

RAD Studio: izrada mobilnih aplikacija

Željko Kovačević
Tehničko veleučilište u Zagrebu
e-mail: zeljko.kovacevic@tvz.hr

Sažetak

Za razvoj mobilnih aplikacija obično su se koristili alati koji su specifični za samo jednu od mobilnih platformi. Tako su programeri najčešće bili prisiljeni unaprijed se opredijeliti za iOS ili Android mobilne aplikacije jer izrada za obje platforme najčešće je značila dvostruki posao. Zato se u ovom radu opisuje najnoviji pristup izradi mobilnih aplikacija pomoću RAD Studija koji omogućava da pomoću Delphi i C++ Builder razvojnih okruženja programeri izrađuju aplikacije za obje platforme upotrebljavajući isti programski kod. Nadalje, taj isti programski kod može se upotrijebiti za izradu Windows i OS X desktop inačica tih mobilnih aplikacija čime se izvlači maksimum u iskoristivosti jednog programskog koda. Ova tehnologija u RAD Studiju dostupna je tek od 2013., tj. 2014. godine za C++ Builder, te kao takva može biti zanimljiva širokom spektru korisnika (programera).

1. Uvod

Neprestanim razvojem mobilnih tehnologija mobilni uređaji postali su više od samih uređaja za telefoniranje. Štoviše, tehnologija je toliko napredovala da današnju generaciju mobilnih uređaja nerijetko zovemo „pametnim telefonima“. Takvi uređaji zapravo su kombinacija osobnih džepnih računala i mobilnih uređaja. Primjena im je raznolika, pa osim za telefoniranje možemo ih koristiti kao rokovnike, preglednike dokumenata i elektroničke pošte, za zabavu (igre) itd. Sve je to moguće zahvaljujući mobilnim aplikacijama koje korisnici pametnih telefona mogu koristiti, a one im olakšavaju svakodnevne poslove i radnje. Mobilni uređaji s Android, iOS i Windows Phone platformama najviše su rasprostranjeni na tržištu te ne čudi da je broj aplikacija za ove platforme u stalnom porastu. Štoviše, i sama se potražnja za mobilnim aplikacijama raznih namjena povećava pa i razvojnih alata za mobilne aplikacije ima sve više i više. Stoga će u ovom tekstu biti opisan jedna od najnovijih skupova alata za izradu mobilnih aplikacija za Android i iOS platforme zvan RAD Studio.

Korištenjem RAD Studio skupa alata programeri su u mogućnosti razvijati mobilne aplikacije za Android i iOS platforme. Međutim, ono što je svakako najzanimljivije jest da se za obje mobilne platforme koristi isti programski kod. Programeri su čak u mogućnosti iskoristiti taj isti programski kod da bi napravili desktop inačice tih aplikacija za Windows i Mac operativne sustave. RAD Studio alati razvijaju se gotovo 20 godina, no tek unazad par godina ova je tehnologija dostupna programerima, što ju čini jednu od najnovijih tehnologija koju treba spomenuti i analizirati.

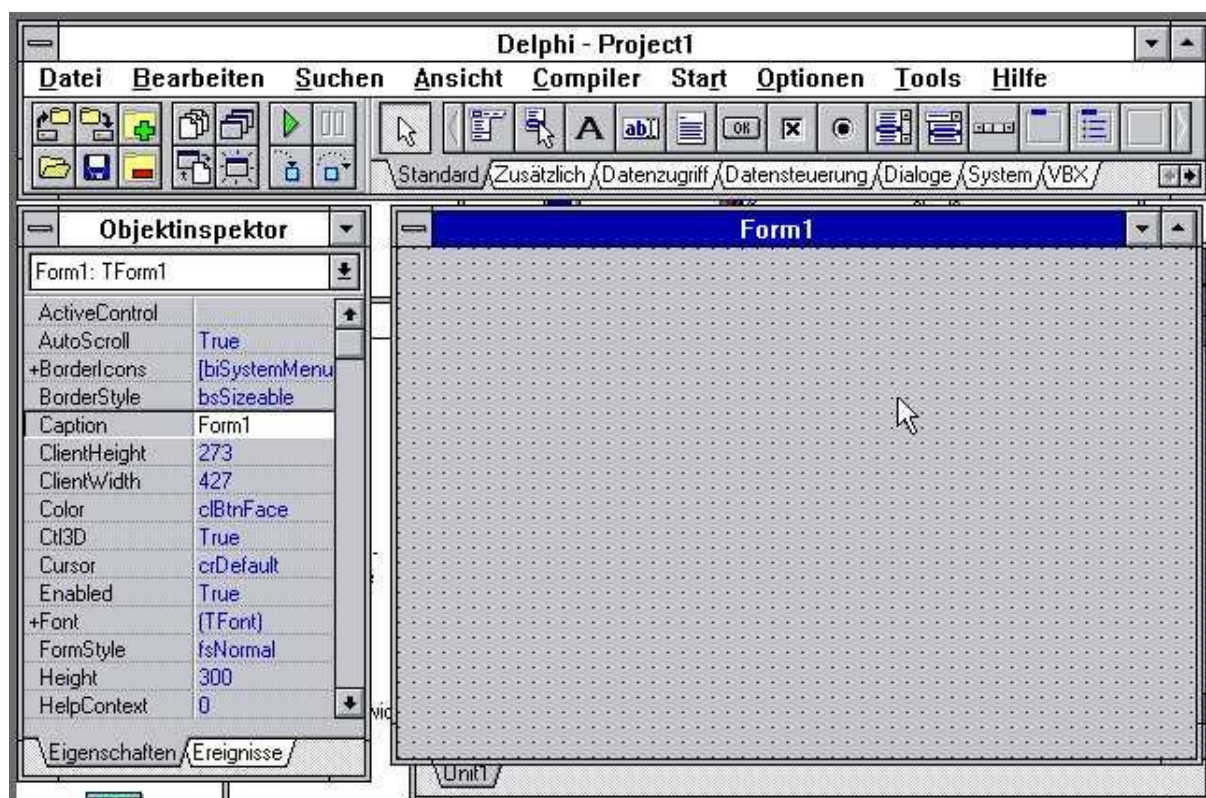
2. Povijest RAD Studio alata

Kada je riječ o RAD (*eng. Rapid Application Development*) Studio skupu alata tada se primarno misli na Delphi i C++ Builder razvojna okruženja. Po izgledu i mogućnostima jako su slični te oba nude mogućnost razvoja izvornih (*eng. native*) mobilnih aplikacija za Android i iOS platforme. Međutim, u samim počecima ovi alati nisu bili niti približno spremni za podržavanje mobilnih platformi pa ćemo kroz kratku povijest ovih alata vidjeti kako se tehnologija razvijala i kako se došlo do podrške za mobilne platforme.

Tvrtka	Razvojno okruženje	Inačica	Godina	Biblioteka	Platforma
Borland	Delphi	1	1995	VCL	Windows
Borland	C++ Builder	1	1997	VCL	Windows
Borland	Kylix	1	2001	CLX	Linux
Embarcadero	Delphi	XE2 (16)	2011	VCL, FM	Windows, OS X
Embarcadero	C++ Builder	XE2 (16)	2011	VCL, FM	Windows, OS X
Embarcadero	Delphi	XE5 (19)	2013	VCL, FM	Windows, OS X, iOS, Android
Embarcadero	C++ Builder	XE6 (20)	2014	VCL, FM	Windows, OS X, iOS, Android

Tablica 1. Povijesni pregled razvoja Delphi i C++ Builder razvojnih alata

Sve je počelo još 1983. godine izlaskom Turbo Pascala 1.0 tvrtke Borland. Turbo Pascal predstavljao je razvojno okruženje u kojemu se moglo uređivati programski kod, pokrenuti prevoditelj te vidjeti greške. Programeri su ga vrlo brzo prepoznali kao jednostavno, brzo i jeftino rješenje za razvoj aplikacija, nakon čega Turbo Pascal prevoditelj postaje jedan od najbolje prodavanijih prevoditelja svih vremena. Ovo je ujedno bio i razlog velikoj popularnosti Pascala na PC platformi. (1)



Slika 1. Delphi 1.0 – novi projekt

Preuzeto sa: <http://delphi.cz/page/Verze-Delphi.aspx>

Nakon nekoliko uzastopnih nadogradnji Turbo Pascala, Borland je 1995. izdao Delphi 1. Ovo razvojno okruženje zamišljeno je za vizualni razvoj aplikacija upotrebljavajući objektno orijentirani Pascal. U to vrijeme Delphi 1 je podržavao 16 bitne Windows aplikacije te je tek u inačici 2 podržavao 32 bitne aplikacije.

Delphi se temeljio na VCL (*eng. Visual Component Library*) biblioteci koja je bila u pripremi od 1993 godine. Već prve skice biblioteke definirale su način na koji će se pristupati razvoju aplikacija. Primjerice, nisu se koristili pokazivači, a nad komponentama su se mogle definirati

akcije za određene događaje (*eng. events*). Također, svaka komponenta imala je i svoju skupinu svojstava (*eng. properties*) koji su je opisivali itd. (2)

Nedugo nakon izlaska Delphija koji je bio baziran na Pascalu, tvrtka Borland izdaje C++ Builder 1997. Iako C++ Builder ima jako puno sličnosti s Delphijem, tj. identičnu razvojnu okolinu i VCL biblioteku, C++ Builder koristi programski jezik C++. Na ovaj je način tvrtka Borland uspjela C++ programerima ponuditi sve pogodnosti Delphijeve VCL biblioteke. Štoviše, C++ Builder u mogućnosti je prevoditi Delphijev programski kod, pa time i potpuno podržavati Delphi komponente. Također, sličnosti Delphija i C++ Buildera bile su vidljive i pri pisanju programskog koda.

Nakon nekoliko uzastopnih izdanja Delphija i C++ Buildera tvrtka Borland htjela je napraviti biblioteku koja bi omogućila podršku i za Linux operacijski sustav. Stoga su 2001. godine izdali Kylix 1.0 koji je koristio CLX biblioteku. Kylix je bio razvojno okruženje za Linux s podrškom za par specifičnih distribucija: Mandrake, Red Hat, SuSE. (3) Pod Windowsima se u Delphiju i C++ Builderu moglo kreirati CLX projekt pa se tako moglo prevesti taj projekt za Windows platformu, a u Kylixu za Linux platformu. Međutim, Borland je od Kylixa brzo odustao zbog slabog odziva programera. Nakon inačice 3 Kylix nije više bio razvijan.

Ubrzo nakon toga tvrtka Borland mijenja poslovnu politiku maksimalno se posvećujući izradi baza podataka te drugim specifičnim programskim rješenjima. Razvoj Delphi i C++ Builder alata bio je zapostavljen te su u konačnici razvojni alati prešli u vlasništvo ogranka CodeGear kojeg je 2008. kupila tvrtka Embarcadero. (4) Nakon ovog ključnog trenutka razvoj Delphija i C++ Buildera naglo se ubrzava. Oba alata izlaze u redovnim vremenskim intervalima pružajući punu programsku podršku te sve više mogućnosti.

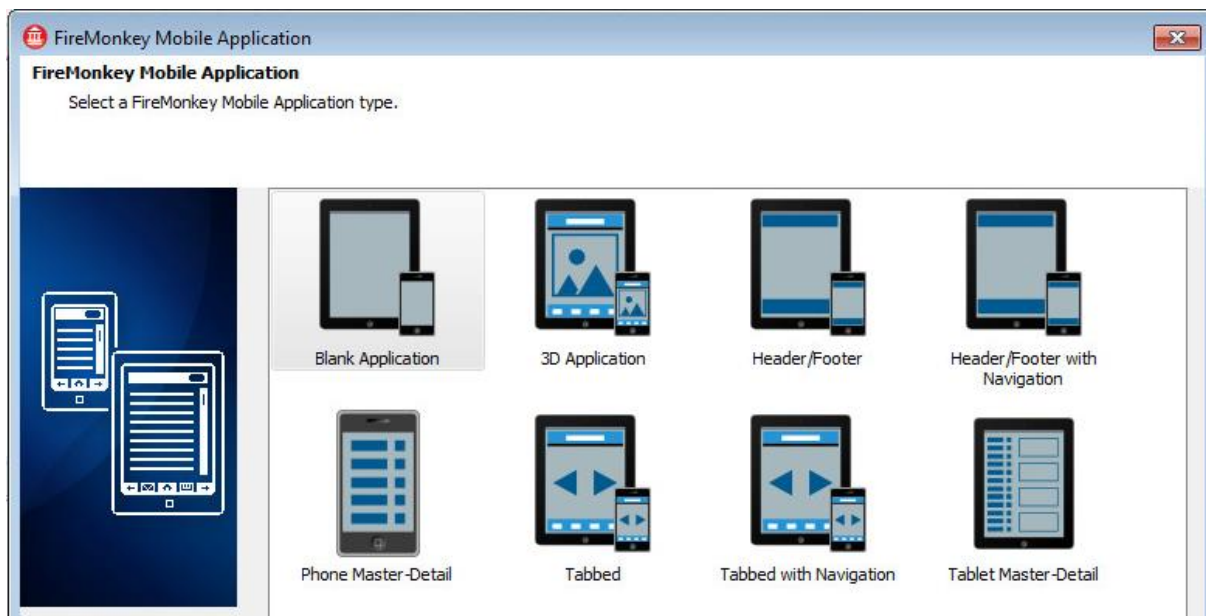
Ubrzo Delphi pruža unicode podršku, kao i podršku za 64 bitne aplikacije, dok 2011. izlaskom Delphija XE2 Embarcadero predstavlja biblioteku FireMonkey. Ova biblioteka postaje baza za aplikacije koje se izvršavaju na različitim platformama počevši od Windows i Mac platforme (Delphi i C++ Builder XE2) do kasnije iOS i Android mobilnih platformi (Delphi XE5). C++ Builder je podršku za Android Firemonkey aplikacije dobio u narednoj inačici XE6.

3. FireMonkey mobilne aplikacije

Biblioteka FireMonkey predstavljena je s izlaskom RAD Studija XE2. U to vrijeme je donosila podršku za Windows i Mac platforme, a postupno je dodana podrška za iOS i Android. Od njene prve verzije biblioteka FireMonkey nudi dva tipa aplikacija, tj. HD (*eng. High Definition*) i 3D aplikacije. Da bi to omogućila, ova biblioteka ima cijeli niz karakteristika i mogućnosti. Štoviše, najvažnije karakteristike biblioteke FireMonkey bile bi sljedeće: (5)

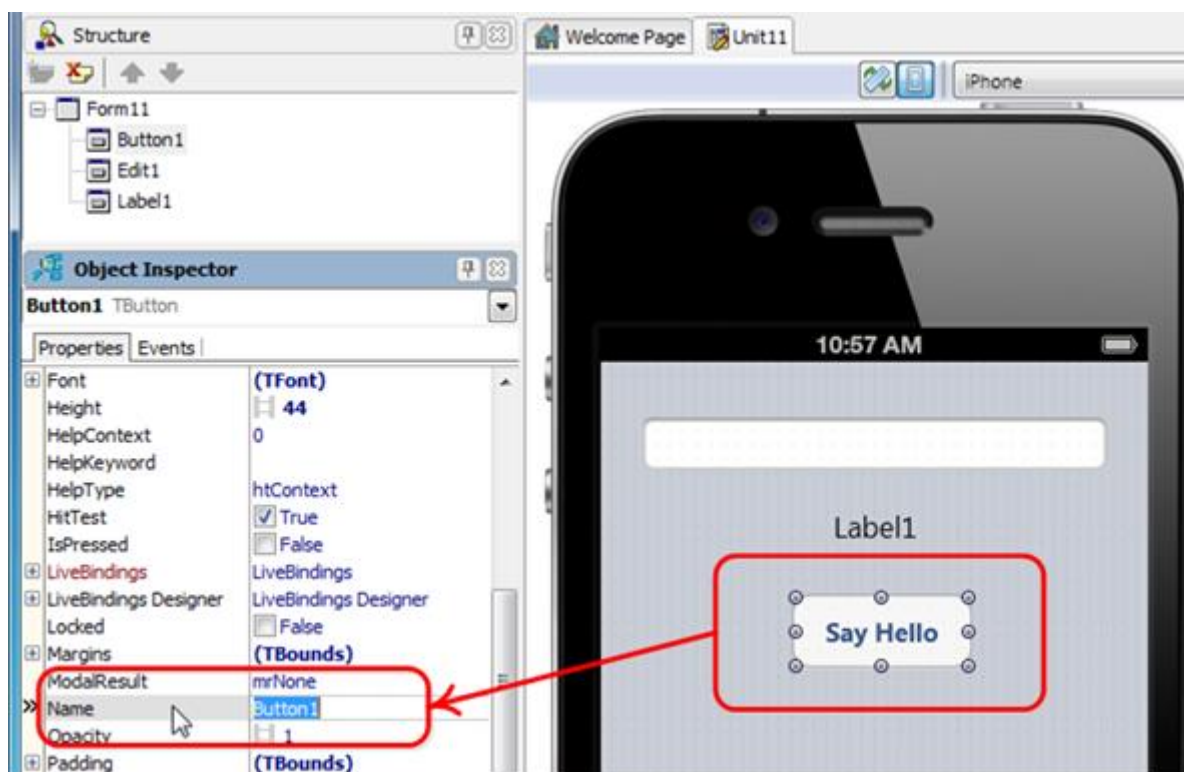
- podrška za prozore, izbornike, brojače vremena i dijaloge na više platformi
- 2D i 3D grafika
- podrška za vektorsku grafiku
- brza AA vektorska grafika u realnom vremenu, neovisna o rezoluciji
- WYSIWYG dizajner i uređivač
- napredan GUI - prozori, gumbi, editori, liste itd.
- GUI teme
- primitivni oblici za 2D grafiku s podrškom za geometriju i transformacije
- napredne animacije; jednostavne za korištenje i precizne sa minimalnim opterećenjem procesora i automatskom korekcijom broja sličica po sekundi
- efekti u slikama uključujući sjene.
- fleksibilni rasporedi komponenti i oblika
- podrška za Unicode
- podrška za čitanje i pisanje JPEG, PNG, TIFF, i GIF formata
- višejezična podrška, uređivač i primjeri

Sve navedene mogućnosti i karakteristike biblioteke FireMonkey mogu se upotrijebiti i pri pisanju mobilnih aplikacija, pa tako i one mogu koristiti napredne grafičke elemente, tj. 2D, 3D i vektorsku grafiku. Da bi napravili mobilnu aplikaciju u RAD Studiju potrebno je kreirati *FireMonkey Mobile Application* projekt.



Slika 2. Kreiranje mobilne aplikacije u RAD Studiju

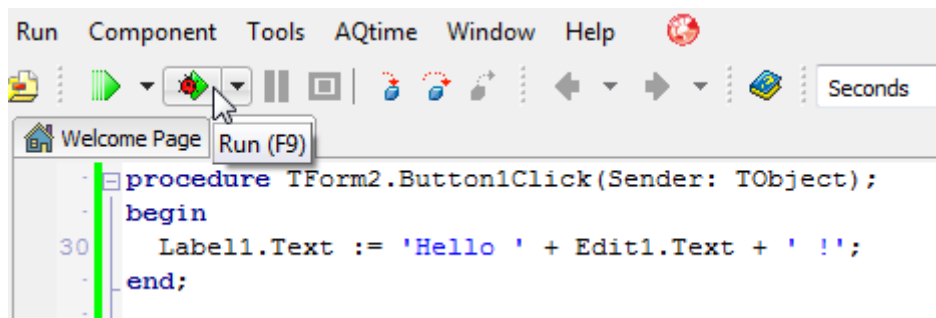
Početni dijalog na slici nam nudi već neka predefinirana sučelja za našu mobilnu aplikaciju. Tako to može biti 3D mobilna aplikacija, aplikacija s podnožjem i zaglavljem, navigacijom itd. ili jednostavno prazna (eng. *blank*) mobilna aplikacija.



Slika 3. Dizajner za mobilnu aplikaciju

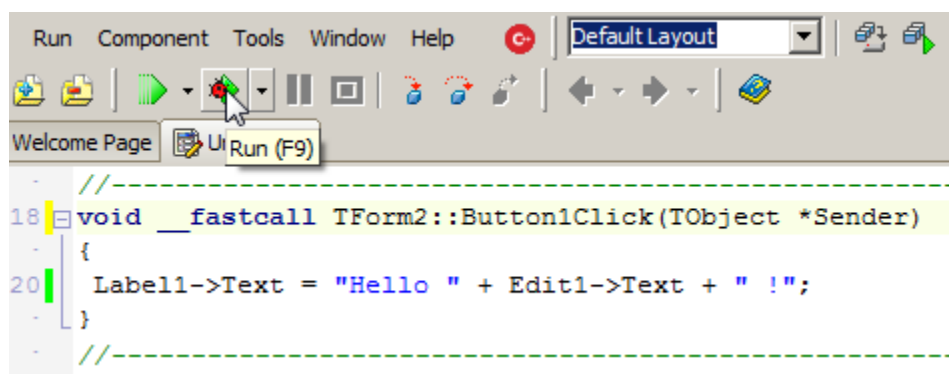
U svrhu demonstracije nismo koristili nikakvo predefinirano sučelje, odnosno odabrali smo *Blank Application*. Ovisno o odabranom tipu mobilne platforme (iOS ili Android) pojavljuje se odgovarajući dizajner. Dizajner automatski prikazuje jedan od ponuđenih mobilnih uređaja za odabranu platformu. Na taj način pri pisanju mobilne aplikacije možemo odmah vidjeti kako bi ona izgledala kada bi se izvršavala na stvarnom mobilnom uređaju. U slučaju iOS platforme možemo odabrati iPhone mobilni uređaj, a u slučaju Android platforme jedan od mobilnih uređaja tog tipa. U oba slučaja ponuđeno je više mobilnih uređaja kako bi korisnik (programer) mogao prilagoditi svoju mobilnu aplikaciju određenom mobilnom uređaju ili više njih, uzimajući u obzir i same dimenzije (rezoluciju) tih mobilnih uređaja.

Iz slike vidimo kako to izgleda kada je u pitanju iOS mobilna platforma. Automatski nam je prikazan iPhone mobilni uređaj te smo odmah u mogućnosti korištenjem palete alata dodavati komponente na prozor. Popis svih komponenti vidljiv je u prozoru s nazivom *Structure*, dok klikom na pojedinu komponentu možemo vidjeti sva njena svojstva i stanja u prozoru s nazivom *Object Inspector*. Ovaj prozor nam ujedno omogućuje da ta svojstva i stanja mijenjamo, tako izbjegavajući pisanje nepotrebnog programskog koda.



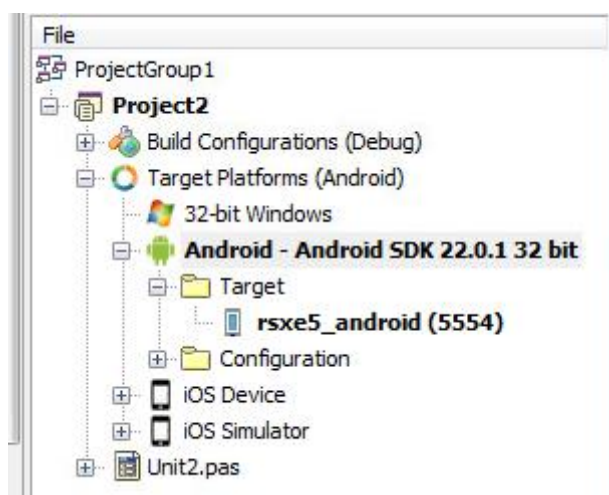
Slika 4. Primjer metode pisane u Delphiju

Na slici možemo vidjeti kako u Delphiju napisati metodu *Button1Click* koja bi izmijenila svojstvo *Text* komponente *Label1*. Delphi koristi objektni pristup vrlo sličan onome u C++ programskom jeziku pa je zato vrlo jednostavno napisati ovaj programski kod u C++ Builderu.



Slika 5. Primjer metode pisane u C++ Builderu

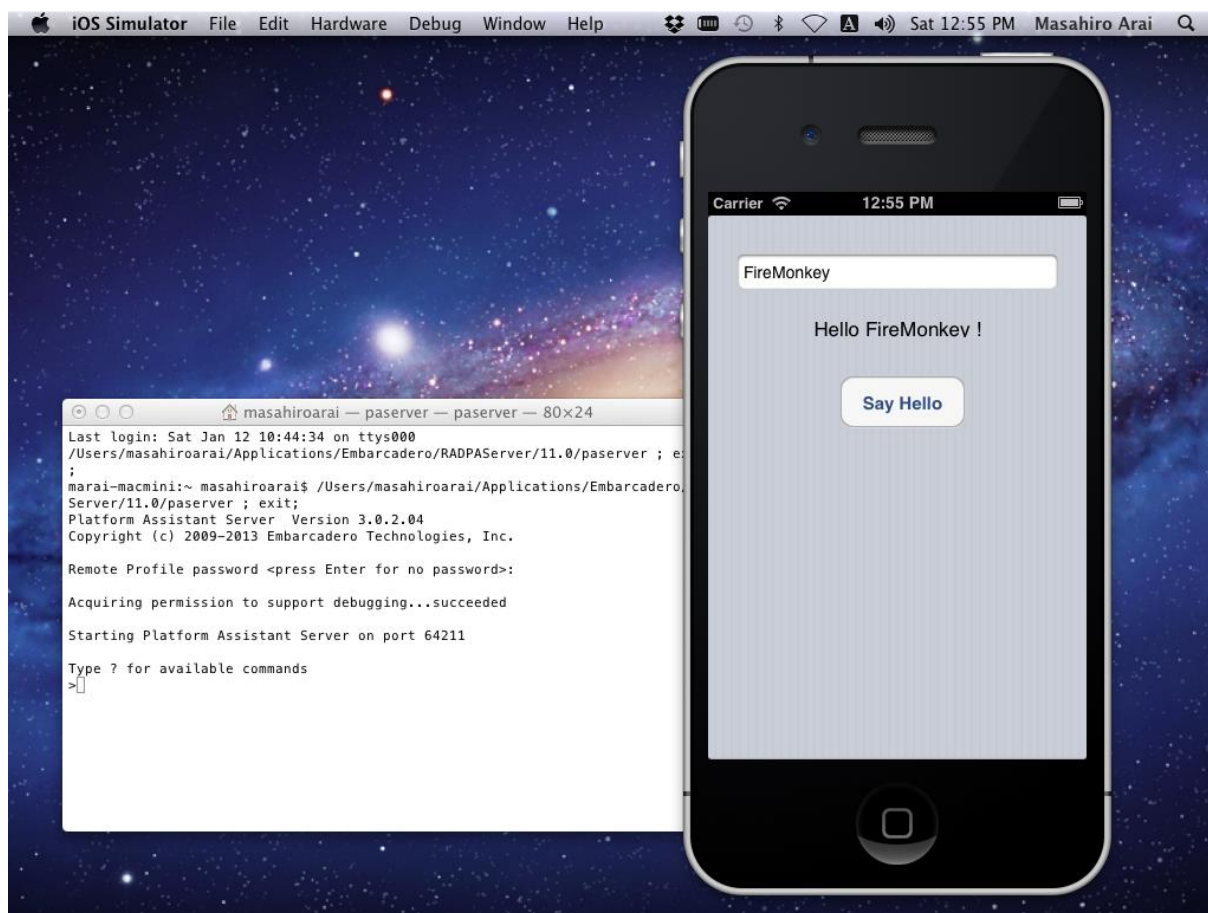
Funkcionalne razlike između programskog koda na prethodne dvije slike nema te je u većini slučajeva Delphi programerima jednostavno razumjeti programski kod pisan u C++ Builderu i obratno. Međutim, razlike ipak postoje jer Delphi ne koristi pokazivače za rad s komponentama, dok C++ Builder ih koristi. C++ Builder u novijim izdanjima ima i podršku specifične C++ biblioteke (Boost, STL itd.) kao i podršku za C++11 standard. U svim tim slučajevima prebacivanje programskog koda u Delphi (Pascal) sintaksu je svakako kompliciranije.



Slika 6. Popis željenih platformi

Ukoliko želimo našu mobilnu aplikaciju prevesti i za Android mobilnu platformu dovoljno ju je tek dodati u popis prikazan u prozoru *Project Manager*. Primijetimo da naš projekt na slici ima mogućnost biti preveden kao Windows 32 bit aplikacija (64 bit također je podržano), te kao Android i iOS aplikacija.

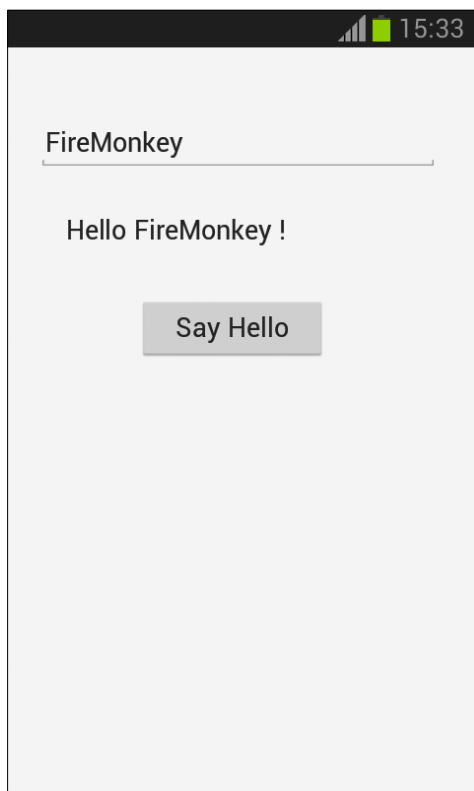
Kada je riječ o samom izvršavanju mobilnih aplikacija nudi nam se nekoliko mogućnosti. Primjerice, možemo koristiti stvarne mobilne uređaje koji su putem USB porta spojeni na računalo ili možemo koristiti iOS simulator, tj. Android emulator. U slučaju korištenja iOS simulatora koji se pokreće u OS X operacijskom sustavu prethodno je potrebno u RAD Studiju kreirati profil za spajanje na željeno Mac računalo. Nakon kreiranja profila, Delphi i C++ Builder su u mogućnosti pomoću aplikacije *Platform Assistant Server* komunicirati s tim računalom te na njemu pokrenuti izvršavanje naše aplikacije, bilo da je riječ o OS X ili iOS aplikaciji pomoću simulatora.



Slika 7. Izvršavanje mobilne aplikacije pomoću iOS simulatora u OS X-u

Nakon pokretanja mobilne aplikacije u iOS simulatoru koji se nalazi na nekom Mac računalu dobili bi rezultat kao na slici. Naš *Platform Assistant Server* prebacio je mobilnu aplikaciju na Mac računalo, tamo pokrenuo iOS simulator te nam omogućio da izravno iz Delphija ili C++ Buildera provjeravamo moguće greške i stanja aplikacije dok se ona izvršava u iOS simulatoru.

Ukoliko bi ovu istu mobilnu aplikaciju pokrenuli na Android uređaju ona bi izgledala ovako.



Slika 8. Izvršavanje mobilne aplikacije na Android mobilnom uređaju

Obje mobilne aplikacije funkcionalno su iste i sadrže istu programski kod. Razlika je tek što rade na različitim mobilnim platformama. U tome i jest tolika prednost RAD Studio alata koji ovakvo što omogućuju.

4. Dostupnost RAD Studio alata

RAD Studio alati nisu besplatni, te dolaze u nekoliko različitih paketa. Moguće je kupiti svaki alat zasebno, ili sve zajedno u RAD Studio paketu. Cijene ovise o izdanju alata, pa tako postoje *Starter*, *Professional*, *Enterprise*, *Ultimate* i *Architect* izdanje. Svako od izdanja je namijenjeno različitim profilima programera. Tako je *Starter* izdanje namijenjeno hobistima i osobama koji tek povremeno razvijaju aplikacije. Njihove potrebe su minimalne, pa tako i sam alat dolazi u dosta ograničenom obliku.

Profesionalno izdanje je namijenjeno pojedincima tj. profesionalcima koji samostalno razvijaju aplikacije. Za razliku od *Starter* izdanja, profesionalno izdanje je mnogo naprednije te uključuje mnoštvo dodatnih alata koji ne dolaze sa *Starter* izdanjem.

Ostala izdanja su namijenjena za manje i veće razvojne timove, te u ovisnosti o tome ta izdanja donose mnoge alate namijenjene razvojnim timovima. Stoga su i cijene ovih izdanja najveće. Distributer za Hrvatsku je tvrtka Konto, čiji cjenik se može pogledati na linku http://embarcadero.konto.hr/?page_id=228 – cjenik za RAD Studio skup alata.

Osim ovih izdanja moguće je za škole i fakultete kupiti akademsko izdanje RAD Studio alata. Akademsko izdanje nudi sve isto kao i profesionalno izdanje, ali se ne može koristiti u komercijalne svrhe. Od svih spomenutih izdanja cijena akademskog izdanja je najmanja i trenutno iznosi svega 30-40 eura po licenci. Za detalje je potrebno kontaktirati distributera za našu regiju tj. tvrtku Konto.

5. Zaključak

Mobilne aplikacije već su duže vrijeme postale gotovo neizbježne, kako za korisnike tako i za tvrtke koje se bave razvojem aplikacija. Štoviše, danas većina tvrtki koje razvijaju desktop aplikacije teže k tome da uz njih mogu ponuditi i mobilne inačice tih aplikacija. Zato se javlja potreba za alatima koji bi omogućili takvo što, odnosno razvoj aplikacija podržanih na više platformi na što lakši i brži način. Upravo zbog toga razvojna okruženja poput Delphi i C++ Buildera su jedna od najboljih jer uz minimalno truda nude upravo to.

Kroz gotovo dva desetljeća primjene i razvoja RAD Studio skupio je na milijune aktivnih korisnika (programera) diljem svijeta. S tolikom već postojećom bazom korisnika ne treba sumnjati da će razvoj mobilnih aplikacija u RAD Studio alatima uskoro postati jedno od najpopularnijih rješenja. Iako je ova tehnologija relativno mlada možda najbolje od svega činjenica je da nije u potpunosti iskorištena te da mnogi uskoro očekuju i dodatna poboljšanja u pogledu podrške za Linux operacijske sustave kao i za mnoge druge uređaje.

Bibliografija

1. **Gajic, Zarko.** Delphi History – from Pascal to Embarcadero Delphi XE 2. [Mrežno] [Citirano: 20. 5 2014.] <http://delphi.about.com/cs/azindex/a/dhistory.htm>.
2. **Intersimone, David.** Historical Document: Visual Component Library First Draft - May 24, 1993 . [Mrežno] 2013. [Citirano: 21. 5 2014.] <http://edn.embarcadero.com/article/32975>.
3. **Long, Brian.** Part 1: Installing And Using Kylix Open Edition. [Mrežno] 2001. [Citirano: 22. 5 2014.] <http://www.blong.com/Articles/Kylix%20Tutorial/Part1/Tutorial1.htm>.
4. **Embarcadero Technologies.** Embarcadero Technologies® to Acquire CodeGear™ from Borland Software. [Mrežno] [Citirano: 22. 5 2014.] <http://www.codegear.com/about/news/embt>.
5. —. FireMonkey Application Platform. [Mrežno] 2014. [Citirano: 23. 5 2014.] http://docwiki.embarcadero.com/RADStudio/XE6/en/FireMonkey_Application_Platform.