

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	专业 (方向)	移动信息工程
学号	15352121	姓名	黄铖杰
开始日期	2017/9/26	完成日期	2017/9/27

一、 实验题目

基于 AS 实现简单的安卓应用基本 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：



只能使用一个 ConstraintLayout 来实现整个布局，具体的要求放在实验结果里面进行说明。

三、 课堂实验结果

(1) 实验截图



上面的结果就是我的根据实验要求的布局结果。

(2) 实验步骤以及关键代码

- 设定标题字体大小 20sp，与顶部距离 20dp，居中。

在 ConstraintLayout 的布局里面，显示文字的控件是 TextView。约束布局里面要想实现 TextView 框和顶部距离 20dp，就将框的顶部的约束条件设定为布局的顶部也就是 parent；居中的效果直接将框的左右约束设定为 parent 就可以了。

```
<TextView
    android:id="@+id/title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="中山大学学生信息系统"
    android:textSize="20sp"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="20dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent" />
```

- **图片和标题的间距为 20dp，居中。**

显示图片的控件是 `ImageView`。和标题设定距离的原理一样，但是建立约束的时候由于图片框要和标题框建立约束，所以约束对象就是标题的 id 号，也就是上图中的 'title'。居中的效果依然是将框的左右约束设定为 `parent`。

值得注意的是怎么让一个 `activity` 读到图片文件。做法就是按照下面的顺序打开文件夹：`app->java->res->drawable`，将图片放在 `drawable` 文件夹里面，然后在 AS 里面就可以看到这个图片资源了。再使用 `android:src` 语句来调用这个图片。

另外假如图片的长宽都使用 `wrap_content` 的话放在整个布局里面大小就不太合适，所以自己就需要来设定长宽的数值。

```
<ImageView
    android:layout_width="130dp"
    android:layout_height="137dp"
    android:src="@drawable/syxsu"
    android:layout_marginTop="20dp"
    app:layout_constraintTop_toBottomOf="@+id/title"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:id="@+id/imageView" />
```

- **输入框整体距离屏幕右边 20dp，上下两栏间距 20dp，内容字体大小 18sp。**

这一个要求需要用到四个控件，分别是两个输入和两个文字显示。输入框用的是 `EditText`。输入框整体距离屏幕右边 20dp 也就是说两个输入框的右边约束设定为 `parent` 以后，再将右边约束距离设定为 20dp，上方输入框的上约束没有要求，为了增强布局的稳定就设定和图片的间隔为 10dp。

下方输入框的上约束为上方输入框，上约束间隔是 20dp，对应的左边文字框使用 `TextView` 来定义，同一行的输入框和文字框的间隔也是自己定义。这里有一个问题就是如果不外加约束，由于 `TextView`

和 EditText 的框内字体位置不一样，所以在整个布局里面文字框的文字和输入框里面的文字就对不齐。一个好的方法是将文字框的上下约束分别设定为输入框的上下边界，这样就可以解决了。

两行输入栏的输入类型不一样，学号栏使用 `inputType="number"` 来约束，密码栏使用 `inputType="textPassword"` 来约束。

为了方便就只贴出密码那一行的代码了。

```
<EditText
    android:id="@+id/input2"
    android:layout_width="240dp"
    android:layout_height="44dp"
    android:textColor="@color/colorPrimary"
    android:textSize="18sp"
    android:hint="请输入密码"
    android:inputType="textPassword"
    android:layout_marginRight="20dp"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="20dp"
    app:layout_constraintTop_toBottomOf="@+id/input1"
    android:layout_marginLeft="20dp" />
```

```
<TextView
    android:id="@+id/hint2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="密码: "
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="@+id/input2"
    app:layout_constraintTop_toTopOf="@+id/input2"
    app:layout_constraintRight_toLeftOf="@+id/input2"
    android:layout_marginRight="8dp" />
```

- 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。

单选按钮使用 `RadioButton` 控件，由于两个单选按钮为一组，所以放在一个 `RadioGroup` 控件里面。

单选按钮的整体布局是在 RadioGroup 里面设定的，呈现要求两个按钮是水平的，因此在 RadioGroup 里面 orientation 要设定为

horizontal；还有整体居中也就是左右约束设定为 parent。

在 RadioButton 里面，两个按钮都要设定自己的 id，由于学生是默认选项，在学生的控件里面要加上 checked='true' 语句。并且两个按钮间隔为 10dp，所以将左边按钮的右约束设定为右边按钮的左边界，约束距离为 10dp 就可以解决了。

下面就只贴出学生按钮的代码了。

```
<RadioButton
    android:id="@+id/id1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="学生"
    android:textSize="18sp"
    app:layout_constraintRight_toLeftOf="@+id/id2"
    android:layout_marginRight="10dp"
    android:checked="true"/>
```

- 两个按钮整体居中，与上方控件间距 20dp，按钮间间距 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。

两个按钮使用 Button 控件。要想将两个 Button 整体居中并且按钮间的间距有要求的话最好就是将两个 button 变成一条 chain，然后左边按钮的左边界约束和右边按钮的右边界约束都为 parent，中间的间隔是 10dp，这就解决了间隔问题。

将按钮的背景图形进行改动，步骤如下：在 drawable 里面 new 一个 resource file。然后在这个文件里面将 'selector' 字样都改为 'shape' 字样，然后在 shape 所在的 <> 框内加上 android:shape="rectangle" 的语句，最后在域内设定圆角半径和背景颜色等。

要设定内边框和文字的间距要用 padding__ 语句来设定。

```
<Button
    android:background="@drawable/button_shape"
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="登陆"
    android:textSize="18sp"
    app:layout_constraintRight_toLeftOf="@+id/button2"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginTop="20dp"
    app:layout_constraintTop_toBottomOf="@+id/id0"
    app:layout_constraintHorizontal_chainStyle="packed"
    android:layout_marginRight="10dp"
    android:paddingBottom="5dp"
    android:paddingTop="5dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"/>
```

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <solid android:color="#3F51B5" />
     <corners android:radius="10dp" />
</shape>
```

(3) 实验遇到的困难以及解决思路

- 不知道怎么让 Activity 读取到想要的图片文件。只要将所需图片放在 drawable 下面就可以在 xml 中通过 arc 的语句来调用这张图片。
- 不知道怎么设置引导线。在这次实验里面不需要认为添加基准线，都只需要通过控件和控件之间的距离或者控件和布局之间的距离就可以完成布局。如果人为添加基准线的话，使用 constraint.Guideline 的控件来设定。
- 不知道怎么形成一条 chain。有两种方法，一种是在可视化界面选中两

个控件然后右键就能看到生成一条 chain 的选项。另一种就是在 xml 里面互相在同一个方向的不同边互相成为约束就可以了。

- 不知道怎么修改按钮的背景图片。在 Baidu 上找到了答案，要自己在 drawable 里面新建一个 resource file 然后进行一系列修改，颜色的修改是通过 solid，四个角的圆化是通过 corners。

四、 课后实验结果

*

五、 实验思考及感想

这是第一个 Android 实验，做下来其实遇到的问题挺多的，一个看似简单的界面其实涉及到知识以及细节都挺多的。比如控件和控件整体居中的同时内部也有距离约束、还有不同控件之间内含的函数或者变量也不一样，需要在网上仔细查找资料才知道要使用什么函数才能达到想要的效果。

其中感受最深的是我们要灵活运用 constraintLayout 的可视化操作的特性，比如设定约束的时候可以直接在可视化视图那里拉一根线就可以了、框的大小也可以直接拉动来设定合适的比例。

其实网上关于 Android 的教程也挺多的，灵活运用好资源仔细做好每一个步骤应该就不会出太大的差错。