



合肥工业大学

计算机与信息学院

实验报告

课 程：移动计算及应用开发

指导老师：唐益明

姓 名：孙辉

学 号：2018211958

专业班级：计算机科学与技术 18-2 班

日 期：12 月 17 日

目录

实验一 Android 开发平台搭建和第一个 Android 程序	3
一、实验内容	3
二、实验过程	3
三、关键程序代码	12
四、实验心得	12
实验二 仿 QQ 客户端登陆界面	14
一、实验内容	14
二、实验过程	14
三、关键程序代码	22
四、实验心得	26
实验三 简单的 Android 游戏程序	27
一、实验内容	27
二、实验过程	27
三、关键程序代码	32
四、实验心得	37
实验四 基于 Android 的综合应用程序	38
一、实验内容	38
二、实验过程	38
三、关键程序代码	45
四、实验心得	56

实验一 Android 开发平台搭建和第一个 Android 程序

一、实验内容

- 1) 在自己的电脑上搭建出 Android 开发平台，记录中间遇到的问题
和解决方法。
- 2) 新建一个项目 First（选择 Empty Activity），修改显示内容（比如
用 Hello, this is mine.），并连接模拟器或手机，得到相应的结果。
- 3) 会进行 Android 模拟器或硬件设备的基本操作，包括修改语言、
修改时间、更换桌面等。

二、实验过程

- 1) 在自己的电脑上搭建出 Android 开发平台，记录中间遇到的问题
和解决方法。

1. 自动配置 SDK 时出错

搭建 android studio 时遇到的第一个问题是在自动配置 SDK 时发生了
报错如下：

```
Preparing "Install Android Emulator (revision: 30.2.6)".
Downloading https://dl.google.com/android/repository/emulator-windows-6962233.zip
javax.net.ssl.SSLException: SSL peer shut down incorrectly
Warning: An error occurred while preparing SDK package Android Emulator: SSL peer shut down
incorrectly.
"Install Android Emulator (revision: 30.2.6)" failed.
Preparing "Install Google APIs Intel x86 Atom System Image (revision: 9)".
Downloading https://dl.google.com/android/repository/sys-img/google_apis/x86-30_r09.zip
javax.net.ssl.SSLException: SSL peer shut down incorrectly
Warning: An error occurred while preparing SDK package Google APIs Intel x86 Atom System
Image: SSL peer shut down incorrectly.
"Install Google APIs Intel x86 Atom System Image (revision: 9)" failed.
javax.net.ssl.SSLException: SSL peer shut down incorrectly
Warning: An error occurred while preparing SDK package Android Emulator: SSL peer shut down
```

于是我根据报错的提示，在 chrome 下载了两个压缩包如下：

然后手动配置一下位置，然后关闭 Android Studio 再次打开自动进行 SDK 配置检查的时候就提示文件已经存在，不再报错，此问题就此解决。

2. 大文件下载失败

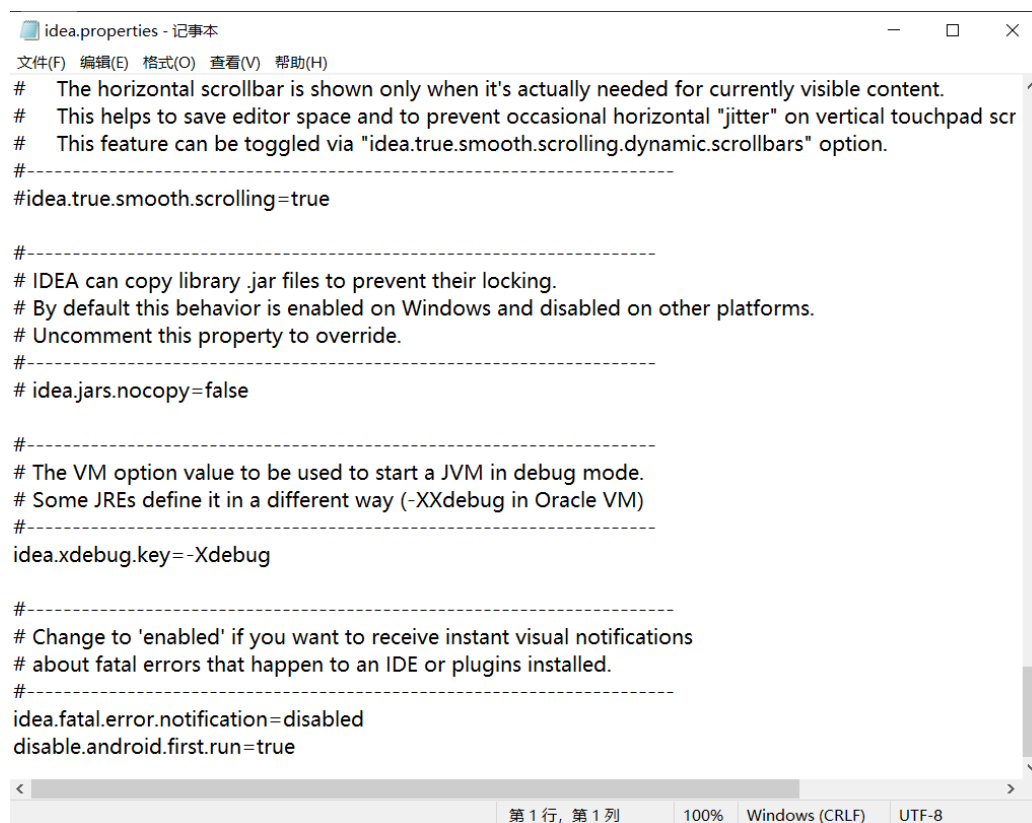
由于之前安装过 java 配置过 JDK，而且科学上网了，所以这次安装基本上没遇到什么大问题，只有些很大文件下载时遇到了下载失败的

 emulator-windows-6962233.zip	2020/12/5 22:15	WinRAR ZIP 压缩...	252,507 KB
 x86-30_r09.zip	2020/12/5 22:15	WinRAR ZIP 压缩...	1,209,280...

问题，只需要记下链接逐个单独下载就行了。

3. 取消默认每次开启 Android Studio 自动检查更新

由于不想每次启动时都更新检查 SDK，所以我们要改一下 Android Studio 的配置文件，打开 Android Studio 安装路径下的 bin 文件夹找



```
idea.properties - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
# The horizontal scrollbar is shown only when it's actually needed for currently visible content.
# This helps to save editor space and to prevent occasional horizontal "jitter" on vertical touchpad scr
# This feature can be toggled via "idea.true.smooth.scrolling.dynamic.scrollbars" option.
#-----
#idea.true.smooth.scrolling=true

#-----
# IDEA can copy library .jar files to prevent their locking.
# By default this behavior is enabled on Windows and disabled on other platforms.
# Uncomment this property to override.
#-----
# idea.jars.nocopy=false

#-----
# The VM option value to be used to start a JVM in debug mode.
# Some JREs define it in a different way (-XXdebug in Oracle VM)
#-----
idea.xdebug.key=-Xdebug

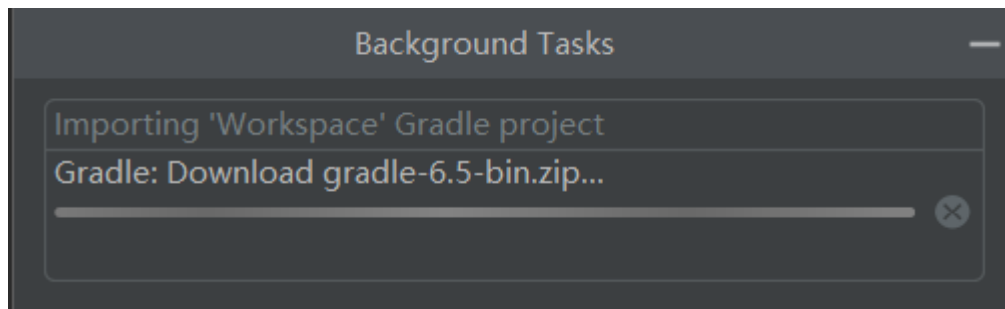
#-----
# Change to 'enabled' if you want to receive instant visual notifications
# about fatal errors that happen to an IDE or plugins installed.
#-----
idea.fatal.error.notification=disabled
disable.android.first.run=true

第 1 行, 第 1 列    100%    Windows (CRLF)    UTF-8
```

到 `idea.properties` 文件，在最后添加代码 `disable.android.first.run=true` 如下，此问题就此解决。

4. 第一次配置 gradle 出错

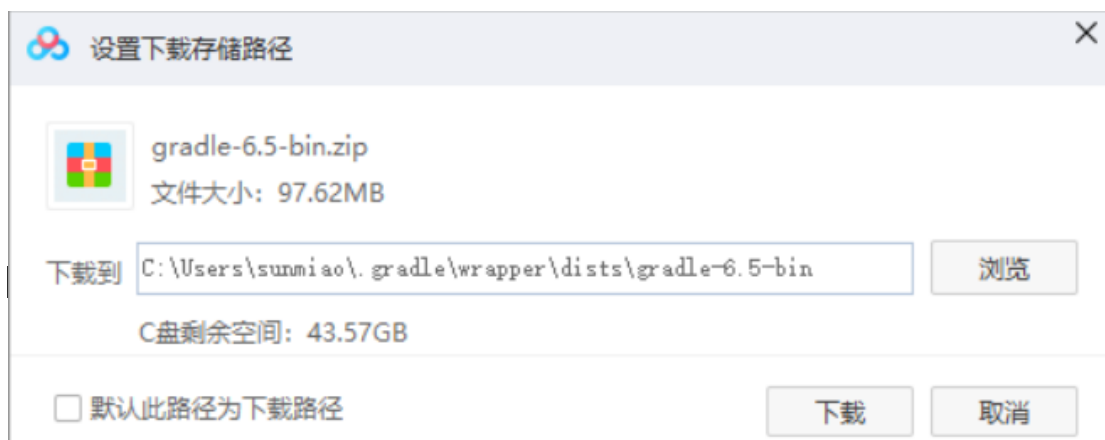
第一次编译运行时需要配置 gradle，如果（外）网络不佳，下载会一直卡在这个地方



最后还会崩掉，报错如下

```
Could not install Gradle distribution from 'https://services.gradle.org/distributions/gradle-6.5-bin.zip'.
```

因为下载很慢还会崩，所以我在网上自己找到了需要的文件并且下载



下载好后放入 Android Studio 的对应路径中，至此就解决了安装过程中遇到的所有的问题了，其他的步骤没有遇到问题，只需要按照步骤下一步就行了

5. 对 Android Studio 的上手

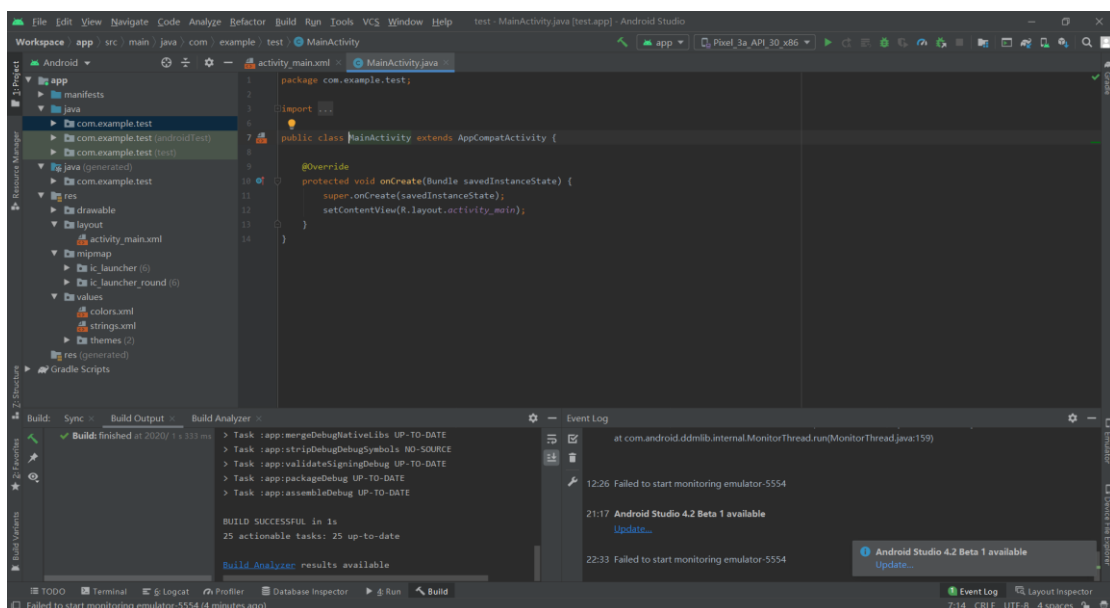
由于在实验室写的软件经常用 matlab 里面的 app designer 的 GUI 设计，pycharm 和这个功能也很像，所以我对 Android Studio 的上手很

快，我一直是坚持需求推动学习的观念，所以我选择看一些简单的帮助文档，对列表栏里面的文件有个大致的功能了解，然后下面就能进行实验了，总结如下：

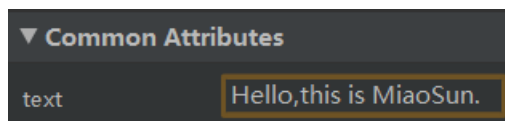
build:构建目录,相当于Eclipse的bin目录
libs:依赖的包
src:
 androidTest:安卓单元测试的目录
 main
 Java: 写Java代码的地方
 res:资源文件
 drawable:图像资源
 layout:布局资源
 menu:菜单资源
 value:
 dimens:css配置文件
 string:字符串资源
 styles:style资源
AndroidManifest.xml:配置文件
build.gradle:Gradle构建脚本

2) 新建一个项目 test（选择 Empty Activity），修改显示内容（比如用 Hello, this is MiaoSun（孙淼）），并连接模拟器或手机，得到相应的结果。

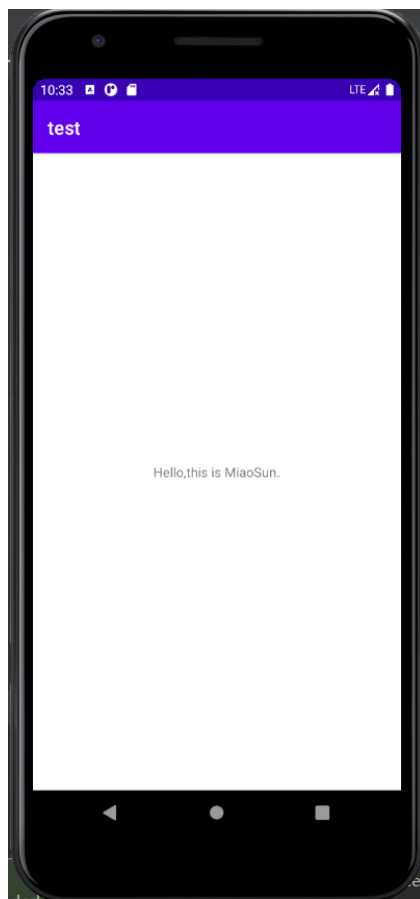
通过网上检索的资料我知道 activity_main.xml, layout 就是布局，所以 activity_main.xml 其实就是一个布局文件。何为布局文件：我们拖



的各种各样的控件在上面，系统就会自动帮我们生成 xml 代码。那能怎么用，拖在上面又不能点，也只能看？如何建立联系？这时候就是 MainActivity.java 要干的事情了。我对他们俩之间关系的理解就是界面以及控件回调函数和对应代码的关系吧，最简单的办法实现就行了，我是直接在 activity_main.xml 里面对文本进行修改的，很轻易的就实现了。



然后运行模拟器，可以看到结果，成功。



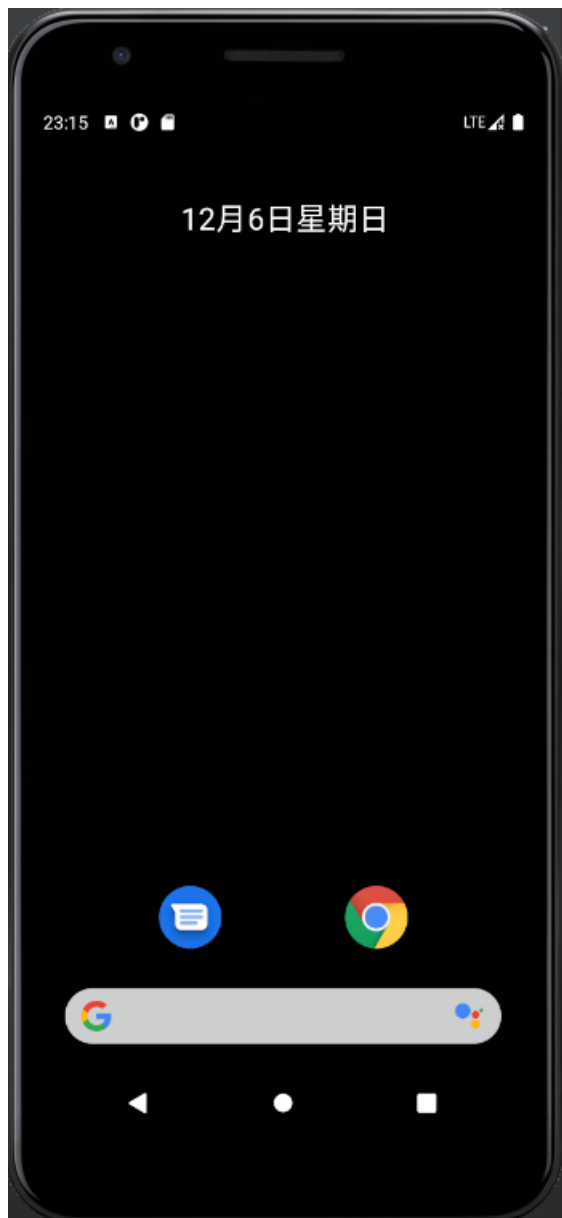
3) 会进行 Android 模拟器或硬件设备的基本操作，包括修改语言、修改时间、更换桌面等。

修改语言：配置语言，然后把简体语言拖动到最上排，此时系统默认的语言更换为简体中文，如下图左侧所示。



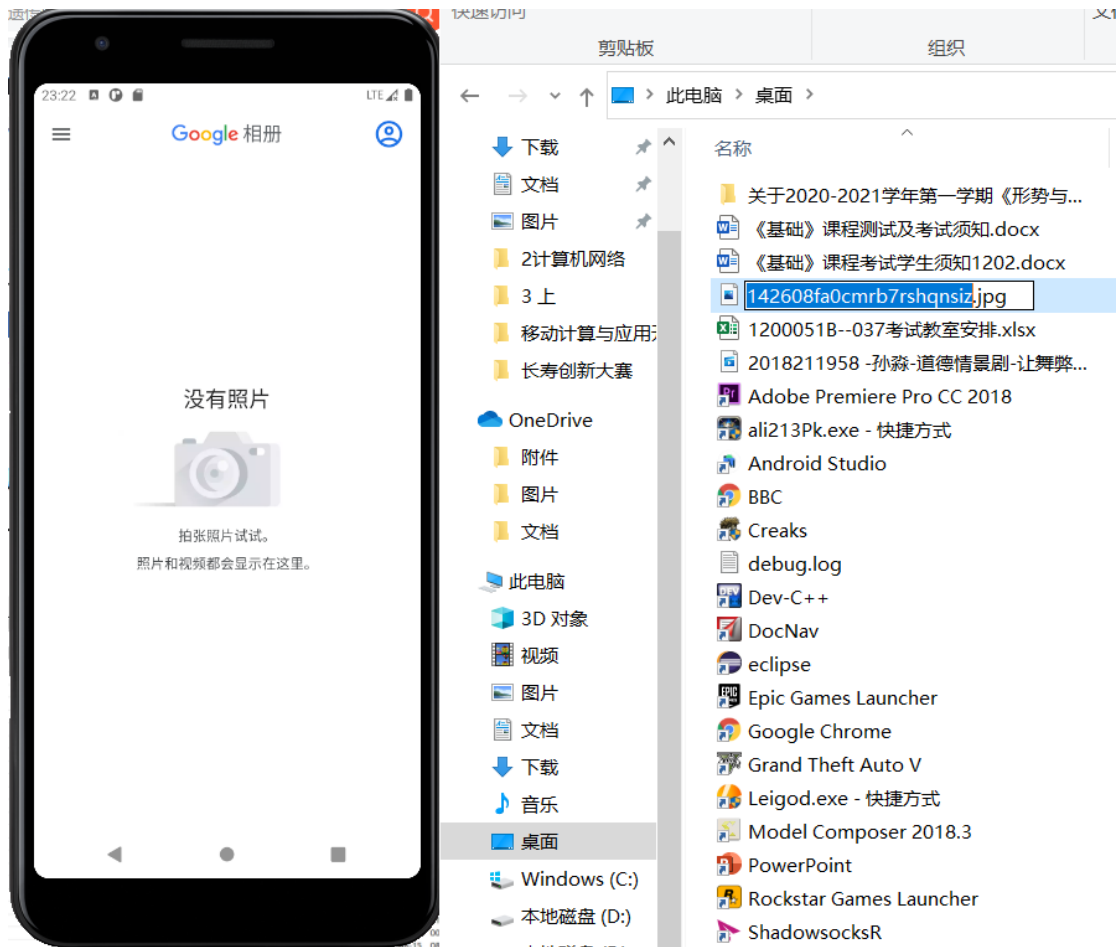
修改时间：Android 模拟器启动后，显示的时间与系统当前时间并不相同，这主要是因为模拟器的时区与系统的不同，我把时区的自动按钮取消，然后选择中国上海作为我们的时区，结果如上图右侧所示，可以看到，和我电脑的时间一致了，都是 22:54。

更改桌面壁纸：我这里遇到了问题，壁纸加载不出来，不能预览，如果就直接选择设置，那么桌面就是全黑色（因为加载不出）



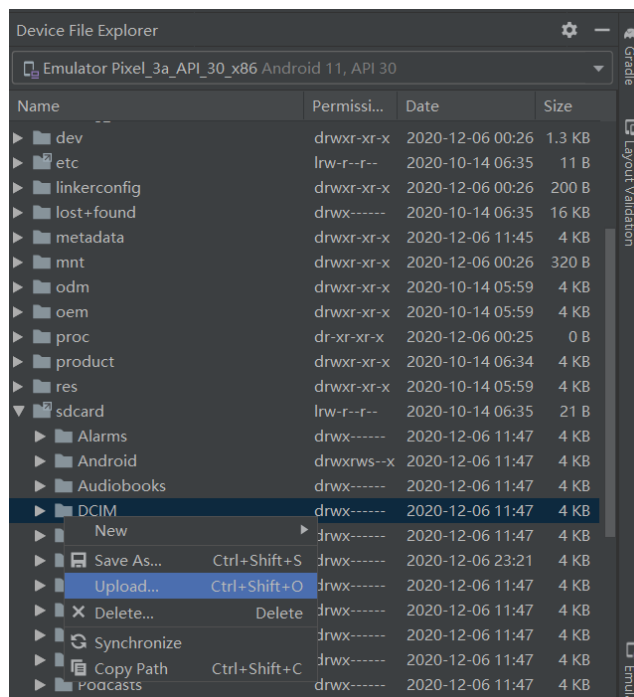
由于不知道是什么原因导致，我就尝试往模拟器图库里导入新的图片试试，在网上检索了张和该 Pixel 分辨率（1080*2220）一致的壁纸，网上提到通过 DDMS 很轻松的把图片拖进去就是了，直接拖动导入图库，然后重启模拟器发现并不行，后来看到这种方式在 Android 3.0

以前可行，在此版本之后，DDMS 已经被 Device File Explore 取而代

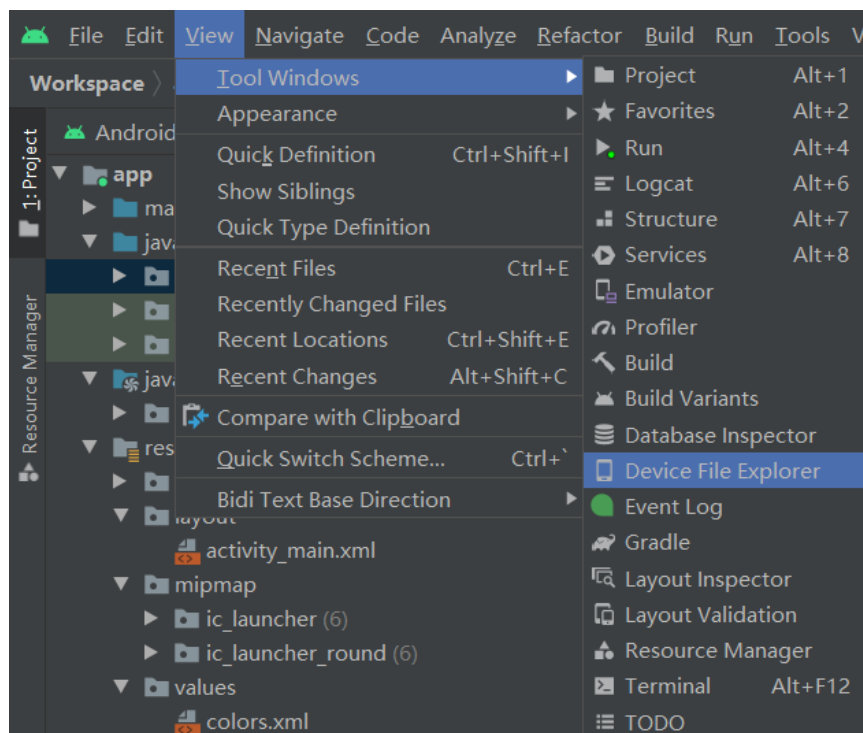
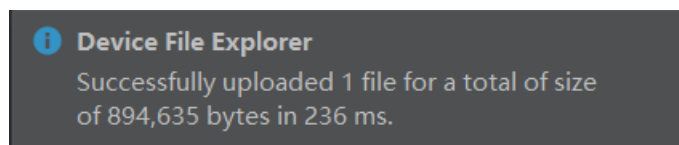


之，我们需要通过 DFE 来导入图片。

我们打开 DFE, 然后选择我们对应的模拟器, 按照下图所示, upload。

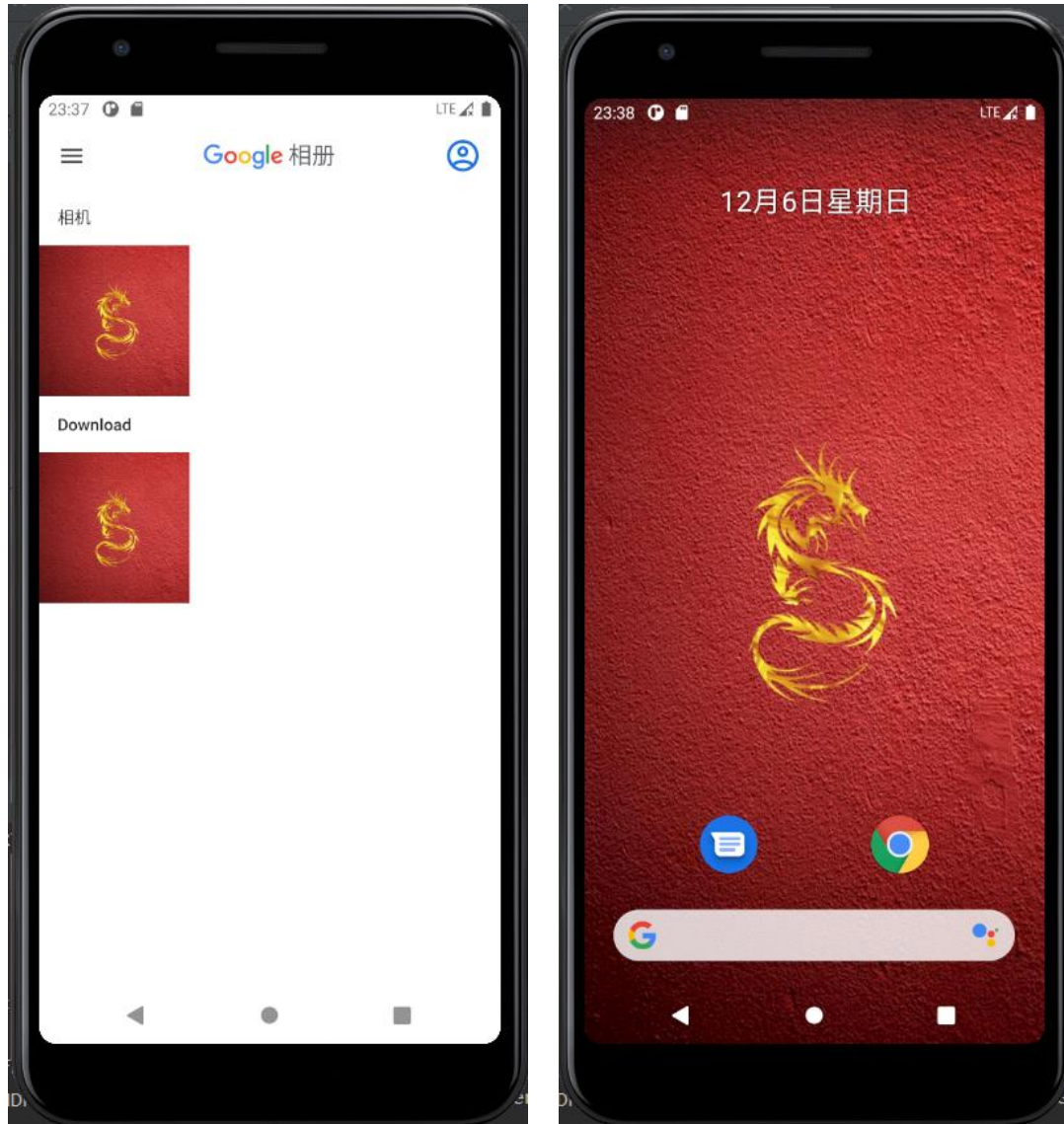


导入我们选好的图片, 提示成功导入



然后我们重启模拟器, 可以看到图片已经成功导入了

此时我们直接选择导入图库图片而不是系统自带的壁纸，设置成功，由此我猜测应该是系统自带的壁纸不能加载或者说并没有内置好的原因



三、关键程序代码

实验一没有涉及代码

四、实验心得

作为一名计算机专业的学生，我自己安装，配置的软件应该不下

50 种了，Google 旗下的 Android Studio 的安装和配置过程实在算不上复杂，以前接触过很多其他的集成开发环境都比这个要复杂、费神许多，这也间接说明了 Android Studio 的开发其实从配置环境开始就对程序员是友好的。

这次实验让我又多了一项技能，学好这门课，也是为当下我正在着手的一个需要 android 开发的项目入个门，积累技术，同时也让我将来的工作方向多了一个选择，所以我很满意自己选择了这门课。

实验二 仿 QQ 客户端登陆界面

一、实验内容

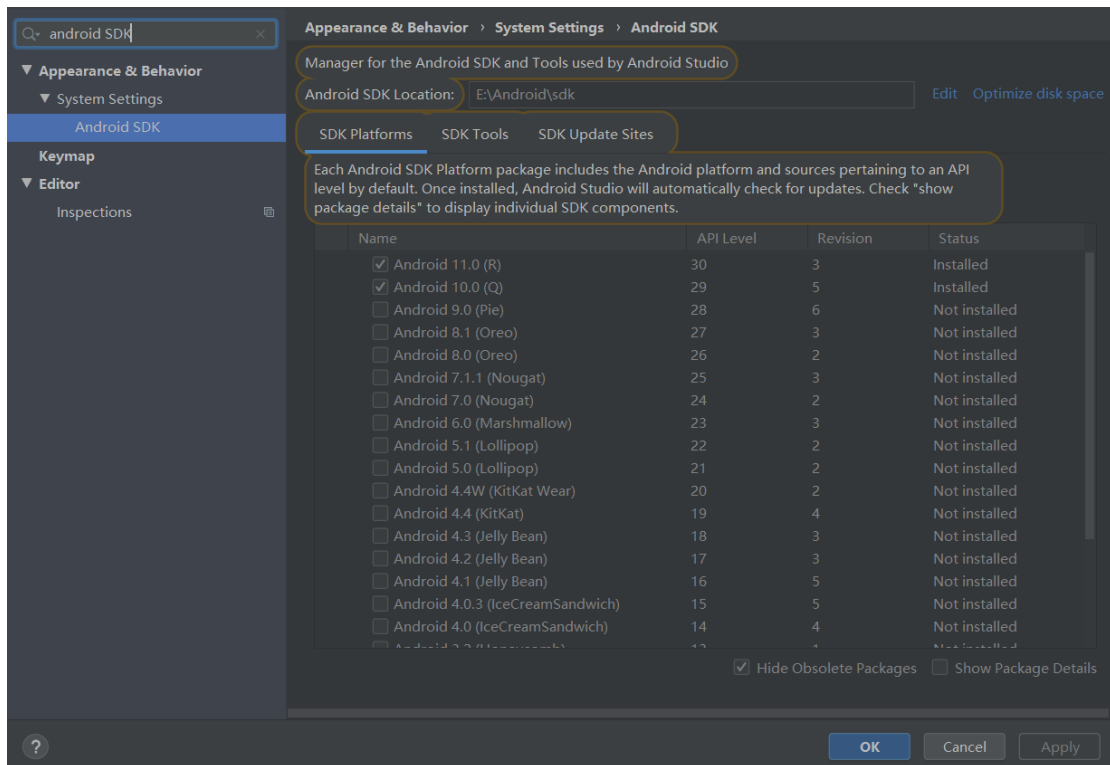
- 1) 模仿以下 QQ 登陆界面，编写布局程序。
- 2) 输入和密码部分必须要有。
- 3) 尽可能多一些细节，越接近越好。

二、实验过程

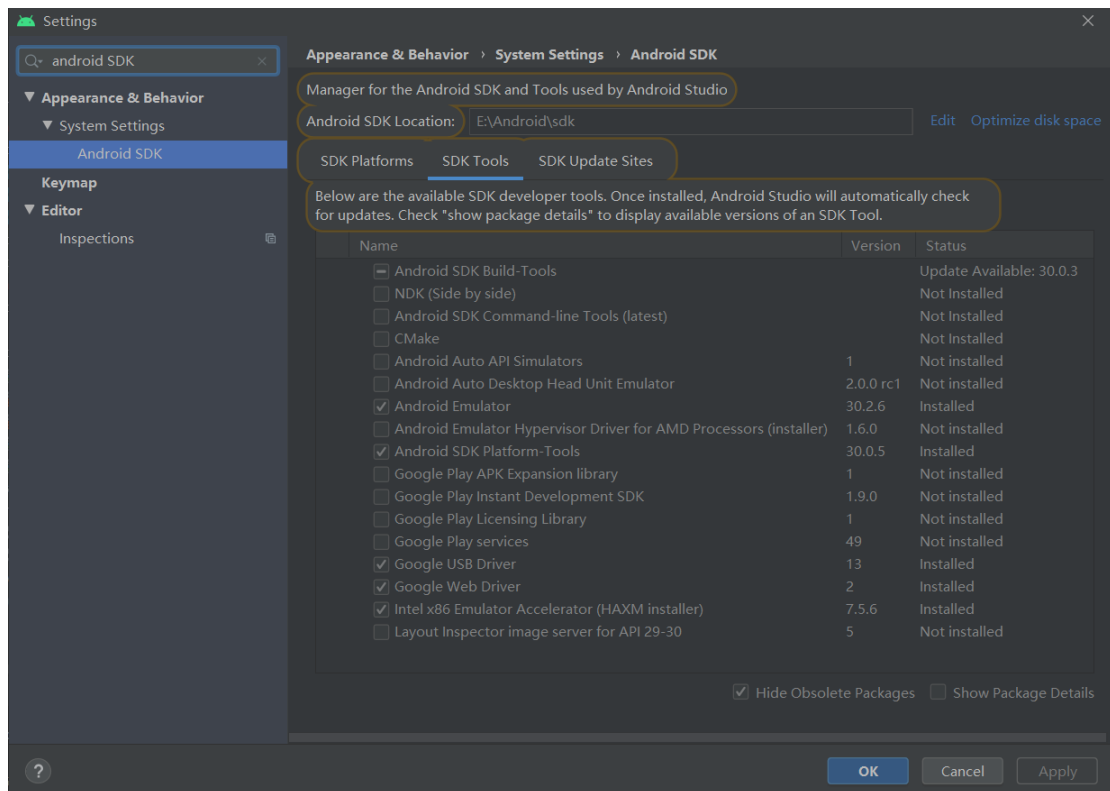
实验进行到这里的时候我已经发现使用 android studio 的模拟器会很卡，为了提高实验进行的效率，我选择使用我的安卓手机进行调试运行，可以看到我的华为 P20 的安卓版本是 10。



所以我在 android studio 里面下载了对应的 android 10 版本，如下



然后在 SDK Tools 里面下载了 Google USB Driver，这些都是真机调试开发的必要准备



最后我们来到手机端，连续点击五下版本号进入开发者模式，并且打

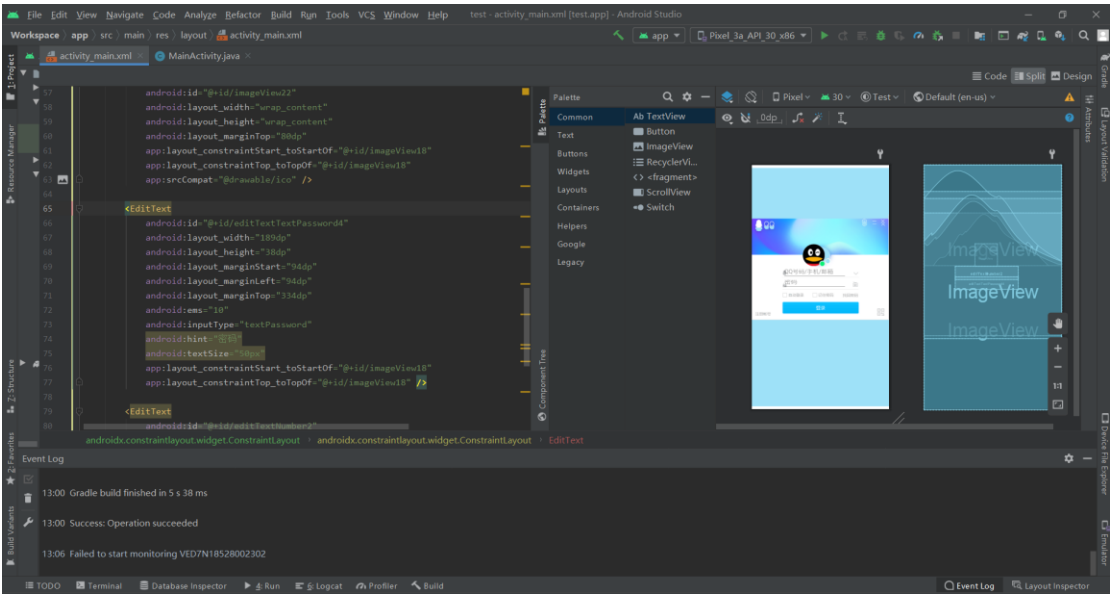
开 USB 调试，至此预设置已经完成，可以开始在安卓手机端进行调试运行了。

















因为之前有做过项目，有设计 GUI 的经验，所以看到实验要求，我就意识到无论是在什么平台（Android Studio 我也是第一次使用），设计这种精细的 GUI 仅仅用代码是很慢的，因为运行和比较现实情况与理想情况的过程会很慢，要是综合 GUI 可视化和代码就很快了，也就是说先通过可视化 GUI 设计出大概的界面，再通过代码做一些精细的调整，那么不仅效率很高，界面也会很逼近目标界面。

令人高兴的是，Android Studio 也有可视化 GUI 设计界面，虽然因为安卓机型多样的原因，这个功能没有 matlab 的 app designer 那么方便，

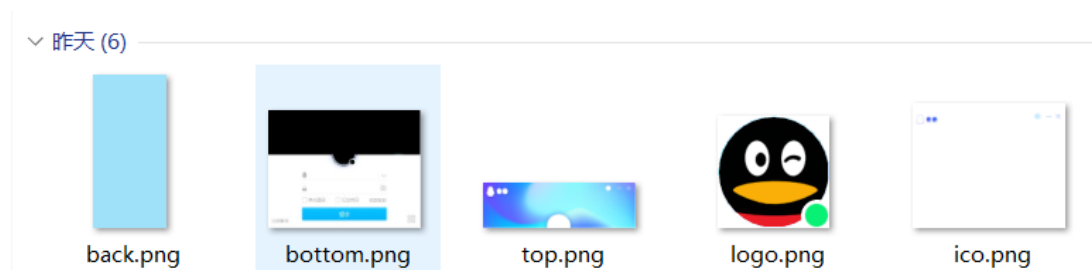
但是也比没有好。



为了达到实验要求中“越接近越好”的要求，我在电脑上直接搜索 qq ico，得到的路径下基本没有登录界面的 ico

	qq.ico	类型: ICO 文件 分辨率: 32 x 32	大小: 31.7 KB
	qunmap-center-icon.png	类型: PNG 文件 分辨率: 30 x 30	大小: 1.21 KB
	icons-sad0f8b42ab.png	类型: PNG 文件 分辨率: 18 x 36	大小: 841 字节
	icons-sb25ee41e81.png	类型: PNG 文件 分辨率: 160 x 580	大小: 49.5 KB
	icon-face-a1d03c.png	类型: PNG 文件 分辨率: 16 x 16	大小: 1.17 KB
	favicon.ico	类型: ICO 文件 分辨率: 16 x 16	大小: 2.79 KB
	icon-face-a1d03c.png	类型: PNG 文件 分辨率: 16 x 16	大小: 1.17 KB
	favicon.ico	类型: ICO 文件 分辨率: 16 x 16	大小: 2.79 KB
	qunmap-center-icon.png	类型: PNG 文件 分辨率: 30 x 30	大小: 1.21 KB
	icons-sad0f8b42ab.png	类型: PNG 文件 分辨率: 18 x 36	大小: 841 字节
	icons-sb25ee41e81.png	类型: PNG 文件 分辨率: 160 x 580	大小: 49.5 KB
	side_icons.png	类型: PNG 文件 分辨率: 18 x 101	大小: 1.03 KB
	icon_all.png	类型: PNG 文件 分辨率: 17 x 40	大小: 578 字节
	ci_icon.gif	类型: GIF 文件 分辨率: 13 x 13	大小: 561 字节

于是我只能到网上检索，检索到了部分 icon 等模块，然后再使用 ps 提取出了 qq 界面上部分模块，最后进行代码和可视化编辑综合起来布局，得到了很逼近的界面，提取的各个界面模块如下



得到界面模块后，除了部分精修的地方（后面会介绍），都可以在 xml 里面实现，具体代码会贴在附录里面，我使用的是最适合这个界面设计的约束布局 `ConstraintLayout`，先用可视化 GUI 设计的方法在 Design 面板上拖动出大概，然后连接实机进行编译运行，根据实际情况对位置进行微调，调整的代码部分主要是每个控件的如下部分

对位置微调的代码

```
android:layout_width="67dp"
android:layout_height="67dp"
android:layout_marginStart="152dp"
android:layout_marginLeft="152dp"
android:layout_marginTop="236dp"
```

根据 api 注释，我们得知 `width` 为该控件的宽，`height` 为该控件的高，`marginStart` 指的是控件距离开头 View 部分的间距大小，`marginLeft` 则指的是控件距离左边 View 部分的间距大小，`marginEnd` 和 `marginRight` 同理。

资料还提到，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，更好了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。

此外，为了保证“越接近越好”的实验要求，我还对账号文本框和密码框加入了 `hint`（即初始默认显示“QQ 号码/手机/邮箱”和“密码”）。

但是直接使用 `hint` 会遇到默认字体太大，有部分被账号栏遮挡的问题，为了解决，我检索了设置 `hint` 字体大小而不影响输入的字体大小的方法，原来在 `Android` 的布局 `xml` 文件中，并没有直接可以设置 `hint` 字体大小的属性。一般来说，提示文字的大小，不希望跟正常字体的大小一样，如果需要改变 `hint` 的字体大小，就只能通过代码的方式来进行处理，代码第三部分，解决问题部分的代码都对应注释了，为了实现“越接近越好”的实验要求，我进行了多次微调，最终实现了可以说是 99% 的相似度。初始运行如下：

test



QQ号码/手机/邮箱

密码

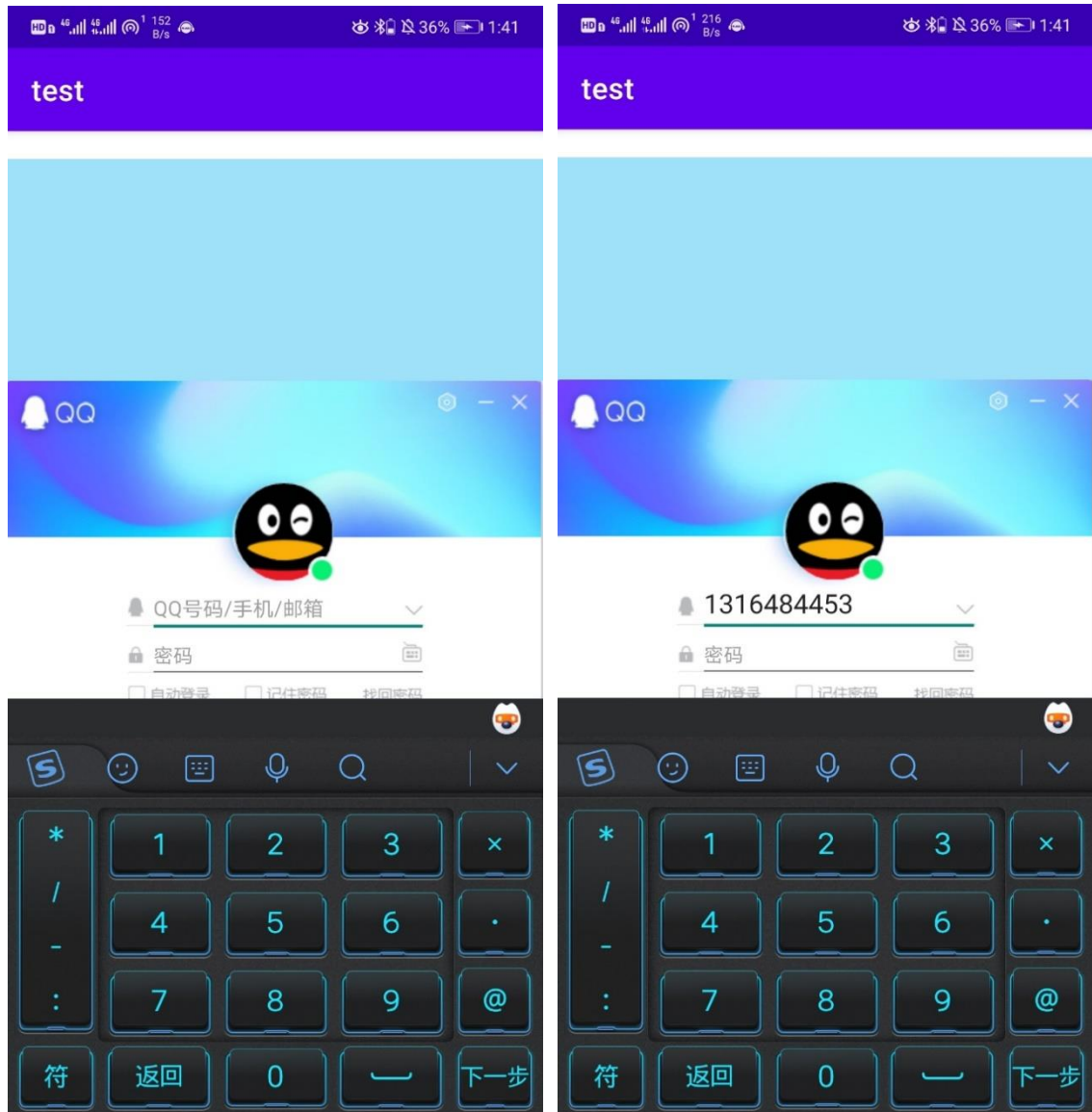
☐ 自动登录 ☐ 记住密码 找回密码

登录

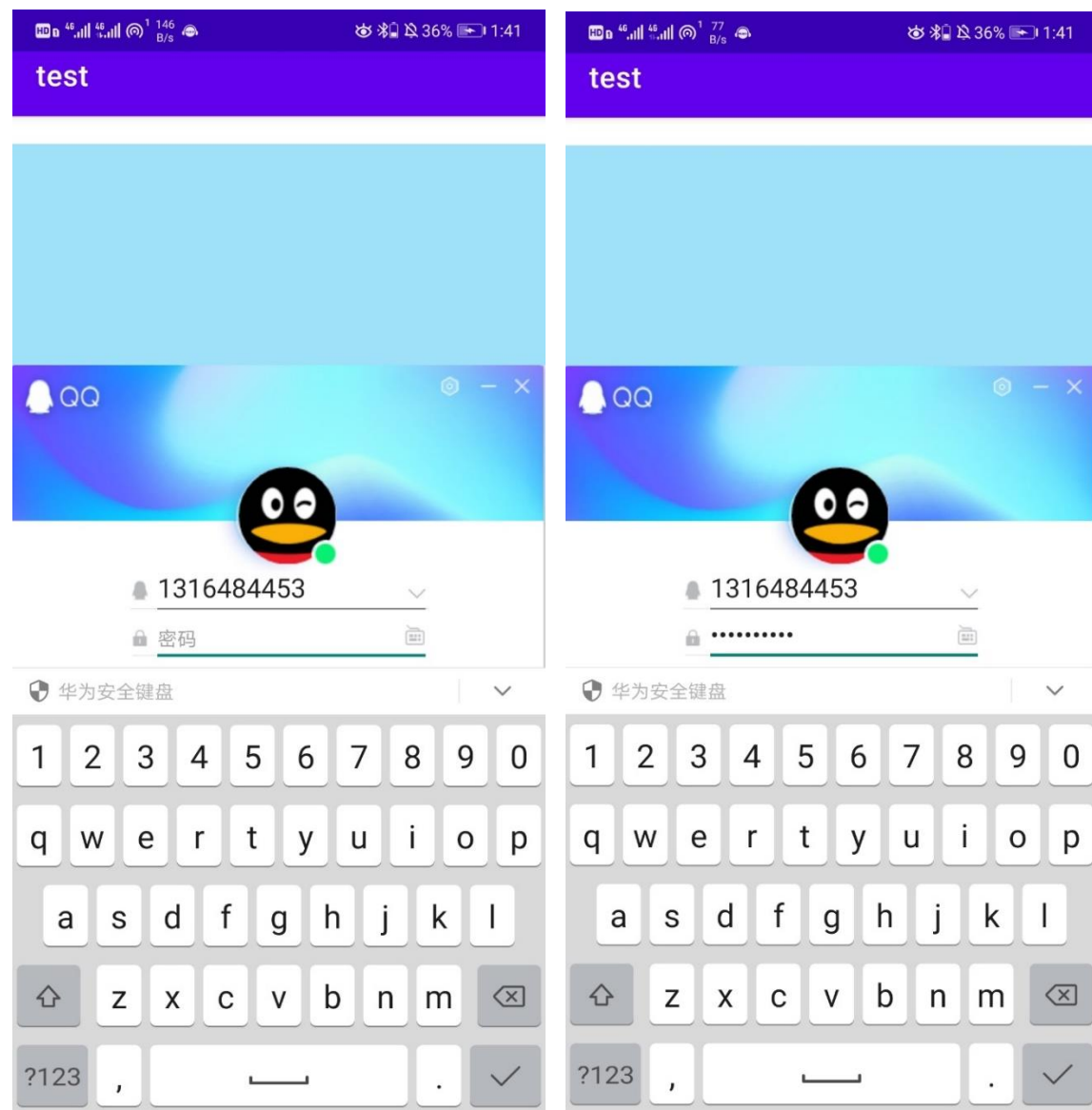
注册帐号



输入账号时的情况如下所示，可以看到输入字符后，账号栏的 hint 消失，密码栏不受影响，且由于我对账号栏的设置是仅数字，所以输入法显示的只能是数字



下面看看密码输入栏情况，如下图所示，可以看到输入字符后，密码栏的 hint 消失，并且输入的密码是不可见的，且由于我对密码栏的设置是所有字符，所以输入法显示的是数字字母符号都可以。



三、关键程序代码

```
activity_main.xml

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
android:overScrollMode="always"
tools:context=".MainActivity">
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteY="213dp">
```

```
<ImageView
```

```
    android:id="@+id/imageView18"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0"
    app:srcCompat="@drawable/back" />
```

```
<ImageView
```

```
    android:id="@+id/imageView19"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="142dp"
    app:layout_constraintStart_toStartOf="@+id/imageView18"
    app:layout_constraintTop_toTopOf="@+id/imageView18"
    app:srcCompat="@drawable/top" />
```

```
<ImageView
```

```
    android:id="@+id/imageView20"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="100dp"
    app:layout_constraintStart_toStartOf="@+id/imageView18"
    app:layout_constraintTop_toTopOf="@+id/imageView18"
    app:srcCompat="@drawable/bottom" />
```

```
<ImageView
```

```
    android:id="@+id/imageView21"
    android:layout_width="67dp"
```

```
android:layout_height="67dp"
android:layout_marginStart="152dp"
android:layout_marginLeft="152dp"
android:layout_marginTop="236dp"
app:layout_constraintStart_toStartOf="@+id/imageView18"
app:layout_constraintTop_toTopOf="@+id/imageView18"
app:srcCompat="@drawable/logo" />
```

<ImageView

```
android:id="@+id/imageView22"
android:layout_width="794dp"
android:layout_height="1335dp"
android:layout_marginTop="80dp"
app:layout_constraintStart_toStartOf="@+id/imageView18"
app:layout_constraintTop_toTopOf="@+id/imageView18"
app:srcCompat="@drawable/ico" />
```

<EditText

```
android:id="@+id/editTextTextPassword4"
android:layout_width="189dp"
android:layout_height="38dp"
android:layout_marginStart="94dp"
android:layout_marginLeft="94dp"
android:layout_marginTop="334dp"
android:ems="10"
android:inputType="textPassword"
android:hint="QQ 号码/手机/邮箱"
android:textSize="50px"
app:layout_constraintStart_toStartOf="@+id/imageView18"
app:layout_constraintTop_toTopOf="@+id/imageView18" />
```

<EditText

```
android:id="@+id/editTextNumber2"
android:layout_width="189dp"
android:layout_height="39dp"
android:layout_marginStart="94dp"
android:layout_marginLeft="94dp"
android:layout_marginTop="302dp"
android:ems="10"
android:inputType="number"
android:hint="密码"
android:textSize="50px"
app:layout_constraintStart_toStartOf="@+id/imageView18"
app:layout_constraintTop_toTopOf="@+id/imageView18" />
```



```
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity_main.java

```
package com.example.test;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.text.Html;
import android.text.SpannableString;
import android.text.Spanned;
import android.text.style.AbsoluteSizeSpan;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

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

        //设置"用户名"提示文字的大小
        EditText etUserName = (EditText)findViewById(R.id.editTextNumber2);
        SpannableString s = new SpannableString("QQ 号码/手机/邮箱");
        AbsoluteSizeSpan textSize = new AbsoluteSizeSpan(14,true);
        s.setSpan(textSize,0,s.length(), Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
        etUserName.setHint(s);

        //设置"密码"提示文字的大小
        EditText etPassword = (EditText)findViewById(R.id.editTextTextPassword4);
        //用 html 的方式来设置这个属性,毕竟都是 String
        etPassword.setHint(Html.fromHtml("<font color=\"\"><small>密码</small></font>"));
        //etPassword.setHint(Html.fromHtml("<font color=\"\"><small><small>密码
</small></small></font>"));与 QQ 界面对比后发现 color 就使用默认值灰色很好
    }
}
```

四、实验心得

在这次实验之前，其实我已经接触过很多需要 GUI 设计的实验和项目，在 Qt, Java 和 matlab 里面都设计过 GUI，我感觉 Android Studio 的界面设计方案是和 matlab 里面的 app designer 有很多相似之处的，都是可以代码和可视化设结合起来来设计界面，这样对程序员是很友好的，不然就需要反复编译和运行代码来观察界面与客户需要的界面之间的差别，费时费力。

在进行实验二的过程中，我逐渐发现，其实 Android Studio 与我更熟悉的 matlab 两者之间有很多地方是可以类比的，仅仅从 GUI 设计的部分来看，Android Studio 的 xml 文件其实就是 matlab 里面的界面代码，Android Studio 的 java 文件其实就是 matlab 里面的控件的回调函数的代码，如此对比一下，我感觉自己对 Android 的界面开发熟悉了很多。

但是，我发现由于 Android 系统的手机有很多机型，对应着各种各样差距很大的分辨率，Android 界面的代码很难做到一劳永逸，也正是因为这个，Android Studio 里面的布局基本上都是相对的而无绝对的。

从上面所说的话，我们不难发现学习是一个触类旁通，举一反三的过程，很多集成开发环境的使用乃至语言之间都有着千丝万缕的联系，我们多学一点，以后在更新换代飞快的计算机领域，学习新知识的时候也不会感到跟不上，因为那个时候我们积累的已经够多了。

实验三 简单的 Android 游戏程序

一、实验内容

- 1) 模仿程序 3.2+3.4，编写游戏程序。
- 2) 有“单击进入游戏”的界面。
- 3) 至少有 2 个可活动的小动物。

二、实验过程

代码中用到了 AlertDialog 类，所以需要导入对应的包，需要注意的是：

```
android.support.v7.app.AlertDialog;
```

```
import android.app.AlertDialog;
```

这两个导入包虽然都含有 AlertDialog，但是 app 的原生的 dialog 会有一个白色扩大边界，而用 v7 包的就不会有扩展边界，导入的时候需要注意根据实验的要求导入，所以我们选择

```
import android.app.AlertDialog;
```

需要注意的是老师的第 23 页 PPT（如下圈红处）的代码有点小瑕疵，继承的应该是 View 而不是 view（与 import android.view.View；对应），还望老师修正。

由于导入的图片背景如果覆盖背景会影响美观，我们需要用 ps 对图

开发自定义的View

第三步，File-New-Java class 创建一个新的类RabbitView2的java类，继承android.view.View类，自动形成如下代码：

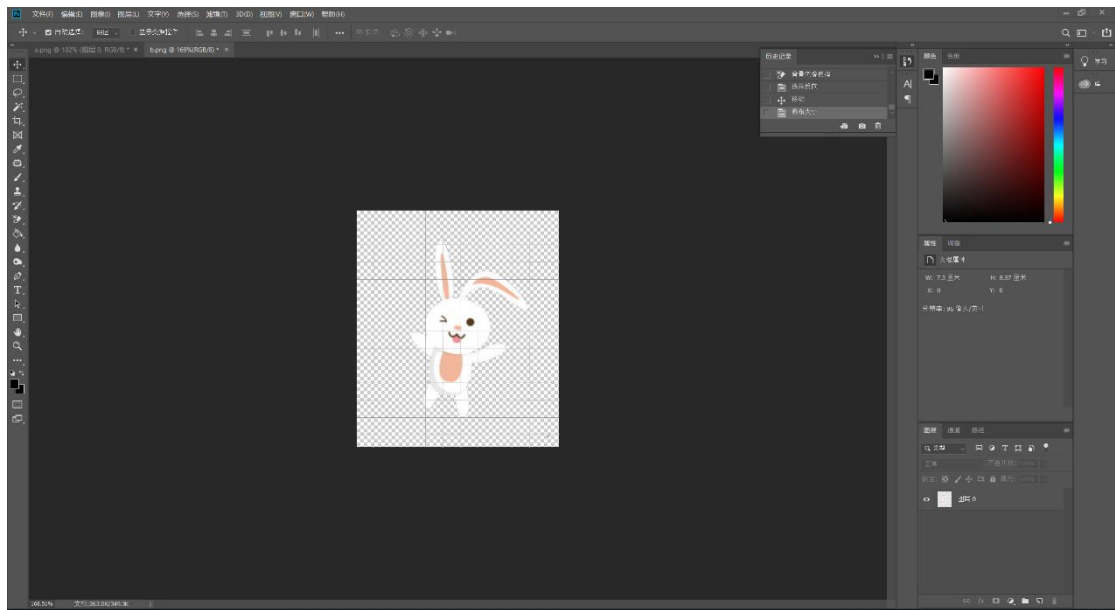
```
package com.example.tang34;  
  
import android.view.View;  
  
public class RabbitView2 extends view{  
}
```

23

片进行背景透明处理



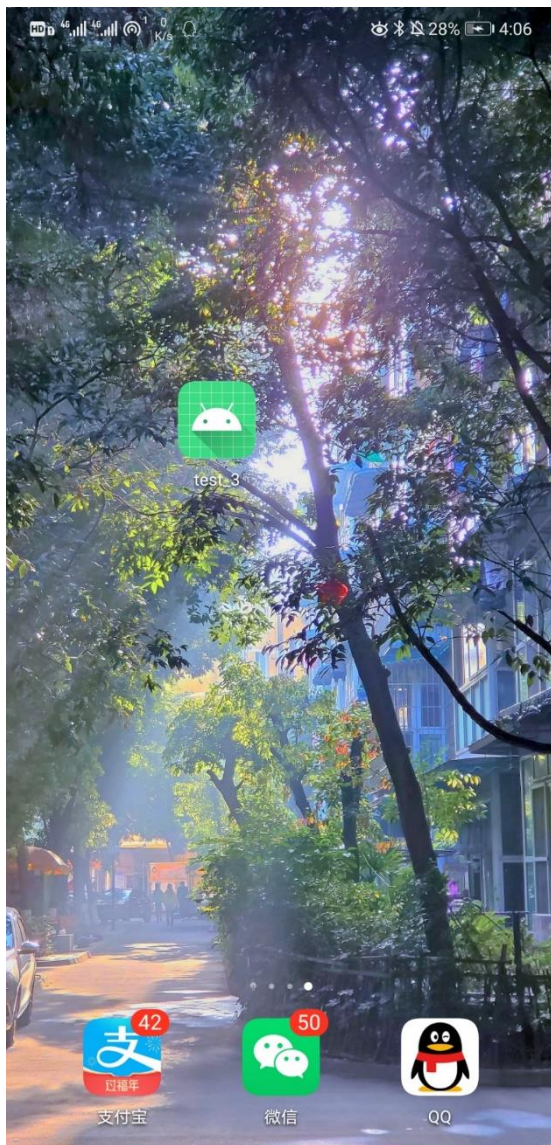
我们使用 PS 对背景进行透明化，过程如下。



实验的要求是至少有 2 个可活动的小动物。那么就有很多种实现的方法，我试着实现了下面两种情况：

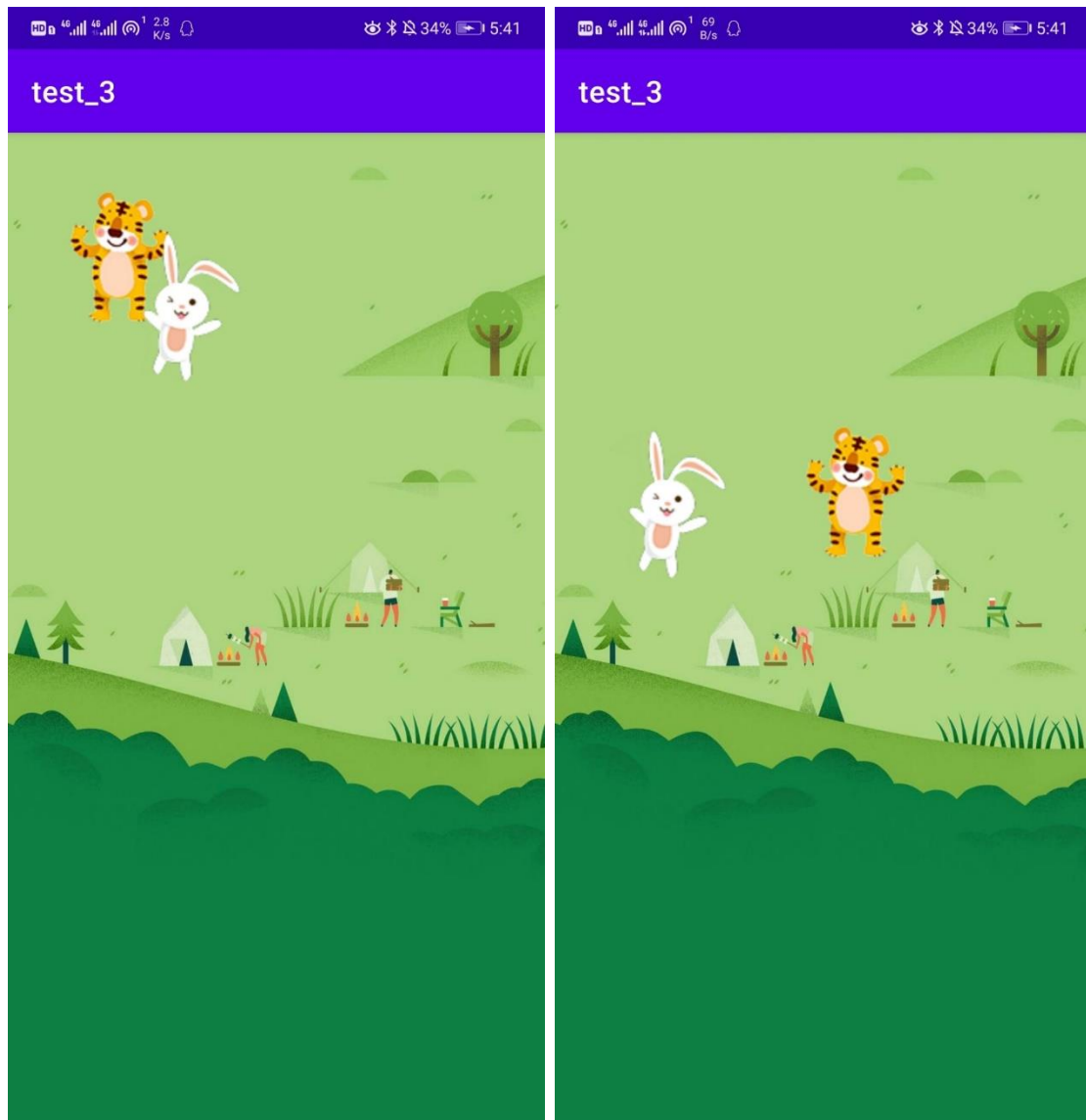
- 1) 两个小动物依次向着点击的位置移动，如：按照兔子，老虎，兔子，老虎...的顺序移动；
- 2) 点击小动物后为选中，然后选中的小动物可以被拖动。

尝试实现第一种情况的时候，会有很高的延迟，可能是我对两个动物的移动顺序实现的算法写的不是很好，所以我最终选择导入到手机上的是第二种情况，软件命名为 test_3，如下。

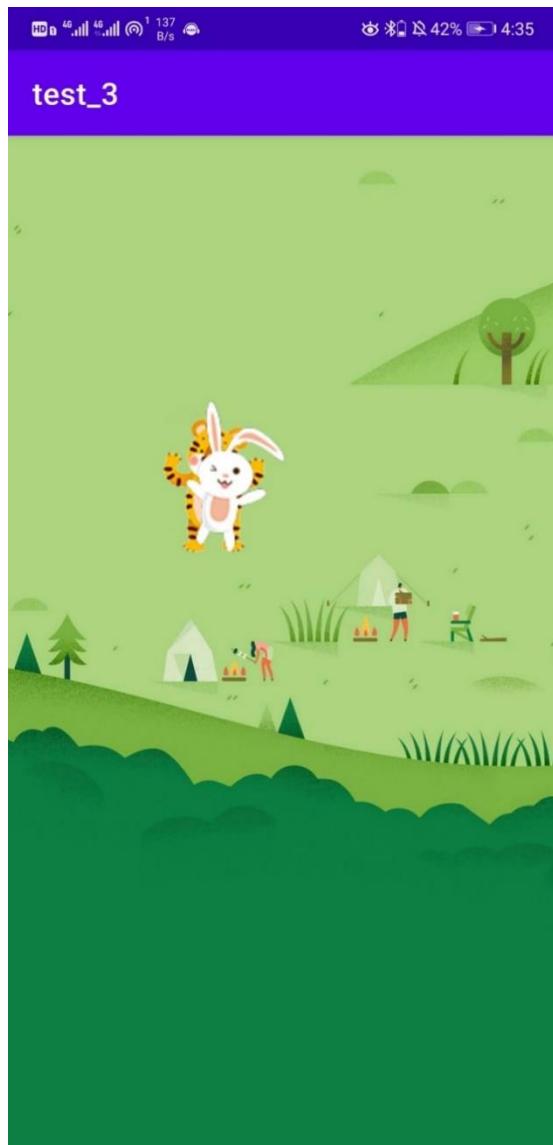


打开软件，进入游戏，我选择设置与老师源码一致的“游戏有风险，进入需谨慎真的要进入吗？”的界面，具体如上图所示。

我们为动物初始设置的位置如下图所示



那么进行随意拖动的结果如下，可以得知两个小动物都可以移动，我是先放入老虎，后放入兔子，所以当两个动物放到一起时，兔子是在老虎上层的，由于是按层传递的判断，所以不会出现有些同学遇到动物放在一起就拆不开的问题。



三、关键程序代码

activity_main.xml

```
activity_main.xml

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/background"
    android:id="@+id/activity_main">
```


</FrameLayout>

MainActivity.java

MainActivity.java

```
package com.example.test_3;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;
import android.widget.FrameLayout;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    private double x,y;
    boolean canmove=false;
    boolean canmove2=false;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        FrameLayout frameLayout = findViewById(R.id.activity_main);
        final String[] array3 = new String[]{"确定","退出"};
        AlertDialog.Builder builder3 = new AlertDialog.Builder(MainActivity.this);
        builder3.setTitle("游戏有风险，进入需谨慎，真的要进入吗？");
        builder3.setSingleChoiceItems(array3, 0, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                if(which==0){
                    Toast.makeText(MainActivity.this,"游戏开始",Toast.LENGTH_SHORT).show();
                }
                else{
                    System.exit(0);
                }
                dialog.dismiss();
            }
        });
        //setCancelable 强制要求点击选项才能返回
        builder3.setCancelable(false).show();
        final RabbitView2 rabbitView = new RabbitView2(this);
        final TigerView2 tigerView = new TigerView2(this);
        tigerView.setOnTouchListener(new View.OnTouchListener() {
```

```

@Override
public boolean onTouch(View v, MotionEvent event) {
    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN: //记录按下位置,若与图片差距
20000, 则向下传递
            if (Math.pow((x - tigerView.bitmapTX), 2) + Math.pow((y -
tigerView.bitmapTY), 2) > 20000) //判定未点击, 直接返回, 并传递下去
            {
                return true;
            }
            canmove2=true; //选中该图片, 移动
            break;
        case MotionEvent.ACTION_MOVE:
            if(canmove2) {
                tigerView.bitmapTX = event.getX();
                tigerView.bitmapTY = event.getY();
                tigerView.invalidate(); //重绘老虎组件
                return true;
            }
            break;
        case MotionEvent.ACTION_UP:
            if (canmove2) //NO
            {
                canmove2=false;
                break;
            }
    }
    return true;
}

});
frameLayout.addView(tigerView);
rabbitView.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN: //记录按下位置,若与图片差距
20000, 则向下传递
                x = event.getX();
                y = event.getY();
                if (Math.pow((x - rabbitView.bitmapRX), 2) + Math.pow((y -
rabbitView.bitmapRY), 2) > 20000) //判定未点击, 直接返回, 并传递下去
                {
                    return false;
                }

```

```

        canmove=true;    //选中该图片，移动
    case MotionEvent.ACTION_MOVE:
        if(canmove) {
            rabbitView.bitmapRX = event.getX();
            rabbitView.bitmapRY = event.getY();
            rabbitView.invalidate();    ///重绘兔子组件
            return true;
        }
        break;
    case MotionEvent.ACTION_UP:    //松开时恢复标志位
        if (canmove)
        {
            canmove=false;
            break;
        }
    }
    return false;
}
});
frameLayout.addView(rabbitView);
}
}

```

RabbitView2.java

```

RabbitView2.java

package com.example.test_3;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.view.View;

public class RabbitView2 extends View {
    public float bitmapRX;
    public float bitmapRY;

    public RabbitView2(Context context) {
        super(context);
        bitmapRX = 200;    //小兔子的默认的 X 坐标
        bitmapRY = 200;    }
}

```

```

@Override
protected void onDraw(Canvas canvas) {
    // TODO Auto-generated method stub
    super.onDraw(canvas);
    @SuppressWarnings("DrawAllocation") Paint paint1 = new Paint();    //创建并实例化 paint 的
对象
    @SuppressWarnings("DrawAllocation") Bitmap bitmap1 =
BitmapFactory.decodeResource(this.getResources(), R.drawable.rabbit);    //根据图片生成位图图像,
改换 mipmap?
    canvas.drawBitmap(bitmap1, bitmapRX, bitmapRY, paint1);    //画小兔子
    if (bitmap1.isRecycled()) {    //判断图片是否回收
        bitmap1.recycle();
    }
}
}
}

```

TigerView2.java

```

TigerView2.java

package com.example.test_3;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.view.View;

public class TigerView2 extends View {
    public float bitmapTX;
    public float bitmapTY;

    public TigerView2(Context context) {
        super(context);
        bitmapTX = 100;    //老虎的默认的 X 坐标
        bitmapTY = 100;    }

    @Override
    protected void onDraw(Canvas canvas) {
        // TODO Auto-generated method stub
        super.onDraw(canvas);
        @SuppressWarnings("DrawAllocation") Paint paint2 = new Paint();    //创建并实例化 paint 的
对象
        @SuppressWarnings("DrawAllocation") Bitmap bitmap2 =

```

```
BitmapFactory.decodeResource(this.getResources(), R.drawable.tiger); //根据图片生成位图图像
canvas.drawBitmap(bitmap2, bitmapTX, bitmapTY, paint2); //画老虎
if (bitmap2.isRecycled()) { //判断图片是否回收
    bitmap2.recycle();
}
}
}
```

四、实验心得

第一次写出一个真正能在自己手机上跑起来的应用程序，掌握的技能又多了一门，心里还是有抑制不住的新鲜感和喜悦感的，也让我对下一次实验更加期待了。

这个实验其实就是对 java 里面的事件监听器和 Android 里面的触屏 Touch 事件监听器的综合考察，之前 java 课程中我们实现过很多次关于事件监听的实验，所以这个实验的难度其实不大，唯一的难度就是涉及到一些我们以前没有接触到的一些包，基本上都是因为 android 的特殊性带来的。

实验四 基于 Android 的综合应用程序

一、实验内容

- 1) 模仿程序 5.3 + 6.2 + 6.6，编写综合程序。
- 2) 有 2 个 Activity，能实现数据的传递（其中有一个是网页）。
- 3) 网页传到第二个 Activity 后可以打开（用一个按钮）
- 4) 同时也可以返回 Home 桌面（用另一个按钮）

二、实验过程

为了在一个界面同时监听多个 Button 并且实现 activity 跳转，我选择了使用下面这种代码：

```
1. private class ButtonListener implements View.OnClickListener{
2.     @Override
3.     public void onClick(View v) {
4.         switch (v.getId()){
5.             case R.id.btnTextView1:
6.                 intent1 = new Intent(MainActivity.this, TestTextViewActivi
7. ty1.class);
8.                 startActivity(intent1);
9.                 break;
10.            case R.id.btnTextView2:
11.                intent2 = new Intent(MainActivity.this, TestTextViewActivi
12. ty2.class);
13.                startActivity(intent2);
14.                break;
15.        }
16.    }
17. }
```

对应到实验四里面就是：

```

1. private class ButtonListener implements View.OnClickListener {
2.     @Override
3.     public void onClick(View v) {
4.         switch (v.getId()) {
5.             case R.id.home_button:
6.                 intent2 = new Intent(MainActivity.this, RegisterActivity
7.                     .class);
8.                 startActivity(intent2);
9.                 break;
10.            case R.id.website_button:
11.                Intent intent_3 = new Intent();// 创建 Intent 对象
12.                intent_3.setAction(Intent.ACTION_VIEW);// 为 Intent 设置动
13.                intent_3.setData(Uri.parse("http://www.hfut.edu.cn/")); / 为 Intent 设置数据
14.                startActivity(intent_3);// 将 Intent 传递给 Activity
15.
16.                /*intent3 = new Intent(MainActivity.this, WebsiteActivit
17.                    y.class);
18.                startActivity(intent3);*/
19.                break;
20.            }
21.        }
22.    }

```

这是最核心的代码，其他都是对老师已给的代码里的一些事件的重命名，对一些问题的解决，先列举如下：

1. Activity 和 AppCompatActivity 的区别

AppCompatActivity，首先从名字上来看，是 compat 系列中的一员，和 ViewCompat, ContextCompat 一样，是提供向下兼容功能的 Activity。在 AppCompatActivity 之前，不同版本的设备中，尤其是 3.0 设备之前，radioButton, EditText 的外观都不一样。所以 Android 引入了 AppCompatActivity 来统一控件的外观风格，使得其在不同的设备上都能统一样式。在代码上，则是把 TextView, EditText, 都自动

转为了对应的 `CompatView` 类。为了美观,我最后统一使用了 `Activity`。

2. `view.OnClickListener` 与 `DialogInterface.OnClickListener` 的冲突和选择

```
1. import android.view.View.OnClickListener;
2. import android.content.DialogInterface.OnClickListener;
```

其实,当我们用某个 `Listener` 时,不一定就要 `import` 它,直接用全名去定义就不需要 `import` 了,例如

```
1. builder.setItems(items, new android.content.DialogInterface.OnClickListener(
2.     {
3.         @Override
4.         public void onClick(DialogInterface dialog, int which)
5.         {
6.             // TODO 自动生成的方法存根
7.         }
8.     });
```

而我在实验四中,没有使用上面的方法,而是直接使用了

```
1. website = (Button) findViewById(R.id.website_button);    //获取 website 按钮
2. website.setOnClickListener(new ButtonListener());
3.
4. home = (Button) findViewById(R.id.home_button);    //获取 home 按钮
5. home.setOnClickListener(new ButtonListener());
```

3. 按钮控件的属性补充和使用

颜色:

1) 使用 `Color` 类的常量

例如: `Int color=Color.BLUE;`//创建一个蓝色

2) 使用 Color 类的静态方法 argb 创建颜色

`int color=Color.argb(127, 255, 0, 255)` ---其中 a 表示半透明度

r 表示红色

g 表示绿色

b 表示蓝色

3) 在 xml 文件中定义颜色值

```
<resources>
```

```
<color name="mycolor.">#7fff00</color>
```

```
</resources>
```

4) 背景颜色:

`android:backgroundTint` 解决

4. 在创建新的 .java 文件时, 要导入父类中的 `AppCompatActivity`, 报错, 无法找到这个父类。

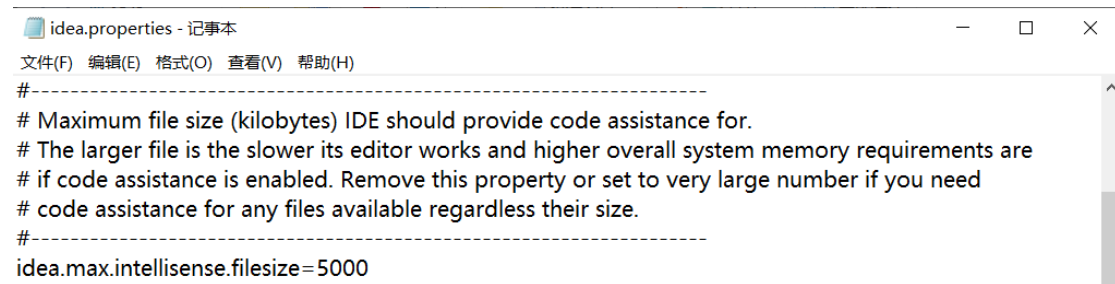
删除 .idea 文件夹, 重启 Android Studio 解决。

5. Cannot resolve symbol 'R' 产生的原因和解决方法

在将 R 文件复制到项目的另外一个位置时, android studio 报出一个提示, File size exceeds configured limit (2560000). Code insight features not available.

原因是 IntelliJ IDEA 对单个文件的大小默认限制为 2500kb, android studio 是基于 IntelliJ IDEA 的, 而 Windows 系统和 OS X 系统对文件大小的计算方式不一致, 前者是 1024 进制, 后者是 1000 进制, 所以进入 android studio 的安装目录, 将里面的

idea.max.intellisense.filesize=2500 数值更改为 5000，如下
重启 android studio ，问题解决。



```
idea.properties - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#-----
# Maximum file size (kilobytes) IDE should provide code assistance for.
# The larger file is the slower its editor works and higher overall system memory requirements are
# if code assistance is enabled. Remove this property or set to very large number if you need
# code assistance for any files available regardless their size.
#-----
idea.max.intellisense.filesize=5000
```

以上就是我在实验过程中遇到的各种问题。

接下来我们看看测试情况，仍是真机调试，打开软件，初始界面如下：

User Name

Password

Confirm Password

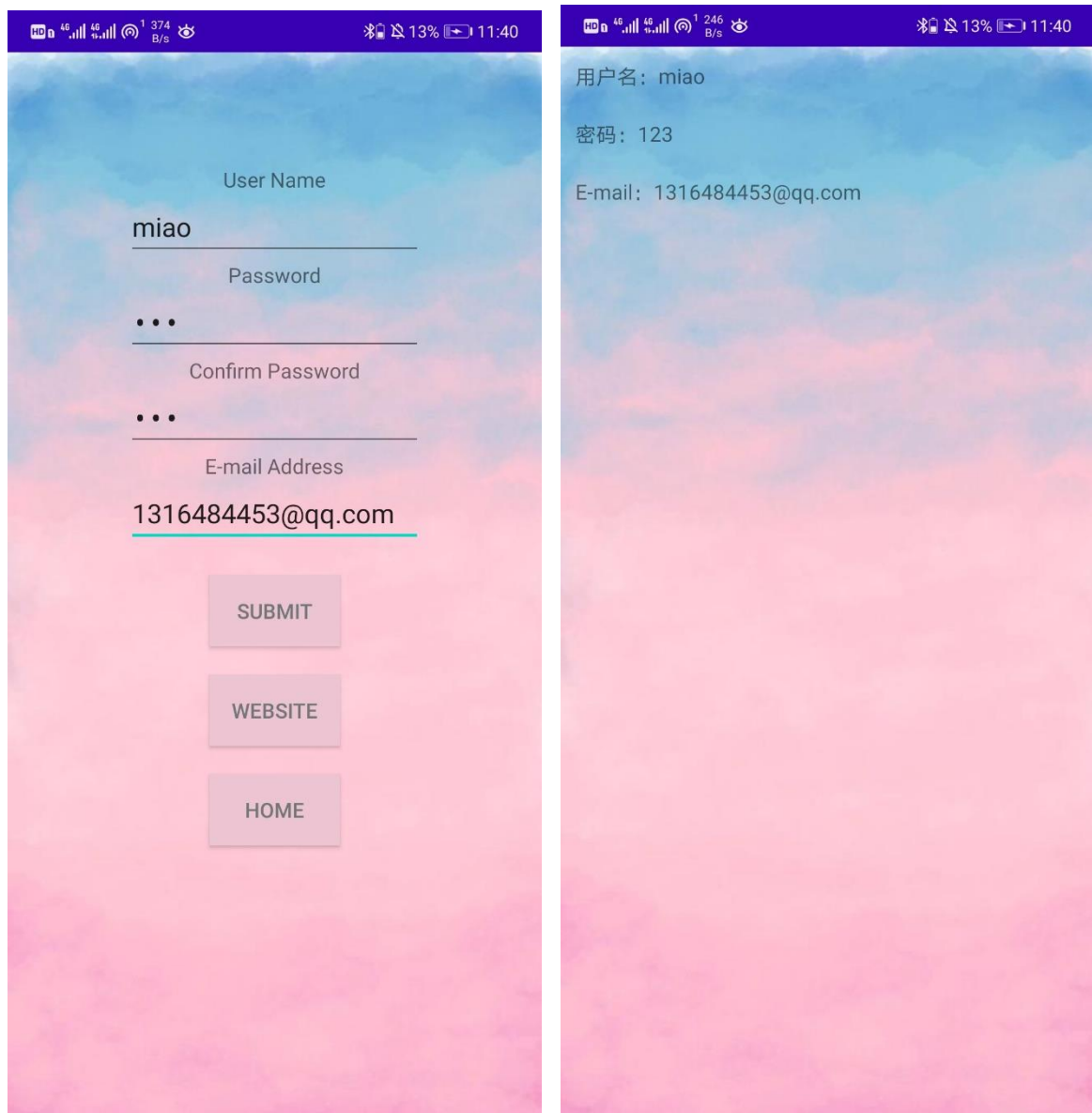
E-mail Address

SUBMIT

WEBSITE

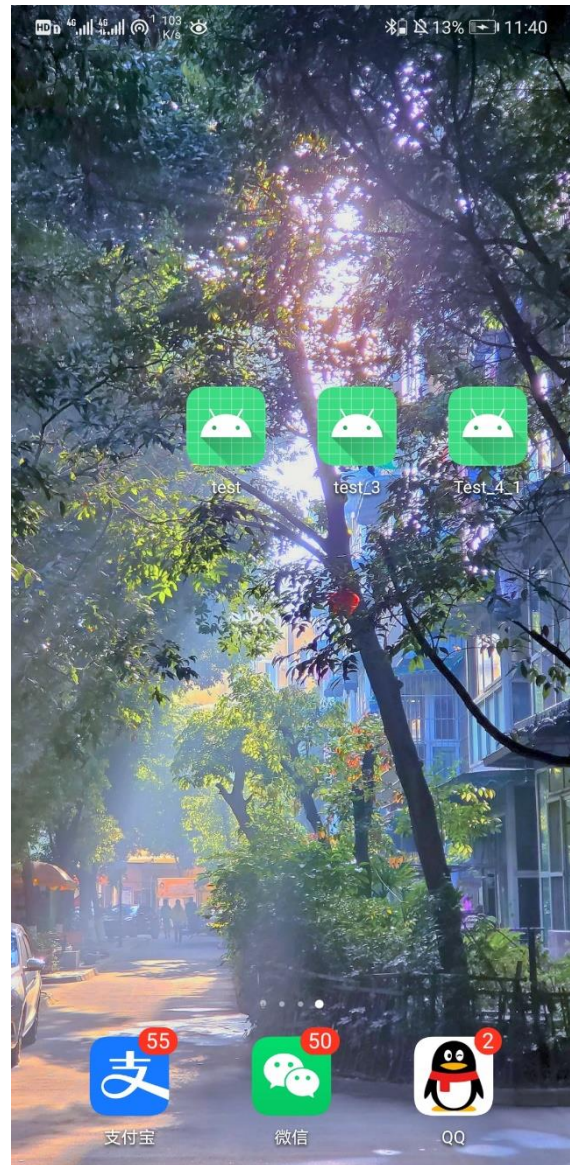
HOME

主界面如上图所示，输入信息如下，点击 SUBMIT，数据传输至第二个 Activity 并且显示。



然后点击 WEBSITE，跳转至合工大官网（最近合工大导师信息的网页有点奔溃，总是会 404，我就没有选择跳转到唐益明老师的信息界面了）

回退至初始界面，点击 HOME，即可回到手机主页，如下。



三、关键程序代码

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    android:orientation="vertical" >
```

```

<TextView
    android:id="@+id/textView_3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<TextView
    android:id="@+id/textView_4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<TextView
    android:id="@+id/textView_5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<TextView
    android:id="@+id/textView_6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="User Name" />
<EditText
    android:id="@+id/user"
    android:minWidth="200dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"/>

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="Password" />
<EditText
    android:id="@+id/pwd"
    android:minWidth="200dp"
    android:inputType="textPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"/>

```

```

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="Confirm Password" />

<EditText
    android:id="@+id/repwd"
    android:minWidth="200dp"
    android:inputType="textPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"/>

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="E-mail Address" />

<EditText
    android:id="@+id/email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:minWidth="200dp" />

<TextView
    android:id="@+id/textView_7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/submit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="#EDC4D2"
    android:rotationX="0"
    android:text="Submit"

```



```

        android:textColor="#747778"
        app:backgroundTint="#EDC4D2" />

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/website_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="#EDC4D2"
    android:rotationX="0"
    android:text="@string/open"
    android:textColor="#747778"
    app:backgroundTint="#EDC4D2" />

<TextView
    android:id="@+id/textView_2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/home_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="#EDC4D2"
    android:rotationX="0"
    android:text="@string/home"
    android:textColor="#747778"
    app:backgroundTint="#EDC4D2" />

</LinearLayout>

```

MainActivity.java


```

package com.example.test_4_1;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends Activity { //改动 AppCompatActivity 为
Activity

    Button submit;
    Button website;
    Button home;
    Intent intent2,intent3;

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

        submit = (Button) findViewById(R.id.submit);    //获取提交按钮
        submit.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v1) {
                String user = ((EditText)
findViewById(R.id.user)).getText().toString();    //获取输入的用户
                String pwd = ((EditText)
findViewById(R.id.pwd)).getText().toString();    //获取输入的密码
                String repwd = ((EditText)
findViewById(R.id.repwd)).getText().toString();    //获取输入的确认密码
                String email = ((EditText)
findViewById(R.id.email)).getText().toString();    //获取输入的 E-mail 地址
                if (!"".equals(user) && !"".equals(pwd)
&& !"".equals(email)) {

                    if (!pwd.equals(repwd)) {    //判断两次输入的密码是否一致
                        Toast.makeText(MainActivity.this, "两次输入的密码不一

```

```

致，请重新输入！", Toast.LENGTH_LONG).show());
        ((EditText) findViewById(R.id.pwd)).setText(""); //
清空密码编辑框
        ((EditText) findViewById(R.id.repwd)).setText("");
//清空确认密码编辑框
        ((EditText) findViewById(R.id.pwd)).requestFocus();
//让密码编辑框获得焦点
    } else { //将输入的信息保存到 Bundle 中，并启动一个新的
Activity 显示输入的用户注册信息
        Intent intent = new Intent(MainActivity.this,
RegisterActivity.class);
        Bundle bundle = new Bundle(); //创建并实例化一个
Bundle 对象
        bundle.putCharSequence("user", user); //保存用户名
        bundle.putCharSequence("pwd", pwd); //保存密码
        bundle.putCharSequence("email", email); //保存 E-
mail 地址
        intent.putExtras(bundle); //将 Bundle 对象添加到
Intent 对象中
        startActivity(intent); //启动新的 Activity
    }
} else {
    Toast.makeText(MainActivity.this, "请将注册信息输入完整!",
Toast.LENGTH_LONG).show();
}

});

website = (Button) findViewById(R.id.website_button); //获取
website 按钮
website.setOnClickListener(new ButtonListener());

/*@Override
public void onClick(View v3) {
    Intent intent = new Intent();// 创建 Intent 对象
    intent.setAction(Intent.ACTION_VIEW);// 为 Intent 设置动作
intent.setData(Uri.parse("http://ci.hfut.edu.cn/2018/0821/c3995a168777/page
.htm"));// 为 Intent 设置数据
    startActivity(intent);// 将 Intent 传递给 Activity
}*/

```

```

        home = (Button) findViewById(R.id.home_button);    //获取 home 按钮
        home.setOnClickListener(new ButtonListener());

        /*@Override
        public void onClick(View v2) {
            Intent intent_2 = new Intent();// 创建 Intent 对象
            intent_2.setAction(Intent.ACTION_MAIN);// 设置 Intent 动作
            intent_2.addCategory(Intent.CATEGORY_HOME);// 设置 Intent 种类
            startActivity(intent_2);// 将 Intent 传递给 Activity
        }
        */
    }

    private class ButtonListener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.home_button:
                    intent2 = new Intent(MainActivity.this,
RegisterActivity.class);
                    startActivity(intent2);
                    break;
                case R.id.website_button:

                    Intent intent_3 = new Intent();// 创建 Intent 对象
                    intent_3.setAction(Intent.ACTION_VIEW);// 为 Intent 设置动
作
                    intent_3.setData(Uri.parse("http://www.hfut.edu.cn/"));//
为 Intent 设置数据
                    startActivity(intent_3);// 将 Intent 传递给 Activity

                    /*intent3 = new Intent(MainActivity.this,
WebsiteActivity.class);
                    startActivity(intent3);*/
                    break;
            }
        }
    }
}

```

register.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/user"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:text="User Name" />
    <TextView
        android:id="@+id/pwd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:text="Password" />
    <TextView
        android:id="@+id/email"
        android:padding="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="E-mail Address" />

</LinearLayout>

```

RegisterActivity.java

```

package com.example.test_4_1;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class RegisterActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.register); //设置该 Activity 中要显示的内容视图
    }
}

```

```

        Intent intent=getIntent(); //获取 Intent 对象
        Bundle bundle=intent.getExtras(); //获取传递的数据包
        TextView user=(TextView)findViewById(R.id.user); //获取显示用户名
的 TextView 组件
        user.setText("用户名: "+bundle.getString("user")); //获取输入的用户
名并显示到 TextView 组件中
        TextView pwd=(TextView)findViewById(R.id.pwd); //获取显示密码的
TextView 组件
        pwd.setText("密码: "+bundle.getString("pwd")); //获取输入的密码
并显示到 TextView 组件中
        TextView email=(TextView)findViewById(R.id.email); //获取显示 E-
mail 的 TextView 组件
        email.setText("E-mail: "+bundle.getString("email")); //获取输入
的 E-mail 并显示到 TextView 组件中

    }
}

```

WebsiteActivity.java

```

package com.example.test_4_1;

import android.os.Bundle;
import android.content.Intent;
import android.net.Uri;
import android.view.View;
import android.widget.Button;
import android.app.Activity;

public class WebsiteActivity extends Activity {

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

        Button button = (Button) findViewById(R.id.website_button); // 通过 ID
值获得按钮对象
        button.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v_3){
                /*Intent intent_3 = new Intent(); // 创建 Intent 对象
                intent_3.setAction(Intent.ACTION_VIEW); // 为 Intent 设置动作

```

```

        intent_3.setData(Uri.parse("http://www.hfut.edu.cn/")); // 为
Intent 设置数据
        startActivity(intent_3); // 将 Intent 传递给 Activity*/
    }
});
}
}
}

```

HomeActivity.java

```

package com.example.test_4_1;

import android.app.Activity;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class HomeActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // 设置页面布局
        Button home = (Button) findViewById(R.id.home_button); // 通过 ID 值获
得按钮对象
        home.setOnClickListener(new View.OnClickListener() { // 为按钮增加单击
事件监听器

            @Override
            public void onClick(View v_2) {
                Intent intent_2 = new Intent(); // 创建 Intent 对象
                intent_2.setAction(Intent.ACTION_MAIN); // 设置 Intent 动作
                intent_2.addCategory(Intent.CATEGORY_HOME); // 设置 Intent 种类
                startActivity(intent_2); // 将 Intent 传递给 Activity
            }
        });
    }
}
}

```

strings.xml

```
<resources>
    <string name="app_name">Test_4_1</string>

    <string name="open">Website</string>
    <string name="action_settings">Settings</string>

    <string name="home">Home</string>
    <string name="action_settings_2">Settings</string>

</resources>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.test_4_1">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Test_4_1">

        <!--</activity>-->
        <activity
            android:label="Test_4_1"
            android:name=".MainActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:label="显示用户注册信息"
            android:icon="@drawable/ic_launcher_background"
            android:name=".RegisterActivity">
        </activity>

        <activity
```

```

        android:name=".WebsiteActivity">
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <activity
        android:name=".HomeActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

</application>

</manifest>

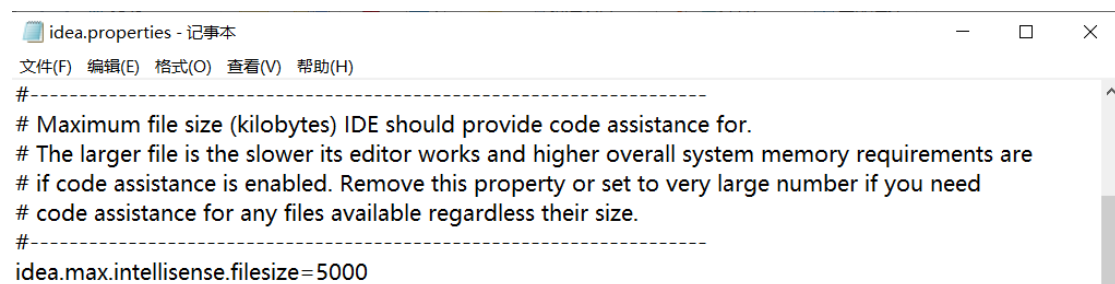
```

四、实验心得

我认为这个实验应该是四个实验中难度最大的一个，从实验下午 5 点公布，到晚上 12 点我才完成全部的功能和修饰，用了我将近 7 个小时才做出来，涉及的知识点也最多的一个，但是有了前几次实验的铺垫，我对各个文件之间的关系已经掌握的很熟练了，所以主要的问题除了为了在一个界面同时监听多个 Button 并且实现 activity 跳转之外，都是在处理各种各样的小报错，比如：

(1) 为了不影响第一版 Test_4 的迭代，我新建了 Test_4_1 并且转移了代码，移植之后就出现了 **Cannot resolve symbol 'R'** 的错误，

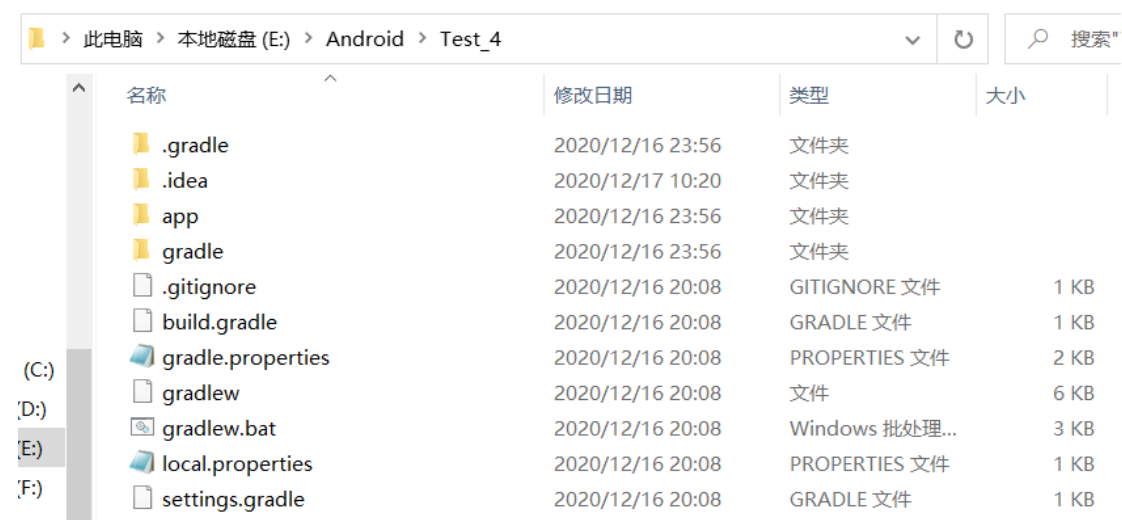
在将 R 文件复制到项目的另外一个位置时，android studio 报出一个提示，File size exceeds configured limit (2560000). Code insight features not available. 半天才终于解决了这个问题，原因是 IntelliJ IDEA 对单个文件的大小默认限制为 2500kb，android studio 是基于 IntelliJ IDEA 的，而 Windows 系统和 OS X 系统对文件大小的计算方式不一致，前者是 1024 进制，后者是 1000 进制，所



```
#-----  
# Maximum file size (kilobytes) IDE should provide code assistance for.  
# The larger file is the slower its editor works and higher overall system memory requirements are  
# if code assistance is enabled. Remove this property or set to very large number if you need  
# code assistance for any files available regardless their size.  
#-----  
idea.max.intellisense.filesize=5000
```

以进入 android studio 的安装目录，将里面的 idea.max.intellisense.filesize=2500 数值更改为 5000，如下
重启 android studio，问题解决。

(2) 在第一版 Test_4 中，我准备通过代码对按钮的颜色进行修改，但是还是一直保持默认的紫色，网上有解释说是默认了黑色背景皮肤的原因，于是我按照教程对默认颜色黑色进行修改为 bright，结果突



名称	修改日期	类型	大小
.gradle	2020/12/16 23:56	文件夹	
.idea	2020/12/17 10:20	文件夹	
app	2020/12/16 23:56	文件夹	
gradle	2020/12/16 23:56	文件夹	
.gitignore	2020/12/16 20:08	GITIGNORE 文件	1 KB
build.gradle	2020/12/16 20:08	GRADLE 文件	1 KB
gradle.properties	2020/12/16 20:08	PROPERTIES 文件	2 KB
gradlew	2020/12/16 20:08	文件	6 KB
gradlew.bat	2020/12/16 20:08	Windows 批处理...	3 KB
local.properties	2020/12/16 20:08	PROPERTIES 文件	1 KB
settings.gradle	2020/12/16 20:08	GRADLE 文件	1 KB

然大面积爆红报错导入父类中的 `AppCompatActivity` 无法找到这个父类等其他错误，即使改成 `Activity` 包也不能解决问题，吓得我回撤，但是仍不能解决问题。

最后看到一种办法是删除 `.idea` 文件，然后给重启，让 `Android Studio` 重新配置删除的文件，才终于解决了这个问题。

这次实验难度虽大，但是收获也比前几次实验多得多，因为这次实验涉及了 `Android` 应用开发的核心 `Intent`。可以说没有 `Intent` 的应用是没有灵魂的，只是一具空壳。

经过了完完整整的四次实验，我对 `Android` 应用开发有了一个总体上的认识，相信我以后入门这方面的方向时，能够更加得心应手，感谢这门让我收获丰富的课程。