

Mobilne tehnologije, razvoj alata i aplikacija

- MIT App Inventor -

Autori:

Sebastian Firi, Veleučilište Velika Gorica

Bruno Valić, Veleučilište Velika Gorica

Tomislav Gligora, Veleučilište Velika Gorica

Sažetak

Danas potražnja za programerima biva sve jača te mnoge razvijene zemlje koje se suočavaju s nedostatkom kvalificiranog radnog kadra, traže načine da osiguraju bolju edukaciju vlastitih stanovnika u budućnosti.

Tokom 2013.g tehnološki tržišni lideri poput Amazona, Microsofta i Googlea su u suradnji s Code.org neprofitnom organizacijom započeli projekt naziva „The Hour of Code“, gdje imaju za cilj upoznati milijune učenika i studenata s osnovama software developmenta. U međuvremenu su izbačena nekolicina rješenja (Khan Academy, CodeAcademy i sl.) koji upravo imaju cilj educirati korisnike o razvoju aplikaciju, i općenito informatičkim tehnologijama. Jedno od najpoznatijih rješenja je i „MIT App Inventor“.

„MIT App Inventor“ je web aplikacija otvorenoga kôda koja je proizvedena od strane Googlea, a sada u vlasništvu Massachusetts Institute of Tehnology (MIT-a). Aplikacija ili bolje rečeno – rješenje, omogućuje zainteresiranima da na veoma jednostavan način, pomoću slaganja vizualnih blokova (objekata), kreiraju aplikacije za AndroidOS. Prednosti ovoga rješenja su zaista mnoge, korisnici prije svega nisu primorani instalirati dodatne aplikacije (IDE sučelja) na računalo da mogu koristiti App Inventor. Potrebit' je samo web-preglednik. „App Inventor“ se nalazi u cloudu čime se stiče doza komfornosti - s kojeg god računala da korisnik pristupa, dočekati će ga isto!

Detaljnije informacije, kao i primjer izrade jedne jednostavne aplikacije za smartphone pogonjen AndroidOS, prikazati će se u radu.

Uvod

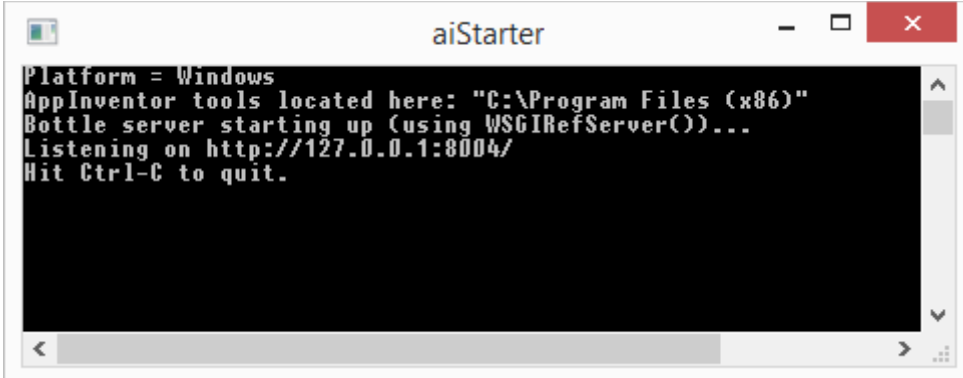
MIT App Inventor se iskazuje kao jedan od najboljih rješenja za edukaciju učenika, kao i studenata bez znanja ili krhkog znanja programiranja. Koristi grafičko sučelje koje je veoma slično Scratchu (konkurent na tržištu) ali i nešto naprednijim rješenjima poput Eclipse-a. Sučelje se može podijeliti na tri dijela, počevši s lijeve strane gdje nalazimo „*Pallete*“ (*Toolbox*), odnosno alate koje se *drag'n'drop* metodom prebacuju na sredinu preglednika, na prikaz ekrana mobilnog uređaja. S desne strane se nalaze komponente i svojstva (*Properties*) odgovarajućih alata.

MIT App Inventor se nalazi na web-adresi: <http://ai2.appinventor.mit.edu/>, važno je napomenuti da je potrebno imati Google račun kreiran, da bi se pristupilo usluzi!

Kao što je spomenuto nije potrebno instalirati nikakve dodatne alate kako biste programirali za AndroidOS, no što ukoliko kasnije želite i testirati novo-kreirane aplikacije a nemate Android mobilni uređaj?! Tu postoji potencijalno jedini problem ovoga alata, a koji je zapravo savršeno riješen! Možete pokrenuti Emulator koji prikaže ekran Android pametnog telefona, i kroz par trenutaka (nakon boot-anja) pokrene vašu aplikaciju koja se nativno nalazi na „virtualnom“ uređaju. Da biste imali mogućnost simulacije, potrebno je skinuti datoteku:

<http://appinventor.mit.edu/explore/ai2/setup-emulator.html>

Nakon instalacije aplikacija se pokreće pritiskom na ikonu te ukoliko je sve prošlo u redu, dobiva se sljedeći prikaz:

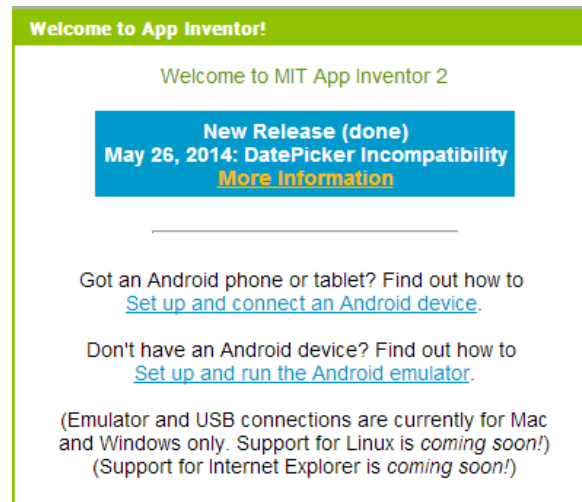


```
aiStarter
Platform = Windows
AppInventor tools located here: "C:\Program Files (x86)"
Bottle server starting up (using WSGIRefServer())...
Listening on http://127.0.0.1:8004/
Hit Ctrl-C to quit.
```

Slika 1. Prikaz konzole Emulatora

Primjer izrade aplikacije

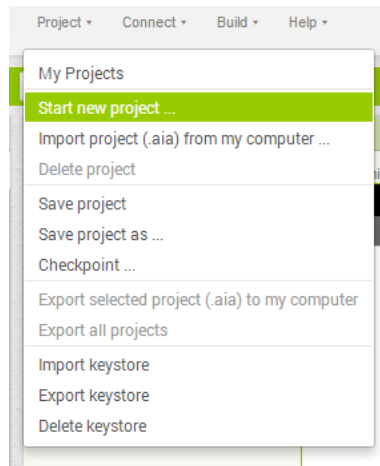
Ukoliko je zadovoljen preduvjet da je Google račun kreiran, može se započeti s kreiranjem aplikacije. Prilikom ulaska prikazuje se tzv. „Welcome Screen“ (Slika 2.) u kojemu sučelje pruža pomoć oko spajanja postojećeg smartphone uređaja, ili kao što je navedeno u prethodnome tekstu – kako pokrenuti Emulator.



Slika 2. Prikaz dobrodošlice

Koristeći App Inventor primjećuje se da se pomoć pruža na svakom koraku. U obliku vodiča (*Tutorial, Guide* i slično) ili u obliku „pop-up“ prozora. Također kao i u slučaju drugih široko korištenih aplikacija, postoje različite community zajednice i forumi gdje se može zatražiti prijedlog ili pomoć.

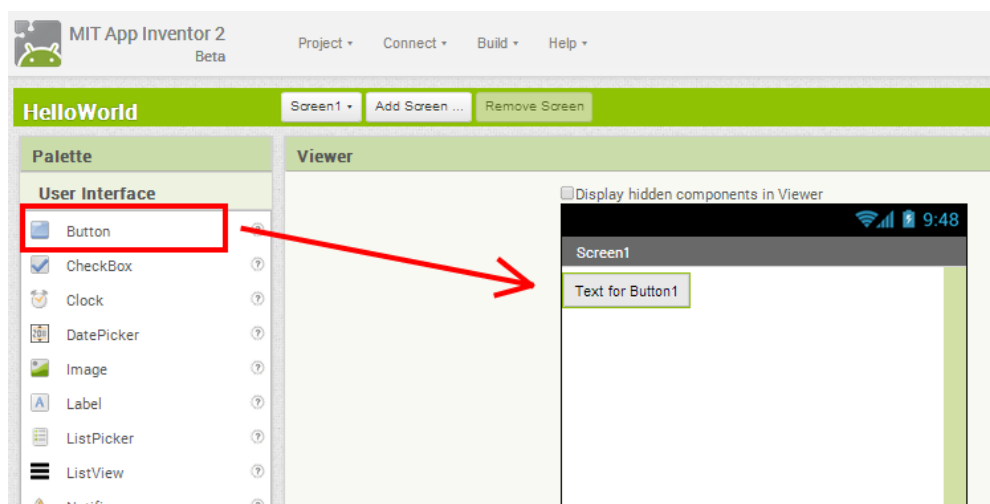
Prvi korak ka izradi aplikacije je kreiranje novog Projekta. Isto se čini pritiskom na 'Project' u gornjem-lijevom kutu, te zatim -> 'Start new project' (Slika 3.). U sljedećem koraku je potrebno upisati naziv, te se dobiva početni prikaz *Designera*.



Slika 3. Prikaz kreiranja novog projekta

Za zadatak će se učiniti aplikacija koja je namijenjena djeci predškolske dobi, kojom bi se olakšalo korištenje smartphone uređaja i omogućilo u najkraće mogućem roku kontakt s roditeljem. Ide se s ciljem minimizirati kompleksnost korištenja telefona i uz pomoć jednog dodira omogućiti djetetu da pošalje SMS (informaciju) roditelju.

Kroz 'Designer' sučelje pridodaju se svi alati koji su potrebni. Prvi korak je *drag'n'drop* metodom dovući *Button* na ekran mobilnog uređaja, u sredini aplikacije. Na Slici 4. je prikaz učinjenog, no potrebno je *Button* još izmijeniti i personalizirati, što se može uraditi mijenjajući svojstva (*Properties*). Krenuti s izmjenom naziva gumba, u polje „Text“ upisati riječ 'Mama', a monotonu sivu boju promijeniti klikom na „BackgroundColor“ i odabrati željenu boju, u ovom slučaju roza (*Magenta*). Radi jednostavnosti i preglednosti poželjno je izmijeniti naziv komponenata (npr. *Button1* → *zoviMamu*), čiji je način prikazan na Slici 5.



Slika 4. Prikaz dodavanja Button alata drag'n'drop metodom

Prednost grafičkog sučelja alata je i u 'live' izvedbi, tj. izmjene koje se učine su momentalno vidljive u *Viewer*-u. Nakon izmjene boje kroz „*BackgroundColor*“ poželjno je povećati komponentu *zoviMama*, radi preglednosti. Pod „*Width*“ je potrebno upisati brojčanu vrijednost u pikselima (u ovom primjeru = 315px) a pod visinu, odnosno „*Height*“ vrijednost = 150px. Da bi estetski dojam bio bolji, izmjenjuje se i boja slova, pritiskom na „*TextColor*“. Izabrati bijelu boju. Oštre rubove preoblikovati u zaobljene pomoću postavke „*Shape*“, u čijem se ispadajućem izborniku izabire 'rounded'. Preostalo je još da povećati veličinu slova, sa zadane 14.0 vrijednosti na 48.

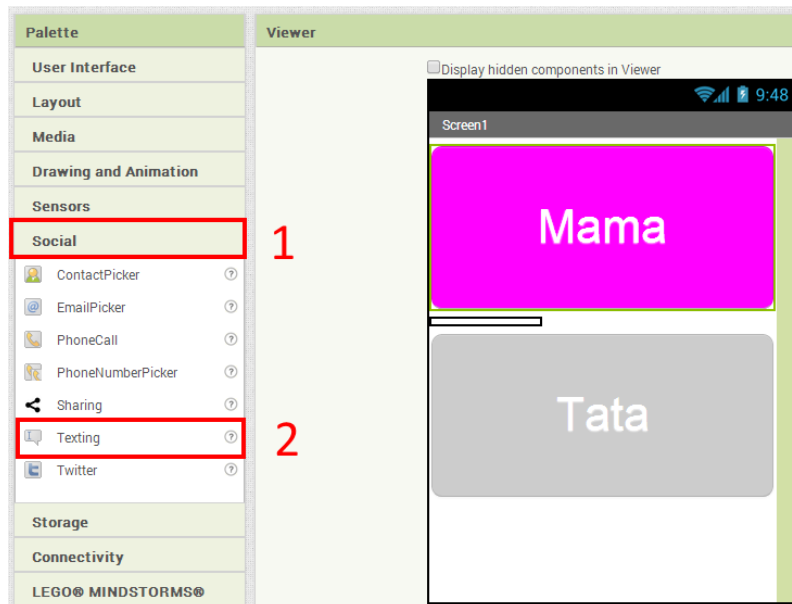


Slika 5. Prikaz preimenovanja komponente

Sličan postupak je potrebno učiniti i u sljedećem koraku, gdje se pridodaje *Button*-u naziv 'Tata'. Font slova i veličina se mijenja kroz svojstva (*Properties*). Što se tiče uređivanja, preostalo je još diferencirati (lagano udaljiti) dvije tipke. To se čini na veoma jednostavan način, s lijeve strane klikom miša odabrati „*Layout*“ te spomenutom „*drag'n'drop*“ metodom pridodati „*VerticalArrangement*“. Kao što je rečeno na prethodnom primjeru, radi jednostavnosti izmijeniti naziv komponente u *vertikalniRazmak*. U svojstvima, vrijednost visine „*Height*“ sa zadane postaviti na vrijednost = 5px.

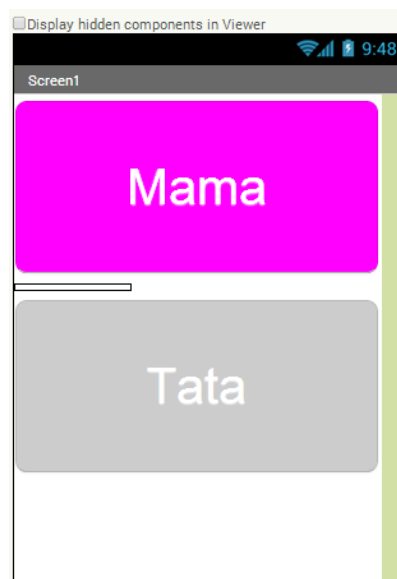
Da se korisniku omogući slanje SMS poruke, klikom se pridodaje „*Social*“ stavka s lijeve izbornika. Bitno je naglasiti da se za sada radi cijelo vrijeme u „*Designer*“ mode-u. Iz izbornika pridodaje se „*Texting*“ stavka dva puta (s obzirom da su u pitanju dva primatelja

poruke), koja spadaju u 'non-visible' komponente, odnosno komponente koje se prikazuju na ekranu ali su „ispod haube“. One u pozadini omogućuju i izvršavaju svoje zadaće. Na Slici 6. se nalazi prikaz dodavanja Texting komponente, kojoj je potrebno(poželjno) izmijeniti naziv.



Slika 6. Prikaz dodavanja *Texting* komponente

Nakon pridodavanja „*Texting*“ stavke potrebno je u svojstvima podesiti primatelja SMS poruke, kao i njen sami sadržaj. U ovom primjeru u polje „*Message*“ je upisano 'Mama nazovi me. Hvala'. Dotični tekst će jednim pritiskom gumba biti proslijeđen majci (ili čiji je broj naveden kao primatelj). Na slici 7. je prikazan dosadašnji proces:

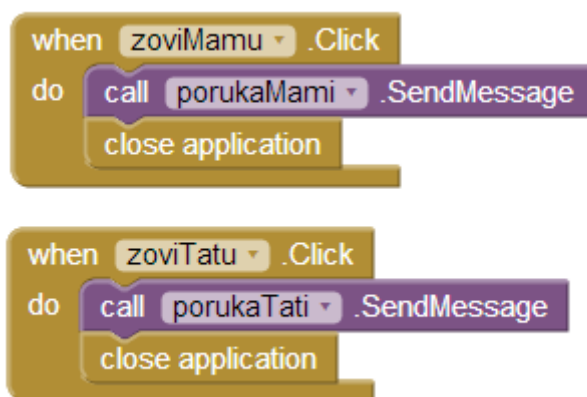


Slika 7. Prikaz gotovog designerskog dijela

Za sada su rađene isključivo designerske preinake, razmještanje *Button*-a, mijenjanje fonta slova, boje i sl. U ovom koraku pridodaje se programska logika. Klikom na „*Blocks*“ u gornjem-desnom kutu ekrana ulazi se u developerski dio aplikacije. Ovaj prikaz se sadrži od dva dijela, blokova s lijeve strane, te „platna“ odnosno *Viewer*-a gdje se postavljaju ti blokovi. Lijevim klikom miša na svaki od već ugrađenih blokova se pruža popis svih mogućnosti. Klikom na „*Control blocks*“ iskazuje se mogućnost pravih naredbi, petlji koje se koriste kod različitih programskih jezika. Primjerice klikom na „*Lists blocks*“ prikazuju se i polja (*Arrays*) koja se mogu koristiti u aplikaciji.

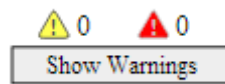
U primjeru ove jednostavnije aplikacije koriste se blokovi koji su pridodani prilikom prvog dizajnerskog dijela. Kako bi se sa sigurnošću znalo koji blok koristiti, treba se prisjetiti osnovnog cilja aplikacije, u ovom slučaju: slanje SMS poruke **nakon** korisnikova klika (dodira) na *Button*. Shodno ovime s lijeve strane izbornika izabire se komponenta '*zoviMamu*' te *drag'n'drop* metodom povlači prvi blok naziva '*when zoviMamu.Click do*'. Uviđamo da je ostavljen prostor (prazna bjelina) u kojoj je moguće pridodati novi blok. U ovom slučaju nakon dodira potrebno je poslati SMS, odnosno pozvati funkciju za slanje SMS poruke. Ona se nalazi pod grupom blokova '*porukaMami*' i naziva je '*call porukaMami.SendMessage*'.
close application'.

Zadatak je korisniku nakon slanja zatvoriti aplikaciju, kako bi nesmetano nastavio koristiti mobilni uređaj. Isto se čini uz pomoć kontrolnog bloka '*close application*'. Isti postupci se čine i za komponentu '*zoviTatu*'. Prikaz platna na kraju treba izgledati ovako:



Slika 8. Prikaz programske logike aplikacije

Izrada aplikacije je sada pri kraju, no potrebno je provjeriti da li je uspješno napisan kôd, tj. da li su blokovi ispravo posloženi. U donjem lijevom kutu ekrana se nalaze indikatori sa brojačima. Obvezno je da nema nikakvih 'Warninga' jer u protivnom aplikacija neće raditi, neće se moći pokrenuti. Na slici 7. je prikaz indikatora gdje je vidljivo da nema upozorenja.

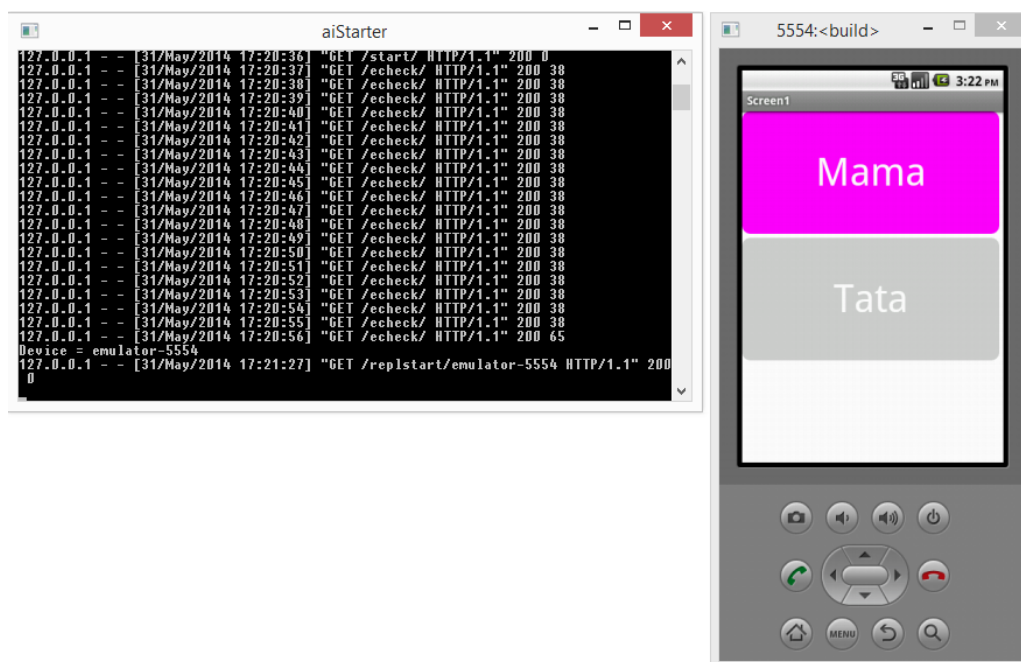


Slika 7. Prikaz indikatora

Zaključak

Aplikacija je sada uspješno završena. Ukoliko učenik posjeduje Android mobilni uređaj može ga slobodno spojiti putem USB sučelja na računalo, ili pokrenuti Emulator. U gornjem lijevom kutu se odabire 'Connect' - > 'Emulator' i ukoliko je student, odnosno učenik, ispravno učinio prethodne korake, dobiva rezultat prikazan na Slici 8.

Korištenjem App Inventor-a stiže se dojam velike perspektive i korisnosti koju alat posjeduje. Na veoma lak način učeniku se pruža mogućnost izrade aplikacije, koje zasigurno daje uvid u osnovne programiraja, ali i „vjetar u leđa“ za daljnje učenje programskih jezika!



Slika 8. Pokrenuta konzola Emulatora s grafičkim prikazom ekrana

Literatura

1. O'REILLY', *App Inventor: Create Your Own Android Apps*, David Wolber, Hal Abelson, Ellen Spertus, Liz Looney, 2011.,
ISBN-10: 1449397484
2. *App Inventor for Android [online]*. Dostupno na:
<http://en.wikipedia.org/wiki/App_Inventor_for_Android>
3. *Getting Started with MIT App Inventor [online]*. Dostupno na:
<<http://appinventor.mit.edu/explore/get-started>>
4. *The MIT App Inventor Library: Documentation & Support [online]*. Dostupno na:
<<http://appinventor.mit.edu/explore/library>>