**Dimitri**

* Application sample for Wifi direct (upload).

**Response**

* A step by step instruction on how to use the Android Wifi P2P is gathered in a nice tutorial on: <http://developer.android.com/guide/topics/connectivity/wifip2p.html> . Android provides in the Samples folder a sample app named: **WiFiDirectDemo**
* A fork of this app can be found in Github - <https://github.com/ahmontero/wifi-direct-demo> and allows you to send images via Wifi P2P.

**Yuri**

* Game sample app
* Game 3D

**Response**

* Check the sample app for SurfaceView game: **Samples/AnimatedSheep**  ( original sample (Romanian) - <http://www.androider.ro/tutorial-surfaceview-ul-si-oaia-care-voia-sa-alerge-3232> ).
* Check the most popular 3D game engine Unity - <http://docs.unity3d.com/Documentation/Manual/index.html> and also a great game programmed with Unity – Guerilla Bob: <https://play.google.com/store/apps/details?id=com.angrymobgames.guerrillabob&hl=de>

**Stepan**

* create the spam filter SMS
* how to secure the application

**Response**

* Check the sample app I created for you: **Samples/SpamFilter**
* Check my presentation on SlideShare - <http://de.slideshare.net/fastlink2/droidcon-eastern-europe-2013-how-secure-is-an-androidapp>

**Vladimir**

* create an app with with Progress bar in the top part
* transitions and animations

**Response:**

* Check the sample app located on: **Samples/TopBarProgressbar**
* Check the animation and the transitions tutorial : <http://www.vogella.com/tutorials/AndroidAnimation/article.html>

**Dasha**

* Networking app to send messages to more than one devices.

**Response:**

* The easy way is to use a server to distribute the messages via push notification to your peers.
* The not so easy solution is to create a server on your phone and to server these messages on your own – check the cool Socket example server from <http://examples.javacodegeeks.com/android/core/socket-core/android-socket-example/>
* The most elegant solution I would choose is a jSocket connetion via a external server, by far the fastest solution. A good example implementation can be found on <https://github.com/koush/AndroidAsync>

**Roman**

* use command tools from the Android application
* to send events - <http://www.pocketmagic.net/2012/04/injecting-events-programatically-on-android/#.UwW6OFt_tfU>
* TestRecorder

**Response**

* To send events check the tutorial from Radu M. - <http://www.pocketmagic.net/2012/04/injecting-events-programatically-on-android/#.UwW6OFt_tfU>
* Official UI testing <http://developer.android.com/tools/testing/testing_ui.html> with alternatives Robotium - <https://code.google.com/p/robotium/> and Testdroid/Testrecorder - <http://bitbar.com/>
* Introduction in monkeyrunner via <http://developer.android.com/tools/help/monkeyrunner_concepts.html>
* Command tools sample bellow (or live from **CC example from Marius**):

command line alternatives + java code alternatives:

adb shell dumpsys batteryinfo

public static void dumpBatteryInfo() {

try {

String cmd = "dumpsys battery";

Process script = Runtime.getRuntime().exec(cmd);

BufferedReader in = new BufferedReader(

new InputStreamReader(script.getInputStream()));

String line = null;

while ((line = in.readLine()) != null) {

Log.i ("BATTERY","Battery stats: " + line);

}

} catch (Exception ex) {

}

}

<uses-permission android:name="android.permission.DUMP" />

**Pavel**

* Simple spread sheat app
* Write your own dialer with conference in 1 click

**Response**

* A good example spreadsheat app is located at <https://github.com/dennis-sheil/android-spreadsheet> . A simple one cann be achieved using the TableLayout example in **Samples/05\_TableLayout**
* Not really so easy doable BUT again for everything it is a way maybe?

via <https://android.googlesource.com/platform/packages/apps/InCallUI/+/master/src/com/android/incallui/InCallPresenter.java>

CallCommandClient.getInstance().merge();

brings you to <https://android.googlesource.com/platform/packages/apps/InCallUI/+/master/src/com/android/incallui/CallCommandClient.java>

and this is using the aidl <https://android.googlesource.com/platform/packages/services/Telephony/+/master/common/src/com/android/services/telephony/common/ICallCommandService.aidl>

you can no TRY to control the <https://android.googlesource.com/platform/packages/apps/InCallUI/+/master/src/com/android/incallui/CallHandlerService.java>

**Nikolay**

* Cloud storage usage in the app

**Response**

* Cloud storage usage in the app via Backup API - <http://developer.android.com/training/cloudsync/backupapi.html> (requires Backup API registration)

**Vladimir**

* quiz app

**Response**

* Sample app is found in folder **Samples/QuizTrivia** and is based on the example from - <http://www.developerfeed.com/simple-quiz-game-andriod>

**Sergey**

* reflection, for something does not exists in the platform

**Response**

* Check the bellow example for the transitions between activities, the transitions are available since API level 5 only.

**public class GeneralActivity extends Activity {**

static Method mDebug\_overridePendingTransition;

**static {**

**initCompatibility();**

**};**

**private static void initCompatibility() {**

try {

mDebug\_overridePendingTransition = Debug.class.getMethod(

"overridePendingTransition", new Class[] { String.class } );

// success, this is a newer device

} catch (NoSuchMethodException nsme) {

// failure, must be older device

}

}

**public void overridePendingTransition(int enterAnim, int exitAnim){**

try {

mDebug\_overridePendingTransition.invoke(null, enterAnim, exitAnim);

} catch (Exception ite) {

}

}

......

//return on main activity

Intent intent = new Intent();

intent.setClass( getApplicationContext(), ApiAccountsMainActivity.class );

**overridePendingTransition(R.anim.fade, R.anim.hold);**

startActivity(intent);

finish();

As alternative you can also use sometimes code similar with the one bellow:

private int isNetworkNotificationDisabled() {

int status= 0;

int notification;

**if(Build.VERSION.SDK\_INT >= 4){**

notification = Settings.Global.getInt(context.getContentResolver(),

Settings.Global.WIFI\_NETWORKS\_AVAILABLE\_NOTIFICATION\_ON, 0);

} else {

notification = Settings.System.getInt(context.getContentResolver(),

Settings.System.WIFI\_NETWORKS\_AVAILABLE\_NOTIFICATION\_ON, 0);

}

if (notification == 0)

status = 1;

update(status);

return status;

}

**Ariana**

* Piano app

**Response**

* A hard one. I attached in **Samples/ChildPlay** a piano sample app derived from <https://github.com/piusvelte/childsplay>

**Anton, Andrey**

* Charting - pie, evolution

**Response**

* Check the official Android sample from **Sample/00\_CustomViewDraw**
* Check live with Marius the library he used in a banking app and a currency app ( <https://code.google.com/p/achartengine/> ). Check also other open source alternatives like <http://androidplot.com/docs/> .

strFromDate = extras.getString( "strFromDate" );

strEndDate = extras.getString( "strEndDate" );

strCurrencyCode = extras.getString( "strCurrencyCode" );

strTitle = extras.getString( "strTitle" );

setTitle(strTitle);

setTitleColor(Constants.COLOR\_ORANGE);

//get the values

valuesCurs = CurrencyUtils.getValuesChart(this, strCurrencyCode, strFromDate, strEndDate);

//display the values calculated

String[] horlabels = new String[] { strFromDate, strEndDate };

**GraphView graphView = new GraphView(this, valuesCurs, "" ,horlabels, null, GraphView.BAR);**

**setContentView(graphView);**

**Dimitri**

* telephony service, AIDL - send a request in hidden manner
* modify the android kernel (internals)

**Response**

* Check the live example from the **sample app DS**
* If you want to modify compile the classic android kernels check <http://source.android.com/source/building-kernels.html> if you are interested in a x86 kernel check <http://www.android-x86.org/documents/customizekernel>
* Check live with Marius the instructions from the chapter **Building Kernel for the Emulator (Goldfish)** from the Android Internals Course.

**Alexander**

* stream music

**Response**

* There are many way music in 2014 is streamed. One of the favorite way to do it is using the **HLS** protocol (variable bitrate, AES encoding etc) but also the classical http progressive streaming remains a great way to do it. A good example is located in **Samples/MediaPlayer** and is based on the example from <http://androidstreamingtut.blogspot.co.uk/2012/08/custom-progressive-audio-streaming-with.html>

**Marina**

* How to work to SMS
* What we should do to improve the security of the app

**Response**

* Check the sample app I created for you: **Samples/SpamFilter** and watch with me some extra coding in **DS app**
* Check my presentation on SlideShare - <http://de.slideshare.net/fastlink2/droidcon-eastern-europe-2013-how-secure-is-an-androidapp> . A good ideea is to have a look also at the project dexguard - <http://www.saikoa.com/dexguard>

**Andrey**

* hidde application from task manager
* enable some functionality by SMS remotely

**Response**

* From taskbar you can hide activities as officialy described on <http://developer.android.com/guide/topics/manifest/activity-element.html#exclude> by using the activity attribute: *android:excludeFromRecents="true"*
* Check the sample app I created for you: **Samples/SpamFilter** and watch with me some extra coding in **DS app**