

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

任课教师：郑贵锋

年级	15 级	专业 (方向)	移动信息工程学院 (互联网)
学号	15352402	姓名	张晨
电话	13609751750	Email	zhangch99@mail2.sysu.edu.cn
开始日期	20170923	完成日期	20170930

一、 实验题目

基本 UI 界面设计

二、 实现内容

实现一个 Android 应用

三、 课堂实验结果

(1) 实验截图(左侧为样例 右侧为我的 app)



(2) 实验步骤以及关键代码

- ① 将各类资源定义在 res 文件夹中的 values 文件夹下，字符串资源定义在 strings.xml 中，颜色资源定义在 colors.xml 中。图片资源则存放在 drawable 文件夹下。

② 整体布局

```
1. <android.support.constraint.ConstraintLayout
2.     xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:app="http://schemas.android.com/apk/res-auto"
4.     xmlns:tools="http://schemas.android.com/tools"
5.     android:layout_width="match_parent"
6.     android:layout_height="match_parent"
7.     android:gravity="center_vertical"
8.     tools:context="com.example.john.myapplication.MainActivity">
```

③ 用 TextView 控件显示标题

④ 用 ImageView 控件显示图片

⑤ 用 TextView 和 EditText 控件显示学号输入框和密码输入框

⑥ 用一组单选按钮 RadioGroup、RadioButton 控件做学生和教职工选项按钮

⑦ 用 button 控件做注册和登录按钮。下图是 drawable 文件夹里的 button_shape.xml 布局文件。定义按钮的样式。在 button 中可以用下面这条语句引用此资源。

android:background="@drawable/button_shape"

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <shape
3.     xmlns:android="http://schemas.android.com/apk/res/android"
4.     android:shape="rectangle"
5.     android:layout_width="match_parent"
6.     android:layout_height="match_parent">
7.     <!-- 填充的颜色 -->
8.     <solid android:color="@color/Aqua" />
9.     <!-- android:radius 弧形的半径 -->
10.    <corners android:radius="10dp" />
11.    <!-- padding: Button 里面的文字与 Button 边界的间隔 -->
12.    <padding
13.        android:bottom="5dp"
14.        android:left="10dp"
15.        android:right="10dp"
16.        android:top="5dp" />
17. </shape>
```

⑧ 然后创建一个 java 文件导入该布局，再在注册文件中注册，将该 Activity 设置为应用启动时第一个加载的 Activity。

(3) 实验遇到困难以及解决思路

① 不同的组件之间的间隔的设定。

采样相对定位 (Relative positioning)的方法，和相对布局的很像，

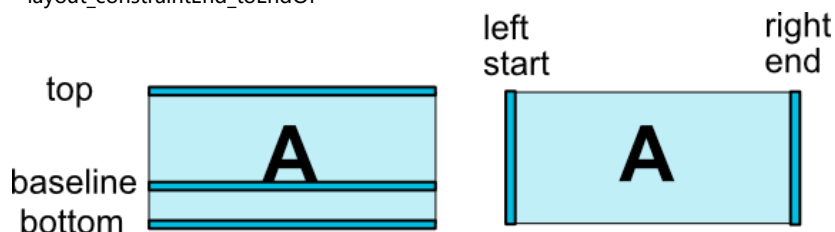
* layout_constraintLeft_toLeftOf * layout_constraintLeft_toRightOf * layout_constraintRight_toLeftOf

* layout_constraintRight_toRightOf * layout_constraintTop_toTopOf * layout_constraintTop_toBottomOf

* layout_constraintBottom_toTopOf * layout_constraintBottom_toBottomOf * layout_constraintBaseline_toBaselineOf

* layout_constraintStart_toEndOf * layout_constraintStart_toStartOf * layout_constraintEnd_toStartOf

* layout_constraintEnd_toEndOf



属性都形如 layout_constraintXXX_toYYYOf, constraintXXX 里的 XXX 代表是这个子控件自身的哪条边(Left、Right、Top、Bottom、Baseline), 而 toYYYOf 里的 YYY 代表的是和约束控件的哪条边发生约束(取值同样是 Left、Right、Top、Bottom、Baseline)。YYY 可以是父控件"parent", 也可以是其他控件, 还可以是 guideline。当 XXX 和 YYY 相反时, 表示控件自身的 XXX 在约束控件的 YYY 一侧, 例如:

app:layout_constraintLeft_toRightOf="@id/button1", 表示的是控件自身的左侧在 button1 的右侧。当 XXX 和 YYY 相同时, 表示控件自身的 XXX 和约束控件的 YYY 的一侧对齐, 例如: app:layout_constraintBottom_toBottomOf="parent", 表示控件自身底端和父控件底端对齐。如下图的代码就是表示两个 button 相距 10dp。

```
1. <Button
2.     android:id="@+id/login"    />
3. <Button
4.     android:id="@+id/register"
5.     app:layout_constraintLeft_toRightOf="@+id/login"
6.     android:layout_marginStart="10dp" />
```

② Chain 的形成条件

把①中的代码做小小的修改, 增加了第 3 行这一句, 结合第 7 行, 建立了链条, 你连着我, 我抱着你。此时然后它们两个成了一个整体, 所以链条左边设 app:layout_constraintLeft_toLeftOf="parent" 使得和父控件左对齐, 右边设置 app:layout_constraintRight_toRightOf="parent" 使得和父控件右对齐, 这样整个链条就居中了, 最后对左控件设置了 margin, 相当于整个链条左边有了 margin, 因为此时链条已为整体了。

```
1. <Button
2.     android:id="@+id/login"
3.     app:layout_constraintRight_toLeftOf="@+id/register" />
4.
5. <Button
6.     android:id="@+id/register"
7.     app:layout_constraintLeft_toRightOf="@+id/login"
8.     android:layout_marginStart="10dp" />
```

学号:

请输入学号

9. 因为把链条视为一体并居中的方法十分简便, 我试着将上面的 TextView 和 EditText 控件也连成 chain。代码如下。但是此时第 12 行报错了, response 显示此时 id 为 input_student_id 的 EditText 控件尚未定义。非常常见的错误。但是如果两个控件属性相同时就可以创建 chain 了。此时我们可以猜想, 除非是创建 chain, 不能引用尚未声明的控件。

```
10. <TextView
11.     android:id="student_id"
12.     app:layout_constraintRight_toLeftOf="@+id/input_student_id" />
13.
14. <EditText
15.     android:id="input_student_id"
16.     app:layout_constraintLeft_toRightOf="@+id/student_id"/>
```

③ 控件居中的方法

a) android:gravity="center_vertical"或者 android:layout_gravity="center_vertical" 但是 emmmm 没有用，不知为何

b) ②中的方法，将链条或者一个控件左右和父控件对齐

c) 倾向 (Bias) 搭配 bias，能使约束偏向某一边，默认是 0.5，有以下属性：

layout_constraintHorizontal_bias (0 最左边 1 最右边)

layout_constraintVertical_bias (0 最上边 1 最底边)

④ ImageView 的 src 属性和 background 属性的区别：名字上来看一个是背景一个是图片来源，但是 src 有 scaleType 这个属性变换更加灵活。详见下面这个博客

ImageView 的 src 与 background 的区别分析 <http://blog.csdn.net/honeysx/article/details/50866478>

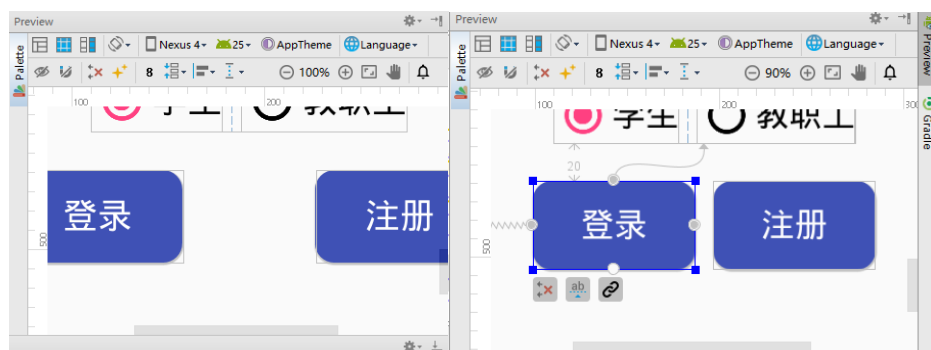
四、实验思考及感想

通过本次实验对 UI 设计有了初步实践。这次实验任务简单，但是可以用这次学到的知识为自己的小组项目开始设计一些操作简单的界面了。如 app 的启动画面、登录界面以及说明信息界面等。今后的目标是对界面除了输入之外的操作的研究，以及界面之间的切换跳转。界面在不同尺寸的屏幕上的适应性也值得留意。

【提交第一次作业后的补充】

看到在课程交流群里有几位同学问 button 怎么居中，我来补充下报告说说我自己的方法。用我上面列举过的方式是可以的。代码如下。把两个 button 连成链条然后把两个 button 左右和父控件相接，这样就可以使两个 button 居中了。但是此时还有一个问题，如果没有第 4 行，呈现的效果如图一，很显然虽然我写了令两个 button 间隔为 10dp，没有办到，即使我令间隔为 0 为负数，两个 button 像是同名磁极一样还是有一定的距离。加了第 10 行代码后，间隔变成 10dp 了，可以调整了。这是由于链条默认情况下是 spread chain，元素将被展开(老师的 PPT 说的)，我猜测被展开意味着一定的距离，而 packed chain 的元素被打包在一起，就不必在意距离了。

```
1. <Button
2.     android:id="@+id/login"
3.     app:layout_constraintLeft_toLeftOf="parent"
4.     app:layout_constraintHorizontal_chainStyle="packed"
5.     app:layout_constraintRight_toLeftOf="@+id/register" />
6. <Button
7.     android:id="@+id/register"
8.     app:layout_constraintRight_toRightOf="parent"
9.     app:layout_constraintLeft_toRightOf="@+id/login"
10.    android:layout_marginStart="10dp" />
```



五、 参考资料

Android 如何设置圆角按钮 <http://blog.csdn.net/lwcumt/article/details/47002067>

ConstraintLayout 属性详解 和 Chain 的使用 <http://blog.csdn.net/zxt0601/article/details/72683379>