

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)	软件工程(移动)
学号	15352133	姓名	黄少豪
电话	13727024545	Email	328730316@qq.com
开始日期	2017/10/2	完成日期	2017/10/3

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：



使用的布局和控件：ConstraintLayout、TextView、EditText、Button、ImageView、RadioGroup、RadioButton

三、 课堂实验结果

(1) 实验截图



(2) 实验步骤以及关键代码

1. 该界面为应用启动后看到的第一个界面:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

这里加载的是 activity_main 布局，所以所有布局代码均需在 activity_main.xml 文件中编写。

2. 要求只用一个 ConstraintLayout 实现整个布局:

```
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.dgl23.helloworld.MainActivity">
```

使用 android.support.constraint.ConstraintLayout 创建一个 ConstraintLayout 布局，之后所有的 View 组件均写在这个布局之下，从而实现 View 元件布局。

3. 标题字体大小 20sp，与顶部距离 20dp，居中；

```
<TextView
    android:id="@+id/headline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="20dp"
    android:textSize="20sp"
    android:text="@string/headline"
    android:textColor="@color/black"/>
```

字体大小 20sp 可以使用 android:textSize="20sp" 实现；与顶部距离 20dp 需要先使用 app:layout_constraintTop_toTopOf="parent" 使标题顶部与窗体顶部关联，之后使用 android:layout_marginTop="20dp" 即可实现与顶部距离 20dp；居中使用 app:layout_constraintLeft_toLeftOf="parent" 和 app:layout_constraintRight_toRightOf="parent" 使标题左右与窗体左右关联，然后在默认情况下，标题会自动左右居中。

4. 图片与标题的间距为 20dp，居中；

```
<ImageView
    android:id="@+id/icon"
    android:layout_width="0dp"
    android:layout_height="100dp"
    android:src="@mipmap/sysu"
    app:layout_constraintTop_toBottomOf="@id/headline"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="20dp"/>
```

使用 `app:layout_constraintTop_toBottomOf="@id/headline"` 使图片顶部与标题底部关联，之后使用 `android:layout_marginTop="20dp"` 实现图片与标题的间距为 20dp；居中原理同标题居中。

5. 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）如图所示，内容字体大小 18sp；

```
<EditText
    android:id="@+id/input_user"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="@id/username"
    app:layout_constraintTop_toTopOf="@id/username"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toRightOf="@id/username"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="40dp"
    android:hint="@string/user_hint"
    android:textSize="18sp"
    android:inputType="number"/>
```

提示内容使用 `android:hint="***"` 实现；这里还要求学号只能输入数字，所以使用 `android:inputType="number"` 使输入框只能输入数字；另外密码输入框还要求输入内容不可见，所以使用 `android:password="true"` 实现密码输入。

6. 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；

```
<RadioGroup
    android:id="@+id/radio_group"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintTop_toBottomOf="@id/password"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="20dp">
```

因为选项数量不唯一，且对两个单选按钮整体有布局要求，所以先定义一个 RadioGroup 圈住两个单选按钮，然后两个按钮的整体布局由 RadioGroup 定义，两个按钮之间的布局则由两个按钮自己内部决定；android:orientation=" horizontal" 表示 RadioGroup 内部组件的排列是水平的。

```
<RadioButton
    android:id="@+id/teacher"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintLeft_toRightOf="@id/student"
    android:layout_marginLeft="10dp"
    android:text="@string/teacher"
    android:textSize="18sp" />
```

这是第二个按钮的布局代码。因为要求间距 10dp，所以 layout_marginLeft 设置为 10dp；默认选中第一个则在第一个单选按钮的布局代码中加入这句 android:checked= "true" 。

7. 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
为了使两个按钮整体居中，同样地，使用 RadioGroup 将两个按钮包住并在 RadioGroup 设置整体样式实现整体居中；

```
<Button
    android:id="@+id/register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintLeft_toRightOf="@id/login"
    android:layout_marginLeft="10dp"
    android:textSize="18sp"
    android:textColor="@color/colorWhite"
    android:text="@string/register"
    android:background="@drawable/circle"/>
```

为了实现按钮背景要求，使用 android:background 设置背景样式，以下是 drawable/circle 的内容：

```

<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners
        android:radius="10dp"/>
    <padding
        android:bottom="5dp"
        android:top="5dp"
        android:left="10dp"
        android:right="10dp"/>
    <solid
        android:color="#3F51B5"/>
</shape>

```

corner 标签的 android:radius 设置背景圆角半径；padding 标签设置背景框的内边距，即边框与文字的距离；solid 标签中的 android:color 设置背景颜色。

（3）实验遇到困难以及解决思路

1. 在实验中提到了使用 chain 样式实现两个 Button 的水平布局，但查阅了一些资料后依然不太会使用，于是模仿了前面两个单选按钮的样式，使用了 RadioGroup 来实现两个 Button 标签的整体样式。
2. 一开始为了实现标题距离顶部 20dp，使用了 constraint.Guideline 在顶部设置一条水平参考线，然后设置标题距离这条参考线 20dp 的距离。后来发现直接使用 app:layout_constraintTop_toTopOf=“parent”使标题与主窗体顶部关联，然后使用 android:marginTop=“20dp”即可实现同样效果。

四、课后实验结果

五、实验思考及感想

本次实验较为简单，在这里作个简单总结：

1. 使用 android:id="@+id/****" 来为 View 元件设置 id，这个 id 号在后面确定 View 组件相对位置时非常有用
2. 使用 app:layout_constraint[_to_]Of 来使两个 View 组件关联并确定位置关系
3. 使用 android:margin[_] 来确定 View 组件外边框距离
4. 需要实现多个 View 组件的整体布局时，可使用 RadioGroup 组件将多个 View 组件组成一个整体并为这个整体设置布局样式。

5. 将字符串值，样式，颜色等定义在资源文件，布局文件采用引用的方式调用这些资源从而实现项目文件之间的解耦