Activity生命周期

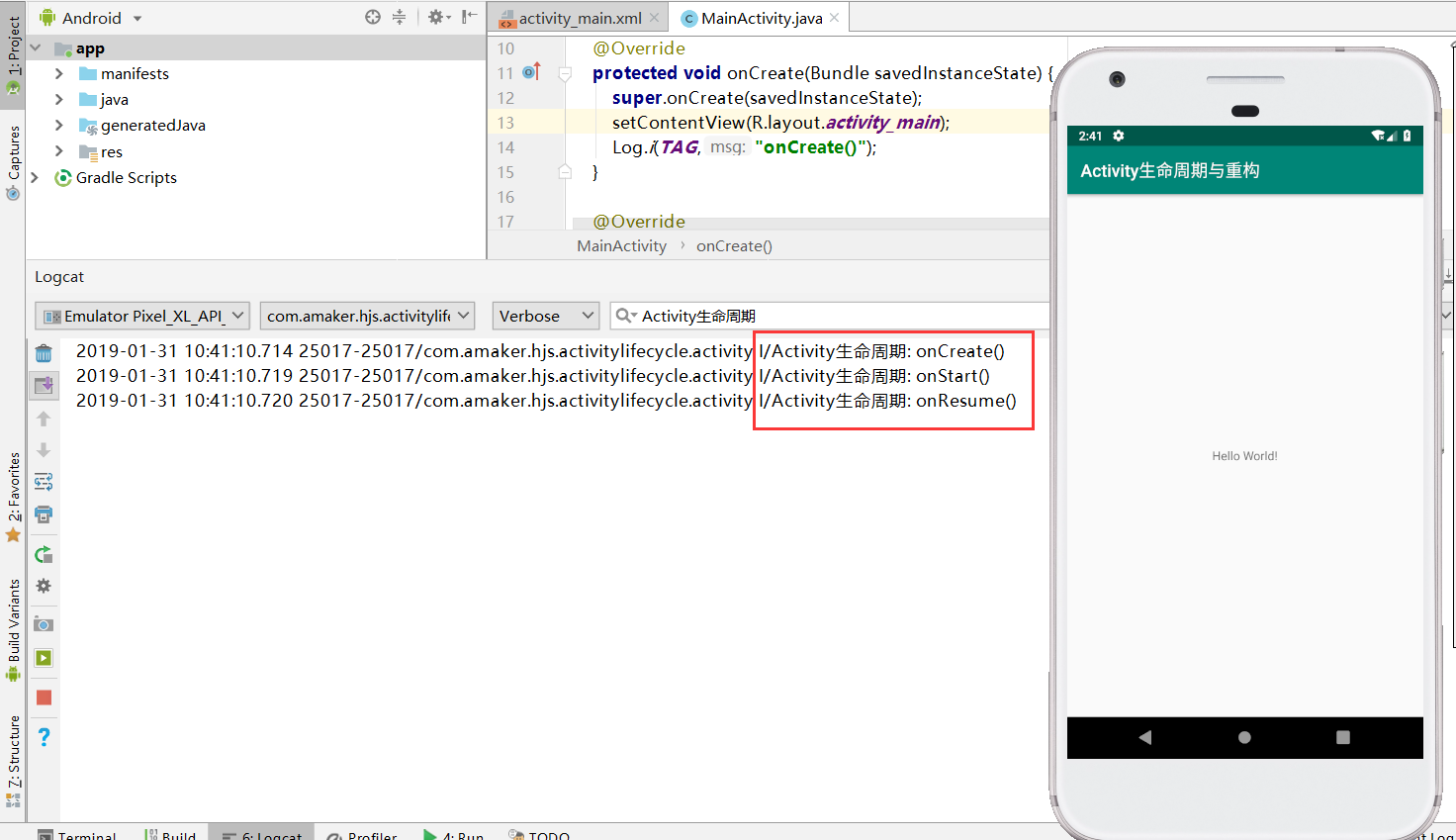


onResume()：显示，获得焦点

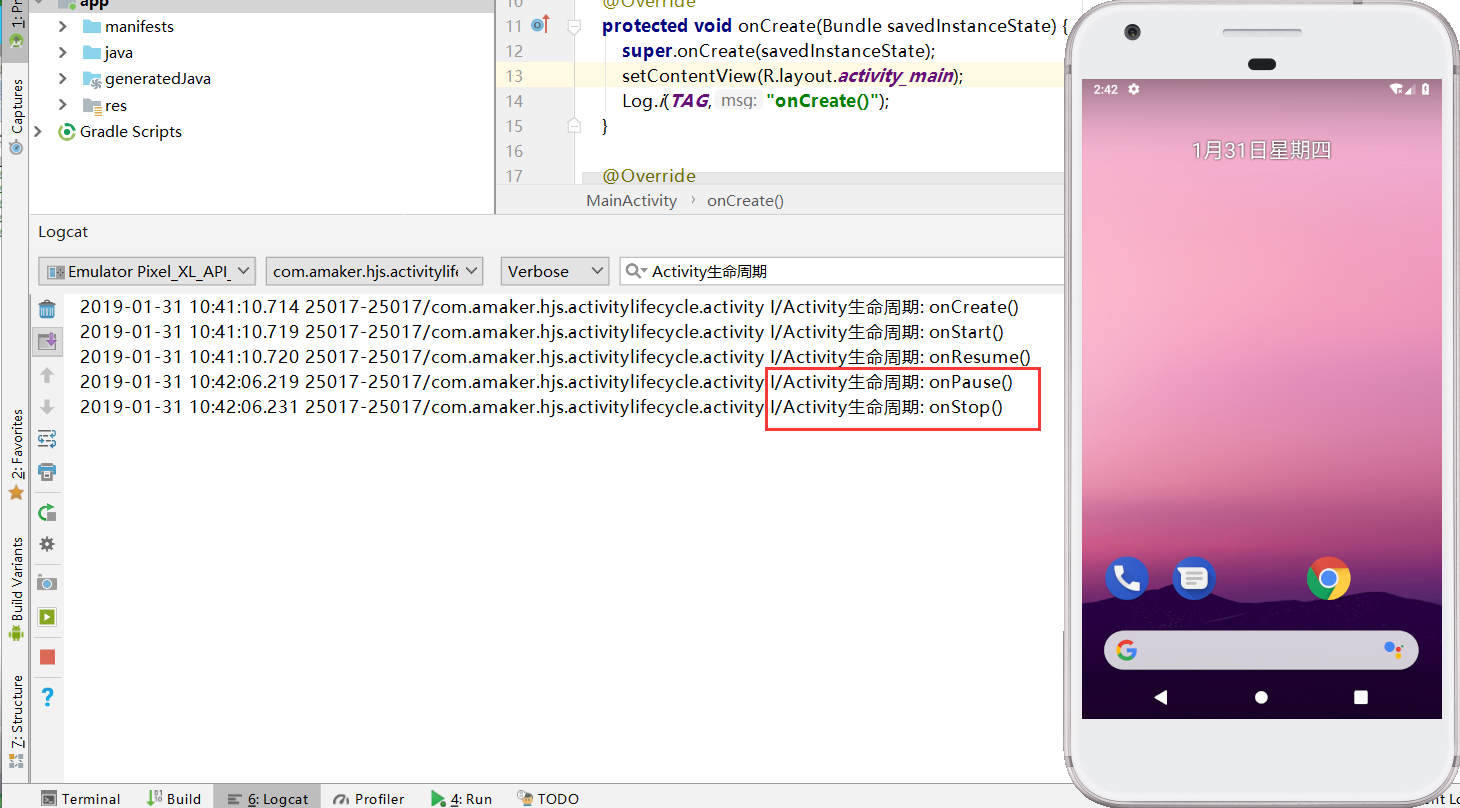
onPause()：暂停，失去焦点。但仍保持当前显示

onStop()：暂停，失去焦点，并且失去当前显示（隐藏在后台）

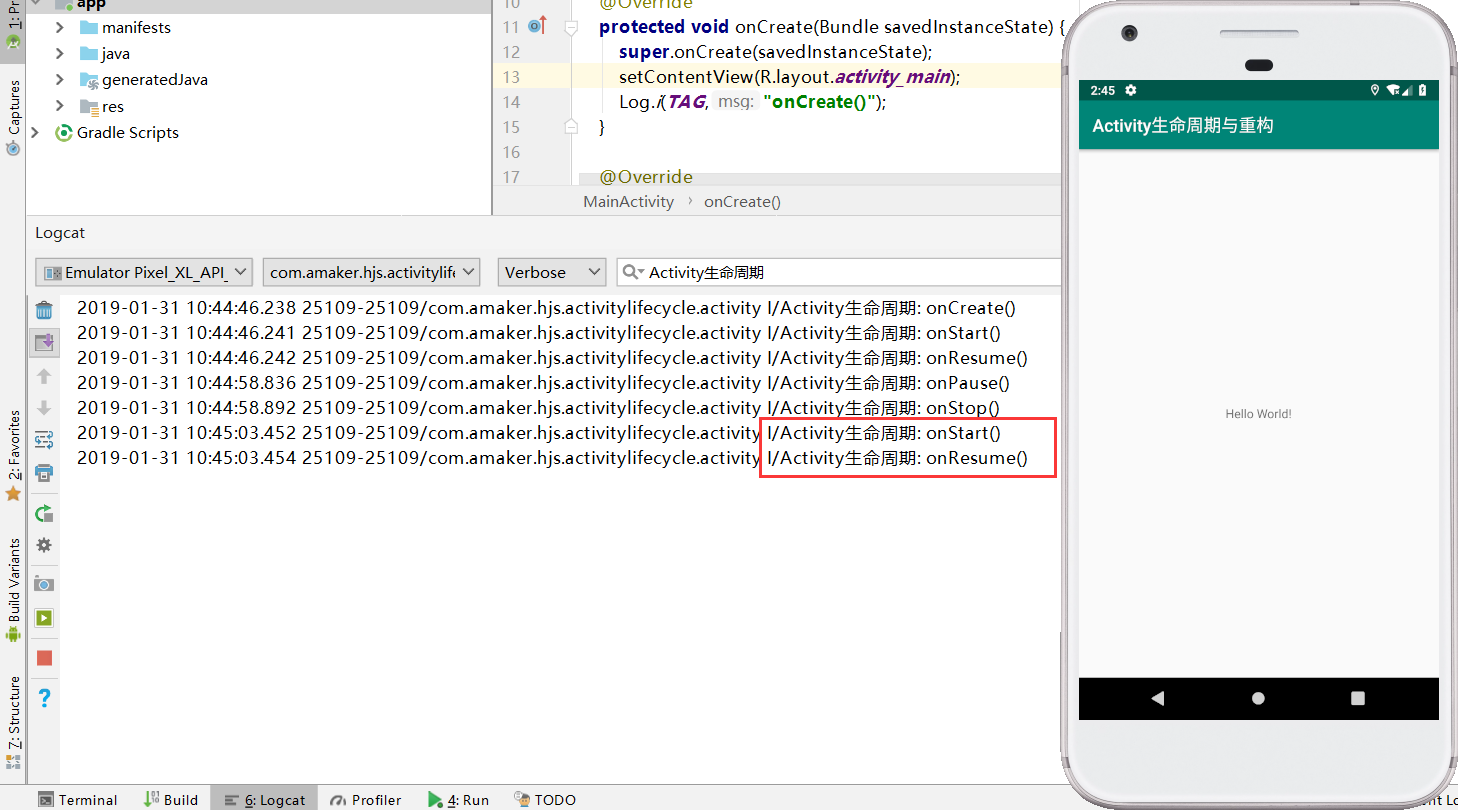
当一开始启动app程序时：



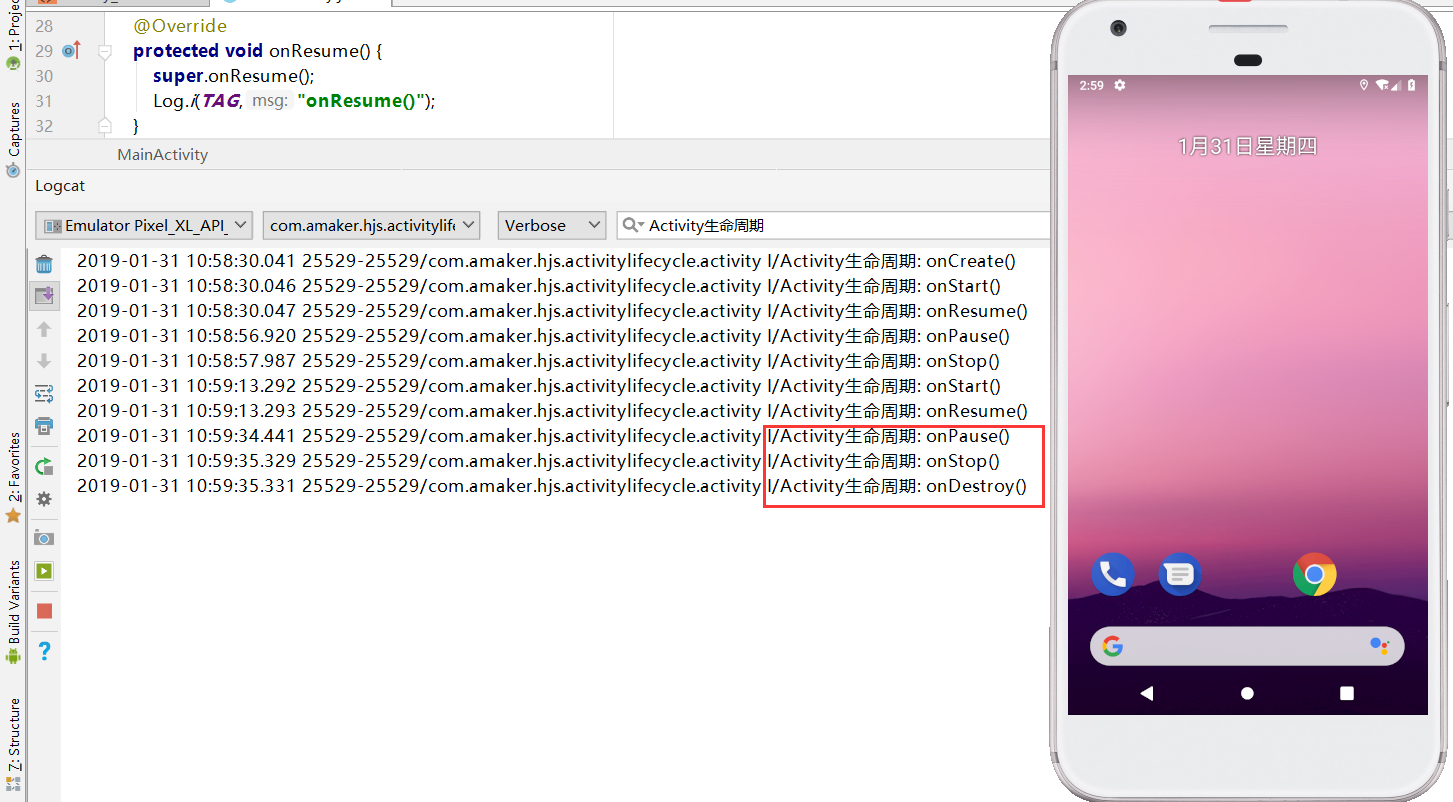
当点击home键/切换界面键，暂时切换app时：



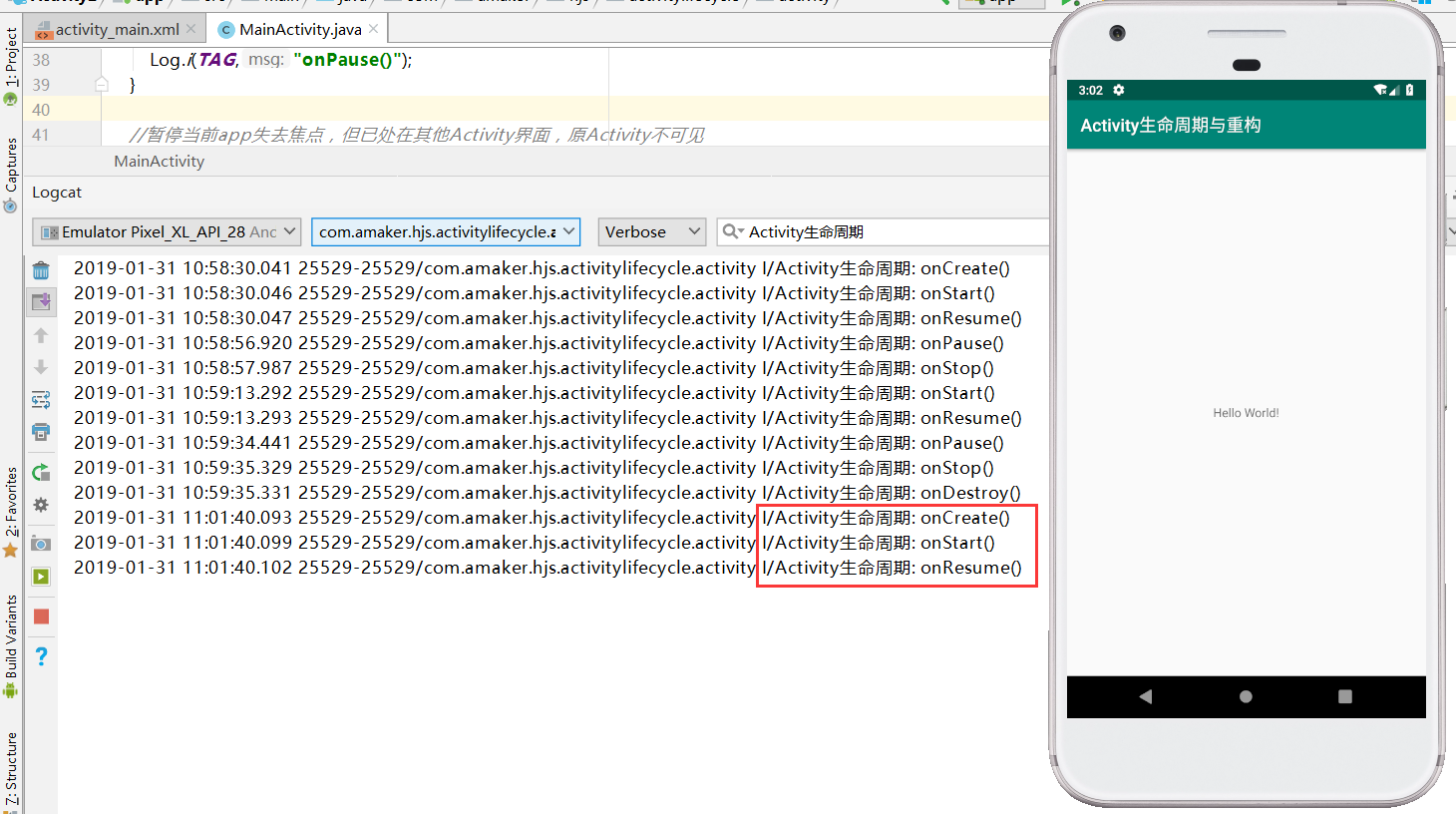
当再次回到app的界面时：



当点击关闭键/退出键，完全退出关闭app时：



当再次打开这个app（activity时）



activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"**>  
  
  
 <**Button  
 android:id="@+id/btnStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="42dp"  
 android:text="启动"** />  
  
 <**TextView  
 android:id="@+id/tvDisplay"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginBottom="300dp"  
 android:text="TextView"  
 android:textColor="#F00"  
 android:textSize="50dp"** />  
</**RelativeLayout**>

MainActivity.java

**package** com.amaker.hjs.activitylifecycle.activity;  
  
**import** android.os.Handler;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**import** org.w3c.dom.Text;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private static final** String ***TAG*** = **"Activity生命周期"**;  
  
 **private** Button **btnStart**;  
 **private** TextView **tvDisplay**;  
  
 **private int count** = 0;  
 **private boolean mainFlag** = **false**;*//标识用于控制线程是否运行* **private boolean isPause** = **true**;  
  
 **private** Handler **handler** = **new** Handler();*//在主线程中实例化Handler，表示这个Handler属于主线程的  
  
 //在Activity的生命周期中，只调用一次onCreate()* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 Log.*i*(***TAG***,**"onCreate()"**);  
 *//启动线程* **new** Thread(){  
 **public void** run(){  
 **mainFlag**= **true**;*//同意线程开始执行* **while**(**mainFlag**) {  
 **if** (!**isPause**) { *//是否暂停* **count**++;  
 **try** {  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 *//工作线程执行耗时任务，当任务完成时，会返回UI线程（主线程），再更新UI，有两种方法：  
 //（1）在子线程的代码中：  
 // handler.sendMessage(msg)  
 // 然后在主线程的Handler代码中接受消息进行处理  
 // 注意：handler是主线程的Handler对象，需要在主线程实例化  
 //（2）在子线程的代码中：  
 // handler.post(Runnable r)  
 // handler的post类方法允许我们排列一个Runnable对象到该handler所属的线程消息队列中，并重写Runnable的run()方法  
 // 所以这个handler必须在主线程中定义，属于主线程！* **handler**.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 *//运行在主线程中* **tvDisplay**.setText(**count**+**""**);  
 }  
 });  
 }  
 }  
 }  
 }.start();  
  
 **btnStart** = (Button)findViewById(R.id.***btnStart***);  
 **tvDisplay** = (TextView)findViewById(R.id.***tvDisplay***);  
  
 **btnStart**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//使线程暂停或继续运行* **isPause**=!**isPause**;  
 }  
 });  
 }  
  
 *//当Activity已经创建起来，下一次想要再运行时可以多次调用onStart()  
 //此时Activity仍不可见* @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 Log.*i*(***TAG***,**"onStart()"**);  
 }  
  
 *//运行时如果想要进行当前显示调用onResume()，获得焦点  
 //此时Activity可见* @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 Log.*i*(***TAG***,**"onResume()"**);  
 **isPause**=**false**;*//重新切换回到该Activity时再继续执行之前的工作* }  
  
 *//暂停当前app失去焦点，但处在当前Activity界面，可见* @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 Log.*i*(***TAG***,**"onPause()"**);  
 **isPause**=**true**;*//切换回到其他Activity时，暂停之前的工作* }  
  
 *//暂停当前app失去焦点，但已处在其他Activity界面，原Activity不可见，被隐藏* @Override  
 **protected void** onStop() {  
 **super**.onStop();  
 Log.*i*(***TAG***,**"onStop()"**);  
 }  
  
 *//在整个生命周期中只调用一次* @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 **mainFlag**=**false**;*//当Activity销毁时，将控制主线程的mianFlag设置为假，使得工作线程的循环结束* Log.*i*(***TAG***,**"onDestroy()"**);  
 }  
}