移动应用开发实验报告(一)

学号	姓名	题目	时间	班级
15352322	王杏婷	基本 UI 设置	2017/9/26	15M3

实验目的

- 1. 熟悉 Android Studio 开发工具操作。
- 2. 熟悉 Android 基本 UI 开发, 并进行 UI 编程。

实验内容 实现一个 Android 应用, 界面呈现如下效果:



要求:

- 1. 该界面为应用启动后看到的第一个界面
- 2. 各控件的要求如下: 图片与标题的间距为 20dp, 居中;
- 3. 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;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;
- 4. 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;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

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

基础知识

ConstraintLayout

约束布局,根据布局中的其他元素或视图,确定 View 在屏幕中的位置,受到三类约束,即其他视图,父容器(parent),基准线(Guideline)。
layout constraint[本源位置] [目标位置]="[目标 ID] "

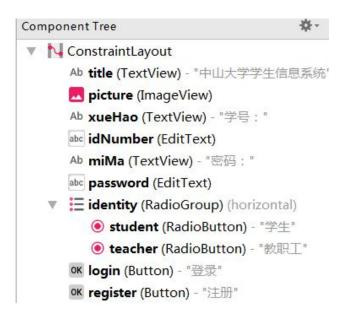
例如:app:layout_constraintBottom_toBottomOf="parent"

约束当前 View 的底部至目标 View 的底部,目标 View 是 constraintLayout. 即, 把当前 View 的底部对齐到 constraintLayout 的底部.

全部边界与 constraintLayout(父容器)边界对齐,则为居中。

实验过程

1. 新建一个 empty 项目,工程名称也即 APP 名称为 lab1,进入工程后样式文件默认为 ConstraintLayout。本次使用的图片 sysu.jpg 要先存在 app\res\mipmap 的目录下,打开 app\res\layout\activity_main.xml,把使用的布局和控件:TextView、EditText、Button、ImageView、RadioGroup、RadioButton 拉入 design 里面, 适当调整控件位置。当然,只是拉控件不靠谱,还是要自己手动写代码来控制控件的相对位置。



2. 标题水平居中的控制语句是:

```
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
与顶部距离 20dp:
app:layout_constraintTop_toTopOf="parent"
android: layout_marginTop="20dp"
标题字体大小 20sp:
android:textSize="20sp"
```

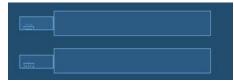
3. 图片与标题的间距为 20dp: 在布局中使用

app:layout_constraintTop_toBottomOf="@+id/title 指定当前控件(ImageView)的顶部约束到

title 的底部,并使用 android:layout_marginTop="20dp" 指定两者相距 20dp.

```
app:layout_constraintTop_toBottomOf="@+id/title"
android:layout_marginTop="20dp" />
```

4. 接下来要实现学号密码输入框:



整个输入框距离屏幕左端 20dp:

```
app:layout_constraintLeft_toLeft0f="parent"
android:layout_marginLeft="20dp"
```

EditText 用于进行学号和密码的输入,其中学号部分需要输入全正整数,使用到xml配置

```
android:inputType="number"
```

而密码框部分使用

android:inputType="textPassword"

限制 "学号输入" 输入框最大输入的字符串长度为 8

android:maxEms="8"

Textview 与学号输入框的左右相邻的约束关系:

app:layout_constraintRight_toLeftOf="@+id/idNumber"

令用于输入项目提示用的 TextView 控件基线对其到输入框的基线,即可取得比较好的对齐效果

app:layout_constraintBaseline_toBaselineOf="@+id/password"

在职业选择部分使用控件 RadioGroup 及 RadioButton , 一般在 RadioGroup 中放置 RadioButton 以达到单选的效果。并且 , 在第一个 RadioButton 使用 android:checked="true" 使得此控件被默认选中。

```
(RadioGroup
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
  (android:orientation="horizontal")
   android:id="@+id/identity"
   app:layout_constraintTop_toBottomOf="@+id/password"
   app:layout_constraintLeft_toLeftOf="parent"
   app:layout_constraintRight_toRightOf="parent"
   android:layout_marginTop="15dp"
   (RadioButton
      android:id="@+id/student"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
     android:checked="true" 默认值
      android:text="学生" />
   RadioButton
      android:id="@+id/teacher"
       android:layout_width="wrap_content"
      android:layout_height="wrap_content"
     android:layout_marginLeft="10dp"
                                 2个button距离10dp
       android:text="教职工"/>
(/RadioGroup)
```

最后是两个按钮"登陆"和"注册",都使用了 Button 控件。为了设置圆角效果,新建一个外部 xml 文件 shape.xml 用来声明实现类型为 rectangle 的 <shape> 后,目录放在 app\res\drawable,在 <shape> 中声明 padding 等样式可调整 Button 中内边框和文字的距离

```
<?xml version="1.0" encoding="UTF-8"?>
<shape shape.xml</p>
   xmlns:android="http://schemas.android.com/apk/res/android"
 android:shape="rectangle">
  〈! 一 填充的颜色 一〉
   <solid android:color="#3f51b5" />
   <!-- 设置按钮的四个角为弧形 -->
   <!-- android:radius 弧形的半径 -->
   <corners android:radius="10dip" />
   <!-- padding: Button里面的文字与Button边界的间隔 -->
   <padding</pre>
      android:left="10dp"
       android:top="5dp"
       android:right="10dp"
       android:bottom="5dp"
</shape>
```

并在 Button 中添加为 background 的方式,

android:background="@drawable/shape"

左边的 button 设置如下:

```
《Button
android:id="@+id/login"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="登录"
app:layout_constraintTop_toBottomOf="@+id/identity"
android:layout_marginTop="20dp"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="@+id/register"
android:layout_marginRight="10dp"
android:background="@drawable/shape"
android:textSize="18sp"
/>
```

2个 button 互相约束, 距离 10dp, 右边的 button 设置如下:

```
Button
android:id="@+id/register"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="注册"
app:layout_constraintTop_toBottomOf="@+id/identity"
android:layout_marginTop="20dp"
app:layout_constraintLeft_toLeftOf="@+id/login"
app:layout_constraintRight_toRightOf="parent"
android:layout_marginLeft="10dp"
android:background="@drawable/shape"
android:textSize="18sp"
/>
```

实验结果

数据线连安卓手机,点击运行结果如下:



实验体会

第一次实验主要是熟悉 Android Studio 以及约束布局(constrainLayout)的基本使用,根据布局中的其他元素或视图,确定 View 在屏幕中的位置,受到三类约束,即其他视图,父容器(parent),基准线(Guideline)。

layout_constraint[本源位置]_[目标位置]="[目标 ID] "

例如:app:layout_constraintBottom_toBottomOf="parent"

约束当前 View 的底部至目标 View 的底部,目标 View 是 constraintLayout.即,把当前 View 的底部对齐到 constraintLayout 的底部. 另外,有不懂的随时百度,大量的博客可以解决初学者的入门问题。