添加 Activity 过渡动画

状态： ☒ 于12月8日晚20:33

我又成功给自己多作了一个死.....

创建动画的方式有很多种，有要求自己去 res/transition 自己写动画的，也有说不需要、可以直接 *new Fade()* 的。

```
// Setup Activity Transition for splashActivity -> mainActivity
getWindow().requestFeature(Window.FEATURE_CONTENT_TRANSITIONS);
// Fade fadeTransition = (Fade)
TransitionInflater.from(this).inflateTransition(R.transition.activity_fade);
// getWindow().setEnterTransition(fadeTransition);
getWindow().setEnterTransition(new Fade());
```

之前写 splashActivity 的时候抄的是网上的代码，其思路是重写了此 Activity 的 onStart 并且 Thread.sleep 了指定时间：

```
@Override
protected void onStart() {
    super.onStart();
    new Thread(new Runnable() {
        public void run() {
            long start = System.currentTimeMillis();
            long costTime = System.currentTimeMillis() - start;
```



```

        if (sleepTime - costTime > 0) {
            try {
                Thread.sleep(sleepTime - costTime);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        startActivity(new Intent(splashActivity.this,
mainActivity.class));
        finish();
    }
}).start();
}

```

但是在加上用于设置过渡动画的部分后：

```

startActivity(new Intent(splashActivity.this, mainActivity.class),
    ActivityOptions.makeSceneTransitionAnimation(splashActivity.this).toBundle());

```

就会报错：

```

E/AndroidRuntime: FATAL EXCEPTION: Thread-2
    Process: com.example.androidapplicationtest, PID: 27746
    java.lang.RuntimeException: Can't create handler inside thread that has not called Looper.prepare()
        at android.os.Handler.<init>(Handler.java:204)
        at android.os.Handler.<init>(Handler.java:118)
        at android.app.ActivityTransitionCoordinator.<init>(ActivityTransitionCoordinator.java:219)
        at android.app.ExitTransitionCoordinator.<init>(ExitTransitionCoordinator.java:70)
        at
        android.app.ActivityOptions.makeSceneTransitionAnimation(ActivityOptions.java:758)
        at
        android.app.ActivityOptions.makeSceneTransitionAnimation(ActivityOptions.java:672)
        at
        com.example.androidapplicationtest.splashActivity$1.run(splashActivity.java:43)
        at java.lang.Thread.run(Thread.java:764)

```

于是转而寻找更为靠谱的方法。后来找到个看着比较靠谱的方法：

法：<https://www.jianshu.com/p/ce3a04e8fbf0>，对里面的代码稍作修改后算是成功了。

- Event 2 - 12月9日18:00

根据我的要求，今天下午在老地方（图书馆三楼手机区）开了一次会。

这次开会除了大概整理了一下目前进度之外还发现了几个新的问题，大概如下：

- Error 6 - 12月9日下午

位置：messageBoardActivity.java

内容：尝试 merge 这个 Activity 并运行后报错

状态：☒

负责写 guessGame（游戏的正式名称叫“Double Rock-Paper-Sissor”）的那位老哥临时决定写个 messageBoard 功能，就是在主页面叫我多加一个入口而已。但是他用的实现方式我没见过，后来研究了一下，发现他用了两个 View，一个负责渲染列表的其中一项，一个负责渲染整个列表，但是即使我检查了两三次在 merge 过程中为了统一文件名和类名而更改的部分，运行的时候仍旧报错。

之后在开会的时候打算给这老哥看一下自己的代码哪里有问题，但是还没等他看明白，我看了一眼报错信息后发现可能是因为在他写的页面文件直接搬过来的过程中针对 Theme.AppCompact 的部分没带过来，而且正好我暂时没给这个 messageBoardActivity 的 View 添加自动识别并改变 Theme 的功能，加上之后就正常了。

- Issue 8 - 12月9日19:05

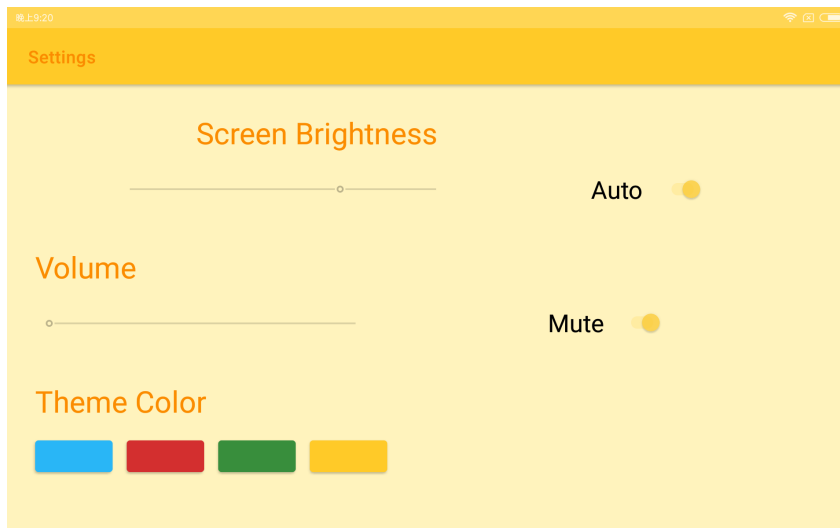
位置：activity_global_settings.xml

内容：修改此页面文件使得其在翻转平板的时候布局不会崩

状态：☒

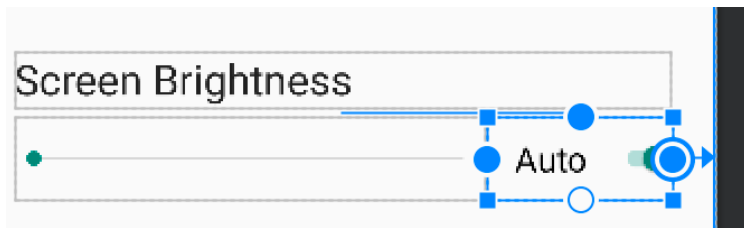
当然我自己很清楚说我们四个写的这些页面文件基本上都可以实现至少在平板转过来的时候不会突然崩溃这一点其实都是无心插柳。而且更别说我估计这些页面文件肯定是没办法 handle 不同尺寸的屏幕的。

目前唯一一个翻转设备到横向会出问题的是设置页面，翻转后会呈现这样的效果：



这显然不是我想要的。

后来检查发现其实和这几个错位的组件设置的 ConstraintView 属性有关，删掉靠右吸附之后就好了。



- Propose 8 - 12月9日21:28

位置：所有旋转平板后显示会出bug的页面

内容：同上

状态：Deprecated

哈，又是一个自立的 flag。

不过这个 flag 我最终没怎么实现，就只是简单的改了那几个旋转设备后非常明显会出现错误的页面就没了。

- Propose 8 - 12月9日19:05

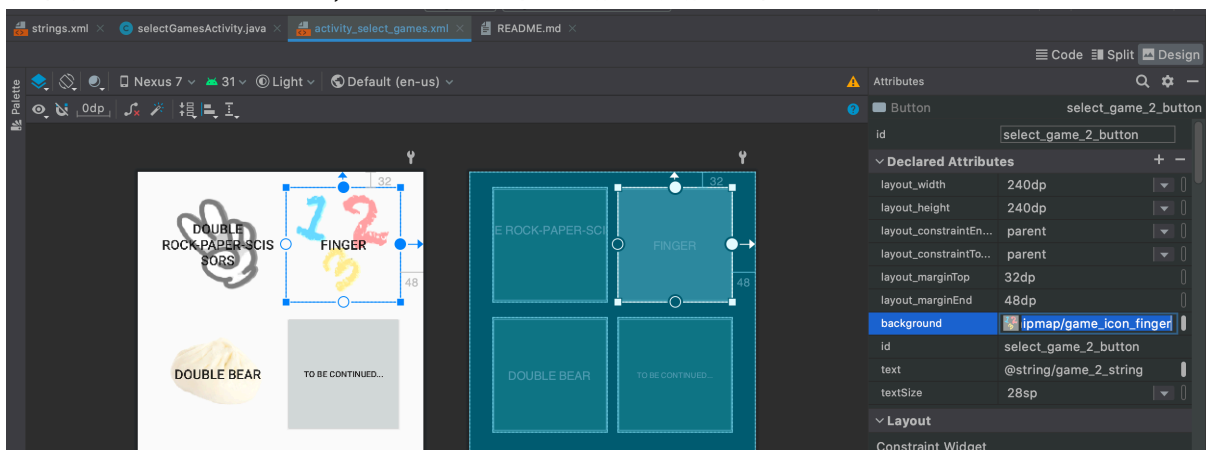
位置：三个游戏 Activities

内容：在 selectGame Activity 的按钮添加背景

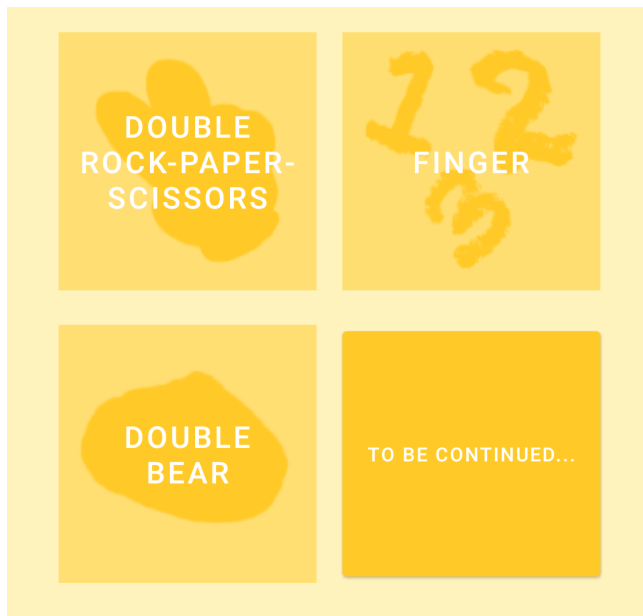
状态：☑

还是负责写 guessGame 的那位老哥临时下的决定.....特么还真把自己当项目经理了.....

后来用在设计 logo 的时候用过的素材做了三张 300*300 的 PNG，尝试放了进去，中途改了一下按钮的大小以及排列，顺便发现说素材文件的文件名不能有大大写字符。



但是很神奇的是，这个效果反而成了这样，虽然不是预想的以及 Design 页面所反映的那样，但是这个反而还挺不错.....



- Issue 9 - 12月9日19:05

位置：三个游戏 Activities

内容：三个游戏各自的小改动（修改页面布局、使用新版本等等）

状态： ☒

在整理了其他三个人的代码之后尝试运行，在跑了第二个游戏发现应该没什么大问题之后跑第三个游戏报错，但是报错信息却和前一个游戏有关：

```
E/WindowManager: android.view.WindowLeaked: Activity
com.example.androidapplicationtest.fingerGameActivity has leaked window
android.widget.PopupWindow$PopupDecorView{632c3e3 V.E..... D 0,0-
1160,1258} that was originally added here
    at android.view.ViewRootImpl.<init>(ViewRootImpl.java:496)
    at
android.view.WindowManagerGlobal.addView(WindowManagerGlobal.java:349)
    at android.view.WindowManagerImpl.addView(WindowManagerImpl.java:94)
    at android.widget.PopupWindow.invokePopup(PopupWindow.java:1433)
    at android.widget.PopupWindow.showAsDropDown(PopupWindow.java:1284)
    at android.widget.PopupWindow.showAsDropDown(PopupWindow.java:1240)
    at android.widget.PopupWindow.showAsDropDown(PopupWindow.java:1219)
    at
com.example.androidapplicationtest.fingerGameActivity.initShowRulePopupWind
ow(fingerGameActivity.java:275)
    at
com.example.androidapplicationtest.fingerGameActivity.onOptionsItemSelected(
fingerGameActivity.java:260)
```

查了一下发现应该是我在写 PopupWindow 的时候没有手动 Dismiss 导致窗口被孤儿了，遂在每个用到这个 PopupWindow 的地方加上一个新的方法：

```
@Override
protected void onDestroy() {
```

```

super.onDestroy();
if (mDialog != null)
    mDialog.dismiss();
}

```

但是加上后仍然报错，查看报错信息并必应：

```

E/AndroidRuntime: FATAL EXCEPTION: main
    Process: com.example.androidapplicationtest, PID: 15607
    java.lang.RuntimeException: Canvas: trying to draw too
    large(224969728bytes) bitmap.
        at
    android.view.DisplayListCanvas.throwIfCannotDraw(DisplayListCanvas.java:229)
        at android.view.RecordingCanvas.drawBitmap(RecordingCanvas.java:97)
        at
    android.graphics.drawable.BitmapDrawable.draw(BitmapDrawable.java:529)

```

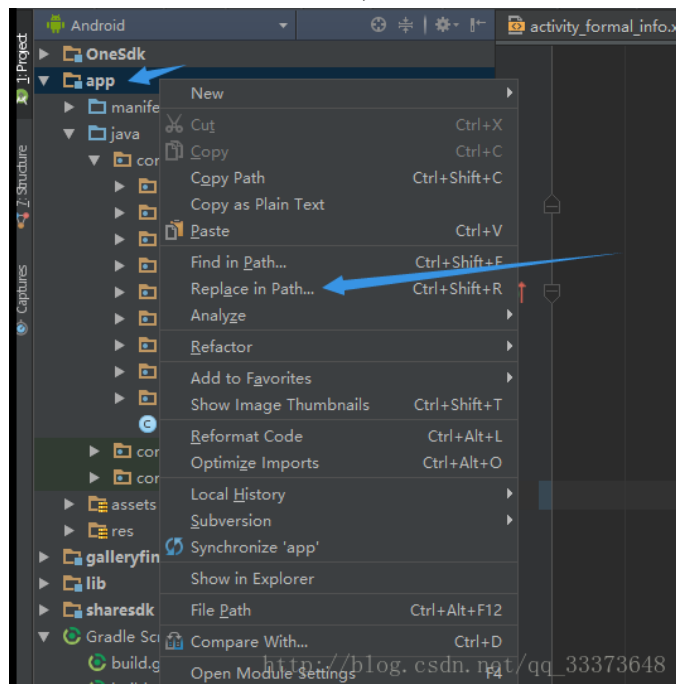
根据这两篇文章的说法：

<https://blog.csdn.net/u011047085/article/details/88179241>

https://blog.csdn.net/guolin_blog/article/details/50727753

说是在安卓系统中绘制一些分辨率比较大的位图时最好把图片资源文件放在 mipmap 文件夹而不是 drawable 里。

不过作为一个负责任的 IDE，它肯定是有批量替换功能的。



- Issue 10 - 12月9日19:25

位置：splashActivity

内容：从 mainActivity 按动返回按钮可以回到 splashActivity

状态：☑ 但是仍旧存在一定遗憾

之前在解决 Propose 7 的时候不知道当时在想些什么居然没杀掉 splashActivity，现在补刀：

```
@Override
public void run() {
    startActivity(new Intent( packageCont
        ActivityOptions.makeSceneTra
        // splashActivity.this.finish();
    }
}, sleepTime);
```

.....但是把这句代码解除注释之后我很快就知道为什么我当时要注释掉这句了。这句使得之前写好的 Fade 转场会出现短暂显示主页面才跳到 mainActiviy 的bug。

于是又一次想到了课上的 *startActivityForResult* 方法，很巧的是，课上老师给的例子也是和这个有关，也是返回一个 RESULT_OK 以 inform 前一个 Activity 自杀。但是这个实现方式也仍旧是有自己的一个 bug，那就是必须得在 MainActivity 下按下返回按钮、并且短暂地回到 splashActivity、把 Intent 要传的 result 传过去、splashActivity 才会被关掉，而这个过程中仍旧会短暂地出现一小会儿的 splashActivity，不过反正也就权当做是一个 goodbye screen 了。

（12月11日晚的突发感想）我忽然间想起来很多安卓软件选择在主页面里提示用户“再按一次返回键退出 app”的原因了.....

- Error 7 - 12月11日下午17:10

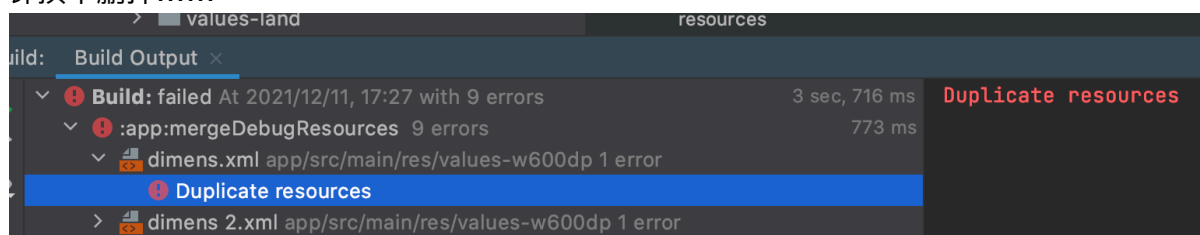
位置：一整个项目

内容：作死产物

状态：☒

.....我再也不把这种会经常改动而且文件依赖性很强的特殊文件（群）存到 iCloud 里了.....

所有的文件全被复制了一份，而且因为暂时不知道 git 怎么搞回滚操作，因此只能花了几分钟挨个删掉.....



- 结尾 - 12月12日

大概也就是在这天晚上宣布结束整个开发流程了。当天晚上把 GitHub 的项目页面设置为公开，完成了 Staff Acknowledgement 的部分，因为已经没时间没精力完成自己之前预想的那种炫酷页面设计因此只是简单的打了几个字放了张图然后放了 GitHub 页面链接上去。在学

校图书馆把最终的代码上传到 GitHub。

然后就是着手开始准备 Group Report、Individual Report 以及后来被改成 Video 的 Presentation 了。