**实验1：Android Studio开发环境搭建及Activity生命周期**

学号： 2021329600006 姓名： 陈昊天

班级： 计算机科学与技术21（4）班 联系电话： 13456982338

日期：2023.9.21

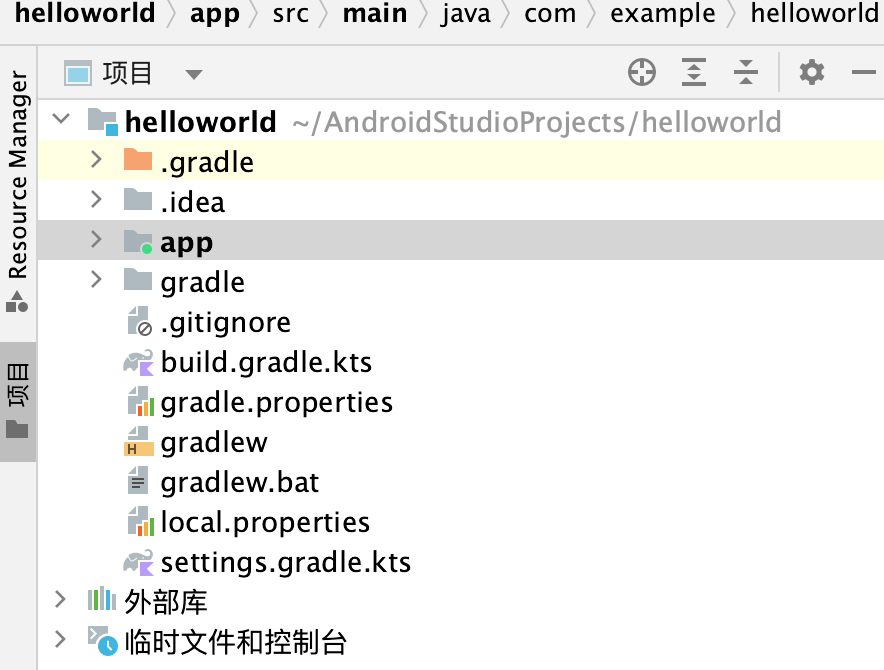
一、实验内容

1. 安装Android Studio，安装SDK和模拟器，修改配置信息，安装位置设置为非系统盘。熟悉AS的基本操作，学会改变编辑器的字体、显示方式等基本操作。
2. 建立新项目，实现输出Hello World！。说明项目各个文件的作用，以及各个关键语句的作用或含义，给出程序的运行结果，请截图并介绍。
3. 设置Activity生命周期的Log日志，分别执行相关操作。
4. 在MainActivity中复写onCreate 、onStart、onResume 、onPause 、onStop 、onDestroy 和onRestart；
5. 分别在各个方法中添加日志监视语句，例如：Log.d(“MainActivity”, "onCreate");
6. 分别执行以下操作，并截图说明。
7. 运行程序，在LogCat的输出日志中检查输出情况，执行顺序是否为onCreate()、onStart()、onResume()；
8. 按下back键，在LogCat的输出日志中检查输出情况，执行顺序是否为onPause（）、onStop（）、onDestory（）；
9. 再启动HelloWorld，在LogCat的输出日志中检查输出情况，执行顺序是否为onRestart（）、onStart（）、onResume（）；

二、实验报告

1. 实验过程

**（2-1）说明项目各个文件的作用**



.gradle：一个目录，存储了Gradle的临时缓存文件。当Gradle执行构建任务时，它会在这个目录中存储一些数据。

.idea：一个目录，由IntelliJ IDEA生成。它包含了有关项目的一些元数据，如代码风格设置、模块等。

app：应用程序的主要源代码目录。它包含了Java/Kotlin源代码、Android资源、manifest文件等。

gradle：这个目录包含了Gradle wrapper的文件，可以对Gradle进行版本控制。

.gitignore：一个文件，用于告诉Git哪些文件和目录不应该被版本控制系统跟踪。

build.gradle.kts 或 build.gradle：项目级别的Gradle构建脚本。它定义了适用于所有模块的构建配置（例如Android Gradle插件的版本）。

gradle.properties：一个可选的属性文件，可以设置一些Gradle的配置项，如JVM参数等。

gradlew 和 gradlew.bat：Gradle Wrapper的执行文件，分别对应于Unix/Linux和Windows系统。使用这些文件，可以在没有安装Gradle的机器上运行Gradle构建。

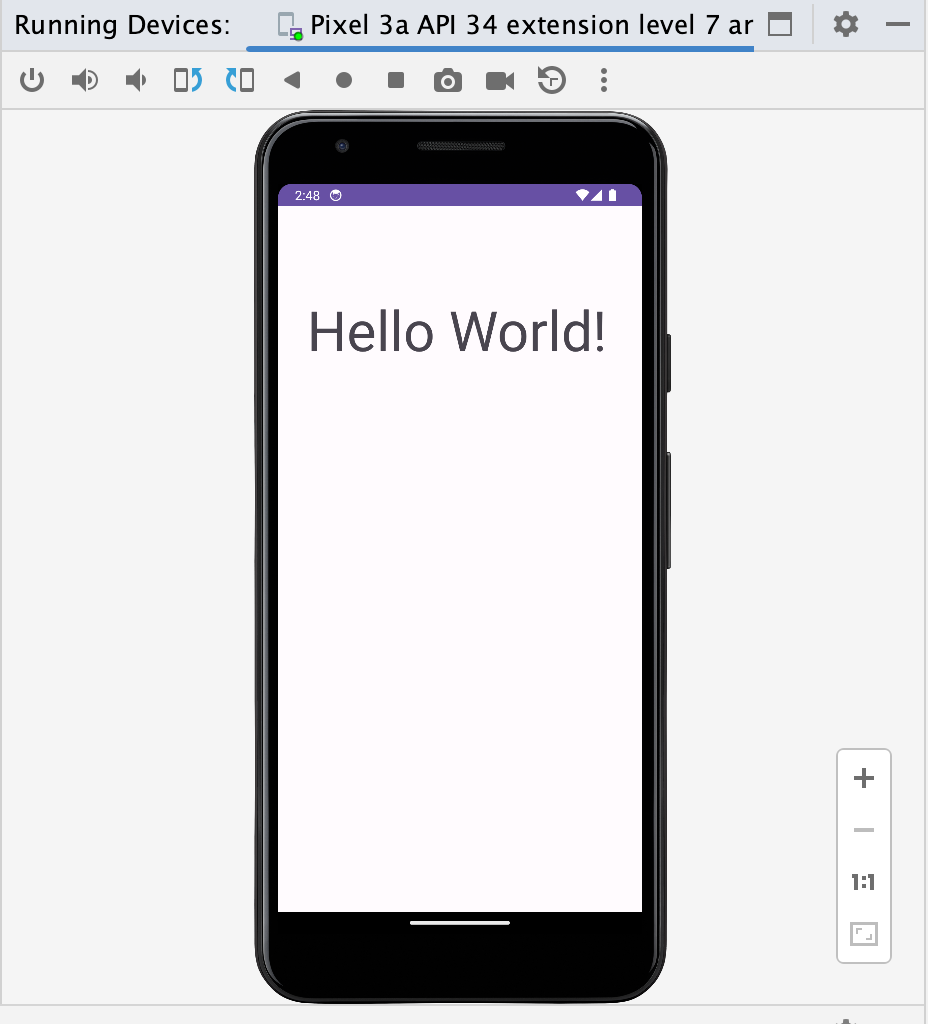
local.properties：一个生成的本地文件，通常包含了SDK的位置等本地特定的配置。

settings.gradle.kts 或 settings.gradle：一个文件，定义了哪些模块应该被包含在项目中。在这个文件中，可以添加或删除模块。

**（2-2）关键语句的作用或含义**

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 *// onCreate() 方法在活动第一次创建时被调用。  
 // savedInstanceState 是一个 Bundle，它包含了活动的前一个实例的状态* **super**.onCreate(savedInstanceState);  
 *// 调用了父类（AppCompatActivity）的 onCreate() 方法* setContentView(R.layout.*activity\_main*);  
 *// 设置了活动的用户界面布局* Log.*d*(**"MainActivity"**, **"onCreate"**);  
 *// 打印一条调试信息*}

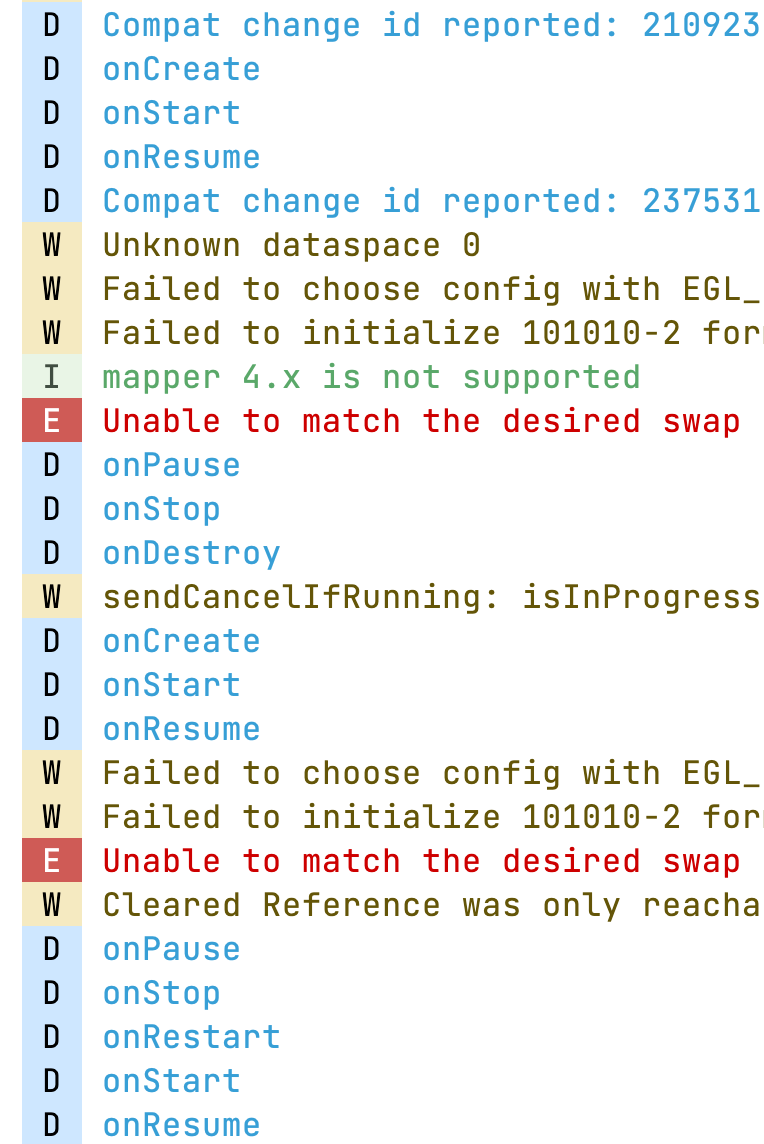
**（2-3）程序的运行结果**



**（3-1）（3-2）复写onCreate 、onStart、onResume 、onPause 、onStop 、onDestroy 和onRestart；添加日志监视语句**

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 *// onCreate() 方法在活动第一次创建时被调用。  
 // savedInstanceState 是一个 Bundle，它包含了活动的前一个实例的状态* **super**.onCreate(savedInstanceState);  
 *// 调用了父类（AppCompatActivity）的 onCreate() 方法* setContentView(R.layout.*activity\_main*);  
 *// 设置了活动的用户界面布局* Log.*d*(**"MainActivity"**, **"onCreate"**);  
 *// 打印一条调试信息*}  
  
@Override  
**protected void** onStart() {  
 **super**.onStart();  
 Log.*d*(**"MainActivity"**, **"onStart"**);  
}  
  
@Override  
**protected void** onResume() {  
 **super**.onResume();  
 Log.*d*(**"MainActivity"**, **"onResume"**);  
}  
  
@Override  
**protected void** onPause() {  
 **super**.onPause();  
 Log.*d*(**"MainActivity"**, **"onPause"**);  
}  
  
@Override  
**protected void** onStop() {  
 **super**.onStop();  
 Log.*d*(**"MainActivity"**, **"onStop"**);  
}  
  
@Override  
**protected void** onDestroy() {  
 **super**.onDestroy();  
 Log.*d*(**"MainActivity"**, **"onDestroy"**);  
}  
  
@Override  
**protected void** onRestart() {  
 **super**.onRestart();  
 Log.*d*(**"MainActivity"**, **"onRestart"**);  
}

**（3-3）执行操作，并截图说明**



1. 学习心得

这次的实验是对Android Studio的初步了解和操作，让我深入理解了Android开发的基本流程和一些关键的概念。

在实验的第一部分中，我成功地安装了Android Studio，并配置了SDK和模拟器。我选择将安装位置设置为非系统盘，避免占用系统盘的空间。在熟悉Android Studio的基本操作过程中，我学习了如何改变编辑器的字体、显示方式等基本操作，这些操作使得开发环境更加符合我的个人习惯，提高了工作效率。

在实验的第二部分中，我创建了一个新的项目，并成功地实现了输出"Hello World!"的功能。在这个过程中，我了解了项目的结构，包括各个文件的作用以及关键语句的含义，这对我理解Android中的程序组织和运行非常有帮助。

在实验的第三部分，我通过设置Activity生命周期的Log日志，详细地了解了Activity的生命周期方法的执行顺序。我在MainActivity中复写了onCreate、onStart、onResume、onPause、onStop、onDestroy和onRestart这些方法，并在每个方法中添加了日志监视语句。通过观察LogCat的输出日志，我明白了在应用启动、停止和重启时，各个生命周期方法的执行顺序。