华南理工大学
South China University of Technology

Android移动应用开发

实验I

zhjing@scut.edu.cn

华南理工大学
South China University of Technology

实验1 ACTIVITY的使用

实验目的

- Android Studio 环境安装
- Activity 的跳转
- 基本 UI 元素的设计与使用

实验内容

- 1、安装配置 Android 移动应用开发环境
- 2、二选一：
  - 设计并实现移动端应用程序
  - 完成整个或部分案例：欢乐写数字t

華南理工大學
South China University of Technology

ANDROID 开发环境

开发环境

- 1、 JDK
  - java --version
- 2、二选一：
  - 自行设计并实现一个移动端应用程序
  - 完成整个或部分案例：欢乐写数字t

华南理工大学
South China University of Technology

欢乐写数字APP开发

欢乐写数字

- 步骤1：使用Empty Activity模板创建WriteNumber。
- 步骤2：设计启动界面
  - 显示项目Logo，启动界面显示5s，跳转至游戏主界面。

本案例来自《Android项目开发实战入门》

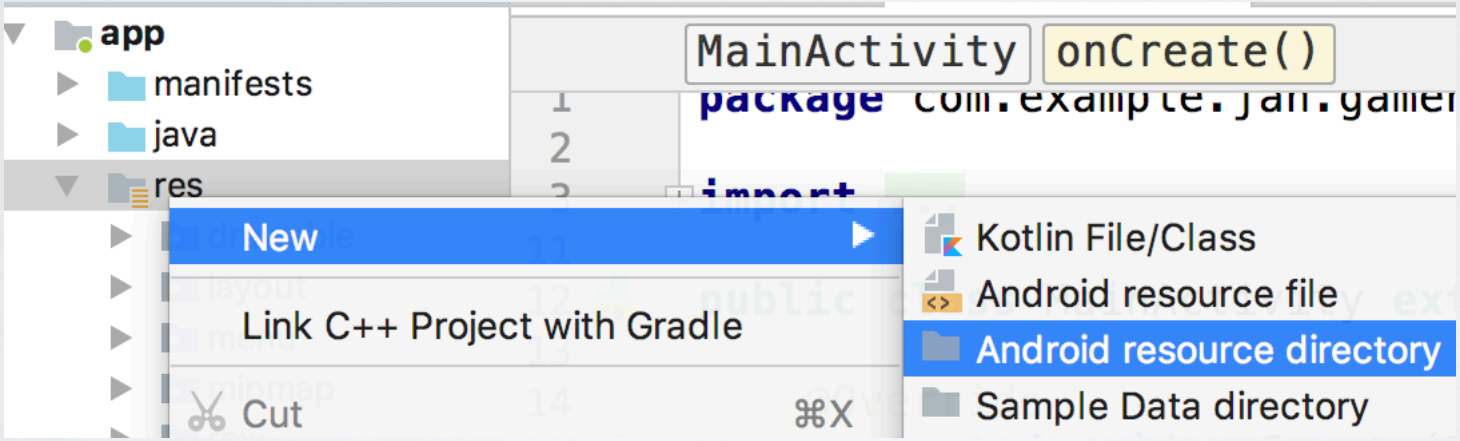
华南理工大学
South China University of Technology

启动界面布局

- 1、添加图片等资源
- 2、修改布局文件
- 3、实现启动界面的全屏显示

添加图片等资源

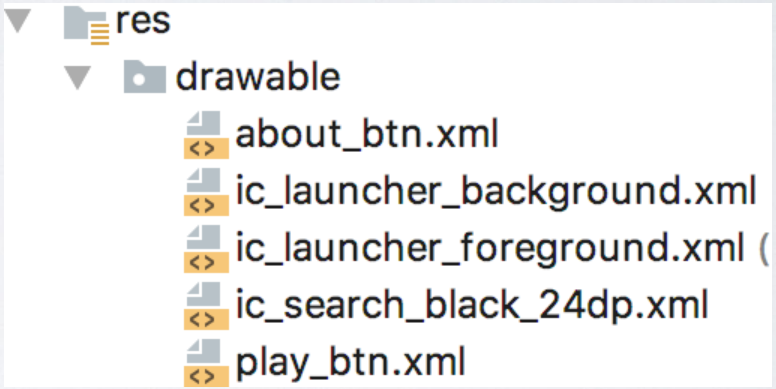
- 如果想将图片等放在自定义目录下，则按下图添加图片资源目录



- 然后将图片 copy+paste 至此目录即可

1、添加图片等资源

- drawable下的文件示例，也可直接将图片拷贝在此目录下。



play_btn.xml，文件中的btn_play*.png 在目录res/mipmap下

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@mipmap/btn_play1" android:state_focused="false" android:state_pressed="false" />
    <item android:drawable="@mipmap/btn_play2" android:state_focused="false" android:state_pressed="true" />
    <item android:drawable="@mipmap/btn_play2" android:state_focused="false" android:state_pressed="true" />
</selector>
```

2、修改布局文件

- 设计布局文件 activity_main.xml
  - 增加/删除控件、修改控件属性等
  - 设置背景图片
- 运行无误，则继续

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="h
    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:background="@mipmap/start_bg"
    tools:context="com.example.jan.gamenumber.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

3、实现启动界面的全屏显示

- 原因：不喜欢界面上还有项目名字。。。
- 1）MainActivity类的基类设置为 Activity；
- 2）在AndroidManifest.xml 中，设置全屏；
- 3）若非Activity基类，请自行百度设置方法

```
<activity
    android:name=".MainActivity"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
```

```
public class MainActivity extends Activity {
```

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="GameNumber"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

游戏主界面

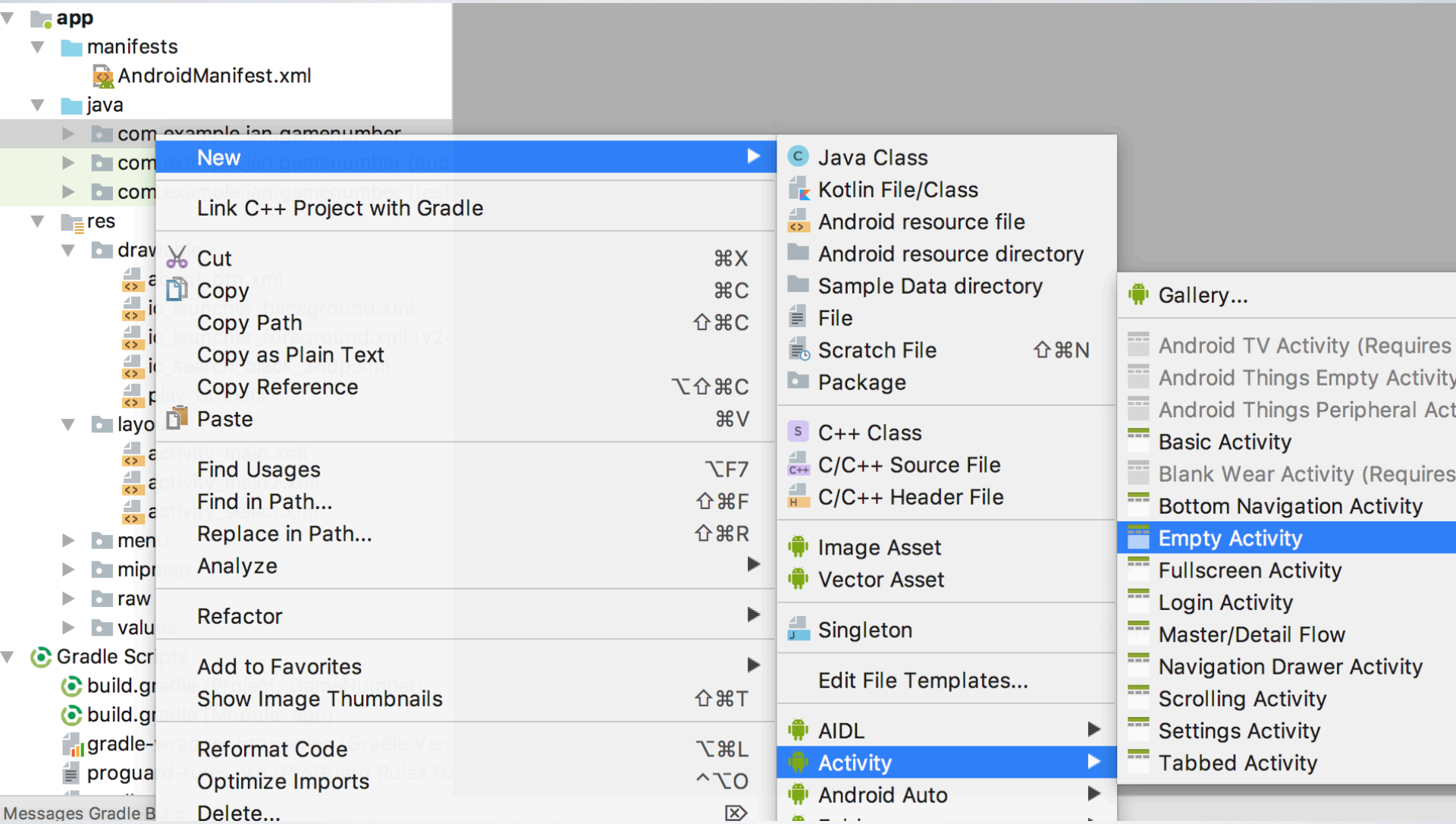
- 4、启动界面跳转到游戏主界面
- 5、游戏主界面设计
- 6、游戏主界面全屏显示

4、启动界面跳转到游戏主界面

- 新建游戏主界面Activity，方法如后页图示
- 用Timer类设置启动界面的显示时间，5s后跳转到游戏主界面 MainActivity2 类。
- 运行无误，则继续

```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Timer timer=new Timer();
        TimerTask timerTask =new TimerTask() {
            @Override
            public void run() {
                startActivity(new Intent( packageContext: MainActivity.this,Main2Activity.class));
                finish();
            }
        };
        timer.schedule(timerTask, delay: 5000);
    }
```


app
manifests
AndroidManifest.xml
java
res
Gradle Scripts
New
Link C++ Project with Gradle
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy as Plain Text
Copy Reference ⌥⇧⌘C
Paste ⌘V
Find Usages ⌥F7
Find in Path... ⇧⌘F
Replace in Path... ⇧⌘R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails ⇧⌘T
Reformat Code ⌥⌘L
Optimize Imports ^⌥O
Delete...
Java Class
Kotlin File/Class
Android resource file
Android resource directory
Sample Data directory
File
Scratch File ⇧⌘N
Package
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Singleton
Edit File Templates...
AIDL
Activity
Android Auto
Gallery...
Android TV Activity (Requires
Android Things Empty Activity
Android Things Peripheral Act
Basic Activity
Blank Wear Activity (Requires
Bottom Navigation Activity
Empty Activity
Fullscreen Activity
Login Activity
Master/Detail Flow
Navigation Drawer Activity
Scrolling Activity
Settings Activity
Tabbed Activity
Messages Gradle B

TimerTask

Summa

```
public abstract class TimerTask
extends Object implements Runnable
```

java.lang.Object
  ↳ java.util.TimerTask

A task that can be scheduled for one-time or repeated execution by a Timer.

Summary

Protected constructors	
`TimerTask()`	Creates a new timer task.

Public methods	
`boolean`	`cancel()` Cancels this timer task.
`abstract void`	`run()` The action to be performed by this timer task.

5、游戏主界面设计

- 图片资源文件，放入drawable或mipmap中
- 新建资源文件目录raw，存放背景音乐mp3文件
- 修改MainActivity2类对应的布局文件
  - button-1：跳转到数字选择界面 onPlay
  - button-2：跳转到关于界面 onAbout
  - button-3：播放音乐 onMusic

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="@mipmap/main_bg"
    tools:context="com.example.jan.gamenumber.Main2Activity">

    <Button
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:layout_above="@+id/btn_music"
        android:layout_centerHorizontal="true"
        android:background="@drawable/play_btn"
        android:onClick="onPlay"/>
    <Button
        android:layout_width="90dp"
        android:layout_height="90dp"
        android:id="@+id/btn_music"
        android:layout_alignParentBottom="true"
        android:layout_margin="10dp"
        android:background="@mipmap/btn_music1"
        android:onClick="onMusic" />
    <Button
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:id="@+id/btn_music2"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_margin="10dp"
        android:background="@mipmap/btn_music1"
        android:onClick="onAbout" />
</RelativeLayout>
```

6、游戏主界面全屏显示

- 方法同前
  - Main2Activity类
  - AndroidManifest.xml

其它要跳转的界面

- 7、数字选择类、关于类的创建
- 8、数字选择类对应的界面设计
- 9、关于类对应的界面设计
- 10、游戏主界面向数字选择界面、关于界面的跳转

7、数字选择类、关于类的创建

- 新建 SelectActivity类，仍使用 Empty Activity 模板。
- 新建 AboutActivity类，仍使用 Empty Activity 模板。

7、数字选择类对应的界面设计

- 图片资源文件，放入drawable或mipmap中
- 资源文件目录raw中存放背景mp3音乐文件

9、游戏主界面向其它界面跳转

- 向选择界面跳转 onPlay()
- 向关于界面跳转 onAbout()

```
public class Main2Activity extends AppCompatActivity {
  … …
  public void onAbout(View view) {
    startActivity(new Intent(Main2Activity.this,AboutActivity.class));
  }

  public void onPlay(View view) {
    startActivity(new Intent(Main2Activity.this,SelectActivity.class));
  }
… …
}
```

10. 音乐播放功能

- 功能1：
  - 用按钮停止播放音乐，或者重新播放。
  - onMusic
- 功能2：启动后自动播放音乐

```
public class Main2Activity extends AppCompatActivity {
  … …
  static boolean isPlay=true;  //Music playing status
  MediaPlayer mediaPlayer;  //Music player object
  Button music_btn;  //Music playing button

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    music_btn=(Button)findViewById(R.id.btn_music);
    PlayMusic();
  }

  public void onMusic(View view) {
    if(isPlay){
        if(mediaPlayer!=null){
            mediaPlayer.stop();
            music_btn.setBackgroundResource(R.mipmap.btn_music2);
            isPlay=false;
        }
    }else {
        PlayMusic();
        music_btn.setBackgroundResource(R.mipmap.btn_music1);
        isPlay=true;
    }
  }
… …
}
```

10. 音乐播放功能

- 功能3：跳转到其它界面时，音乐停止播放
- 功能4：返回当前界面时，音乐继续播放

```
public class Main2Activity extends AppCompatActivity {
  … …
  @Override
  protected void onStop() {
    super.onStop();
    if(mediaPlayer!=null){
        mediaPlayer.stop();
    }
  }

  @Override
  protected void onDestroy() {
    super.onDestroy();
    if(mediaPlayer!=null){
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer=null;
    }
  }

  @Override
  protected void onRestart() {
    super.onRestart();
    if(isPlay==true){
        PlayMusic();
    }
  }
… …
}
```

Activity的生命周期函数

MediaPlayer

added in API level 1

Summary: Nested Classes | Constants | Fields | Ctors | Methods | Protected Methods | Inherited Methods

```
public class MediaPlayer
extends Object implements VolumeAutomation
```

java.lang.Object
  ↳ android.media.MediaPlayer

MediaPlayer class can be used to control playback of audio/video files and streams. An example on how to use the methods in this class can be found in `VideoView`.