How to make runnable/thread pause when switching to next activity and then again resume when going back to same activity in which thread lies?

Activity\_main.xml

<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"*

android:paddingBottom=*"@dimen/activity\_vertical\_margin"*

android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*

android:paddingRight=*"@dimen/activity\_horizontal\_margin"*

android:paddingTop=*"@dimen/activity\_vertical\_margin"*

tools:context=*"com.example.waitnotifyandroid.MainActivity"* >

<Button

android:id=*"@+id/btnGoNextActivity"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"Go to next activity"* />

</RelativeLayout>

MainActivity.java

**package** com.example.waitnotifyandroid;

**import** android.content.Intent;

**import** android.os.Bundle;

**import** android.os.Handler;

**import** android.support.v7.app.ActionBarActivity;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.Toast;

**public** **class** MainActivity **extends** ActionBarActivity {

MonitorObject mSync = **new** MonitorObject();

Thread t;

**int** threadNameCounter = 0;

Runnable work;

Handler handler = **new** Handler();

Button btnGoNextActivity;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

btnGoNextActivity = (Button) findViewById(R.id.*btnGoNextActivity*);

btnGoNextActivity.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View arg0) {

// **TODO** Auto-generated method stub

startActivity(**new** Intent(MainActivity.**this**, NextActivity.**class**));

}

});

work = **new** Runnable() {

**boolean** myRunning;

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**synchronized** (mSync) {

myRunning = mSync.running;

}

**while** (myRunning) {

runOnUiThread(**new** Runnable() {

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

**synchronized** (mSync) {

**if** (mSync.mustBePost) {

Toast.*makeText*(MainActivity.**this**,

mSync.message, Toast.*LENGTH\_SHORT*)

.show();

mSync.mustBePost = **false**;

**if** (mSync.message

.equals("Main Activity is going to pause")) {

mSync.running = **false**;

}

}

Toast.*makeText*(getApplicationContext(),

"Thread is running", Toast.*LENGTH\_SHORT*)

.show();

}

}

});

**synchronized** (mSync) {

myRunning = mSync.running;

}

**try** {

Thread.*sleep*(1000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

};

t = **new** Thread(work, "My name is " + String.*valueOf*(threadNameCounter));

t.start();

// handler.postDelayed(work, 0);

}

@Override

**protected** **void** onPause() {

// **TODO** Auto-generated method stub

**super**.onPause();

**synchronized** (mSync) {

mSync.mustBePost = **true**;

mSync.message = "Main Activity is going to pause";

}

}

@Override

**protected** **void** onResume() {

// **TODO** Auto-generated method stub

**super**.onResume();

threadNameCounter++;

**synchronized** (mSync) {

mSync.running = **true**;

mSync.mustBePost = **true**;

mSync.message = "Main Activity is going to resume";

}

t = **new** Thread(work, "My name is " + String.*valueOf*(threadNameCounter));

t.start();

}

**public** **class** MonitorObject {

**public** **boolean** running = **true**;

**public** String message = "";

**public** **boolean** mustBePost = **true**;

}

}

* When activity switches from MainActivity to next Activity, onPause() method on MainActivity is called and again when switching back from NextActivity to MainActivity, onResume() method on MainActivity is first one to invoke and that’s where Thread starts.