

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：

年级	1501	专业（方向）	移动信息工程
学号	15352015	姓名	曹广杰
电话	13727022190	Email	1553118845@qq.com

Contents

一、	实验题目	1
二、	实现内容	1
三、	课堂实验结果	2
	实验截图	2
	实验步骤以及关键代码	2
	实验遇到困难以及解决思路	5
四、	课后实验结果	5
五、	实验思考及感想	5

一、 实验题目

基本 UI 界面设计

二、 实现内容

实现一个 Android 应用，界面呈现如下效果：



- 1) 该界面为应用启动后看到的第一个界面；
- 2) 要求只用一个 ConstraintLayout 实现整个布局：
 - a. 标题字体大小 20sp，与顶部距离 20dp，居中；
 - b. 图片与标题的间距为 20dp，居中；输入框整体距屏幕右边间距 20dp，上下两栏间距 20dp，内容（包括提示内容），内容字体大小 18sp；

- c. 学号对应的 EditText 只能输入数字，密码对应的 EditText 输入方式为密码；
- d. 两个单选按钮整体居中，字体大小 18sp，间距 10dp，默认选中的按钮为第一个；
- e. 两个按钮整体居中，与上方控件间距 20dp，按钮间的间距 10dp，文字大小 18sp。按钮背景框左右边框与文字间距 10dp，上下边框与文字间距 5dp，圆角半径 10dp，背景色为#3F51B5；

三、课堂实验结果

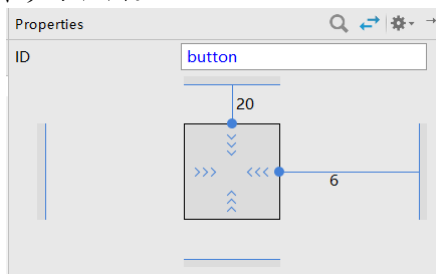
实验截图



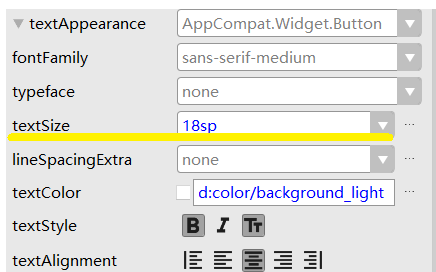
实验步骤以及关键代码

- 1) 实现所有关于字体大小以及控件之间的距离的要求，这些要求可以在右侧的 properties 中手动设置，不需要理解代码：

设置该部分的区域在 design 图形的右侧。

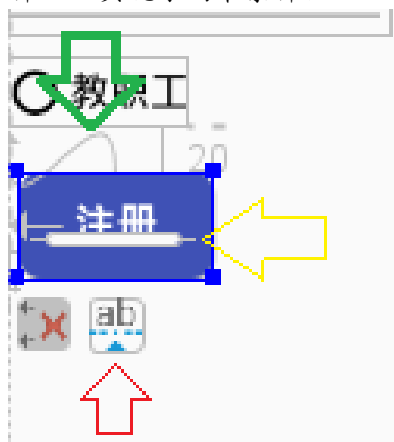


矩形周遭的数字可以用于表示该控件与其他控件的相对距离。而在右下方则可以用于设置字体、字号等信息。



如果需要将两个控件横向对齐，则可以使用 design 图形化的操作方式。先点击红色箭头所指的图形，上书“ab”，这时候控件上出现一个梭形，再点击黄色箭头所指的梭形，最后将梭形与其他控件连接起来，就可以实现两个控件之间的距离信息约束，这时候就会出现绿色箭头所

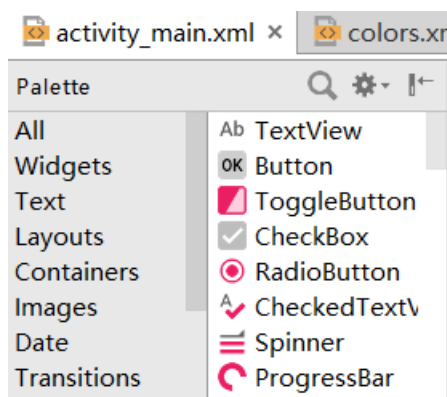
指的曲线，以此来表示这两个控件已经实现了约束条件：



这种约束可以用于控件对齐以及控件上下之间的固定距离，类别上，既不属于父容器约束也不是基准线约束，应该划分为视图约束。

2) 控件选择：

本次实验的所有控件有：TextView、EditText、Button、ImageView、RadioGroup、RadioButton，选择区域如下：



位于 design 图形的左上方。

3) 设置单选按钮：

RadioButton 是单选按钮，一组单选按钮需要包括在一个 RadioGroup 中，并且需要对 RadioGroup 和其包括的每一个 RadioButton 都设置 id，RadioButton 才能正常工作。

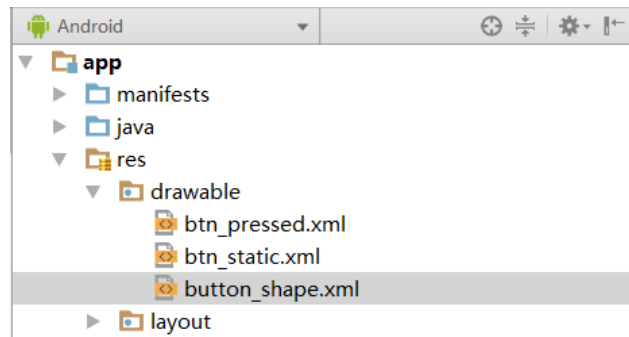
```
<RadioGroup
    <RadioButton
        android:id="@+id/radioButton"
    <RadioButton
        android:id="@+id/radioButton2"
```

关于按钮的属性可以按照之前的方法在右侧的 property 中设置，但是这种格式需要在代码中进行设置，好在实验课的 PPT 上都有相关的代码示例。

4) 设置圆角按钮：

设置圆角按钮需要添加新的布局文件。

1. 在 drawable 中新建 xml 文件，命名时候需要自己添加 “.xml” 后缀：



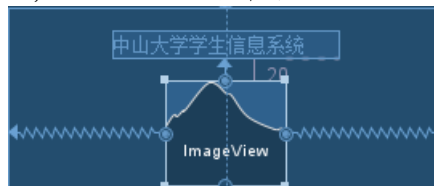
2. 在“activity_main.xml”的该控件信息位置处修改 background 信息为“@drawable/你的名字.xml”：“`android:background="@drawable/button_shape"`”。
3. 编写.xml 文件，笔者在该文件中使用 selector 进行图形不同状态设置的切换，使用 item 用于检测是否该图形被触摸，以此实现两个状态的转换。同样的，在 selector 中，笔者对于图形在触摸时与非触摸时：

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Button 正常状态下的背景 -->
    <item android:drawable="@drawable/btn_static" android:state_pressed="false"/>
    <!-- Button 按下时的背景 -->
    <item android:drawable="@drawable/btn_pressed" android:state_pressed="true"/>
</selector>
```

4. 使用代码“`<corners android:radius="10dp" />`”在新建.xml 文件中添加即可设置圆角；

5) 如何设置居中：

笔者在设置居中格式的时候使用的是链样式设置，因为笔者目前只知道链样式设置。即在控件的左端（或者右端）会有一个小圆圈，拖动时会出现链条，指向需要对齐的控件或者 parent，在左右同时对齐的时候，就可以实现居中了。



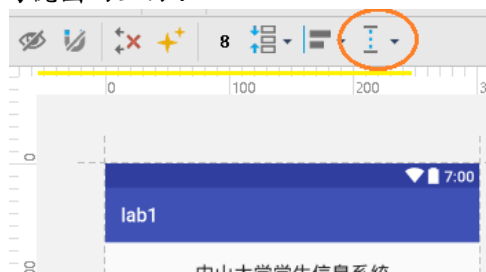
从代码角度而言，使用的是 layout constraint，即视图约束布局。左侧约束到 parent 的左侧，右侧约束到 parent 的右侧，则实现居中。

```
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
```

6) 如何设置整体居中：

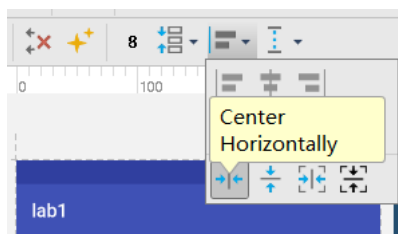
使用基准线约束布局。

先添加基准线信息，在布局视图的上方：



点击基准线端部的三角符号，基准线的基准会不断变化，我们需要将其参数调节为整个屏幕的 50%，然后将需要整体居中的两个控件设置为关于 vertical 基准线对称——得到整体居中效果。

设置对称：



实验遇到困难以及解决思路

第一次做 Android 开发的实验，笔者的实验过程不是特别顺利。

1. 设置单选按钮。笔者在最初仅仅是搭建了一个框架，关于单选按钮的设计是在看 PDF 之后才知道应该如何入手的，可是在未动手实践之前，教程上的介绍总是不能理解，而动手之后，常常又忽略了教程的重要性，可能在观摩教程之前自己随便玩几分钟会收获更好的效果吧。
2. 设置整体居中。“整体居中”这个字眼让笔者一度非常困惑，后来明白也许是关于中心线对称的，继而笔者添加的基准线约束。这种表意不清的困难实在是让笔者不知道如何避免。这真的不是没有困难条件下的创造困难嘛，换成“关于中位线对称”是不是好一点。
3. 添加圆角按键。按键的设置还是比较常规的，但是设计为圆角就比较困难了，笔者在查阅了相关的博客之后，发现是需要自己添加新的布局文件的，这个难度就要高一些了。笔者首先还是把博客上的代码复制过来，预先 run 一遍，结果显示没有检测到引用新建布局文件——在代码不处问题的情况下，依然无法引用到，这可能是文件关联的问题。经过询问同学得知在命名时需要自己添加文件格式。所以，在信息管理上报错时，系统应对的往往是文件或者较为系统的部分，而代码相比之下则太过于微观了。

四、 课后实验结果

如图，笔者在按下该按钮之后，其颜色会变成代表原谅的绿色。



五、 实验思考及感想

本次实验实现的仅仅是 Android 应用的界面设计，但是在实现的时候就已经有很多的困难，几度陷入不知如何入手的窘状。

这些困难在之前已经提到过了，而笔者对于文件命名的问题始终不能释怀。很显然这种问题在 **blog** 上很少提及，而避免这种问题浪费时间的主要办法还是依靠开发人员的开发经验，而使用者则需要对于软件的文件关联额工作原理有所理解，这是一劳永逸的办法。

关于软件的使用体验，**Android studio** 还是很让笔者惊喜的，开发过程相对流畅，通过视图约束可以在 **text** 中自己编写代码简直让笔者眼前一亮，虽然由于连线过程无序导致的代码无序化严重，可读性降低，但是还是大大地减轻了开发者的工作量以及上手难度。

对于本次实验的一些感受，本次实验虽然设计的是一个非常简单的界面，但是此后设计静态界面都不会太困难，应该说设计得还是非常有趣，深入浅出。对于适应 **Android** 开发过程以及开发语言都非常有帮助。