

Laboratorij za komunikacijske naprave

Študijsko gradivo in navodila
za laboratorijske vaje pri predmetu
Terminali in aplikacije

Bolonjski študijski program (1. stopnja VS)
MMK – Vrtojba

š.l. 2013/2014

doc. dr. Grega Jakus
(grega.jakus@fe.uni-lj.si)

1

Vhodne in izhodne naprave pri interakciji človeka in računalnika

1. Uvod

Interakcija je proces izmenjave informacij. Ljudje med seboj vsakodnevno komuniciramo preko različnih vhodnih in izhodnih kanalov, na podoben način pa poteka tudi naša interakcija z računalnikom oziroma z drugimi podobnimi napravami. Prenos informacij med človekom in računalnikom ter v obratni smeri poteka preko *vhodnih* in *izhodnih naprav*. Danes so najpogostejše takšne naprave tipkovnica, miška, LCD zaslon in tiskalnik. V vaji bomo spoznali še nekatere druge vhodne in izhodne naprave, ki so po večini namenjene bolj specifični uporabi.

2. Vhodne naprave

Vhodne naprave na grobo razdelimo na tri vrste:

- **naprave za vnos znakov in besedila**

Predstavljajo najpogostejši način interakcije s terminalom (npr. pisanje dokumenta v urejevalniku besedil, pošiljanje e-mail ali SMS sporočil, vnašanje števil v kalkulator,...).

Med naprave za vnos znakov in besedila uvrščamo razne tipkovnice (alfanumerično, številčno, tipkovnico mobilnega terminala, posebne tipkovnice itd.), naprave za ročno risanje (grafične tablice) ter prepoznavo pisave in govora.

- **kazalne naprave**

Sem uvrščamo naprave, s pomočjo katerih direktno ali indirektno »pokažemo« na nek objekt (na zaslonu) in z njim manipuliramo (npr. premaknemo okno). Najbolj razširjena naprava iz te skupine je miška.

- **naprave za navidezno resničnost in 3D interakcijo**

To so posebne naprave za interakcijo v 3D prostoru, ki je pogosto uporabljen v aplikacijah z navidezno resničnostjo.

3. Izhodne naprave

Izhodne naprave delimo na:

- **prikazovalnike**

Sem sodijo predvsem namizni zasloni in zasloni, namenjeni širšemu krogu uporabe, v zadnjem času pa tudi digitalni papir.

- **tiskalnike**

- **zvočne naprave**

4. Priprava na vajo in potek vaje

Za gladek potek vaje se ustrezno pripravite:

- **temeljito predelajte snov s predavanj**
- **temeljito predelajte to gradivo**
- **doma rešite 1. in 2. nalogo v nadaljevanju**
- **gradivo natisnite in ga prinesite s seboj na vaje**

V prvem delu vaje bo vsaka skupina preizkusila pripravljene vhodne in izhodne naprave. Pri izdelavi poročila iz prvega dela vaje:

- uvrstite napravo v ustrezno skupino in podskupino
- določite lastnosti naprave (npr. število gumbov, občutljivost na dotik, premik, ...)
- določite funkcionalnosti naprave (kaj s to napravo lahko počnete, za kakšne aplikacije bi bila po vašem mnenju primerna,)
- poslikajte napravo in način, na katerega jo uporabljate
- opišite vaše vtise o napravi

V drugem delu bo vsaka skupina dobila nalogo, da izbere vhodne in izhodne naprave, ki bi bile primerne za izvedbo interakcije z določeno aplikacijo. Izbira mora biti ustrezno utemeljena (glej 3. nalogo v nadaljevanju). Vsaka skupina bo rezultate na kratko predstavila ostalim ter **jih vključila v končno poročilo.**

Poročilo vaje naj bo kratko in jedrnato, z vso zahtevano vsebino in naj ne bo daljše od 700 besed. Ocenjena bo kvaliteta poročila in ne njegova kvantiteta.

5. Naloge

1. naloga (doma): Uvrstite naslednje vhodne naprave v ustrezno skupino in podskupino (pozor: nekatere naprave lahko uvrstimo celo v več kategorij).

- alfanumerična tipkovnica
- numerična tipkovnica
- miška
- sledilna ploščica (ang. *touchpad*)
- sledilna kroglica (ang. *track ball*)
- igralna palica
- zaslon na dotik
- optični čitalnik (ang. *scanner*)
- prepoznavna pisave
- telefonska tipkovnica
- prepoznavna govora
- smerne tipke (kurzorji) na tipkovnici
- sledilna paličica (ang. *pointing stick*)
- zaslon na dotik s palčko (ang. *stylus*)
- daljinski upravljalnik
- 3D miška
- računalniški volan
- podatkovna rokavica (ang. *dataglove*)

2. naloga (doma): Uvrstite naslednje izhodne naprave v ustrezno skupino.

- LCD zaslon
- laserski tiskalnik
- LED zaslon
- zaslon na katodno cev (CRT zaslon)
- digitalni papir
- 3D očala
- projektor
- bralnik besedil

3. naloga (na vajah)

Predlagajte vhodne in izhodne naprave, ki so najprimernejše za načrtovanje vmesnikov za komunikacijo med človekom in terminalom (elektronsko napravo). Predloge utemeljite.

1. Osebni prenosni komunikacijski sistem (primer)

Osebni prenosni komunikacijski sistem bi uporabniku omogočal komunikacijo kjerkoli in kadarkoli. Deloval bi s pomočjo naprave, ki bi bila nekakšna kombinacija mobilnega telefona in prenosnega računalnika. Naprava bi morala biti lahka in takšnih dimenzij, da bi jo lahko uporabnik nosil v žepu suknjiča. Naprava bi morala podpirati različne vrste komunikacije (govor, tekstovna sporočila, video sporočila itd.), hkrati pa nuditi podporo za shranjevanje kontaktnih informacij v obliki telefonskih imenikov itd. Predlagamo naslednje načine interakcije:

- majhen LCD zaslon z majhno porabo energije,
- majhna numerična tipkovnica, namenjena pogosti, osnovni uporabi,
- interakcija s pomočjo palčke ali zaslona na dotik, ki bi omogočala napredno uporabo aplikacij na napravi,
- možnost priklopa dodatne (alfanumerične tipkovnice), ki bi omogočala udobno uporabo sistema v okoljih, ki bi to dopuščala (npr. v hotelski sobi).

2. Informacijski terminal v turističnem središču
3. Terminal za nakup študentskih bonov
4. Multimedijski predvajalnik v osebni avtomobilu
5. Vmesnik (nadzorna plošča) za krmiljenje industrijskega robota
6. Simulator letenja
7. Naprava za škropljenje poljščin
8. Prenosni urejevalnik besedil
9. Elektronska alarmna naprava
10. Letalski kokpit
11. Nadzorni sistem kontrole letenja
12. Naprava za izdelavo vozovnic na vlaku (pri sprevodniku)
13. Nadzorna plošča stroja za struganje kovin
14. Vmesnik za upravljanje z robotsko roko
15. Spletna trgovina
16. Satelitska navigacija
17. Osebni, prenosni vodnik po muzeju
18. Spomnite se še svoj primer ...

Viri in literatura

1. A. Dix, J. Finlay, G. D. Abowd, R. Beale, *Human-Computer Interaction*, 3. izdaja, Prentice Hall, 2004
2. S. Smith-Atakan, *Human-Computer Interaction*, Thomson, 2006
3. Wikipedia

2

Izdelava spletnih strani ob upoštevanju uporabniške izkušnje

1. Pripravite se na vajo doma

1. Poiščite spletno stran, ki ste jo izdelali pri predmetu Spletne tehnologije. Potrebovali bomo le del, ki deluje na odjemalcu: HTML, CSS in JS. Če imate dinamično spletno stran (PHP), potem shranite tvorjen HTML in ga prinesite na vaje.
 2. Če nimate lastne spletne strani, potem poiščite preprosto spletno stran na spletu.
 3. Ponovite HTML, CSS in JS, saj vam v nasprotnem primeru ne bo uspelo opraviti vaje.
 4. Seznanite se z enim izmed brezplačnih urejevalnikov spletnih strani, npr. PageBreeze, ki ga boste lahko uporabljali na vajah.
 5. Preberite gradivo in si zamislite, na kak način boste upoštevali priporočila na vaši spletni strani.
- Gradivo povzema izbrana priporočila za izdelavo spletnih strani z upoštevanjem uporabniške izkušnje. Obsežnejša priporočila najdete v literaturi.
 - Vseh priporočil, opisanih v gradivu, ne boste mogli upoštevati na vašem spletnem mestu. Poznavanje teh naj ostane za izpit ter izdelavo spletnih strani v prihodnosti.
 - Cilj vaje ni izdelava v celoti delujočega spletnega mesta, ampak njegova zasnova, ki bo prinašala največjo izkušnjo obiskovalcev.
 - **Oddali boste izdelane spletne strani in poročilo. V poročilu zelo na kratko predstavite, katera priporočila ste upoštevali in kje se to na spletnem mestu opazi. S poročilom boste asistenta usmerjali pri ocenjevanju vaše vaje.**

Kako brati navodila?

- Znak opisuje poudarke.
- ✓ Znak opisuje priporočilo, ki ga morate upoštevati.

➤ *Na ta način so predstavljene dodatne naloge, ki jih morate pri nekem priporočilu opraviti.*

- 🔗 Primeri ali dodatno gradivo na spletu.

Pa začnimo...

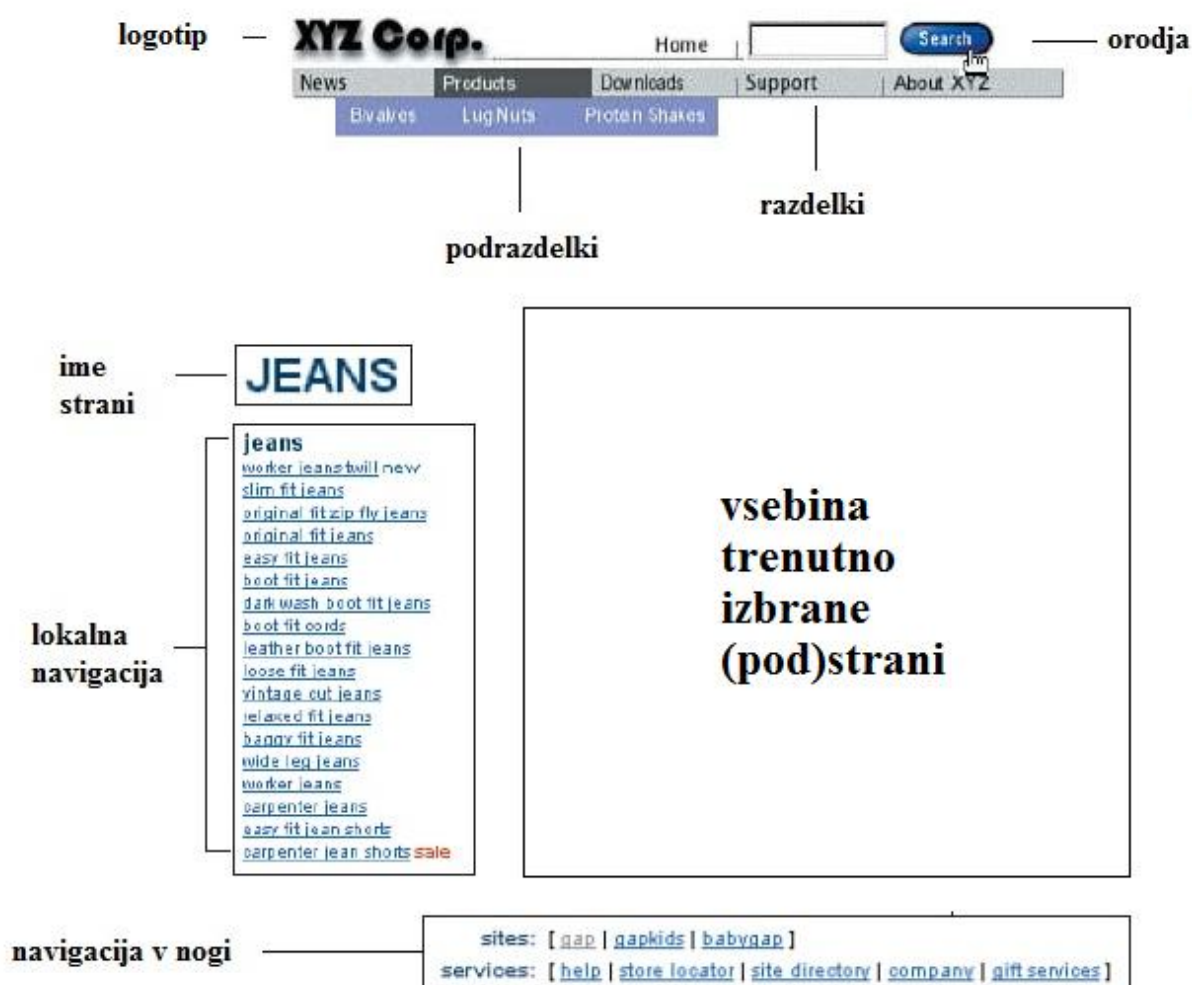
2. Spletno mesto smiselno strukturirajte

Spletno mesto razdelite na domačo stran in podstrani

- ✓ Domača stran naj ponuja pregled nad celotnim spletnim mestom, podstrani pa naj se osredotočijo na določeno temo. S smiselnim strukturiranjem spletnega mesta dosežete, da obiskovalci ne izgubljajo časa z vsebino, ki jih ne zanima.

Posamezno stran razdelite v jasno določena področja

- ✓ Kot vodilo naj vam služi spodnja slika.



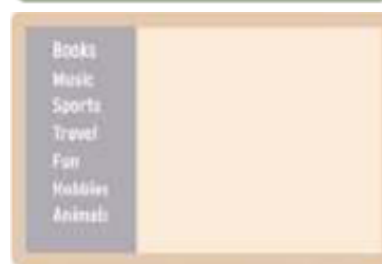
Slika 1: Značilna področja spletne strani

- ✓ Vsebino posamezne strani razbijte v več manjših enot. Uporabite naslove, podnaslove, sezname ter različne načine združevanja vsebine (glej točko c).

Izdelajte jasno vizualno hierarhijo elementov na strani

Spletna stran mora biti prilagojena hitremu pregledovanju. Poskrbite, da izgled posameznih elementov na strani jasno in natančno odraža odnose med temi elementi. Jasno mora biti razvidno, kateri elementi so povezani in kateri so del drugih elementov. Z drugimi besedami, vsaka stran mora imeti jasno razvidno vizualno hierarhijo.

- ✓ **Bolj kot je neka vsebina pomembna, bolj naj bo vidna.** Na primer, najpomembnejši naslovi naj bodo večji, okrepljeni, v izstopajoči barvi, okoli njih naj bo več praznega prostora ali/in naj bodo bližje zgornjega roba strani.
- ✓ **Elementi, ki so povezani logično, naj bodo povezani tudi vizualno.** Da so določene stvari podobne, lahko na primer nakažete tako, da jih združite v poglavje, jasno definirano področje ali pa jih prikažete v podobnem vizualnem stilu.
- ✓ **Z vizualnim gnezdenjem elementov pokažite, kateri elementi so del drugih elementov.** Na primer, naslov poglavja »računalniške knjige« se mora pojaviti pred naslovi posameznih knjig. Obenem mora naslov vizualno zaobjeti vso vsebino, ki jo pokriva, saj je vsaka navedena knjiga del omenjenega poglavja. Podobno mora tudi naslov posamezne knjige zaobjeti vsebino, namenjeno opisu oziroma predstavitvi knjige.



- *Izdelajte po en primer vsakega izmed teh treh načinov (ni potrebno preoblikovati celotnega spletnega mesta).*
- *V poročilu navedite, kje na spletnem mestu so izdelani primeri vidni.*

Upoštevajte področja največje pozornosti

Na podlagi sledenja očem je bilo ugotovljeno, da uporabniki največjo pozornost najpogosteje namenijo trem področjem spletne strani, ki v grobem tvorijo obliko črke F (spodnja slika).

- Obiskovalec najprej preleti zgornji del področja z vsebino v horizontalni smeri.
- Obiskovalec se nato premakne nižje in ponovno preleti vsebino v horizontalni smeri.
- Nato obiskovalec preleti levi del področja z vsebino navpično navzdol.



Slika 2: Področja največje pozornosti na treh spletnih straneh.

Področja največje pozornosti lahko na vašem spletnem mestu upoštevate na sledeče načine:

- ✓ Prva dva odstavka morata vsebovati najpomembnejše informacije na posamezni strani.
- ✓ Vsa poglavja, odstavke, alineje in povezave začnite z besedami, ki nosijo najpomembnejše informacije. Tako jih bo uporabnik opazil, ko bo preletel stran v navpični smeri. Najpomembnejši sta prvi dve besedi.

✚ Več na: <http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>

- *Eno izmed strani prilagodite, tako da boste upoštevali področje največje pozornosti (ni potrebno preoblikovati celotnega spletnega mesta).*
- *V poročilu opišite, kje in na kakšen način ste območje največje pozornosti upoštevali.*

Vsako stran poimenujte

- ✓ Z imenom strani poimenujete vsebino strani, zato naj bo ime strani tudi vizualno nadrejeno vsebini strani (in ne pa npr. navigaciji, ki služi kot infrastruktura).



Slika 3: Ime strani je vizualno nadrejeno vsebini strani

- ✓ Le poudarjeno ime trenutne strani v navigaciji ni dovolj, ime strani naj bo vidno. Uporabite ustrezen položaj, velikost, barve in pisavo. V večini primerov bo ime strani največji tekst na strani.
- ✓ Ime strani naj ustreza tekstu na povezavi, ki obiskovalca pripelje na to stran.

V naslov spletne strani (<title>) vključite kratek opis strani

- ✓ Naslovi strani (pa tudi poglavij) naj jasno nakazujejo vsebino na strani. Ta bo tako jasna, tudi ko bodo naslovi prebrani izven spletnega mesta, na primer na seznamu rezultatov iskalnikov (glej spodnjo sliko).
- ✚ Glejte še <http://www.imdb.com/>, <http://www.rtvsl.si/>, <https://www.facebook.com/>



Slika 4: Naslov spletne strani v znački *title* se prikazuje na seznamu rezultatov iskanja.

Na vsako stran postavite logotip

- Logotip predstavlja spletno mesto.
- ✓ Logotip postavite v zgornji levi kot vsake podstrani (glej Sliko 5).
- ✓ Logotip naj hkrati služi kot povezava, ki vodi na domačo stran.
- ✓ Prilagojen in pomanjšan logotip uporabite kot ikono spletne strani, ki je prikazana zgoraj levo v zavihku brskalnika (glej Sliko 4) ter pri shranjevanju zaznamkov.

3. Uporaba spletnega mesta naj bo obiskovalcem prijazna

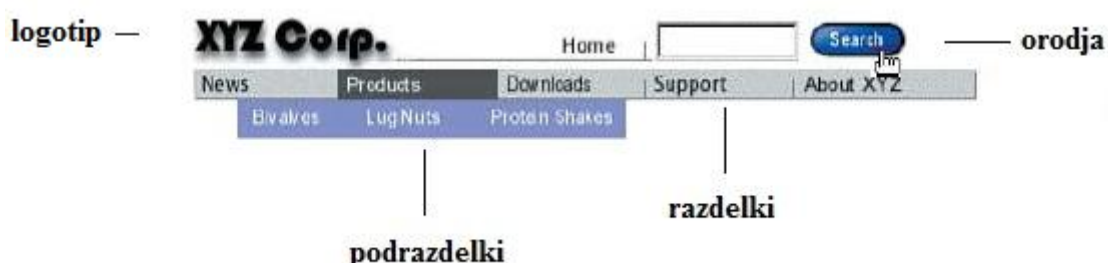
Obiskovalci spletno mesto v splošnem uporabljajo na dva načina:

- s pomočjo brskanja prek navigacije ali
- s pomočjo neposrednega iskanja na podlagi ključnih besed.

V obeh primerih mora biti uporaba spletnega mesta prilagojena uporabniku, saj bo ta, v primeru, da iskane stvari v zelo kratkem času ne bo našel, spletno mesto zapustil in se ne bo več vrnil.

Uporabite uveljavljene načine navigacije

- Ker so se ti razvili skozi čas, jih je uporabnik že navajen.



Slika 5: Horizontalna navigacija skupaj z logotipom in iskalnikom na vrhu strani.

Navigacija naj bo enaka po vseh straneh, razen:

- ✓ **na domači strani** (glejte razdelek za domačo stran) in
- ✓ **na straneh z obsežnimi obrazci** (»naročilo«, »registracija«, ...). Na teh straneh navigacija le odvrta pozornost, saj uporabnik strani ne bo zapustil, dokler ne bo izpolnil obrazcev. Na teh straneh zato poskrbite samo za logotip (s povezavo na domačo stran) ali morebitna orodja za izpolnjevanje obrazca (nazaj, naprej, ...).

- Za potrebe vaje bo dovolj le ena vrsta glavne navigacije (levo ali zgoraj). Vključite logotip, kategorije in podkategorije, ime strani in navigacijo v nogi.
- Lahko si pomagajte z že pripravljeno kodo za »pull down« meni.

Nakažite obiskovalcu, kje na spletnem mestu se trenutno nahaja

- ✓ V navigaciji jasno označite obiskovalčev trenutni položaj na spletnem mestu. To lahko storite z uporabo vizualnih učinkov, npr. z **izstopajočo barvo** in **krepko pisavo**.
- ✓ Na kompleksnejših straneh lahko uporabite princip »krušnih drobtinic« (ang. breadcrumbs). Glejte na primer www.cnet.com.

Ponudite iskalnik

Iskalnik je »rešilna bilka« uporabnikov pri uporabi kompleksnih spletnih strani. Iskalnik

- uporabnikom daje občutek, da imajo brskanje pod svojim nadzorom;
- je izhod v sili, ko se uporabnik izgubi.
- ✓ Iskanje mora zaradi tega biti dostopno na vsaki (in ne le na domači) strani. Ne moremo namreč vnaprej predvideti, na kateri strani se bodo uporabniki izgubili.
- ✚ Preverite: <http://www.amazon.com/>, <http://www.imdb.com/>

Uporabniki imajo natančno določena pričakovanja, kako naj bi iskalnik izgledal in deloval. Vsak odmik od teh predstav lahko vodi v zmedo. Iskalnik naj vsebuje sledeče tri komponente:

- polje za vnos iskalnega niza;
- gumb, ki sproži iskanje; ter
- linearen seznam rezultatov, ki se prikaže na novi strani.

Vnosno polje. Ko obiskovalec spletnega mesta potrebuje iskalnik, išče vnosno polje, v katerega bo lahko takoj začel tipkati, podobno, kot je to navajen na spletnih iskalnikih.

- ✓ na domači strani mora iskalnik ponujati vnosno polje (in ne le povezave na iskalnik).
- ✓ na ostalih straneh lahko uporabite povezavo na iskalnik, če gre za zelo preproste strani, V nasprotnem primeru tudi tu uporabite vnosno polje;
- ✓ vnosno polje mora biti dovolj široko, da sprejme povprečen iskalni niz;
- ✓ za označevanje iskalnika namesto napisa zraven vnosnega polja uporabite napis na gumbu. Dodatni elementi lahko namreč zmedejo obiskovalca (velja »manj je več«).

Gumb. Tudi pri izdelavi gumba je pomembno, da sledite pričakovanjem uporabnikov:

- ✓ gumb naj se nahaja desno od vnosnega polja v isti vrstici;
- ✓ gumb poimenujte »search« ali »išči«.

Rezultati iskanja. Kvaliteten seznam rezultatov iskanja je zelo pomemben, saj obiskovalci pogosto ocenjujejo spletno stran po kakovosti iskanja. Pogosto bodo obiskovalci stran zapustili, če ne bodo že pri prvem iskanju dobili zelenih rezultatov. Poleg tega obiskovalci navadno pogledajo le prvo stran rezultatov. Zato je pomembno, da prva stran vsebuje najbolj relevantne rezultate, ki so razvrščeni po jasnem kriteriju.

Enostavno ali napredno iskanje? Povprečna dolžina iskalnega niza sta dve besedi. Uporabniki imajo raje enostavno iskanje v primerjavi z naprednim, saj naprednega iskanja pogosto ne razumejo.

- ✓ Ne ponujajte naprednega iskanja na domači strani. Če je to nujno potrebno, omogočite napredno iskanje kot dodatno možnost na strani z rezultati iskanja v smislu "*Niste našli, kar ste iskali? Poskusite napredno iskanje.*".

Obseg iskanja določa dele spletnega mesta, ki bodo vključeni v iskanje (celotno spletno mesto, posamezne podstrani, kategorije izdelkov itd). Na preprostih spletnih straneh

ponujanje obsega iskanja ni smiselno, saj lahko to vnese zmedo (obiskovalec lahko spregleda izbran obseg). Na kompleksnejših straneh je smiselno omejiti obseg iskanja, pri čemer:



- ✓ nastavite privzet obseg na celotno spletno mesto;
- ✓ ko uporabnik spremeni obseg, ga izpišite na vrhu seznama rezultatov;
- ✓ na strani z rezultati omogočite takojšnjo spremembo obsega;
- ✓ če iskanje vrne preveč rezultatov, predlagajte uporabniku, naj obseg iskanja omeji.



Slika 6: Primer enostavnega iskanja (<http://www.nngroup.com/>) in iskanja z določitvijo obsega (<http://www.imdb.com/>)

- *Na spletno stran vključite navidezni iskalnik, pri tem pa sledite priporočilom.*
- *Iskanje bomo le simulirali. Rezultati iskanja naj bodo zato vnaprej pripravljeni, izpišejo pa naj se na novi (pod)strani.*

Povezave naj bodo uporabne

- ✓ **Povezave izpostavite** z uporabo **krepega** in podčrtanega besedila. Tako bo obiskovalec hitro našel orientacijo, kam lahko gre s trenutne strani.
 - ✓ **Ne podčrtujte vseh povezav.** To velja predvsem za povezave v navigaciji ali v seznamih povezav, oziroma kjerkoli je jasno razvidno, da gre za povezave. Če so povezave predstavljene v barvah, ki povzročajo težave ljudem z barvno slepoto (na primer rdeča in zelena), potem takšne povezave podčrtajte v vsakem primeru.
 - ✓ **Spremenite barve obiskanih povezav.** Povezave so ključne za orientacijo na spletnem mestu. Na podlagi barvanja že obiskanih povezav lahko obiskovalci prepoznajo povezave, ki so vodile do neustrezne vsebine in jih tako ne obišejo več ali pa znova obišejo vsebine, ki so bile uporabne. Upoštevajte sledeče:
 - barva neobiskanih povezav naj bo bolj živa, svetla ali nasičena.
 - barva obiskanih povezav naj bo pusta, izprana oziroma naj deluje izrabljeno.
 - ✓ **Ne uporabljajte stila povezav v navadnem tekstu.** Ne podčrtujte teksta (pa čeprav povezav sicer sploh ne podčrtujete) in ne uporabljajte barv povezav v navadnem tekstu.
 - ✓ **V besedilu povezave jasno predstavite, kam povezava vodi.** Tako lahko obiskovalec predvideva, kam povezava vodi, ne da bi jo uporabil.
 - ✓ **Jasno nakažite, da je na gumb mogoče klikniti.** Sprememba oblike miškega kazalca nad gumbom, ni dovolj, saj mora uporabnik prej uganiti, da gre za gumb. Gumb naj torej izgleda kot gumb (npr. senčeno), torej  in ne .
 - ✓ **Ne prikazujte vsebine v novih zavihkih ali oknih brskalnika.**
- ✚ Več na : <http://www.nngroup.com/articles/guidelines-for-visualizing-links/>

4. Vsebina strani naj bo namenjena hitremu pregledovanju

Znebite se »šuma«

- ✓ **Ne uporabljajte elementov, ki preveč pritegujejo pozornost** (animacije, utripanje, klicaji, velike črke, žive barve, intenzivna ozadja, preveč izrazite obrobe itd.)
- ✓ **Znebite se odvečnih besed** ter na ta način bolj izpostavite vsebino, ki je pomembna.
- ✓ **Zmanjšajte dolžino strani**, tako da je večina vsebine vidna brez drsenja (scrolling).
- ✓ **Upravičite pričakovanja uporabnikov**. Spletna stran naj dejansko vsebuje tisto vsebino, ki jo obiskovalcu obljublja v naslovu strani in na domači strani.

Poskrbite za berljivost vsebine

- ✓ **Uporabite berljivo pisavo** (font).
- ✓ **Izogibajte se fiksni velikosti pisave**, določeni v CSS, saj s tem onemogočite spremembo velikosti pisave v brskalniku.
- ✓ **Ne uporabljajte PDF dokumentov za prikaz vsebine**, saj to prekine uporabnikov tok brskanja. Format PDF dokumenta je drugačen od dimenzij spletne strani. Ker je prikazovanje PDF dokumentov v brskalniku mogoče le preko vtičnika, standardne funkcije brskalnikov pri prikazovanju PDF dokumentov ne delujejo.

Obrazci

- ✓ **Iz obrazcev izločite nepotrebna polja** (npr. za spol, če podatka zares ne potrebujete).
- ✓ **Nastavite fokus tipkovnice na prvo polje v obrazcu**.
- ✓ **Za obvezna nastavite le tista polja, za katera je to resnično potrebno**.
- ✓ **Ko uporabnik pošlje nepopolne podatke, ga o tem obvestite** tako, da označite tista polja, v katerih so bili podatki nepravilno vpisani. V ostalih poljih naj bodo že vpisane vrednosti, ki jih je izpolnil uporabnik v prvem poskusu. Na ta način uporabniku obrazca ne bo potrebno izpolnjevati v celoti.

Konsistentnost

- ✓ Ena najpomembnejših načel pri izdelavi spletnih strani je konsistentnost z obstoječimi načini uporabe spleta. Na podlagi predhodnih izkušenj, uporabnik namreč pričakuje, da se bodo stvari obnašale podobno na vseh spletnih straneh. Če opazite, da se neka stvar dobro obnese na večini drugih spletnih strani, potem jo posnemajte. Zapomnite si Jakobov zakon o uporabniški izkušnji na spletu: *spletni uporabniki preživijo večino časa na drugih spletnih straneh in tam izoblikujejo pričakovanja, kako splet deluje*.
- ✓ Pred objavo spletno stran testirajte s pravimi uporabniki, saj boste le na ta način ugotovili, kako stran deluje v resnici.
- ✓ Poskrbite, da bo stran delovala v vseh (ali vsaj najpogosteje uporabljenih) brskalnikih.

5. Domača stran naj izkazuje namen spletnega mesta

Domača stran je pogosto prvi stik obiskovalca s spletnim mestom. Pri večini obiskovalcev bo stik z domačo stranjo morda vaša edina priložnost ustvarjanja dobrega vtisa. Domača stran mora zato jasno nakazovati, kako obiskovalec pride do vsebine, ki jo išče, obenem pa tudi predstaviti vsebino, ki je obiskovalec sicer ne išče, bi ga pa lahko kljub temu zanimala.

- ✓ Domača stran mora povedati **bistvo o vašem spletnem mestu**: čemu je namenjeno in, če je mogoče, nakazati, kaj ima uporabnik od tega, da je tukaj in ne kje drugje oziroma po čem se razlikuje oziroma je boljša od konkurenčnih strani.
- ✓ Domača stran mora nakazati, **kaj se nahaja na spletnem mestu**:
 - vsebino – kaj lahko obiskovalec na spletnem mestu najde
 - možnosti – kaj lahko tukaj počne
 - kako je spletno mesto organizirano (to lahko nakažete z navigacijo)
- ✓ **Vključite iskanje** (glej razdelek o iskanju)
- ✓ **Vključite namige** oziroma primere “bombastične” vsebine (t.i. “teasers”), ki jo lahko uporabnik najde na spletnem mestu (pomislite na naslovnice revij). Vključite promocije najnovejše, najboljše ali najbolj popularne vsebine, ki jo spletno mesto ponuja (t.i. “featured” vsebino).
- ✓ **Poskrbite za ažurno vsebino** – vsebino, ki je pogosto posodobljena. Na ta način bodo obiskovalci ob ponovnem obisku dobili občutek, da je spletno mesto živo.
- ✓ **Vključite bližnjice** na najbolj obiskane oziroma popularne vsebine.
- ✓ **Vključite prijavo in registracijo**. Ta naj vključuje tudi indikacijo uporabniku, da je prijavljen (“Welcome back, Chuck Norris!”).

Namen spletnega mesta lahko predstavite na več načinov, še posebej pa sta pri tem pomembna dva elementa:

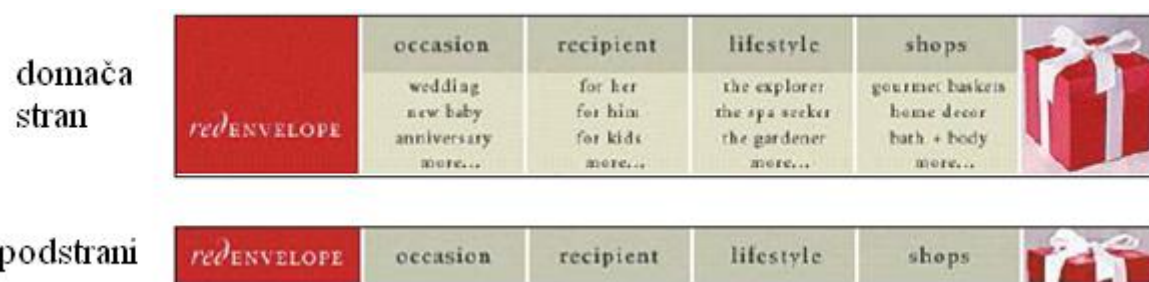
- ✓ **Slogan** (ang. “tag line”) – kratek stavek ali besedna zveza, ki jasno povzema, kaj uporabnik najde na spletnem mestu. Slogan naj bo postavljen ob logotip spletnega mesta, saj je tam najbolj izpostavljen obiskovalčevi pozornosti, obiskovalec pa tako tudi ve, da je slogan povezan s spletnim mestom. Dobri slogani so **kratki, jasni** in **informativni**. Ne zamenjajte slogana z motom. Moto izraža načelo, cilj ali ideal, slogan pa izraža **dodano vrednost**. Če obiskovalec ne ve, o čem je spletna stran, mu to lahko pove slogan, ne pa moto.
- ✚ Glej npr. <http://www.jango.com/>, <http://www.nngroup.com/>, <http://www.onvia.com/>, <http://www.domovanje.com/>, <http://www.domenca.com/>, <http://1nadan.si>.
- ✓ **Nagovor obiskovalca** naj bo jasno viden na izpostavljenem delu domače strani. Nagovor je začetna gesta vsakega obiska, zato mu tudi obiskovalci spletnega mesta posvetijo pozornost. Bolj kot vljudnostna gesta pa lastniku spletnega mesta nagovor služi kot vnovična priložnost za jedrnat opis spletnega mesta. Poskrbite, da nagovoru namenite dovolj prostora.

Slika 7: Slogan ob logotipu (<http://www.nngroup.com/>)

Navigacija na domači strani

Zaradi edinstvenih značilnosti domače strani je navigacija lahko na domači strani drugačna (a ne preveč!) v primerjavi z ostalimi stranmi. Razlike, ki jih lahko napravite, so naslednje.

- ✓ Poimenovanje podstrani. Ker mora domača stran razkriti čim več o spletnem mestu, lahko k samim imenom podstrani v navigacijo dodate tudi opisno besedno zvezo, če imate dovolj prostora pa lahko celo izpišete nekatere podkategorije (glej sliko).



Slika 8

- ✓ Uporabite lahko drugačno orientacijo (na primer vertikalno namesto horizontalne).
- ✓ Logotip je na domači strani navadno večji kot na drugih straneh, poleg njega pa se navadno nahaja še slogan. Ker tako oba zavzmeta več prostora kot na ostalih straneh, lahko v skladu s tem prilagodite tudi navigacijo.

Pri kakršnih koli spremembah pa upoštevajte sledeče

- ✓ Ohranite ista imena podstrani in njihovo enako razvrščenost.
- ✓ Ohranite isti vizualni stil: isto pisavo, barve in kapitalizacijo (velike/male črke).

Več priporočil za izdelavo domače strani najdete na <http://www.nngroup.com/articles/113-design-guidelines-homepage-usability/>

6. Slike in grafični elementi

- ✓ Izogibajte se množici velikih slik. Raje uporabite majhne slike (t.i. »thumbnails«, ki služijo kot povezave na večje slike.
- ✓ Pri izdelavi majhnih slik pred zmanjšanjem izrežite nepomembno vsebino.
- ✓ Ne uporabljajte elementov, ki spominjajo na oglase: pojavna okna, utripanje, animacije itd.
- ✓ Uporabite slike in grafiko za prikaz prave vsebine in ne le za okras. Na primer, prikažite slike pravih ljudi ali izdelkov, ne pa slik modelov, ki jih kupite na spletu.
- ✓ Ne uporabljajte »Flash-a«, če ga zares ne potrebujete.

Literatura

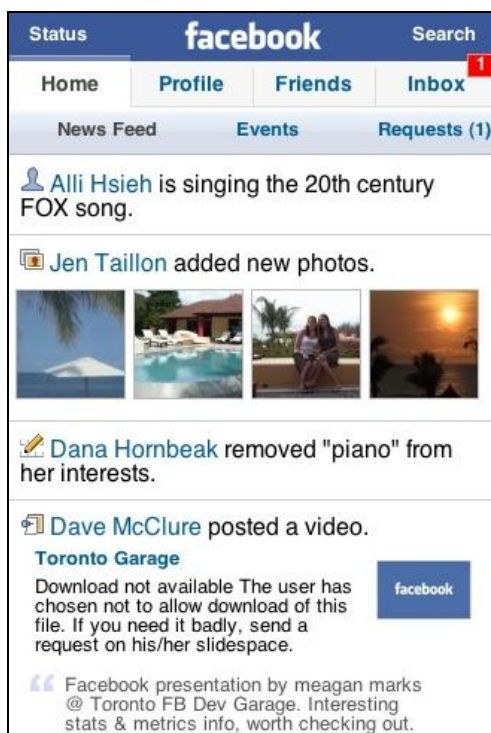
- Top 10 Mistakes in Web Design
<http://www.nngroup.com/articles/top-10-mistakes-web-design/>
- Steve Krug (2006) Don't Make Me Think! A common sense approach to web usability. New Riders Publishing.

3

Razvoj mobilnih spletnih strani

1. Uvod

Izraz 'mobilni splet' se nanaša na dostop do spleta preko brskalnika na mobilnem telefonu, ki je v internet povezan preko brezžičnega mobilnega omrežja. Brskanje po spletu z uporabo mobilnega terminala ima svoje specifične in probleme. Največja problema sta interoperabilnost in uporabnost.



Slika 9: Primer spletne strani, prilagojene za prikaz na mobilnem terminalu
(vir: <http://blogohlert.typepad.com/photos/uncategorized/2007/08/28/fbiphone1.jpg>)

Težave z interopreabilnostjo izhajajo iz velikega števila medsebojno nezdržljivih platform, na katerih teče programska oprema mobilnih terminalov, operacijskih sistemov, brskalnikov, pa tudi strojne opreme različnih mobilnih terminalov. Problemi z uporabnostjo so povezani z majhnimi dimenzijami mobilnega terminala, kar posledično vodi v omejitve prikaza vsebine in interakcije uporabnika s terminalom.

2. Izdelava mobilnih spletnih strani

a. Omejitve mobilnih terminalov

Omejitve mobilnih terminalov lahko razdelimo v več kategorij:

- **Majhni zasloni**
 - Velikost zaslona - Na mobilnem terminalu je skoraj nemogoče videti besedilo in slike, ki so namenjene prikazu na običajnemu namiznemu računalniku.
 - Okna - Namizni računalniki imajo sposobnost prikaza enega ali več oken hkrati, med katerimi lahko uporabnik preklaplja. Mobilni terminal lahko prikazuje le eno okno, spletne strani pa lahko uporabnik pregleduje samo v zaporedju, tako kot so bile prvotno dostopane.
- **Omejena interakcija uporabnika s terminalom**
 - Vhodne naprave - Večina mobilnih naprav nima vhodne naprave, podobne miški, ampak ima zagotovljeno le osnovno premikanje po zaslonu v dveh ali štirih smereh. Zaradi tega je navigacija po straneh omejena.
- **Omejena moč procesiranja in količine spomina**
 - Odsotnost JavaScripta in piškotkov - Večina mobilnih terminalov (razen t.i. »pametnih telefonov«) ne podpira skriptnih jezikov na odjemalcu (JavaScript) in »piškotkov«. Tako je onemogočena identifikacija uporabnikov, personalizacija vsebine, validacija vnesenih podatkov itd.
 - (Ne)dostopnost spletnih strani in vsebin – Veliko strani, ki so dostopne prek namiznega računalnika je nedostopnih preko mobilne naprave. Veliko mobilnih terminalov še vedno namreč ne podpira varnih povezav, Flash-a, PDF, videa ter drugih tehnologij in vsebin.
- **Omejena pasovna širina**
 - Hitrost – Večina mobilnih terminalov je povezanih v omrežja z relativno majhnimi prenosnimi hitrostmi, ki so pogosto počasnejše od (fiksne) klicnega dostopa do interneta (t.i. dial-up).
 - Cena – Poleg počasnih povezav uporabnike mobilnega interneta omejujejo tudi relativno visoke cene prenosa podatkov v primerjavi s tistimi v fiksni omrežjih.

Spletne strani, ki so v osnovi namenjene prikazu na namiznih računalnikih s hitrimi povezavami do interneta, so v mobilni domeni prilagojene gornjim omejitvam in so zato vsebinsko in oblikovno okrnjene.

b. Arhitektura mobilne spletne strani

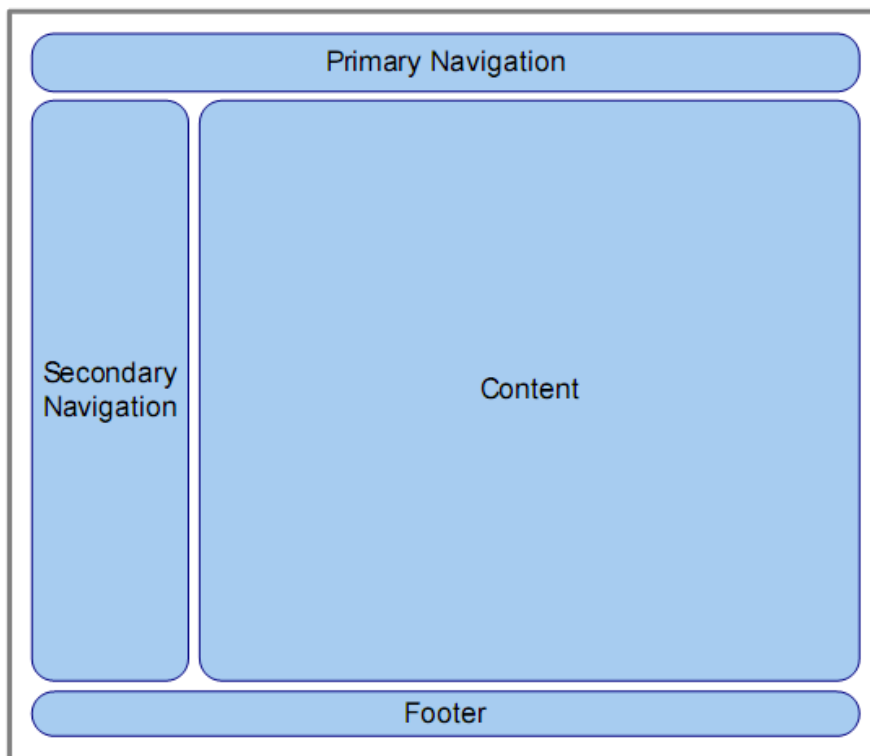
Zaradi predstavljenih omejitev je pri izdelavi aplikacij v spletni domeni pomembno identificirati najpomembnejše informacije in te nato čim bolj preprosto strukturirati. Pri tem je potrebno upoštevati sledeče:

- Na mobilni spletni strani je potrebno predstaviti le najbolj relevantno vsebino.
- Vsebina naj bo predstavljena v okviru dobro definiranih in označenih kategorij. Spletna stran naj bo tako razbita po podstraneh.
- Kategorij naj bo čim manj, poleg tega naj bo tudi čim manj nivojev kategorij (največ pet).
- Povezave na kategorije (podstrani) naj bodo omejene na največ 10 na stran. Omogočena naj bo izbira preko številčne tipkovnice (0-9).
- Kategorije naj bodo urejene glede na »popularnost«. Kategorije, ki so najpogosteje obiskane, naj bodo prikazane pred ostalimi, tako da jih bo uporabnik prej našel.
- Ko identificiramo vsebino, ki je relevantna za prikaz na mobilni spletni strani, si je pred izdelavo same spletne strani smiselno skicirati arhitekturo strani, kot to prikazuje Slika 10.

c. Dimenzije mobilnih spletnih strani

Ena izmed specifičnosti mobilnih terminalov je raznolikost dimenzij njihovih zaslonov, kar vpliva tudi na prikaz spletnih strani. K problemu prikaza spletne strani na mobilnem terminalu lahko pristopimo na dva načina:

- Če se odločimo, da bomo vsebino prilagajali različnim terminalom in velikosti njihovih zaslonov, je smiselno te uvrstiti v kategorije glede na dimenzije zaslona, kot to prikazuje Slika 11.
- Če se odločimo, da bomo pripravili samo eno mobilno stran za vse mobilne terminale, potem je smiselno skleniti kompromis med omenjenimi kategorijami. Na Sliki 11 je ta kompromis prikazan črtkano.



Slika 12: Struktura običajne spletne strani



Slika 13: Struktura mobilne spletne strani

d. Navigacijske sheme

Običajne spletne strani uporabljajo različne sheme navigacije, npr. z zavihki, različnimi vrstami menijev itd. Te sheme uporabniku dajejo vizualne vtise o tem, kje znotraj spletnega mesta se nahaja in kako od tu pride na drugo mesto. V mobilnem svetu je zaradi vseh omejitev navigacija po spletni strani otežena.

Najenostavnejša in na mobilnih straneh najpogostejša je **vertikalna razpostavitev** vsebine in možnosti, ki jih ima uporabnik (Slika 13). Dobra praksa je, da posameznim izbiram priredimo tudi dostop prek tipk številčne tipkovnice (0-9). Prikaz različnih nivojev navigacije v mobilnih straneh ni primeren, saj odžira dragocen prostor. Bolje je prikazati le tiste možnosti, ki so bistvene znotraj posamezne podstrani. Poleg njih mora vsaka podstran imeti tudi povezave, preko katerih uporabnik lahko »pobegne« bodisi na nadrejeno kategorijo, bodisi na naslednjo ali izhodiščno stran. Te povezave se navadno nahajajo na dnu strani. Na ta način uporabniku omogočajo prehod na želeno mesto, ne da se bi mu bilo potrebno pomakniti nazaj na vrh strani.

Vertikalna razpostavitev vsebuje (Slika 13):

- glavo (naslov strani itd.),
- navigacijo (kot je opisana zgoraj),
- vsebino (najpomembnejše informacije v zvezi z izbrano kategorijo),
- ponovljeno navigacijo,
- »nogo« (avtorji strani, avtorske pravice itd.)

e. Mobilni standardi

V razdelku so na kratko predstavljeni pomembnejši mobilni spletni standardi.

WAP

WAP (*Wireless Access Protocol*) je bil prvi široko razširjen skupek standardov za mobilni splet. WAP združuje tako označevalni jezik za zapis strani (WML, XHTML-MP) kot tudi protokole za njegov prenos. Obstajata dve pomembnejši različici:

- WAP 1.0 (uporablja WML kot primarni označevalni jezik)
- WAP 2.0 (uporablja XHTML-MP kot primarni označevalni jezik)

XHTML MP / XHTML Basic

XHTML-MP (*eXtensible Hypertext Markup Language – Mobile Profile*) je posebna različica XHTML, namenjena prikazu na mobilnih terminalih. XHTML-MP je razširitev XHTML Basic 1.1.

Wireless CSS in CSS Mobile Profile 2.0

Tudi v mobilnem svetu obstaja dobra praksa ločitve vsebine od njene oblike. Za določanje slednje sta v uporabi jezika WCSS (*Wireless Cascading Style Sheets*) in CSS Mobile Profile 2.0, ki sta okrnjeni različici »klasičnega« CSS.

3. Priporočila pri izdelavi mobilnih spletnih strani

Razdelek predstavlja povzetek priporočil (oz. »dobrih praks«) pri izdelavi mobilnih spletnih strani. Vsako priporočilo je opremljeno z oznako, ki služi kot referenca. Natančnejše opise priporočil in dobrih praks dobite v literaturi.

a. Navigacija in povezave

[URI_NASLOVI]

URI naslovi vstopnih točk na spletno mesto naj bodo kratki.

[NAVIGACIJA]

Zagotovite le osnovno navigacijo, ki naj bo postavljena na vrhu in dnu strani. Uporabite enak mehanizem navigacije po vseh podstraneh.

[URAVNOTEŽENOST]

Sklenite kompromis med številom postavk v navigaciji in številom povezav, ki jim mora uporabnik slediti, da pride na željeno stran.

[TIPKE_DOSTOPA]

S pomočjo atributa *accesskey* omogočite uporabo številčnih tipk mobilnega terminala v povezavah, navigaciji in za dostop do najpogosteje uporabljene funkcionalnosti.

[POVEZAVE]

Uporabniku jasno nakažite, kam vodi posamezna povezava, tako da se ta lahko odloči, ali je zanj vsebina pomembna. Če povezava vodi do datoteke, ob njej navedite format datoteke, če gre za format, različen od XHTML, GIF ali JPG. Če gre za veliko datoteko, navedite tudi njeno velikost.

[POP_UP_OKNA]

Ne uporabljajte pop-up ali kakršnih koli drugih pojavnih oken.

[ZUNANJI_VIRI]

Obdržite minimalno število zunanjih virov, ki so navedeni v HTML (slik, CSS skript ali drugih objektov). Za prenos vsakega od omenjenih virov je namreč potrebna ločena zahteva strežniku, kar poveča čas nalaganja strani.

b. Vsebina in izgled strani**[PRIMERNOST_IN_JASNOST]**

Poskrbite, da je vsebina spletne strani primerna za uporabo v mobilnem svetu. Uporabniki tu namreč ponavadi iščejo točno določene informacije in ne »brskajo« tako kot po klasičnem spletu. Osredotočite se na vsebino, ki jo je uporabnik zahteval. Uporabljajte preprost in jasen jezik.

[VELIKOST_STRANI]

Sklenite kompromis med dolžino strani in številom strani, če neko stran razbijete na manjše. Strani naj ne bodo predolge, saj ima mobilni terminal omejen pomnilnik, po drugi strani pa naj jih ne bo preveč, ker je »klikanje« za uporabnika nepraktično. Posamezna stran naj ne preseže velikosti 10 kB.

[DRSENJE]

Omejite drsenje strani (scrolling) v eno smer (navzdol).

[OSREDNJA_VSEBINA]

Vsebina, ki je osrednjega pomena za spletno stran, naj bo prikazana pred ostalo vsebino (sekundarna navigacija, dekorativne slike, oglasi itd.). Uporabnik naj ne bi preveč »drsel« po strani, da bi videl vsebino, ki je na strani bistvena.

[GRAFIKA]

Ne uporabljajte grafičnih elementov za prostorsko ločevanje vsebine. Izogibajte se slikam z visoko ločljivostjo.

[BARVE_IN_OZADJE]

Zagotovite, da so informacije, ki so podane z barvo, posredovane tudi drugače. Uporabite barvne kombinacije, ki zagotavljajo zadosten kontrast med ozadjem in ospredjem. Če uporabljate sliko za ozadje, zagotovite primeren kontrast, tako da je vsebina berljiva.

Mobilne naprave ne omogočajo visokega barvnega kontrasta, poleg tega pa se navadno uporabljajo v okoljih, kjer svetlobne lastnosti niso najboljše. Informacija, ki jo posredujete z barvo, morda ne bo vidna uporabnikom. Če barva nakazuje neko lastnost, potem naj bo ta lastnost izražena tudi kako drugače. Ne uporabljajte modre ali škrlatne barve, ker bi uporabnik

besedilo v tej barvi lahko zamenjal s hiperpovezavami, še posebno na napravah, ki ne podčrtujejo povezav.

c. Definicija strani

[NASLOV_STRANI]

Zagotovite kratek a opisen naslov strani (v elementu *title*). Naslov mora biti kratek zato, da ga brskalnik lahko izpiše v eni vrstici. Brskalniki navadno uporabijo naslov strani tudi kot oznako v zaznamkih, kjer je prostor spet omejen. V naslovu se osredotočite predvsem na opis vsebine.

[OKVIRJI]

Ne uporabljajte okvirjev (frames).

[STRUKTURA]

Uporabljajte elemente, ki nakazujejo logično strukturo dokumenta, na primer elemente za poglavja in podpoglavja (*h1*, *h2* itd.), namesto elementov za stil (*b*, *u*, *i* itd.). Poglavja naj bodo pravilno gnezdena.

[TABELE]

Ne uporabljajte tabel.

[TEKSTOVNE_ALTERNATIVE]

Zagotovite nadomestno tekstovno vsebino vsem ne-tekstovnim elementom.

Slednje je še posebej pomembno pri slikah. Ker prenos slik povzroča dodatno zakasnitev pri prikazu strani in ceni prenosa, se uporabnik lahko odloči, da bo njihov prenos izklopil. Tudi če se za to ne odloči, je tekstovna vsebina uporabna, saj je na strani prikazana, dokler se ne v celoti prenese slika.

[OBJEKTI_IN_SKRIPTI]

Ne uporabljajte vgrajenih objektov ali skript (ali pa se vsaj ne zanašajte na njihovo pravilno delovanje).

[SLIKE]

Če so slike, ki jih želite prikazati na spletni strani, prevelike, jih za prikaz na mobilnem terminali zmanjšajte (zmanjšajte njihovo ločljivost). Poleg tega v HTML dokumentu s pomočjo atributov *width* in *height* v elementu *img* določite dimenzije za njihov prikaz.

[VELJAVNOST_OZNAČB]

Sestavljajte dokumente, ki so v skladu s formalno slovnico. V nasprotnem primeru je prikaz dokumentov nepredvidljiv in potencialno nepopoln. Uporabljajte validatorje, npr.

<http://www.w3.org/QA/Tools/#validators>

[MERE]

Ne uporabljajte absolutnih enot (npr. px) za določanje dimenzij pri oblikovanju dokumenta (razen pri slikah). Raje uporabite odstotke ali relativne oznake, kot so *em*, *ex*, *bolder*, *larger* itd.

[OBLIKOVANJE]

Uporabite CSS za oblikovanje strani. CSS dokument naj bo kratek in učinkovit. Organizirajte strani tako, da bodo berljive tudi brez uporabe CSS.

[MINIMIZACIJA]

Zapis v HTML naj bo učinkovit. Odstranite nepotrebne znake, kot so presledki, nove vrstice itd. Za oblikovanje strani ne uporabljajte t.i. »inline« stilov, ampak se poslužujte HTML atributa *class* za določanje stila v CSS.

[KODIRANJE_IN_TIP_DOKUMENTA]

Da bodo vsi znaki prikazani, kot pričakujete, uporabljajte ustrezno kodiranje. Najbolj je priporočljiva uporaba kodne tabele UTF-8. Na začetek XHTML dokumenta dodajte direktivo:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Da bo brskalnik stran prikazal pravilno, določite pravilen tip dokumenta (doctype). Tip dokumenta brskalniku pove, kako naj stran prikaže, pravila ki jih pri tem uporabi in kako striktno naj se drži teh pravil. Primer doctype deklaracije za XHTML Basic 1.1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.1//EN"
"http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd">
```

d. Uporabniški vnos**[UPORABNIŠKI_VNOS]**

Zmanjšajte uporabniške vnose (še posebej tekstovne) na minimum. Kjerkoli je potrebna izbira, uporabite radio gumbe, izbirne sezname ali druge gradnike, ki ne zahtevajo tekstovnih vnosov. Pri uporabnikovih izbirah le-te omogočite z uporabo številčnih tipk.

[OZNAČEVANJE_GRADNIKOV_OBRAZCA]

Uporabite HTML element **label** za označitev vseh gradnikov v obrazcu. Poskrbite, da bo oznaka vedno neposredno ob gradniku.

4. Potek in pripomočki za vajo

Obstoječo spletno stran bomo preoblikovali v mobilno spletno stran. Potrebovali bomo:

- Spletni brskalnik
- HTML urejevalnik
- Program za delo s slikami
- Emulator mobilnega brskalnika

5. Priprava na vajo

- Poiščite spletno stran, ki ste jo izdelali kot projekt pri Spletnih tehnologijah, in jo prinesite s seboj na vaje.
- Premislite, kako bi svojo spletno stran preoblikovali v mobilno.
- Skicirajte izgled in strukturo (zemljevid) mobilne spletne strani. Skice prinesite na vaje.
- Prilagodite vsebine, ki jih boste obdržali v mobilni spletni strani (smiselno si je na primer že pripraviti slike – njihove dimenzije prilagoditi mobilni strani).
- Izberite si HTML urejevalnik, ki ga boste uporabljali na vajah (npr. PageBreeze).
- **Ponovite jezika HTML in CSS, ki smo ju obravnavali pri Spletnih tehnologijah.**
- **Temeljito predelajte priporočila in dobre prakse pri izdelavi mobilnih spletnih strani.** Seznanite se tudi z ostalimi tehnologijami, ki so omenjene v tej skripti.

6. Navodila za vajo

S pomočjo zgoraj navedenih pripomočkov izdelajte mobilno različico spletne strani. Pri tem upoštevajte sledeče:

- Spletna stran naj bo primerna za prikaz na katerem koli mobilnem terminalu (stran naj ne bo namenjena specifičnemu terminalu).
- Upoštevajte vse dobre prakse, ki so navedene v tej skripti.
- Kjer je mogoče, smiselno upoštevajte tudi nekatera priporočila iz 2. vaje (uporabniška izkušnja).
- Upoštevajte specifikacije XHTML Basic in WCSS (za povezave glej literaturo).
- Stran mora vsebovati
 1. vsaj 4 podstrani, ki so med seboj povezane preko kazala/navigacije, ki se nahaja na vsaki podstrani (na vrhu strani in v nogi);
 2. povezave med podstranmi v navigaciji, pri čemer mora biti omogočena tudi navigacija preko številčnih tipk;
 3. nekaj slik;
 4. obrazec;
 5. povezavo na vaš e-mail in telefonsko številko.
- Uporabite pripravljeno predlogo mobilne spletne strani, ki jo najdete na naslovu <http://www.lkn.fe.uni-lj.si/gradiva/TA/vaja3/>

Delovanje mobilne spletne strani lahko med razvojem preizkusite v brskalniku ali na lastnem mobilnem telefonu

- Preko spletne strani <http://www.lkn.fe.uni-lj.si/mobi/upload.html> prenesite na strežnik vse vaše datoteke v novo mapo z vašim imenom (sledite navodilom).
- Povezavo, ki vam jo po prenosu prikaže spletna stran, vpišite v brskalnik v svojem mobilnem telefonu.

Vaja je priznana, če je rezultat delujoča mobilna spletna stran, izdelana na podlagi gornjih zahtev.

7. Literatura

- W3C, Mobile Web Best Practices, 1.0 <http://www.w3.org/TR/mobile-bp/>
 - XHTML™ Basic 1.1 W3C specifikacije <http://www.w3.org/TR/xhtml-basic/>
 - DotMobi <http://mtd.mobi/>
 - DotMobi emulator <http://mtd.mobi/emulator.php>
 - Wireless CSS specifikacije
http://www.openmobilealliance.org/technical/release_program/docs/Browsing/V2_3-20080331-A/OMA-WAP-WCSS-V1_1-20061020-A.pdf
 - CSS Mobile Profile 2.0 <http://www.w3.org/TR/css-mobile/>
 - Comparison of CSS 2.1, CSS MP, WCSS and CSS Level 1
<http://mobiforge.com/designing/story/comparison-css-21-css-mp-wcss-and-css-level-1>
- Opera Mini Simulator <http://www.opera.com/mini/demo/>

4

Razvoj aplikacij na mobilni platformi Android

1. Uvod

V okviru vaje bomo razvili preprosto aplikacijo na Googlovi mobilni platformi Android. Razvoj aplikacij na omenjeni platformi poteka v programskem jeziku Java z uporabo že pripravljenih javanskih knjižnic, ki so del t.i. aplikacijskega programskega vmesnika (*Application Programming Interface*, API). API je vmesnik med izvajalnim okoljem platforme Android in razvijalcem aplikacije. API izpostavlja funkcionalnosti platforme, tako da jih lahko razvijalec enostavno uporabi s pomočjo že pripravljenih razredov, metod in ostalih elementov objektnega programiranja.

Mobilne naprave imajo nekatere specifičnosti, ki jih je potrebno upoštevati pri razvoju aplikacij. V primerjavi z namiznimi računalniki imajo mobilne naprave:

- manjšo moč procesiranja,
- omejen delovni pomnilnik,
- omejene kapacitete trajnega shranjevanja podatkov,
- manjše zaslone z manjšo ločljivostjo,
- manj zanesljive podatkovne povezave,
- višje stroške prenosa podatkov,
- počasnejši prenos podatkov z večjimi zakasnitvami,
- omejeno trajanje baterije.

Ker podrobnejša obravnava vseh teh vidikov presega okvir predmeta, se bomo v vaji le bežno dotaknili nekaterih izmed njih. Obravnavali bomo:

- umestitev aplikacije na platformo,
- uporabo multimedijskih virov (slike, zvok),
- izdelavo grafičnega uporabniškega vmesnika,
- uporabniško interakcijo z uporabo t.i. »dogodkov«,
- uporabniško interakcijo z uporabo menijev in dialogov.

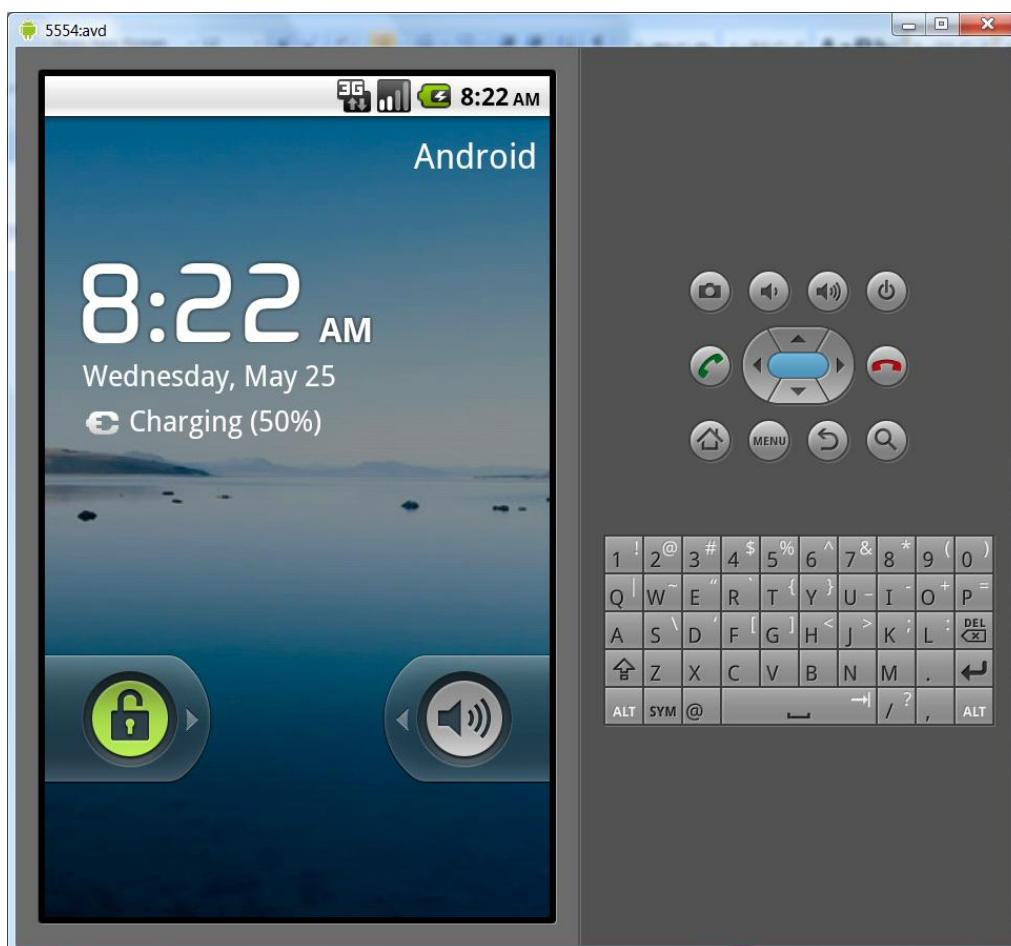
Rezultat vaje bo preprosta igra. Večina njene funkcionalnosti bo pripravljena v obliki delčkov kode, ki jih bo potrebno povezati v delujočo aplikacijo.

2. Razvoj aplikacij na platformi Android

Razvoj aplikacij na platformi Android je mogoč s pomočjo brezplačnega razvojnega kompleta (SDK, *Software Development Kit*). Razvojni komplet sestavljajo orodja, od katerih so za nas najpomembnejša:

- **Programske knjižnice**, ki predstavljajo »most« med platformo Android in našo aplikacijo.
- **Razvojno okolje** v obliki dodatka za Eclipse.
- **Emulator** – računalniški program, s katerim dosežemo posnemanje delovanja prave mobilne naprave. Emulator služi za testiranje aplikacij brez uporabe mobilnega terminala.

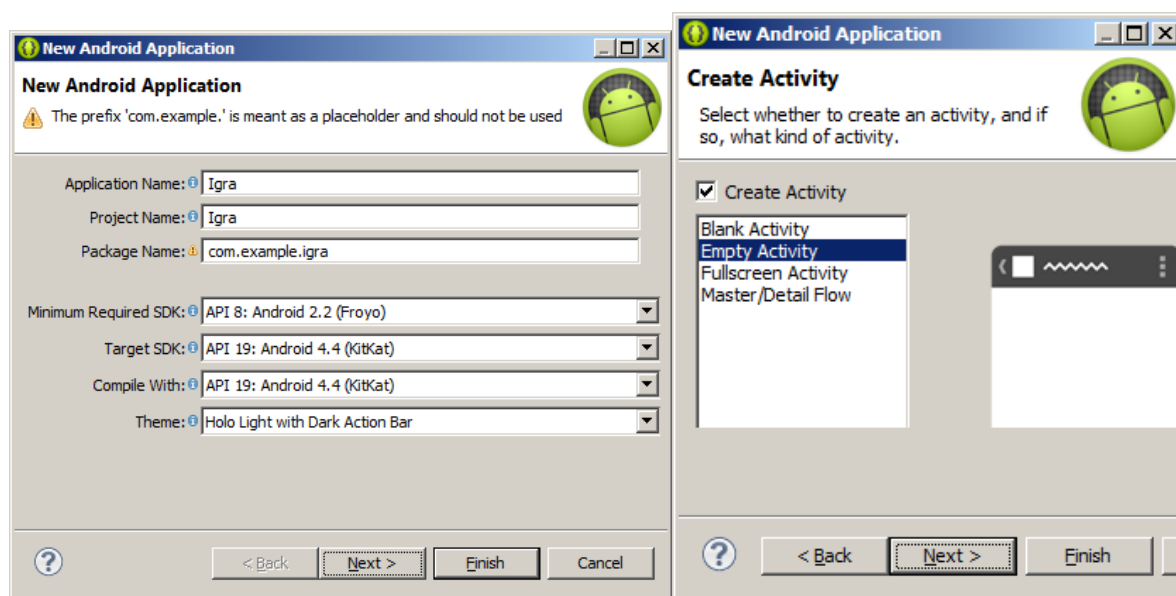
Postopek vzpostavitve razvojnega okolja je opisan v prilogi.



Slika 14: Androidov emulator

3. Razvojno okolje in viri, ki tvorijo aplikacijo

Ko je razvojno okolje pripravljeno za uporabo, lahko ustvarite nov Androidov projekt (*File/New/Project.../Android/Android Application Project*). Polja v oknu, ki se vam prikaže, izpolnite tako, kot je to prikazano na Sliki 15, nato pa izberite trikrat **Next** in **Finish**. Če vam okolje Eclipse javlja napake, ga ponovno poženite.

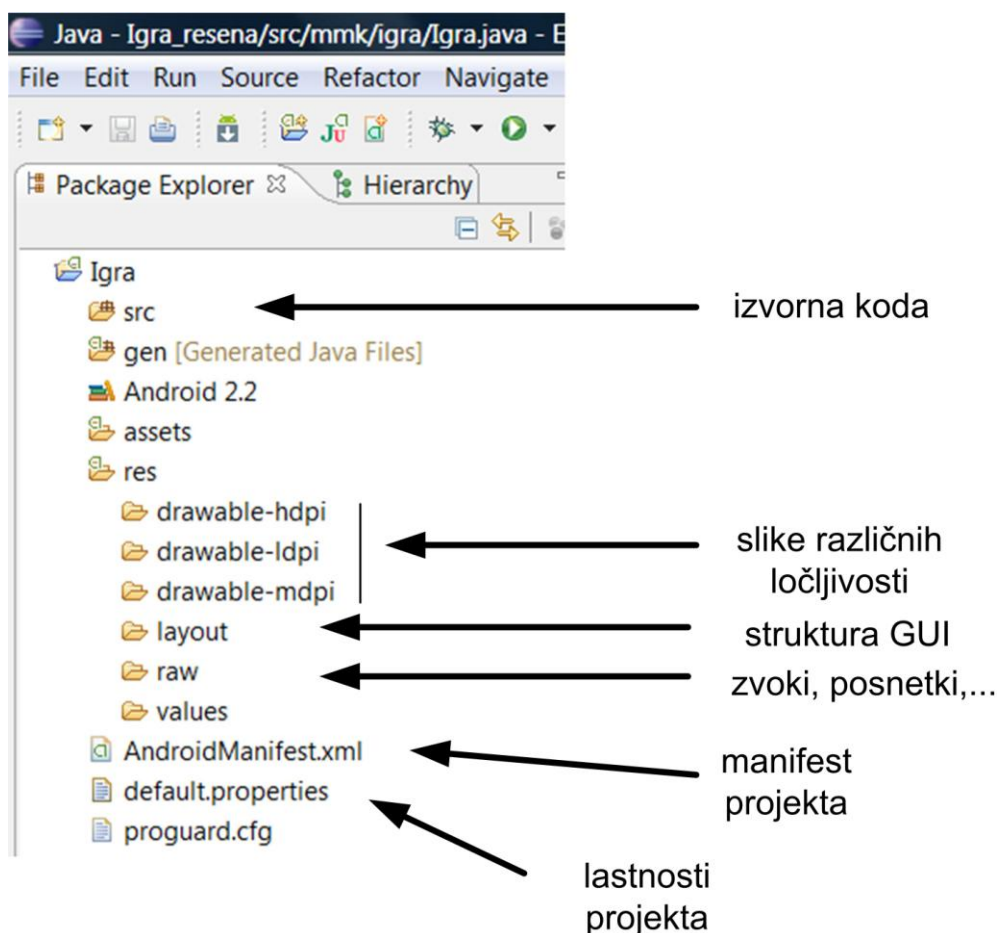


Slika 15: Ustvarjanje novega projekta

Slika 16 prikazuje strukturo Androidovega projekta. Pomembnejše mape v projektu so sledeče:

- mapa *src* vsebuje izvorno kodo
- v mapo *res* damo ostale vire, in sicer
 - slike različnih ločljivosti v mape *drawable-*,
 - razporeditev gradnikov določimo v mapi *layout*,
 - zvočne datoteke postavimo v mapo *raw*,
 - v mapi *values* so vrednosti (tekstovne oznake itd.), ki se pojavljajo znotraj aplikacije.

Med razvojem lahko aplikacijo preizkušamo v emulatorju, tako da z desnim miškinim gumbom kliknemo na ime projekta in izberemo *Run As/Android Application*.



Slika 16: Struktura Androidovega projekta v razvojnem okolju

4. Razvoj aplikacije

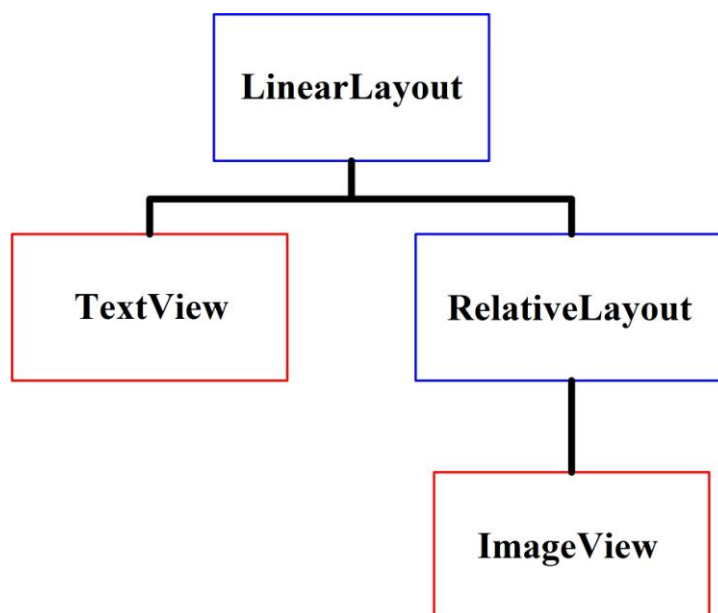
Aplikacije za platformo Android so sestavljene iz komponent, ki jih izvajalno okolje ustvari in požene. Za naše potrebe je pomembna le komponenta **Activity** (prevedeno: *aktivnost*), ki jo potrebujemo pri izdelavi aplikacij z grafičnim uporabniškim vmesnikom. Vsaka konkretna različica komponente **Activity** je izdelana kot potomec razreda **Activity**, kar v programskem jeziku java nakažemo z uporabo rezervirane besede **extends**. Razvojno okolje navadno ustvari ogrodje različice komponente **Activity** (Slika 17), ko ustvarimo nov projekt. Z vidika razvijalca aplikacij je vstopna točka aplikacije metoda **onCreate**. Z drugimi besedami, v to metodo vključite programsko kodo, za katero želite, da se izvede najprej.


```
1 package AT.AT;
2
3 import android.app.Activity;
4
5
6 public class AT extends Activity {
7     /** Called when the activity is first created. */
8     @Override
9     public void onCreate(Bundle savedInstanceState) {
10        super.onCreate(savedInstanceState);
11        setContentView(R.layout.main);
12        //tu se začne vaša aplikacija
13    }
14 }
```

Slika 17: Aktivnost je osnovna komponenta Androidove aplikacije z grafičnim uporabniškim vmesnikom. Aktivnost je razred, ki je potomec razreda *Activity*, ta pa je del aplikacijskega programskega vmesnika. Gradnja uporabniškega vmesnika se začne v poveženi metodi *onCreate*.

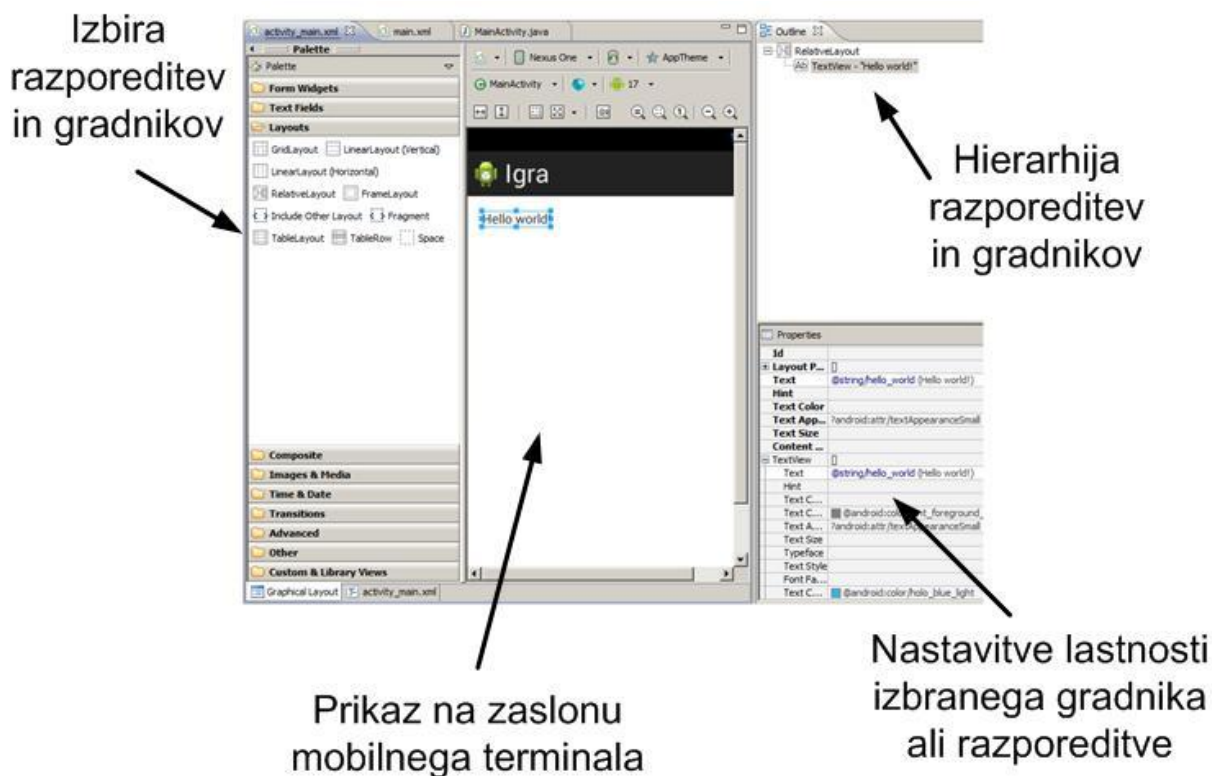
5. Grafični uporabniški vmesnik

Vsaki *aktivnosti* pripada svoj zaslon. Vizualna vsebina zaslona je strukturirana v hierarhiji *pogledov* (*views*) ter ostalih gradnikov. Način razporejanja gradnikov znotraj starševskega gradnika dosežemo z *razporeditvami* (*layouts*), kot to shematsko prikazuje Slika 18.



Slika 18: Primer strukture uporabniškega vmesnika: v linearno razporeditev (*LinearLayout*) sta vključena pogled za prikazovanje teksta (*TextView*) in absolutna razporeditev (*RelativeLayout*), v katero je postavljen pogled za prikazovanje slik (*ImageView*).

Struktura uporabniškega grafičnega vmesnika je opisana v XML datoteki, ki se nahaja v mapi *res/layout*. Ko kliknemo na datoteko *main.xml*, se prikaže urejevalnik, s katerim lahko načrtujemo grafični uporabniški vmesnik aplikacije (Slika 19).



Slika 19: Urejevalnik za načrtovanje grafičnega uporabniškega vmesnika Androida v okolju Eclipse.

1. Urejevalnik vam na levi strani ponuja nabor gradnikov in razporeditev, s katerimi lahko sestavite uporabniški vmesnik.
2. Gradnik ali razporeditev uporabite tako, da ga/jo z miško povlečete na želeno mesto v sredinskem oknu, ki predstavlja zaslon mobilnega terminala.
3. Okno »Outline« prikazuje hierarhijo uporabljenih razporeditev in gradnikov.
4. Okno »Properties« prikazuje lastnosti trenutno izbranih gradnikov in razporeditev. Gradniku ali razporeditvi lahko nastavite lastnosti tako, da ga/jo najprej izberete v desnem oknu.

5. Interakcija uporabnika preko dogodkov

»Dogodek« je pojem, ki je povezan z uporabnikovo interakcijo z določenim gradnikom uporabniškega vmesnika. Tipični dogodki so na primer:

- dotik zaslona,
- dvojni klik na ikono,
- zasuk mobilnega terminala za 90 stopinj.

Za izkoriščanje dogodkov v aplikaciji moramo **dogodke »prestreči«**. To storimo tako, da:

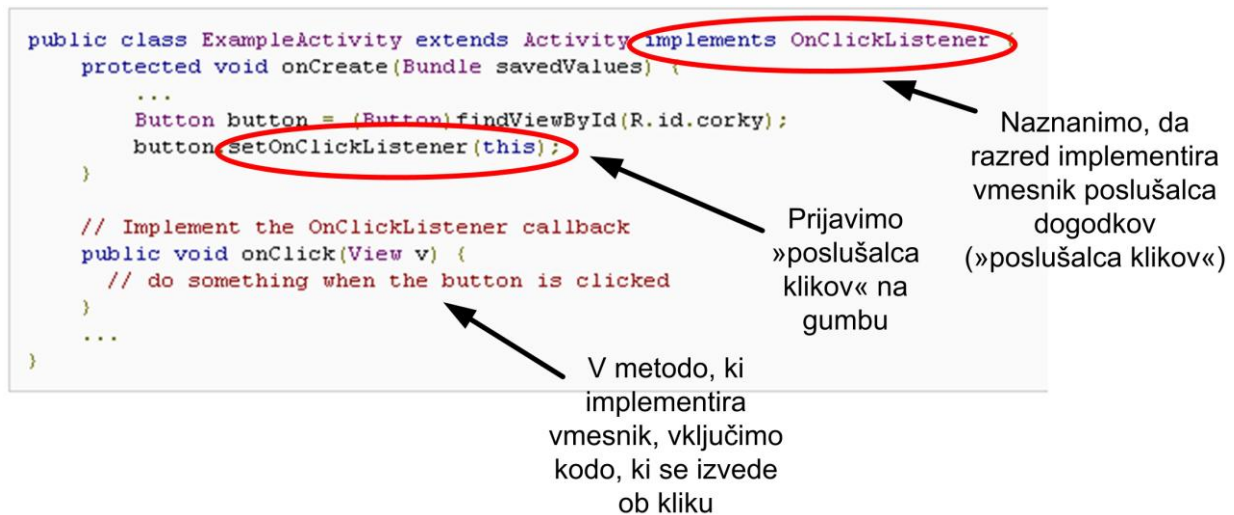
- **Prijavimo »poslušalca« izbranega dogodka na izbranem gradniku.**
- V argumentu metode, s katero prijavimo poslušalca, **navedemo, kje se nahaja upravljalnik dogodka**, ki ga prijavljamo;
- **izdelamo »upravljalnik dogodka«**. Upravljalnik dogodka je del kode, v katerem je sprogramiran odziv na dogodek. Ta koda se izvede, ko se dogodek zgodi.

Primer uporabe dogodka prikazuje koda na Sliki 20. Aktivnost *ExampleActivity* vsebuje gumb *button*, na katerem je registriran poslušalec *onClickListener*, ki posluša klike na ta gumb¹. Če bi pogledali v specifikacije poslušalca *onClickListener*, bi ugotovili, da mora biti koda, ki se izvede ob kliku, vključena v metodo *onClick* (metoda *onClick* je torej upravljalnik dogodka »klik«). Ob prijavi poslušalca z rezervirano besedo *this* nakažemo, da bomo metodo *onClick* vključili v isti razred², znotraj katerega pravkar registriramo poslušalca (v našem primeru v razred *ExampleActivity*).

Na podoben način lahko »poslušamo« dogodke tudi na drugih gradnikih grafičnega uporabniškega vmesnika. Imena metod, v katerih mora biti sprogramirana reakcija na posamezen dogodek, najdemo v dokumentaciji aplikacijskega programskega vmesnika.

¹ Če se v predstavljeni kodi ne znajdete, ponovite osnove objektnega programiranja! Omenjene principe ste obravnavali pri gradnji grafičnih uporabniških vmesnikov. Za naše potrebe bo zadostovala »centralna obravnava dogodkov«.

² Natančneje: rezervirana beseda *this* se nanaša na konkretno različico razreda, torej na objekt, ki v našem primeru predstavlja aktivnost, ki je »trenutno« aktivna.



Slika 20: Prijava in implementacija poslušalca dogodka.

6. Uporaba menijev in dialogov

Meniji

S pomočjo menijev lahko aplikaciji nastavljamo funkcije in lastnosti. Platforma Android ponuja že nekaj pripravljenih oblik menijev, ki jih lahko uporabimo.

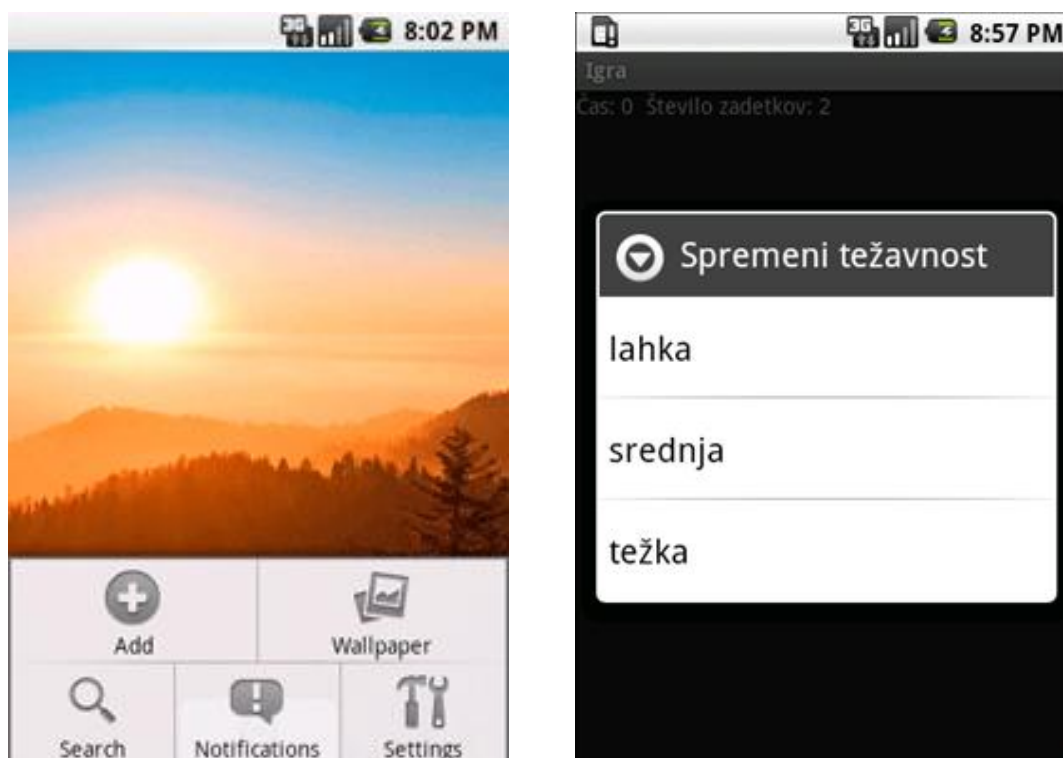
Najpogostejši je meni, ki se prikaže ob pritisku na temu namenjeno tipko na tipkovnici mobilnega terminala. Pravimo mu »options« meni, prikazan pa je na levem delu Slike 21. V ta meni lahko vključimo ikone in tekst za največ šest menijskih postavk. Kadar ima aplikacija veliko število nastavitev, ki so organizirane po tematskih področjih, lahko znotraj »options« menija uporabimo tudi »podmenije« (Slika 21, desno).

Primer izdelave »options« menija dobite na:

<http://developer.android.com/guide/topics/ui/menus.html#options-menu>

Primer izdelave »podmenija« pa na:

<http://developer.android.com/guide/topics/ui/menus.html#submenu>



Slika 21: »Options« meni in podmeni

Dialogi

Dialog je majhno okno, ki se pojavi v trenutni aktivnosti. Obstaja več vrst dialogov, ki so namenjeni za pozive, zbiranje podatkov ali obveščanju uporabnika. V naši aplikaciji bomo uporabili dialog z obvestilom (*alert*) (Slika 22, desno).

Več o dialogih na:

<http://developer.android.com/guide/topics/ui/dialogs.html>



Slika 22: Dialogi

7. Priprava na vajo in potek vaje

Za gladek potek vaje se pripravite:

- **temeljito predelajte to gradivo**
- **na domačem računalniku namestite razvojno okolje za razvoj aplikacij na platformi Android in se z njim seznanite (glejte prilogo)**
- **ustvarite nov (prazen) Androidov projekt in poženite emulator**
- **temeljito ponovite osnove objektnega programiranja v Javi**
- **razvojno okolje (zip arhiv iz točke 1 v prilogi) prenesite na prenosni medij in ga prinesite na vaje**

Na vajah bomo izdelali preprosto igro, v kateri bomo uporabili nekatere funkcionalnosti platforme Android. Potek vaje je sledeč:

1. Vzpostavili bomo Androidovo razvojno okolje.
2. Ustvarili bomo nov Androidov projekt in novo aktivnost.
3. Izdelali bomo ogrodje grafičnega uporabniškega vmesnika.
4. Z uporabo dogodkov bomo omogočili interakcijo uporabnika z elementi grafičnega vmesnika.
5. Dodali bomo slike in zvok.
6. Dodali bomo menije in dialogno okno.
7. Izdelano igro bomo testirali na emulatorju in pravem mobilnem terminalu.
8. Tekmovali bomo, kdo bo boljši.

Večina funkcionalnosti bo pripravljena v obliki delčkov kode, ki jih bo potrebno povezati v delujočo aplikacijo.

8. Naloga

Po zaslonu se premika »balonček« z različnimi hitrostmi. Naloga uporabnika je, da ga s prstom »poči«, s tem pa dobi točko. Igra ima omejen čas in težavnost (hitrost premikanja balončka).



Slika 23: Igra na mobilni platformi Android

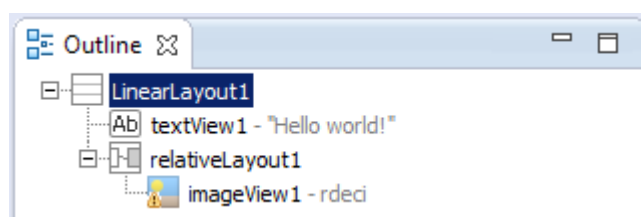
a) Priprava

1. S spletnega mesta <http://www.lkn.fe.uni-lj.si/gradiva/TA/vaja4/> prenesite delčke pripravljene izvorne kode in ostale vire aplikacije.
2. Namestite Androidovo razvojno okolje (glejte prilogo).
3. Ustvarite nov projekt in ga poimenujte »Igra«. Obrazec izpolnite tako, kot kaže Slika 15.
4. Razglejte se po vsebini projekta v *Package Explorer*-ju.
5. V mapi *res* (Slika 16) ustvarite mapo *raw*. V to mapo boste shranjevali zvoke.
6. Prenesite slike in zvočne datoteke v ustrezne mape, tako kot je to predstavljeno v 3. poglavju.
7. Odprite manifest projekta. Kot ikono aplikacije nastavite sliko balončka.

b) Grafični vmesnik

1. V razvojnem okolju odprite XML datoteko, v kateri je definirana struktura grafičnega vmesnika. Prikaže se vam okno, podobno tistemu na Sliki 19.
2. V naši aplikaciji bo vmesnik uporabljal razporeditev *LinearLayout*, v katero bomo umestili pogled *TextView* in razporeditev *RelativeLayout*. V slednjo bomo dali še pogled *ImageView* (Slika 24).

Pozor: razporeditve in poglede morate poimenovati natančno tako, kot so poimenovani na Sliki 24, ker so ta imena uporabljena v pripravljeni kodi!



Slika 24

3. V zavihku *Properties* gradnikom nastavite naslednje lastnosti:

gradnik	lastnost	vrednost
LinearLayout1	Orientation	vertical
LinearLayout1	Layout height	fill_parent
LinearLayout1	Layout width	fill_parent
LinearLayout1	Background	<i>datoteka s sliko neba</i>
textView1	Layout height	wrap_content
textView1	Layout width	wrap_content
relativeLayout1	Layout height	wrap_content
relativeLayout1	Layout width	wrap_content
imageView1	Layout height	50dp
imageView1	Layout width	50dp
imageView1	Src	<i>datoteka s sliko balončka</i>

4. Ko definirate strukturo grafičnega vmesnika, lahko zometke aplikacije že preizkusite v emulatorju (z desnim miškinim gumbom kliknemo na ime projekta in izberemo *Run As/Android Application*).

Nasvet: Zaganjanje emulatorja traja kar precej časa, zato ga ne izklaplajte med posameznimi testiranjmi aplikacije.

c) Ogrodje aplikacije

V nadaljevanju bomo sestavili aplikacijo na podlagi že pripravljenih delčkov kode.

Vsak delček pripravljene kode vključite v ločeno metodo. Metode boste nato dopolnili in jih medsebojno povezali po spodnjih navodilih!

1. Nadomestite vsebino datoteke *MainActivity.java* s kodo, ki je že pripravljena pod oznako **[aktivnost]**. Koda vsebuje ogrodje aplikacije.
2. Aktivnost dopolnite tako, da se ob zagonu izvede inicializacija, nato pa se začne igra. To storite tako, da kodo z inicializacijo **[inicializacija]** in začetkom igre **[zacniIgro]** najprej vključite v novi metodi, ki ju nato pokličite iz metode `onCreate`.
3. Kodo **[zacniIgro]** dopolnite tako, da sprožite premikanje balona **[premikanje]** in odštevanje časa **[odstevanje]**. V pripravljeni kodi imate s komentarji označena mesta, kjer morate kodo dopolniti.
4. Preverite delovanje aplikacije v emulatorju.

d) Poslušalci

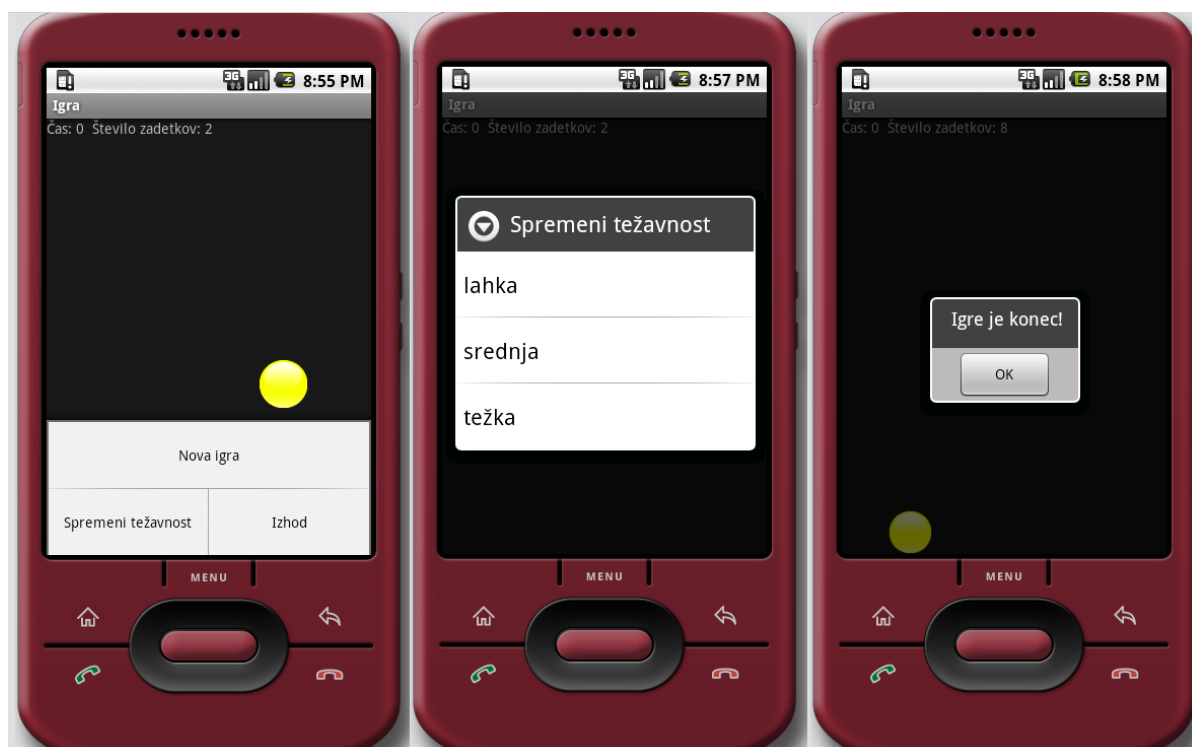
1. Prijavite poslušalca dotika na pogledu, ki vsebuje sliko balona (najprej ugotovite, kateri je ta pogled!). Poslušalca prijavite na označenem mestu v **[inicializacija]**.
2. Izdelajte upravljavnik dogodka. Upravljavnik naj bo izdelan kar v razredu aktivnosti (pomagajte si s Sliko 20).
3. Sprogramirajte reakcijo na dogodek:
 - a. Povečajte števec zadetkov.
 - b. Sprožite zvočni efekt **[zvocniEfekt]**.

e) Meniji in dialogi

1. Ustvarite menije. V dokumentaciji poiščite ime metode, v kateri ustvarimo »options« meni, in vanjo vključite kodo **[ustvariMeni]**.
2. V dokumentaciji poiščite ime metode, v kateri določimo, kaj se zgodi, ko je izbrana določena menijska postavka, in vanjo vključite kodo (**[izberiMeni]**, **[koncajIgro]**).
3. Dopolnite menije tako, da bodo podobni tistim na Sliki 25.
4. Ko se čas izteče **[odstevanje]**, končajte igro **[koncajIgro]** in obvestite uporabnika **[dialog]**.

f) Testiranje igre v emulatorju in na mobilnem telefonu

Doma igro nadgradite, nastali izdelek pa predstavite pred pisnim izpitom.



Slika 25: Meniji in dialog v igri.

Opis pripravljene kode

Spodnja tabela prikazuje seznam pripravljenih delčkov kode in njihovo funkcionalnost. **V programu pripravite metode, v katere boste kodo vključili in jo na primernem mestu v programu klicali.** Nekatere delčke kode boste morali dopolniti sami (v kodi so ta mesta označena), medtem ko so nekateri izdelani že v celoti.

[aktivnost]	Vsebuje deklaracijo spremenljivk in konstant, ki potrebujete v aplikaciji.
[inicializacija]	Prijava poslušalcev, inicializacija predvajalnika, shranjevanje referenc na gradnike uporabniškega vmesnika itd.
[zacniIgro]	Klici metod, ki sprožijo začetek igre.
[premikanje]	Vsebuje kodo, ki določa premikanje balončka po zaslonu.
[odstevanje]	Vsebuje kodo, ki odšteva čas igre.
[zvocniEfekt]	Sproži zvočni efekt.
[ustvariMeni]	Gradnja menija.
[izberiMeni]	Določa dejanja, ki se izvedejo, ko je izbrana posamezna postavka v meniju.
[koncajIgro]	Opravila, ki jih moramo narediti, ko se igra konča.
[dialog]	Koda prikaže dialog s sporočilom o koncu igre.

9. Viri in literatura

Nadaljnje informacije za delo s platformo Android dobite v sledeči literaturi:

Uradna spletna stran za razvijalce aplikacij za Android

<http://developer.android.com/guide/index.html>

Knjiga za začetnike in za naprednejše razvijalce:

R. Meier: *Professional Android Application Development*, Wiley Publishing, 2009

Priloga: namestitev Androidovega razvojnega okolja

Namestitev razvojnega kompleta Android SDK

1.) S spletnega mesta <http://developer.android.com/sdk> prenesite na svoj računalnik Androidov SDK (»ADT bundle for Windows«, 32-bit).

2.) Vsebino *zip* datoteke razširite v mapo `c:/android`.

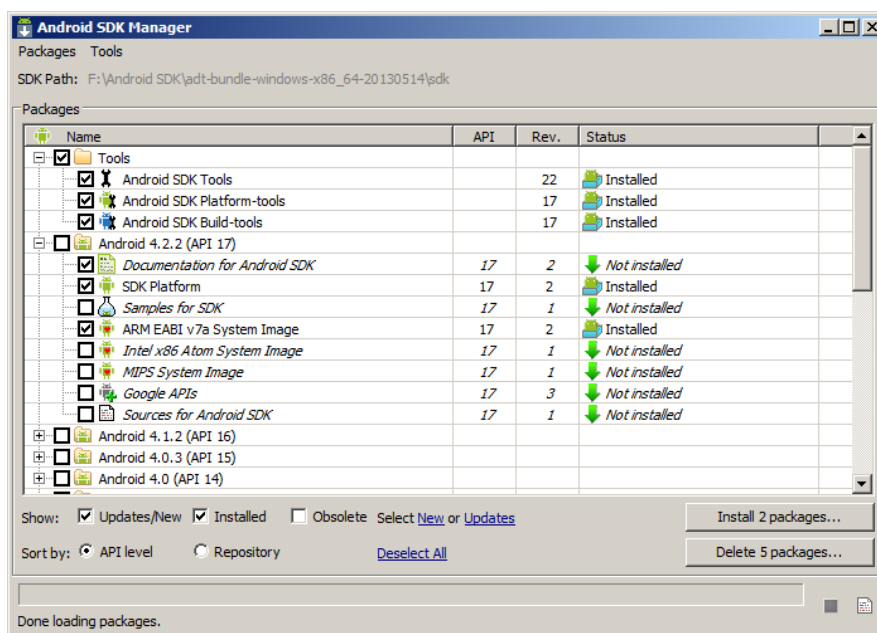
POZOR: Androidov SDK potrebuje javanski razvojni komplet JDK (samo izvajalno okolje JRE ni dovolj!). Prepričajte se, ali ga imate nameščenega, v nasprotnem primeru ga namestite. Najdete ga na: (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

3.) Razglejte se po vsebini mape. Ta vsebuje

- Okolje Eclipse, pripravljeno za delo na platformi Android (ADT)
- Razvojni komplet SDK
- Upravljalnik razvojnega kompleta SDK (SDK Manager)

4.) Poženite Eclipse ADT

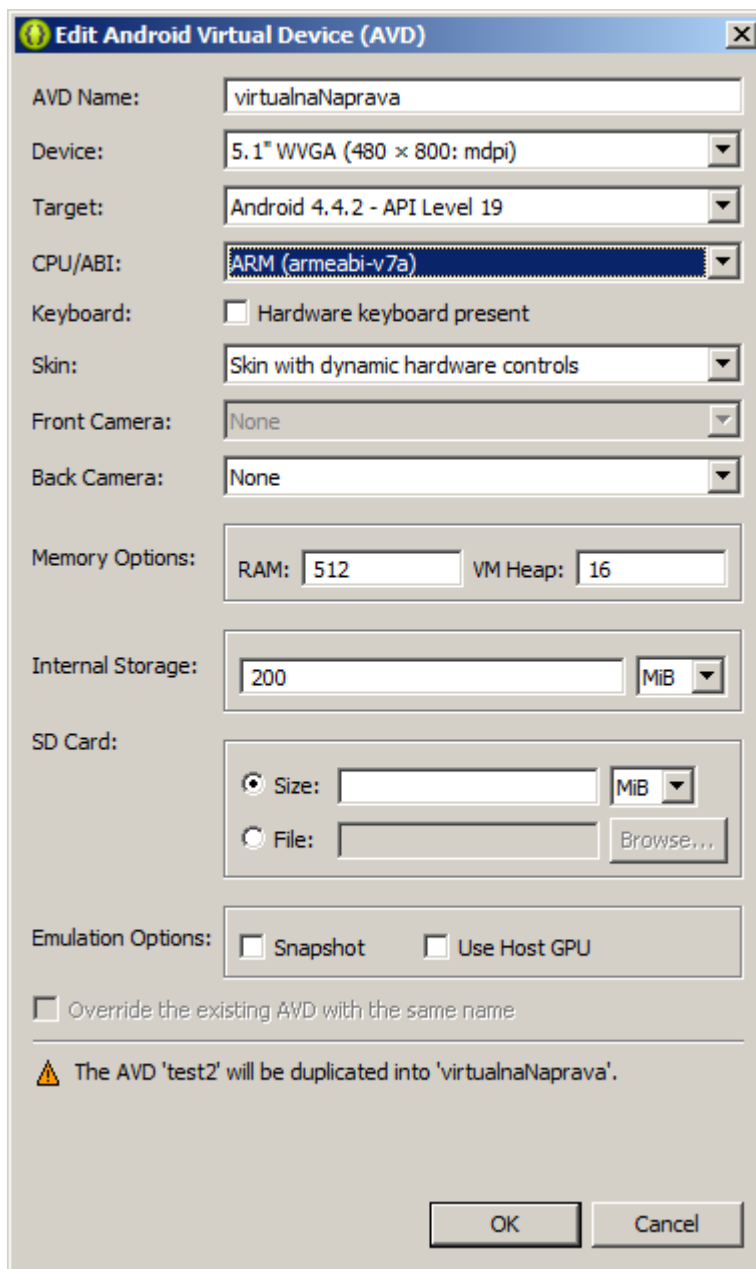
5.) Izberite Window → Android SDK Manager in izberite namestitev komponent, kot to prikazuje slika.



Slika 26

Namestitev emulatorja

- 1.) Izberite **Window > Android Virtual Device Manager**
- 2.) Izberite **New**, izpolnite polja, kot to prikazuje Slika 27, in nato izberite **OK**.
- 3.) Namestitev je zaključena. Sedaj lahko ustvarite prvi Androidov projekt.



Slika 27