

中山大学移动信息工程学院本科生实验报告

(2017 学年秋季学期)

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	15352320	姓名	王诗语
电话	15626007477	Email	1532079717@qq.com
开始日期	2017.09.23	完成日期	2016.09.23

【实验题目】

——基本 UI 界面设计

【实验工具】

Android studio 2.3.3: constraint layout

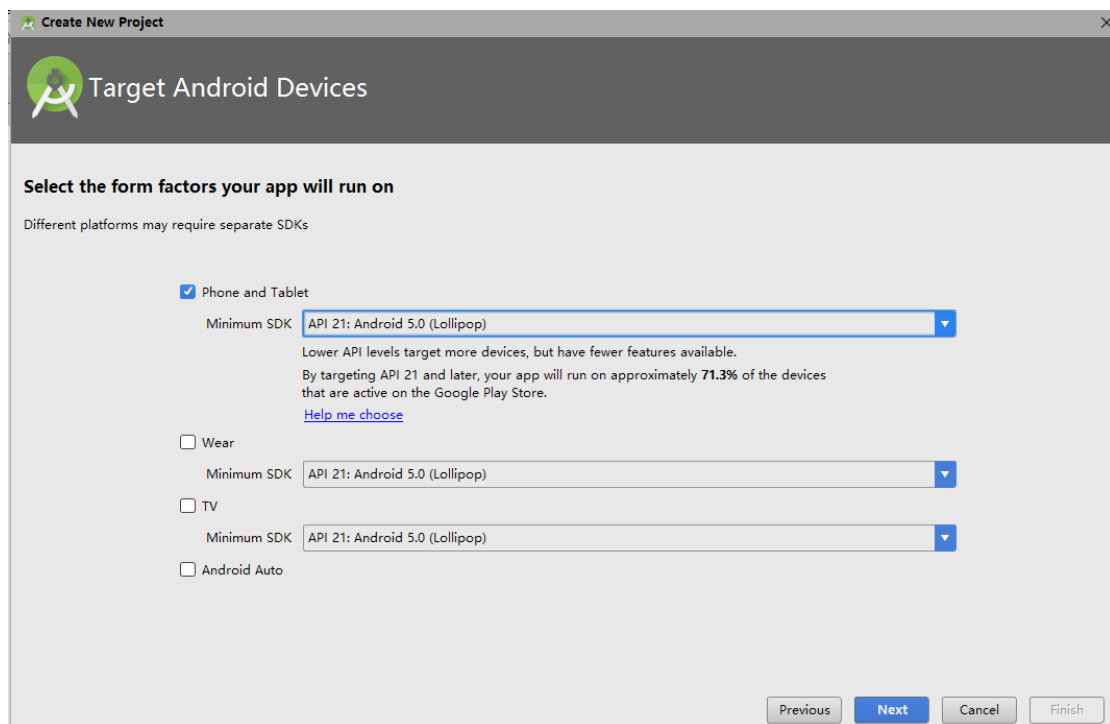
【实验目的】

- 1.熟悉 Android Studio 开发工具操作^[1]
- 2.熟悉 Android 基本 UI 开发,并进行 UI 基本设计

【实验过程】

1. 环境搭建

新建 android project 时，SDK 的选择上一般选择的最小 SDK 是 API15:Android 4.0.3，这一项可以根据自己的手机配置来选择，具体信息可以在手机 about 中查看，顺便，手机的分辨率也可以一起记录一下。



2. Constraint Layout 简介

约束布局由 Google I/O 在 2016 年推出，虽然发布时间较短，但从支持力度而言，将成为主流布局样式。而且因为约束布局减少了布局的层级，很大程度上优化了渲染性能，所以将完全代替其他布局。

Constraint Layout 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)**。设置约束的基本语法如下：

layout constraint[本源位置]_[目标位置]="[目标 ID]"

例如语句

`app:layout_constraintBottom_toBottomOf="parent"`（即父容器）

的效果就是把当前 View 的底部对齐到 Constraint Layout 的底部。

3. UI 实现

UI 界面效果如下图。



在 `constraintLayout` 的 `design` 界面中每个部件都可以通过拖动以及连接约束来定位。如果是代码，每个控件都有自己的 ID，当你要将两个控件彼此间建立约束的话，只要改变上述例子中的 `parent` 为控件 ID 就能实现。

代码中每设置一次约束，如连接到图片底部：

```
app:layout_constraintTop_toBottomOf="@+id/imageView"
```

距离的设置（如果需要）就要配套写一行：

```
android:layout_marginTop="20dp"
```

（`margin` 这个概念我留在最后的心得里与 `padding` 一起比较）

左方 `Palette` 窗口中，可以选择需要的东西拖入容器中，相当于建立好了一个控件。

点击一个控件，在右端 `Properties` 窗口中，可以更改长宽、文字等

属性，上方正方形试图可以通过设置与约束部件的距离来调整当前控件在 view 中的位置。一般默认 View 的长宽采用 wrap_content，如果采用 match_content，则会自动匹配到与父容器等长(宽)。在 Properties 窗口最下方找到 View all properties，可以对控件各种详细的属性，如内边距，颜色，背景框等进行设置。

但是很多人反映这个 design 界面常常碰一下就歪，异常地玄学，所以大都采取了直接打代码的方式，虽然慢了点不过确实很稳健。最好的方法当然是两者的结合，不过这大概需要经验的积累，才能清楚什么操作在哪个界面比较方便。

至少，上面我说的那些在 design 界面的拖拽和属性设置功能可以帮助自动生成大量繁琐的代码。

下表是本次基本 UI 实现所需要的控件。

效果	控件
标题	TextView
文本输入框	EditText
图片	ImageView
单选框	RadioGroup
按钮	Button

图片文件我这次实验里我放在 mipmap 下，当然这与放在 drawable 下是有不同的，分辨率就是一个问题，不过中大图标本身就那么点大。如果只是小图标的话建议放在 mipmap 下，mipmap 新建图片时会自动生成适合图标使用的大小并提供五种分辨率，drawable 则是直接导

入图片，使用情况有所不同。

EditText 中学号和密码分别需要选为 `number` 和 `textPassword` 类型，效果是只能输入数字和密码显示为实心圆点。关于输入框，可以使用

```
android:ems="10"
```

设置 `TextView` 或者 `EditText` 的宽度为 10 个字符的宽度。当设置该属性后，控件显示的长度就为 10 个字符的长度，超出的部分将不显示。

单选按钮需要建立一个 `RadioGroup`，一个 `Group` 中可以包含若干个 `RadioButton`，可以使用语句：

```
android:orientation="horizontal"
```

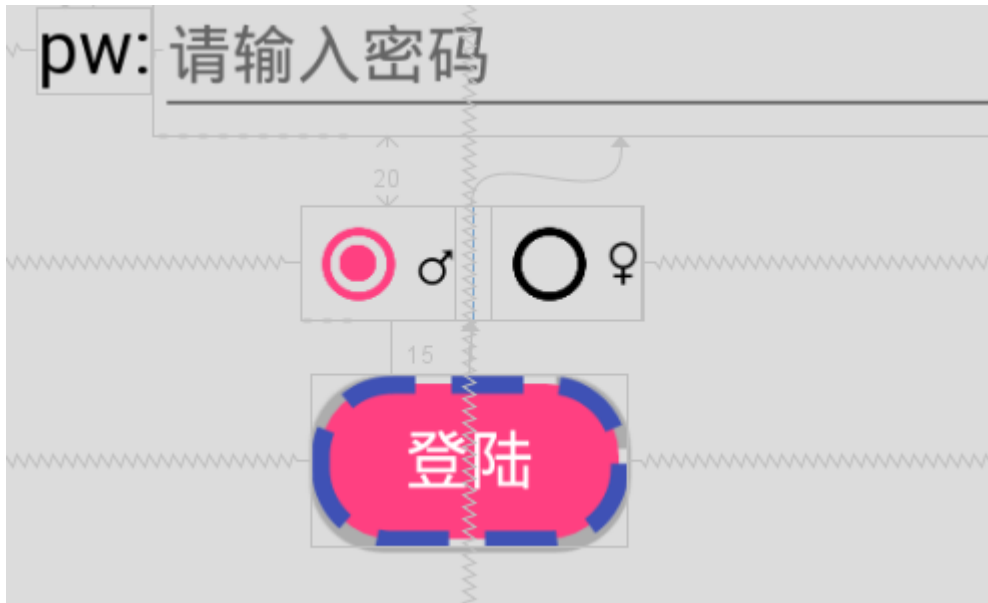
来决定各个 `RadioButton` 是水平排列还是垂直排列。

整个 `Group` 的效果在容器中体现为只能选择一个选项。很方便的一点是要改变这一套选择按钮在布局中的位置只需要调整 `Group` 的位置即可。

`Button` 的属性设置建议在 `Properties` 窗口下进行，包括 `padding` 的设置，背景的导入，都十分方便。这里矩形框可能看不出来 `padding` 的作用，实际上 `padding` 可以定位按钮中文本的位置，比如做了一个奇形怪状的 `button`，就需要为文本挑一个顺眼的位置放着。

Properties			🔍 ↺ ⚙️ →
☆	id	button2	
	layout_width	wrap_content	
	layout_height	wrap_content	
	▶ Constraints		
	▶ Layout_Margin	[?, 10dp, 20dp, 80dp, ?]	
	▼ Padding	[?, 10dp, 5dp, 10dp, 5dp]	
	padding		
	paddingBottom	5dp	
	paddingLeft	10dp	
	paddingRight	10dp	
	paddingTop	5dp	
	paddingEnd		
	paddingStart		
	▶ Theme		
	elevation	1dp	
	background	@drawable/button_frame	
	text	注册	
	textColor	<input type="checkbox"/> @android:color/background_light	
	textSize	18sp	
	accessibilityLiveRegion		
	accessibilityTraversalAfter		
	accessibilityTraversalBefore		
	allowUndo	<input type="checkbox"/>	

关于自定义 Button 的背景边框，当需要将一个 button 设置为圆角矩形时，光设置 button 的属性是达不到效果的，需要定义一个背景边框来达到这个效果。



在 drawable 文件夹下新建一个 Drawable resource file，填写 file name，然后把自动生成的 selector 标签改为 shape，具体怎么使用参见这篇教程：

<http://blog.csdn.net/sysukehan/article/details/52022307>

【实验结果】

安卓手机展示：



Lab01

中山大学学生信息系统



学号:

密码:

☒ 学生 ☐ 教职工

登陆

注册



【心得体会】

1. 关于 **Start/End** 和 **Left/Right**

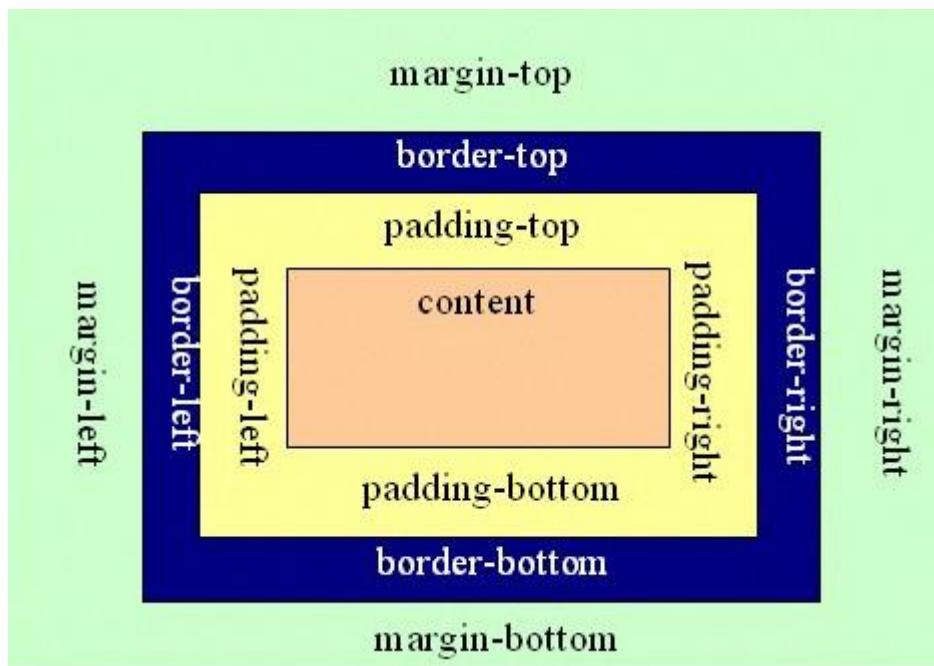
一般情况下，View 开始部分就是左边，但是有的语言目前为止还是按照从右往左的顺序来书写的，例如阿拉伯语，在 Android 4.2 系统之后，Google 在 Android 中引入了 RTL 布局，更好的支持了由右到左文字布局的显示，为了更好的兼容 RTL 布局，google 推荐使用 **MarginStart** 和 **MarginEnd** 来替代 **MarginLeft** 和 **MarginRight**，这样应用可以在正常的屏幕和由右到左显示文字的屏幕上都保持一致的用户体验。

参考 <http://blog.csdn.net/zhufuing/article/details/40181815>

2. 关于 **padding** 和 **magin**

概念：**padding** 是站在父 view 的角度描述问题，它规定它里面的内容必须与这个父 view 边界的距离。**margin** 则是站在自己的角度描述问题，规定自己和其他（上下左右）的 view 之间的距离。

一张图可以看懂：



3. 控件居中

- ① 如果是水平线上就一个 View,则将 View 与 parent 两边建立左右约束就会自动居中，不放心的话可以右键单击 View，选择 Central Horizontally (in parent)，体现在代码里就是

```
app:layout_constraintHorizontal_bias="0.5"
```

0.5 指的是居中，是一个比例值，试着改成 0.1 或 0.8，View 就会偏左或偏右。如下：一个 0.3 的偏移



- ②利用 guideline

```
<android.support.constraint.Guideline
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/myguide"
    android:orientation="vertical"
```

```
app:layout_constraintGuide_begin="22dp"  
/>
```

可以在距 layout 左边界 22dp 处画一条竖直的 guideline

③多个 View 并列，可以建立链式连接：

通过 `app:layout_constraintHorizontal/Vertical_chainStyle` 设置链接的样式。

其中，一种可自动居中的链接为 `packedchain`，将 `chainStyle` 设置为 “packed”，关于 `packedchain` 的定义如下：

```
CHAIN_PACKED -- the elements of the chain will be packed together. The  
horizontal or vertical bias attribute of the child will then affect the positioning of the  
packed elements
```

可见设置 `bias` 可以实现居中。

4. 小结

Constraint Layout 这种布局方式比起相对布局和线性布局，灵活度更高。这次的 UI 完成的过程也比较顺利，感觉安卓应用开发没有想象中那么枯燥，在看到手机上展示出自己界面的时候也挺有成就感的。所以做完之后也着手查资料打算学习一下按钮点击的实现等，毕竟挺想做出一个自己的 app 的。