

Pavlin.si -> same zanimive stvari

Šola, MTB, potepanja, elektronika, foto

Višinski diagram	Manjka GPX zapis.
------------------	-------------------

Archos 101 internet tablet

Pred časom sem bolj iz firkca, kot zares pokukal na stran <http://developer.android.com/index.html> z namenom, da pogledam, kako android zgleda v praksi. S tem seveda ne mislim uporabniško prakso, ampak s strani izdelave aplikacije za android platformo.

Po prvem navdušenju je sledilo razočaranje, saj sem ugotovil da kar nekaj stvari ni podrtih v emulatorju. Razvoj je namreč podprt s sicer odličnim emulatorjem, ki zna marsikaj, ne zna pa npr. uporabljati bluetooth strojne opreme, ki je na PC-ju. Po svoje je to tudi logično, saj se hardware na PC-jih zelo razlikuje od modela do modela, na resnični tablici pa za to poskrbijo hardverski nivoji operacijskega sistema (torej proizvajalec tablice).

Zaradi teh omejitev sem začel razmišljati o nabavi tablice. Najprej sem gledal **najcenejše**, češ, saj bom samo poskusil, pa verjetno nikoli nič zares rabil. V tej kategoriji [so tile modeli](#). Pa tudi modeli z [začetka tega seznama](#).

Potem sem malo pogledal, kaj je na voljo v bolj **“razsutem” stanju**. V to kategorijo padejo npr:

IGEPv2

BeagleBoard

ali celo kakšna bolj **obskurna izvedba**.

še najbolj pa mi je ustrezala [tale “huda” platka](#). Za ceno raje ne vprašajte 😊

Potem sem malo razmišljal, da bo na koncu to spet ena nova **razvojnja plata** brez prave uporabnosti za vsakodnevno delo, saj bi vsem tem platom moral dodati še LCD, baterijo, ohišje... ce bi hotel imeti prenosno napravo. Sicer je **samogradnja** dober stimulacijski faktor, vseeno pa se je ustavilo pri ceni, saj za isti hardver v lepem ohišju z velikim LCD in v povsem delujočem stanju lahko odštejem manj, kot za samo eno razvojno “golo” ploščico. Samogradnja popolnoma iz “nule” seveda odpade, saj je to v zglednem času mission impossible, ker so v glavnem vsi cortex-M8 v takih BGA ohišjih,

ki za garažno delo niso najbolj prijazni.

Potem sem malo pobrskal po raznih nemških, angleških in ameriških "ceneje.si"-jih in na koncu izsjal proizvajalca **archos**, ki je dobro podprt s strani uporabnikov (forumi, recenzije, testi), ima vse, kar je potrebno za razvoj aplikacij, obenem pa ni predrag ob kar največji kvaliteti (glede na razne spletne ocene). Zastavil sem si še zahtevo, da mora imeti LCD s čim večjo ločljivostjo in kapacitivni touch, saj je bojda uporaba androida brez multi-touch GUI bolj dolgačasna.

Zadnje cene modela **archos 101 s 16GB** notranjega flasha so tukaj.
Pa še model **archos 101 z 8GB** notranjega flasha.

Od archos-a obstaja še model s 7" LCD z 8GB in model z 250GB diskom, ki je bolj namenjen kakšnim filmofilom.

So pa tudi še manjši, npr. s 4,3" LCD ali celo 2,8" LCD, s **ceno pod 100€**.

Potem sem naročil model **101 s 16GB** in v nekaj dneh je bil dostavljen preko UPS.

Sledilo je seznanjanje in prvi preizkus....

Prvi občutek, ko sem tablico prijel v roke je bil malenkost "kitajski". Velik LCD ima obrobo iz tanke eloksirane aluminijaste pločevine, ki je bila na sredini malenkost privzdignjena, ravno toliko, da je zgledalo čudno. Nežen odločen pritisk na dveh mestih je stvar uredil. Sledilo je polnjenje baterij in vklop.

Že iz forumov in opisov sem vedel, da tablica nima uradnega dostopa do android marketa, ki je spletno skladišče aplikacij za androida. Ker je tablica prišla z mnameščenim androidom 2.2 (froyo), sem sledil navodilom na [tej strani](#). **Android market** je bil takoj za tem dostopen, tako da uradna nedostopnost **ne pomeni nikakršne ovire**.

Sledila je ena huda napaka in sicer **inštalacija angry birds**. Seveda so otroci sloneli poleg, ko sem tablico "oživljal" in po namestitvi ptičev je sledil doooooolg premor v uporabi tablice. Ko sem si jo **v naslednjih dneh priboril nazaj**, sem počasi začel s priklopom na PC z namenom zagona lastnega "hello world" programčka.

Tule bom opisal, kaj sem naredil, da sem s PC-ja zagnal "hello world" na tablici.

Najprej je potrebno namestiti **razvojno okolje**. Najenostavneje je uporabljati eclipse. Okolje je malenkost samosvoje, vendar spet ne toliko, da se ne bi dalo navaditi uporabe v nekaj urah dela ob pomoči kakšnega step-by-step priročnika, ki jih je veliko.

1. Najprej rabimo **JDK** (JRE ni dovolj!)
2. potem namestimo **eclipse**. Jaz sem namestil najnovejšo različico **indigo**. Sicer je potrebno izbrati Eclipse Classic.
3. Sledi namešcanje **android SDK**.

Do tu smo uporabili namestitvene programe, ki so delovali samodejno. Sledi prilagajanje eclipse za delo z android-om.

4. Sedaj moramo "dopovedati" eclipse, da bomo delali z androidom. Za ta namen je potrebno namestiti Android Development Tools (ADT). To storimo iz samega eclipse. Za naslednje korake proedpostavljam, da smo namestili eclipse 3.5, 3.6 ali 3.7.

4.1. Zaženi eclipse in izberi **Help > Install New Software**, izberi gumb "**Add**"

4.2. V dialogu vpiši v prvo polje (Name): **ADT Plugin**, Za Location vpiši: <https://dl-ssl.google.com/android/eclipse/>

4.3. V dialogu "**Available Software**" obkljukaj "**Developer Tools**" in pritisni **Next**

4.4. Pojavi se seznam, kjer samo klikneš **Next**

4.5. Izpiše se licenca. **Potrdi, da sprejemaš pogoje** in pritisni **Finish**

4.6. Po namestitvi ADT zapri in **ponovno zaženi eclipse**.

5. Po namestitvi ADT je potrebno tega še nastaviti. To narediš tako, da izbereš v glavnem menuju **Window > Preferences** in na levem seznamu klikneš na **Android**

5.1. Pri vnosnem polju "**SDK Location**" klikni na gumb "**Browse**" in poišči direktorij, kjer je nameščen SDK. V WIN XP je privzeto na lokaciji **C:\Program Files\Android\android-sdk-windows** v drugih win-sih pa bo drugje.

5.2. Pritisni **Apply** in **OK**

Evo, to je to. Sedaj lahko začnemo... ali pa tudi ne

Mogoče za zacetek še kakšna **beseda o androidu**. Vse, kar pišem tule, sem spoznal v glavnem na [teh straneh](#), tako da tule le povzemam zapiske, ki sem si jih naredil kot **popolni začetnik**. Ker vse skupaj vsaj za nekoga, ki je vsaj malo povohal orodja za razvoj softvera sploh ni nekaj posebej zapletenega, pa tudi za vse ostale, ki se želijo preizkusiti v programiranju, je uporaba opisanih orodij relativno enostavna naloga.

Android je v bistvu neka nadgradnja linux 2.6, ki naredi pisanje aplikacij popolnoma neodvisno od strojne opreme preko standardiziranih knjižnic in na njih slonečega aplikativnega okostja (application framework). To še najlepše ponazori tale slika:



Do rdečega nivoja običajno nimamo dostopa, se pa "vsevedi" mladci radi pohvalijo, da so "rootali" nek telefon. To ne pomeni nič drugega, kot da je omogočen superuser terminalski dostop do linux operacijskega sistema. Uporabniško ime je v tem primeru "root" - od tod tudi glagol "rootati".

Kakorkoli, mi se bomo tule ukvarjali z nivoji, ki ležijo nad rdečim na siki zgoraj, ko bomo pa v Društvu elektronikov pripravili svojo prvo razvojno ploščico z npr. TMS320C6A8167 pa se bo nekdo moral pozabavati tudi s tem nivojem.

Načeloma aplikacije pišemo v javi, izjemoma pa tudi v C/C++ s pomočjo [android NDK](#). Slednjega še nisem poskusil, tako da bo opis tega počakal še kakšen dan ali dva.

Priprava target-a (ciljnega sistema) za zagon nove aplikacije

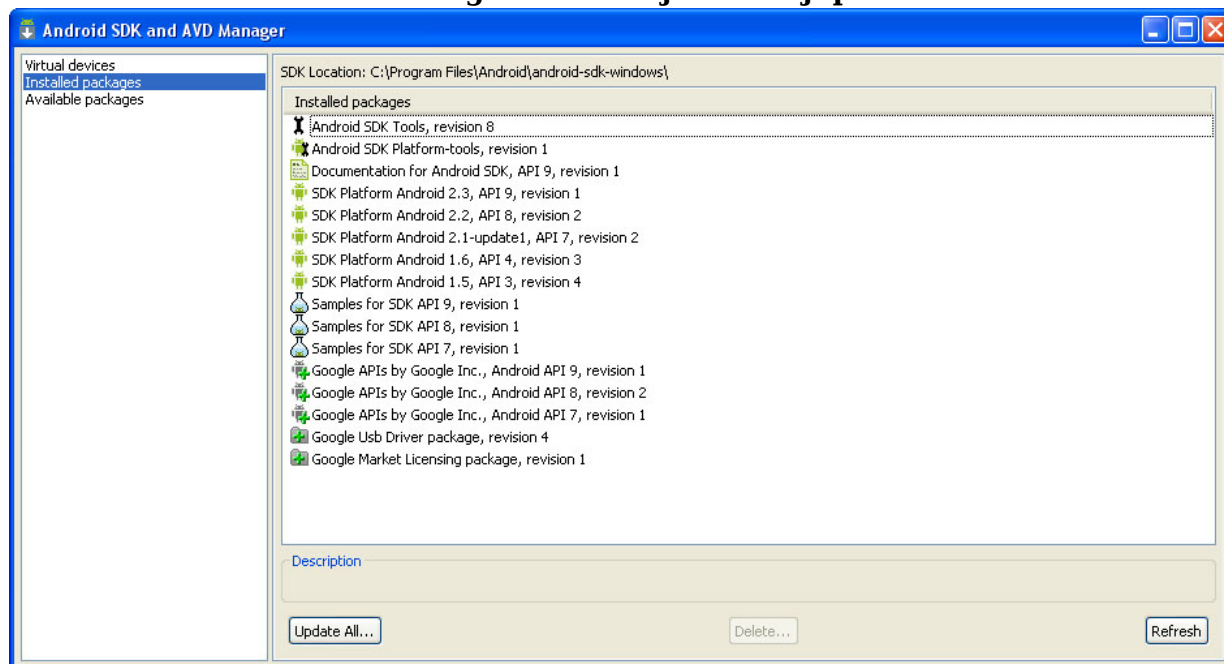
Za preizkus naše bodoče "hello world" aplikacije potrebujemo najprej ciljni sistem, na katerem bomo to preizkusili. Pisanje programa za windows-e je v tem pogledu enostavna, saj imamo razvojno orodje in ciljni sistem na istem računalniku. Preizkušanje android aplikacije pa je bolj podobno programiranju mikrokontrolerjev. V bistvu je tablica embedded sistem, na katerem deluje android, tako da smo bolj v "domačih" - elektronskih vodah kar se tice samega dela s sistemom.

Pri izbiri ciljnega sistema imamo dve možnosti. Lahko izberemo emulator, ki je del SDK ali pa dejansko tablico (hardware).

Če izberemo **emulator**, moramo najprej ustvariti **Android Virtual Device (AVD)**. Zaženemo "Android SDK and AVD Manager". Lahko kar direktno iz eclipse (**Window > Android SDK and AVD Manager**) ali pa poiščemo SDK manager (zeleno ikonco) -

pri meni na XP je to "**C:\Program Files\Android\android-sdk-windows\SDK Manager.exe**" Ob prvem zagonu SDK managerja moramo namestiti nekaj komponent, ki omogočajo delo z SDK.

Za delo z archos tablico so zagotovo dovolj naslednji paketi:



Da dopolnimo seznam, izberemo na levi **available packages** in pokljukamo komponente, da se ujemajo z zgornjo sliko.

Če tablice nimamo, s pomočjo AVD vseeno lahko preizkusimo aplikacijo na virtualnem sistemu. V SDK managerju izberi **Virtual devices** in pritisni **New**. Vnesi **ime** in **verzijo** androida, ki ga želiš podpreti z virtualno napravo, določi **velikost virtualne SD kartice** in **ločljivost LCD-ja** na virtualni napravi. Spodaj lahko dodaš **podporo raznim dodatkom**, ki jih podpira android, npr. **tipkovnico, GPS, akcelerometer** in podobno.

Tako ustvarjeno virtualno napravo potem **uporabimo kot ciljni sistem pri zagonu naše nove aplikacije** (o tem kasneje).

Bolj kot virtualna pa je seveda zanimivejša resnična tablica.

Za priklop archos-a moramo storiti dvoje

1. prenesti **USB gonilnik za archos-a**. Ta se nahaja tukaj. Sledi priklop na USB. Windowsi seveda zajamrajo, da je na USB neznana naprava. Izberemo ročno namestitev gonilnika in pokažemo na lokacijo na disku, kamor smo razpakirali USB gonilnik z zgornje povezave. Več podrobnosti **glede na verzijo WIN**, ki jo uporabljaš je tule. Jaz sem sledil korakom za namestitev na XP in je delovalo iz prve.
2. Na tablici sem še pognal **settings** in izbral **Application settings->Development** in obkljukal "**USB Debugging**". Obkljukal sem še "**Stay awake**", da se mi v času, ko je tablica priključena na USB oz. polnilec ne ugaša LCD.

Koda in uporabniški vmesnik

Sedaj, ko je vse priključeno in pripravljeno, se pa res lahko **lotimo "hello world"**.

Poženi eclipse in zaženi čarodeja: **File > New > Project**.

Izberi **Android > Android Project , Next**

Za **Project Name** vpiši **HelloWorld**

Izberi **Create new project in workspace**

Obkljukaj **Default location**

Obkljukaj **Android 2.2** (če boš testiral na archos tablici)

Pod **application name** vnesi poljubno ime aplikacije

Pod **package name** vnesi svojo domeno, jaz sem dal test.pavlin.si, lahko je pa karkoli, vsebovati mora vsaj eno piko

Pod **main activity** vnesi **HelloWorld**, je pa to glavna *Activity*, ki se požene ob zagonu programa.

Pritisni **Next** in **Finish**.

Na levi se pokažejo direktoriji:

src: tu so .java izvorne datoteke. V našem primeru: **HelloWorld.java** - naš glavni program

gen: **R.java** generira prevajalnik - tega ne urejaj.

Android Library: tu je le ena datoteka: **android.jar** in vsebuje vse potrebne knjižnice.

res: tu so vsi viri (resource) za našo aplikacijo. Vsebuje pod-direktorije, ki si jih bomo ogledali v nadaljevanju

AndroidManifest.xml - tu nastavimo lastnosti aplikacije, npr. dovoljenja, ki jih potrebuje aplikacija (npr. dostop do SD kartice ali uporaba bluetooth).

Sedaj lahko ustvarimo uporabniški vmesnik (GUI). Android omogoča ustvarjanje GUI s pomočjo kode ali pa z uporabo XML. Slednje je priročno, saj lahko učinkovito ločimo delovna je programa od izgleda. V pod-direktoriju **res/layout** odpri datoteko **main.xml**. Urejanje je lahko grafično ali tekstovno. Za tekstovno urejanje **klikni spodaj** pod urejevalnikom na **main.xml** in pokaže se koda:

Koda:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
    />
</LinearLayout>
```

Več podrobnosti je v [tej dokumentaciji](#).

Sedaj **popravi XML** tako, da **dodaš gumb in vnosno polje**:

Koda:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />

<EditText
    android:id="@+id/txt_name"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    />

<Button
    android:id="@+id/btn_clickme"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/btnclickmetext"
    />

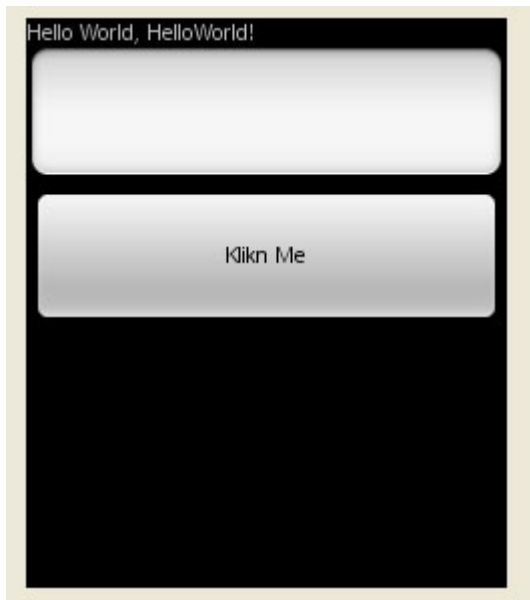
</LinearLayout>
```

V direktoriju **res/values** popravi **strings.xml**:

Koda:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, HelloWorld!</string>
    <string name="btnclickmetext">Klikn Me</string>
    <string name="app_name">Helou</string>
</resources>
```

Sedaj mora GUI izgledati takole:



`<EditText>` prikazuje **besedilo**, `<Button>` pa uporabimo za “klikabilni” **gumb**. Ve o widgetih je [tukaj](#).

Sedaj lahko dodamo še **programsko kodo**, ki nadzoruje **delovanje** naše mini aplikacije:

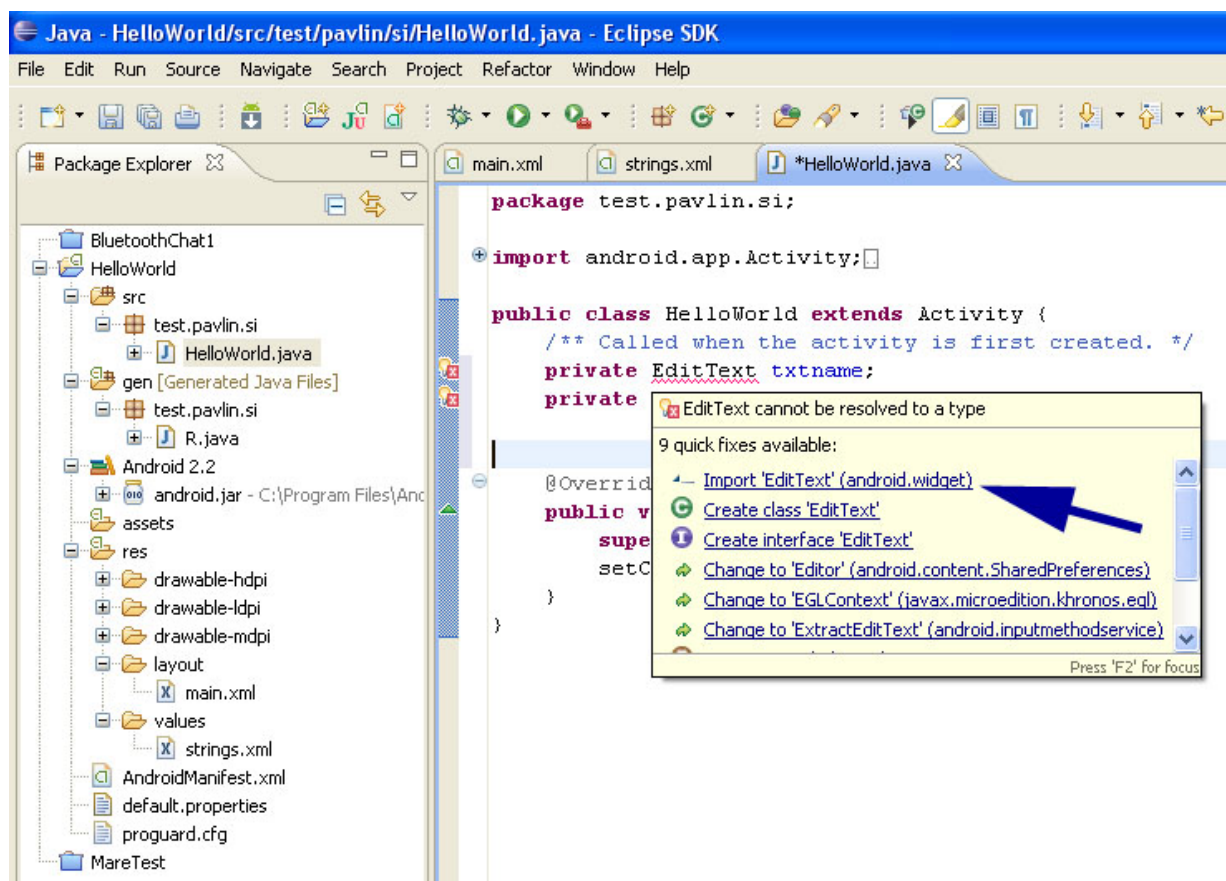
Najprej odpri datoteko s kodo **HelloWorld.java** Sam se z java nisem nikoli kaj resno ukvarjal. Kolikor razumem stvari je Activity v bistvu en ekran, ki se trenutno prikazuje in nekaj počne. Je pa stvar zelo podobna C#... Lahko pa da se motim. Tule bo potrebno še nekaj branja 😊

Kakorkoli, najprej **deklariramo objekte iz GUI**:

Koda:

```
private EditText txtname;
private Button btnclickme;
```

Ker se java sproti prevaja, takoj vidimo, da je z novima objektoma nekaj narobe. **Z miško greš nad ime objekta in pojavi se menu**, ki ponudi rešitve. Izberi **prvo opcijo, ki doda referenco** na ustrezno knjižnico:



Isto naredi še za "knof".

Vnosno polje poiščemo med widgeti našega ekrančka s pomočjo funkcije findViewById():

Koda:

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    txtname = (EditText) findViewById(R.id.txt_name);
    btnclickme = (Button) findViewById(R.id.btn_clickme);
}
```

Tu lahko vidimo, kako s pridom uporabljamo samo-zaključevanje kode (auto complete).

Na koncu dodamo še kodo, ki se izvede ob kliku na gumb (prikaz malega popup okna) in končni izgled celotne kode je tak:

Koda:

```
package test.pavlin.si;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class HelloWorld extends Activity {
    /** Called when the activity is first created. */
    private EditText txtname;
    private Button btnclickme;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

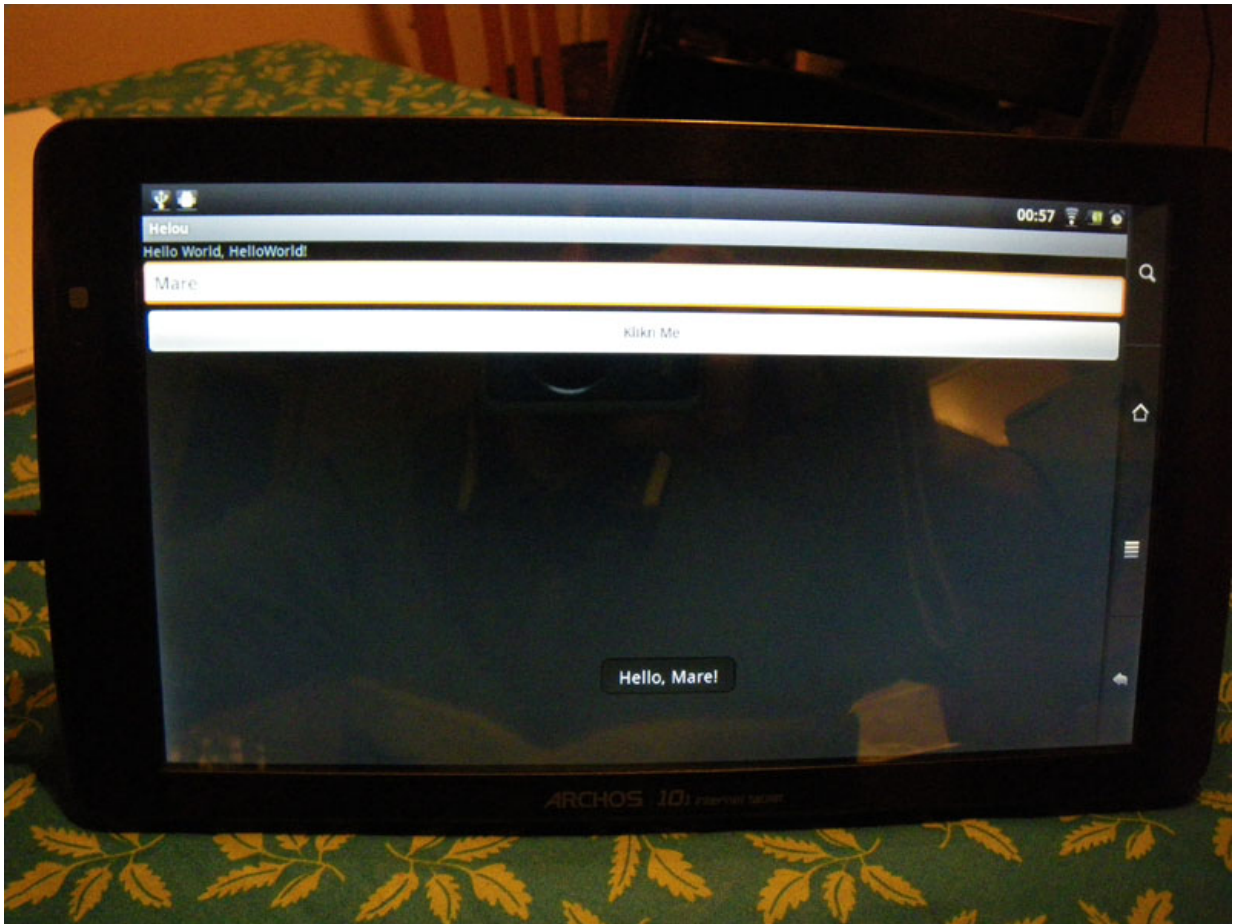
        txtname = (EditText) findViewById(R.id.txt_name);
        btnclickme = (Button) findViewById(R.id.btn_clickme);

        btnclickme.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                String str = "Hello, " +
                    txtname.getText().toString() + "!";
                Toast.makeText(getBaseContext(), str,
                    Toast.LENGTH_SHORT).show();

            }
        });
    }
}
```

Na koncu v eclipse pritisni **Run** ali **Ctrl-F11** in če je vse OK, se na tablici požene naš nov programček:



Stay tuned for more....