Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matevž Bartol

Aplikacija uganke Sudoku za Android

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: doc. dr. Mira Trebar

Ljubljana, 2012

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

[1] V dogovorju z mentorjem lahko kandidat diplomsko delo s pripadajočo izvorno kodo izda tudi pod katero izmed alternativnih licenc, ki ponuja določen del pravic vsem: npr. Creative Commons, GNU GPL. V tem primeru na to mesto vstavite opis licence, na primer tekst [**?**]

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00183/2011
Datum: 02.12.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MATEVŽ BARTOL**

Naslov: **APLIKACIJA UGANKE SUDOKU ZA ANDROID**
SUDOKU GAME APPLICATION FOR ANDROID

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Z razvojem pametnih mobilnih naprav se je vse bolj razširila uporaba številnih zabavnih aplikacij, kot so različne igre in uganke. Operacijski sistem Android predstavlja poseben izziv pri razvoju in programiranju enostavnih, kot tudi bolj zapletenih problemov. Kandidat naj v svoji diplomski nalogi predstavi in analizira uganko Sudoku ter potrebna programska orodja za razvoj aplikacije, ki bo delovala na operacijskem sistemu Android. Reševanje uganke naj bo implementirano za različne težavnostne stopnje, ki vključujejo tudi poznane postopke namigov in pomoči pri izpolnjevanju uganke.

Mentor:

doc. dr. Mira Trebar

Dekan:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Matevž Bartol, z vpisno številko **63040007**, sem avtor diplomskega dela z naslovom:

Aplikacija uganke Sudoku za Android

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 18. aprila 2011 Podpis avtorja:

Zahvala

Rad bi se zahvalil staršem in sestri, ki so me spodbujali ter podpirali med študijem. Posebna zahvala tudi mentorici doc. dr Miri Trebar, ki mi je pomagala ter me usmerjala.

Kazalo

Povzetek

V diplomski nalogi je opisan razvoj aplikacije za uganko Sudoku na operacijskem sistemu Android. Najprej so opisane osnove igre Sudoku ter njena zgodovina. Sledijo opisi tehnologij, ki so nepogrešljive pri izdelavi androidovih aplikacij. Pred izdelavo aplikacije se ustavimo in opišemo osnove operacijskega sistema Android. Razvoj aplikacije se začne s kreiranjem grafičnega uporabniškega vmesnika večinoma v jeziku XML, za ostale dele pa je uporabljen programski jezik Java. Po končanem grafičnem uporabniškem vmesniku se implementira logika igre. Ta je narejena v programskem jeziku Java. Da bi bila aplikacija boljša kot ostale, je potrebno, da vključimo različne pripomočke za premagovanje težavnosti in izboljšavo funkcionalnosti. V aplikaciji je implementiran algoritem za reševanje Sudoku ugank s pomočjo eliminacije stolpca, vrstice in kvadrata. V diplomski nalogi je realizirana aplikacija, namenjena povprečnim uporabnikom, ki si želijo igrati igro Sudoku na svojih pametnih telefonih ter tabličnih računalnikih.

Ključne besede: Android, aplikacija, Sudoku, uporabniški vmesnik, Java, XML

Abstract

This thesis will discuss the development of an Android Sudoku application, for use on Smartphones or Tablet Computers. The application was designed for an average user who wishes to play games. The thesis will describe the basics of the Sudoku game and its history. It will go on to describe technologies, development tools and basics of the Android operating system. The development of the application began with the creation of an appropriate graphical user interface using mostly XML and some Java. The game's logic is implemented in Java programming language. It was important to ensure that the Sudoku application would be superior to others available on the market. Therefore, it was developed to include accessories to help users solve the puzzle. To achieve this, the game includes a column, row and mini-grid elimination algorithm.

Key words: Android, application, Sudoku, user interface, Java, XML

Poglavje 1

Uvod

Človeštvo je v zgodovini preživelo kar nekaj zlatih mrzlic, začenši s kalifornijsko sredi devetnajstega stoletja. Zadnja v svetu računalništva je programiranje za mobilne naprave, ki jo je pognal Applov revolucionarni iPhone.

Pred leti sem v prostorih Fakultete za računalnistvo in informatiko našel list papirja. Na njem je bila kvadratna mreža velikosti 9 krat 9 z napisom Sudoku. Približno ena tretjina kvadratkov je že imela vpisano številko. Ta številska uganka me je izredno pritegnila. Vsakodnevno sem jo reševal med odmori s svinčnikom na papirju. Kasneje sem odkril, da obstajajo tudi Sudoku zborniki, ki obsegajo od 100 do 200 različnih ugank. Vse kar potrebujemo, je svinčnik in obilico prostega časa.

Tako kot drugi mediji, se je tudi Sudoku preselil na splet, v elektronsko obliko. Veliko spletnih strani ponuja Sudoku uganke. Obstajajo plačljive ter brezplačne, vendar za igranje velike večine izmed njih potrebujemo konstantno internetno povezavo. Zato sem se odločil, da bom naredil svojo Sudoku aplikacijo, ki potrebuje le enkratno povezavo na Google Play - Android Market. Tja se bo potrebno povezati ter si naložiti aplikacijo na napravo in ta bo pripravljena na uporabo. Torej nič več ne potrebe po papirju in svinčniku. Zamenjala jih bo mobilna naprava, ki si jo lasti že skoraj vsak prebivalec v

razvitih državah.

Kot se spodobi za sodobno aplikacijo, je potrebno narediti ustrezen uporabniški vmesnik, implementirati dober algoritem za nadzor pravilnosti delovanja ter izbrati operacijski sistem in naprave, ki bodo podpirale aplikacijo.

Poglavje 2

Sudoku

V tem poglavju bomo opisali osnove ter zgodovino igre Sudoku. Omenili bomo tudi dogodke, ki jih je povzročila popularnost Sudokuja.

2.1 Opis sudokuja

Sudoku je logična uganka, pri kateri je potrebno zapolniti kvadratno mrežo, ki je običajno velikosti 9×9, s števili od 1 do 9. Vsako število se lahko pojavi samo enkrat v vsakem stolpcu, vsaki vrstici in vsakem manjšem kvadratu velikosti 3 krat 3. V mreži so nekatera števila že podana. Namesto števil lahko nastopajo tudi črke ali znaki.

Ker je sudoku priljubljena igra, so si ljudje zamislili kar nekaj vrst:

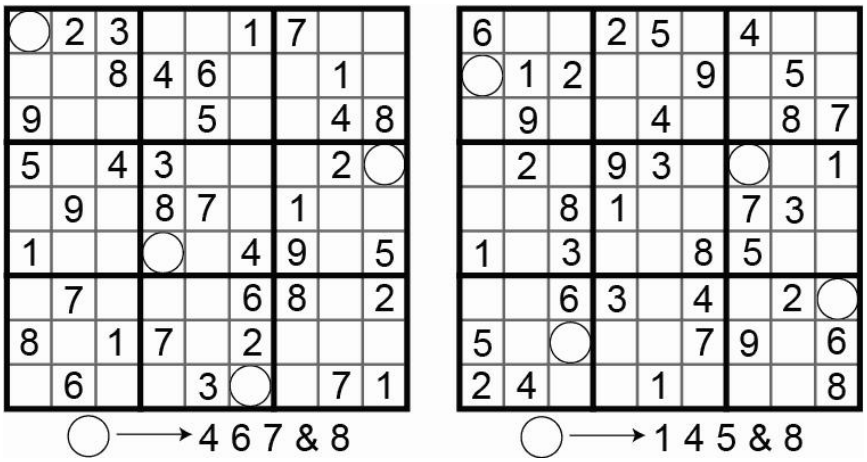
- sudoku-X,
- sodo-lihi sudoku,
- geometrijski sudoku,
- sudoku z drugačnimi dimenzijami.

Za rešitev uganke je potreben logičen razmislek in malo potrpežljivosti, pri težjih pa tudi nekoliko kombinatorike. Uganke so lahko sestavljene ročno

ali pa s pomočjo računalniškega programa. Zahtevnost uganke je načeloma obratno sorazmerna s številom že vpisanih števil, lažje uganke jih imajo preko 30, težje pa med 20 in 30. Število vseh možnih mrež pri velikosti 9×9 je 6.670.903.752.021.072.936.960 ali krajše 6.67×10^{21}. Kolikšno je najmanjše število podanih števil, da je rešitev še vedno enolična, ni znano, trenutni minumum je 17 [4].

2.2 Zgodovina sudokuja

Sudoku je sestavil Howard Garns, upokojeni arhitekt in sestavljavec križank. Prvič ga je objavil leta 1979. Čeprav ga je predvidoma navdihnil latinski kvadrat Leonharda Eulerja, je Garns dodal tretjo dimenzijo tej matematični konstrukciji in jo predstavil kot križanko. Poleg tega je Sudoku delno izpolnjen in zahteva, da vstaviš še manjkajoče številke tako, da so v vsaki vrstici, stolpcu in kvadratu 3×3 številke od 1 do 9, ne da bi se ponovile. Sudoku je bil prvič objavljen pri založbi Dell Magazines, v njihovi reviji Dell Pencil Puzzles and Word Games, pod naslovom »Number Place« (sl. Številčni prostor) v New Yorku. Prva primera igre Sudoku Howarda Garnsa sta prikazana na sliki 2.1 [9].



Slika 2.1: Sudoku Howarda Garnsa iz leta 1979.

Leta 1984 je Japonsko podjetje »Nikoli« predstavilo to številsko uganko svojim bralcem. Naslov križanke si je izmislil Kaji Maki, predstavnik Nikolija. Poimenoval jo je »Suuji wa dokushin ni kagiru«. Prevedeno to pomeni »številke morajo biti edine« oz. »številke se morajo pojaviti le enkrat«. Ker pa je bilo to ime predolgo, so ga poimenovali »Sudoku«. Sestavljeno je iz dveh besed. »SU« pomeni številka in »DOKU« pomeni edina oz. samska [15, 16].

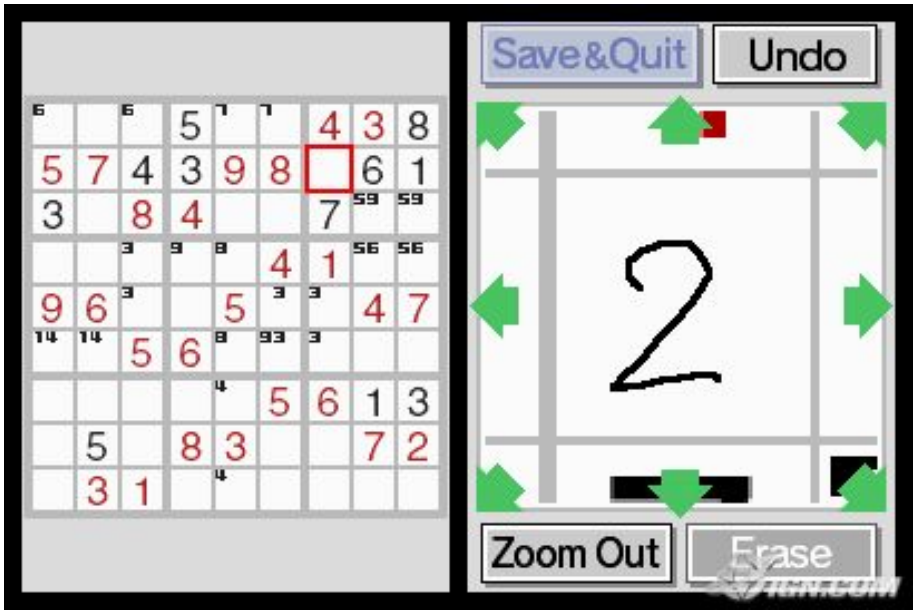
V Sloveniji že od leta 1999 to uganko pod imenom »Devet« redno objavljajo v slovenski ugankarski reviji Modro razvedrilo. Istega leta se je uganka pod imenom »Magični kvadrat« pojavila v reviji Logika in razvedrilna matematika. V juniju 2005 pa se je pojavila tudi v Slovenskih novicah. V avgustu 2006 se je pojavila tudi v Nedeljskem dnevniku [7].

Leta 2005 je Sudoku postal mednarodno popularen. V Veliki Britaniji je pristal na prvih straneh časopisov. Temu je sledila tudi televizijska oddaja namenjena Sudokuju. Imenovala se je »Sudoku Live«. Prvič je bila predvajana 1. julija 2005 na programu Sky One. Za promocijo tega šova so na hribu blizu mesta Bristol naredili največji sudoku na svetu. V višino je meril 80 metrov [17, 18, 20].

Prvo svetovno prvenstvo v Sudokuju je potekalo v Italiji 10. in 11.marca 2006, zmagala pa je 31 letna češka računovodkinja Jana Tylova pred diplomantom s Harvarda Thomasom Snyderjem in programskim inženirjem pri Googlu Wei-Hwa Huangom. Preizkusili so se v klasičnem 9×9 Sudokuju in v 18 različnih variantah le-tega [21].

Sudoku je kmalu postal popularen tudi v elektronski obliki. Ena izmed prvih aplikacij, ki je vsebovala igro Sudoku, je bila »Brain Age: Train Your Brain in Minutes a Day«. Razvilo jo je japonsko podjetje Nintendo. Izdana je bila 16.aprila 2006 za igralno konzolo Nintendo DS . Poleg igralne kon-

zole Nintendo DS potrebujemo tudi pisalo za Nintendo DS (ang. Nintendo DS Stylus Pen), s katerim vpisujemo številke, znake ali črke na ekran. V aplikaciji je vgrajen algoritem prepoznavanja rokopisa, ki loči ali je vpisana navadna številka ali pa je le opomba (ang. note). Loči se jih na podlagi kje v kvadratku vpišemo številko. Opombo je potrebno vpisati v levi zgornji kot. Vpis v preostali del kvadrata pa pomeni navadno številko, kot je prikazano na sliki 2.2 [19].



Slika 2.2: Brain Age: Sudoku na konzoli nintendo DS.

Poglavje 3

Razvojna orodja

Najprej bomo opisali programska jezika v katerih je aplikacija napisana. To sta Java in XML. Sledi razvojno okolje Eclipse s potrebnimi dodatki, na koncu pa še opis operacijskega sistema Android.

3.1 Programski jeziki

3.1.1 Java

Java je programski jezik, ki ga je leta 1995 razvil James Gosling iz podjetja Sun Microsystems. Od aprila 2009 je Sun Microsystems podružnica podjetja Oracle. Spada v skupino visokonivojskih programskih jezikov. Značilnost Jave je objektna usmerjenost programiranja. Večji del sintakse pa izhaja iz programskih jezikov C in C++, vendar vsebuje java preprostejši objektni model in manj nizkonivojskih ukazov. Aplikacije napisane v programskem jeziku Java so običajno sestavljene iz prevedenih razredov v binarni obliki (ang. bytecode), ki lahko delujejo na katerem koli javanskem navideznem stroju (ang. Java Virtual Machine - JVM)[22].

3.1.2 XML

XML je tričrkovna okrajšava za angleški izraz Extensible Markup Language, razširljiv označevalni jezik. XML je preprost računalniški jezik podoben HTML-ju, ki nam omogoča format za opisovanje strukturiranih podatkov ali arhitektura za prenos podatkov in njihovo izmenjavo med več omrežji. XML spreminja mnogo aspektov računalništva, še posebej na področju komuniciranja aplikacij in strežnikov. Da pa se ga tudi razširiti, saj ima namreč to možnost, da si lahko sami izmislimo imena značk (ang. TAG). Zelo je uporaben za komunikacije, saj ima zelo preprosto in pregledno zgradbo [23, 2].

3.2 Eclipse

Za razvoj aplikacije operacijskega sistema Android smo uporabili razvojno okolje Eclipse Classic 3.6.2, ki podpira razvoj aplikacij v številnih programskih jezikih, npr. Java, C, C++, Python itd. Na voljo so verzije za različne operacijske sisteme (Windows, MacOS, Linux), kot tudi različne verzije za 32-bitno in 64-bitno arhitekturo. Medtem ko so verzije Androida poimenovane po raznih sladicah, so verzije Eclipsa poimenovane v povezavi z astronomijo. V Googlu so za razvoj aplikacij za platformo Android pripravili skupek knjižnic in orodij (ang. Software development kit - SDK), ki nam zelo olajšajo razvoj. Vse skupaj so povezali z zelo razširjenim razvojnim okoljem Eclipse, da je delo za razvijalca bolj udobno. Po uspešni namestitvi je treba v okolje Eclipse namestiti še vtičnike iz paketa Android SDK. Prvi vtičnik, ki ga moramo namestiti, je ADT (ang. Android Development Toolkit), ki je vmesnik za nameščanje drugih androidnih vtičnikov. Omogoča nam, da [12, 13]:

- Ustvarimo nove projekte za Android aplikacije.
- Dostopamo do emulatorjev in naprav.
- Prevajamo in razhroščujemo aplikacije.

- Izvažamo aplikacije v tako imenovane Android pakete (ang. android packages - apk), ki jih lahko nato zaženemo na mobilni napravi. Tako namestimo aplikacijo in jo lahko začnemo uporabljati oz. testirati.
- Ustvarimo lahko digitalne certifkate za podpisovanje naših aplikacij.
- Gradimo uporabniške vmesnike na dva načina. Lahko pišemo XML datoteko ali z miško postavljamo posamezne komponente na prikazan zaslon.

Tako je okolje za razvoj androidnih aplikacij pripravljeno. Sedaj je potrebno odločiti na kakšen način bomo preizkušali našo aplikacijo Sudoku. Možnosti so:

1. *Na androidni napravi.* Tu je potreben še en korak: namestitev gonilnika, ki omogoča razhroščevanje in enostavno poskusno poganjanje aplikacij na napravi.
2. *Na emulatorju androidnega okolja (ang. Android Emulator).* V razvojnem okolju Eclipse je potrebno pripraviti tudi virtualno napravo. Nato izberemo ime naprave, različico operacijskega sistema, ki jo želimo imeti, velikost zaslona, količino pomnilnika in druge možnosti.

3.3 Operacijski sistem Android

Android je odprtokoden programski jezik in operacijski sistem za pametne telefone ter ostale prenosne naprave. Zgrajen je na Linux-ovem jedru. Za razvoj Androida je najzaslužnejši Google, ki je ravno v ta namen ustanovil poslovno združenje več podjetij, imenovano Open Handset Alliance (OHA). Poslovno združenje so ustanovili in predstavili javnosti 5. novembra 2007. Namen je prizadevanje skupnega razvoja odprtih standardov na področju telefonije ter ostalih prenosnih naprav, saj poslovno združenje teži k razvoju inovacij na področju mobilne telefonije in prenosnih naprav.

Ker je Android odprtokoden, omogoča cenejše in lažje razvijanje programov. Je enostaven, odziven in omogoča večopravilnost ter se samodejno sinhronizira z Google-ovimi storitvami. Razvojni model, ki ga ponuja Android, zaradi svoje preprostosti privlači veliko proizvajalcev. Primer takih podjetij sta Motorola in Sony Ericsson. Slednji podjetji in mnogo drugih so rešitev videli ravno v Androidu. Le-tega bo podjetje uporabilo za operacijski sistem, strojno opremo pa bodo še naprej razvijali sami. Občutno prednost tu občutijo tudi uporabniki, saj so programi za ta operacijski sistem večinoma brezplačni.

Prednost pri izbiri Androida je v enotnem pristopu pri razvoju aplikacij. V tem proizvajalci vidijo možnost, da se zoperstavijo pohodu iPhone-a ter iPad-a, ki sta zavzela velik delež trga.

3.3.1 Zgradba operacijskega sistema

Operacijski sistem Android je sestavljen iz petih elementov: aplikacije, njihovo ogrodje, knjižnjice, prevajalnik in jedro Linux.

Aplikacije

Napisane so v programskem jeziku Java. Poleg njih se za izdelavo programov uporablja tudi xml. Aplikacije so shranjene v Android paket s končnico apk. Vsaka aplikacija se požene v svojem procesu. Operacijski sistem požene proces takrat, ko mu je poslana zahteva za izvajanje aplikacije. Proces se zaustavi, ko z izvajanjem aplikacije zaključimo. To omogoča rabo pomnilnika tudi drugim aplikacijam. Vsak posamezni proces se prevede posebej, kar omogoči izoliranje aplikacij, da med seboj delujejo neodvisno. Vsaki aplikaciji se ob zagonu ustvari tudi lastna identifikacijska koda, ki se je nato dodajajo pravice za uporabo strojne opreme. Med jedrne (ang. core) aplikacije spadajo:

- email klient,
- program za kratka sporočila (ang. Short Message Service - SMS),

- koledar,
- kontakti,
- zemljevid,
- spletni brskalnik in tako dalje.

Aplikacijsko ogrodje

V aplikacijskem ogrodju se nahajajo vse sistemske aplikacije, ki se uporabljajo za koordiniranje aplikacij. Te aplikacije so:

- upravljalec aktivnosti,
- upravljalec pomnilnika,
- upravljalec lokacij in
- upravljalec obvestil.

Knjižnice

So temeljni del sistema Android. Navaden uporabnik operacijskega sistema do njih ne more dostopati. Uporabljajo jih razvijalci za dostop do strojnih komponent naprave. Za našo aplikacijo je pomembna grafična knjižnica za podporo 2D grafike, predvsem paketa *android.graphics.drawable* in *android.view.animation*. Razlog je v tem, da Android ne podpira paketov AWT ter Swing [24].

Prevajalnik

Operacijski sistem Android uporablja za prevajanje kod aplikacij prevajalnik JIT (ang. Just in time compiler). To omogoča prenos aplikacij na več različnih prenosnih naprav brez ponovnega pisanja izvorne kode.

3.3.2 Android market - Google Play

Je licenčna aplikacija podjetja Google, ki se uporablja za prenos oz. nalaganje aplikacij. To se izvršuje na dva načina. Lahko jih nalagamo (ang.

download) neposredno preko aplikacije Android Market - Google Play ali jih naložimo preko kode QR (ang. Quick Response Code), katero preberemo s prenosno napravo. Po branju nato Android market sam poišče aplikacijo na njihovem strežniku, jo prenese ter namesti na prenosno napravo. Aplikacija je bila predstavljena na Googleovi konferenci oktobra 2008. Je zelo priročna aplikacija za distribucijo aplikacij[11].

3.3.3 Verzije operacijskega sistema Android

Zgodba o Androidu se začne z beta verzijo novembra 2007. V tem času je bilo izdanih veliko popravkov in veliko novih funkcionalnosti. Od aprila 2009 se vsaka nova verzija Androida poimenuje po neki sladici. Po angleškem abecednem redu so bile izdane verzije:

- Astro 1.0
- Bender 1.1
- Cupcake 1.5
- Donut 1.6
- Eclair 2.0-2.1
- Froyo 2.2
- Gingerbread 2.3
- Honeycomb 3.0
- Icecream Sandwich 4.0

Prvi dve verziji, ki jih zaradi licenc in lastniških pravic niso uporabili sta Astro in Bender. Verzija Honeycomb je bila narejena posebej za tablične računalnike. Trenutno najnovejša verzija Icecream Sandwich je narejena z namenom poenotenja operacijskega sistema pametnih telefonov ter tabličnih računalnikov.

Poglavje 4

Razvoj aplikacije

V tem poglavju najprej opišemo koncept uporabniškega grafičnega vmesnika. Za njegovo izdelavo se nam ponujata dve možnosti, ki ju opišemo, primerjamo ter ovrednotimo. To sta proceduralni ter deklarativni način. Sledijo opisi posameznih delov grafičnega uporabniškega vmesnika ter funkcionalnosti aplikacije Sudoku.

4.1 Grafični uporabniški vmesnik

Koncept grafičnega uporabniškega vmesnika (ang. graphical user interface - GUI), kakršen je v uporabi danes, je nastal že v 70. letih dvajsetega stoletja. Razvili so ga pri podjetju Xerox, v raziskovalnem centru v Palo Altu. Od takrat je bilo sprememb relativno malo. Izboljšal se je le izgled in dodala kakšna nova vrsta gradnika. Dosedanje poskuse večjih sprememb, kakršen je bil na primer vmesnik Microsoft Bob - 1995 in Microsoft Surface - 2008, uporabniki povečini niso podprli. Uporabniški vmesnik služi kot vezni člen med uporabnikom in računalniškim sistemom in omogoča njegovo uporabo. Zahteva, ki ji mora zadostiti kakovosten vmesnik, je predvsem nezahtevnost za uporabnika. To pomeni omogočanje čim bolj učinkovito, enostavno in prijetno uporabu računalniškega orodja. Vse našteto zajemata pojma uporabnost in uporabniška prijaznost, s katerima navadno opisujemo vmesnike [1, 8].

4.1.1 Izdelava uporabniškega vmesnika

Uporabniški vmesnik lahko ustvarimo na dva načina:

(a) Proceduralni način: Tu je z nizom navodil podan postopek, kako pridemo do rešitve. V našem primeru je to uporaba programskega jezika Java, v kateri elementi GUI pripadajo različnim instancam razredov.

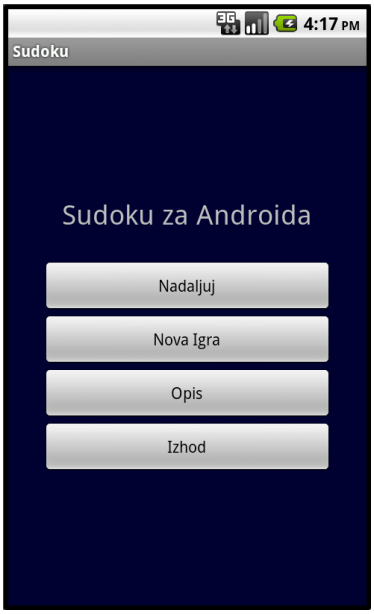
(b) Deklarativni način: Opisati je treba, kaj želimo dobiti kot rezultat, ne pa kako pridemo do njega. Torej uporaba XML.

Katerega torej uporabiti? Android je optimiziran za mobilne naprave z omejeno strojno opremo. Google priporoča, da uporabljamo deklarativni XML, kjerkoli je to mogoče. XML koda je v večini primerov krajša in tudi lažje razumljiva kot ekvivalentna javanska koda. To seveda ne pomeni, da je XML boljši v zmogljivostih, saj ni ravno znan po učinkovitosti ter prijaznosti do strojne opreme. Zakaj ga potem Google priporoča? Odgovor je preprost, človeski faktor. Ker pa človeški faktor ni edino merilo, je potrebno najti tudi ostale razloge. Težavo z zmoglivostjo je odpravil Eclipsov prevajalnik virov AAPT (ang. Android Asset Packaging Tool). Njegova prednost je v tem, da ko programiramo, vidimo podatke v XML, vendar pa AAPT predprocesira XML v zgoščen binarni format. V tem formatu se shrani na napravo [6, 14].

Kompromis med deklarativnim ter proceduralnim! Android podpira kompromis med proceduralnim ter deklarativnim načinom, saj nam omogoča, da uporabniški vmesnik naredimo na oba načina. Vprašanje je samo, kateri način je bolj optimalen. Po temeljitem razmisleku smo večino grafičnega uporabniškega vmesnika naredili s pomočjo XML, programski jezik Java pa smo uporabili pri izrisovanju Sudoku mreže.

4.2 Glavni meni

Glavni meni je izhodišče programa oz. aplikacije, kjer je uporabniku na voljo seznam najbolj pomembnih možnosti. Te možnosti so »Nadaljuj«, »Nova Igra«, »Opis«, »Izhod« ter »Nastavitve«. Slednjo možnost se prikliče s pritiskom na gumb MENU, ki ga ima vsaka naprava z operacijskim sistemom Android. Izgled glavnega menija (ang. main menu) je definiran v datoteki /res/layout/main.xml, prikazan pa je na sliki 4.1. Eden izmed njegovih gradnikov, gumb »Nadaljuj«, je predstavljen v kodi 4.1.



Slika 4.1: Glavni meni.

Koda 4.1: Del main.xml datoteke.

```
<Button
        android:id="@+id/nadaljuj_gumb"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/nadaljuj_label" />
```

Da ima gumb »Nadaljuj« napis, v strings.xml dodamo vrednost labele, kot je prikazano na kodi 4.2. Ker pa je v naši igri veliko besedil oz. nizov, je smotrno, če jih definiramo v eni datoteki, to je /res/values/strings.xml.

Koda 4.2: Nizi - del strings.xml datoteke.

```
<resources>
        <string name="nadaljuj_label">Nadaljuj</string>
        ....
        <string name="opis_title">O igri Sudoku.</string>
        <string name="opis_text">\ Sudoku je sestavljanka, ki
            temelji na logiki ter kombinatoriki. Namen igre je
            zapolniti polje 9x9.</string>
</resources>
```

Barve v operacijskem sistemu Android so predstavljene s štirimi 8 bitnimi števili. Števila predstavljajo prosojnost, rdečo, zeleno ter modro barvo (ang. alpha, red, green, blue - ARGB). Alpha je merilo za prosojnost. Najnižja vrednost je 0, kar pomeni popolno prosojnost. Najvišja vrednost FF_{16} oz. 255_{10} pa pomeni, da je barva motna oz. neprosojna. Barve lahko predstavimo tudi s tremi 8-bitnimi števili: rdeča, zelena, modra (ang. red, green, blue - RGB). Ta način pa ima veliko pomankljivost, obarvanje kvadratkov na mreži povzroči slabšo vidljivost vpisanih števil. ARGB vrednosti barve ozadja glavnega menija je definirana v kodi 4.3. Ostale tri barve pa bodo omenjene kasneje[25].

Koda 4.3: Vrednosti barvnih parametrov - del colors.xml datoteke.

```
<resources>
        <color name="background">#350000ff</color>
        <color name="celica_navadna">#ff000000</color>
        <color name="celica_pomoc">#6000ff80</color>
        <color name="celica_izbrana">#50fd0000</color>
</resources>
```

S temi osnovnimi gradniki smo izdelali izgled glavnega menija. Sedaj so gumbi vidni, lahko jih tudi kliknemo, vendar se nič ne zgodi. V ta namen definiramo metodo onClick(). Za izbiro gumbov smo uporabili switch stavek,

kot je prikazano v kodi 4.4.

Koda 4.4: Switch stavek - del datoteke Sudoku.java .

```
public void onClick(View v){
        switch (v.getId()){
        case R.id.nadaljuj_gumb:
                zacniIgro(Igra.nadaljuj);
                break;
        case R.id.opis_gumb:
                Intent i = new Intent(this, Opis.class);
                startActivity(i);
                break;
        case R.id.novaigra_gumb:
                novaIgraDialog();
                break;
        case R.id.izhod_gumb:
                finish();
                break;
        }
}
```

4.2.1 Nadaljuj igro

Ker uporabnike med reševanjem uganke sudoku lahko prekine kak dogodek, smo se odločili in naredili tudi sprotno shranjevanje trenutnega stanja igre. Za to je potrebna vrstica iz kode 4.5.

Koda 4.5: Shranjevanje trenutnega stanja - del datoteke Igra.java .

```
getPreferences(MODE_PRIVATE).edit().putString(PREF_SUD, vString(
    sud)).commit();
```

4.2.2 Nova igra

Ko pritisnemo na gumb »Nova Igra«, se nam ponudi možnost izbire težavnosti. V naši igri imamo 3 težavnosti: lahka, srednja in težka. Vsaka stopnja težavnosti ima 5 primerov, ki so fiksno določeni. Z izbiro ene od teh začnemo novo igro.

Koda 4.6: Dialog izbire težavnosti - del datoteke Sudoku.java .

```
private void novaIgraDialog(){
        AlertDialog.Builder(this)
        .setItems(R.array.tezavnost,
           new DialogInterface.OnClickListener(){
        void onClick(DialogInterface dialoginterface,int i){
                zacniIgro(i); }}).show();
}
```

Rezultat kode 4.6 je prikazan na sliki 4.2. Igra se zaključi, ko je vseh 81

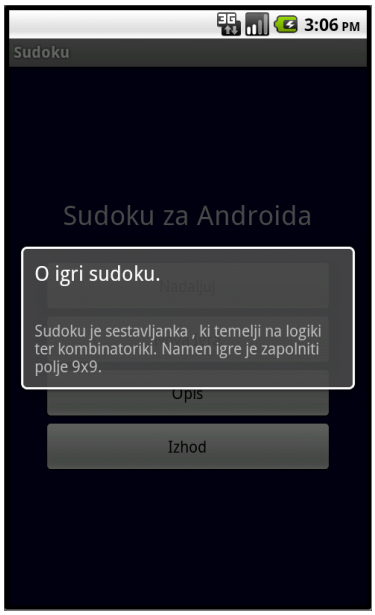
Slika 4.2: Izbira težavnostne stopnje.

polj pravilno izpolnjenih. Če se med igranjem zmotimo, lahko to napako

odpravimo z zamenjavo števila. V primeru, da to ne pomaga, lahko začnemo reševati od začetka. Vendar s tem zgubimo vse prejšne rešitve.

4.2.3 Opis igre v aplikaciji

Ko uporabnik izbere gumb »Opis«, pomeni da se ga je dotaknil na zaslonu oz. preko ostalih vmesnikov. To dejanje sproži novo okno, ki vsebuje osnovne informacije o igri Sudoku.Izgled je prikazan na sliki 4.3, niza pa sta omenjena v kodi 4.2. Ko se želi uprabnik vrniti na prejšnjo stran, preprosto pritisne



Slika 4.3: Opis igre.

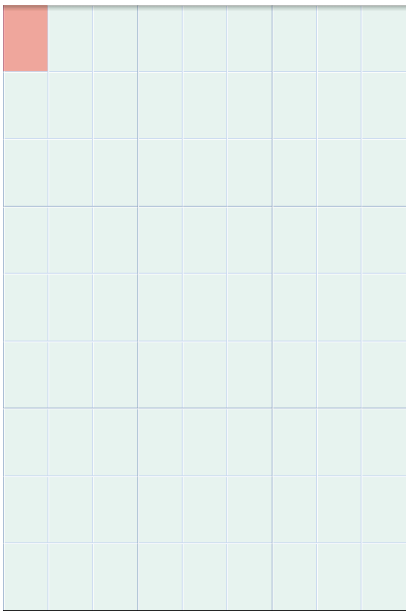
gumb na BACK (sl. nazaj).

4.2.4 Izhod

Gumb »Izhod«, v resnici deluje kot gumb »BACK« ali »HOME« (sl. domov), prekine trenutno aktivnost in naloži se naslednja aktivnost iz sklada (ang. stack). V kodi 4.6 je prikazan del onClick() metode, ki se nanaša na gumb »Izhod«.

4.3 Kvadratna mreža

Igralna plošča je edini del GUI, ki ni izdelan v XML. Kvadratke bi lahko izdelali kot gumbe ali pa jih definirali kot mrežo razreda ImageView. Vendar je v tem primeru javanska koda bolj primerna. Vse se začne s prazno površino kvadratne oblike. To komponento, na katero lahko rišemo in pišemo, imenujemo Canvas. Za izris mreže 9 krat 9 potrebujemo 8 vodoravnih in 8 navpičnih črt. Za vizualno ločevanje mini kvadratov, uporabimo temno različico sive barve. Za črte med kvadrati pa svetlo sivo barvo. Da bi še bolj poudarili kvadratke, smo jim na robovih dodali izredno svetlo barvo. Te robovi v povezavi s sivimi barvami izstopajo ter nam dajejo občutek globine. Kot prikazuje slika 4.4 je izrisan tudi trenutno izbrani kvadratek. Njegova velikost je vedno ena devetina dolžine in širine ekrana. Slabost tega načina



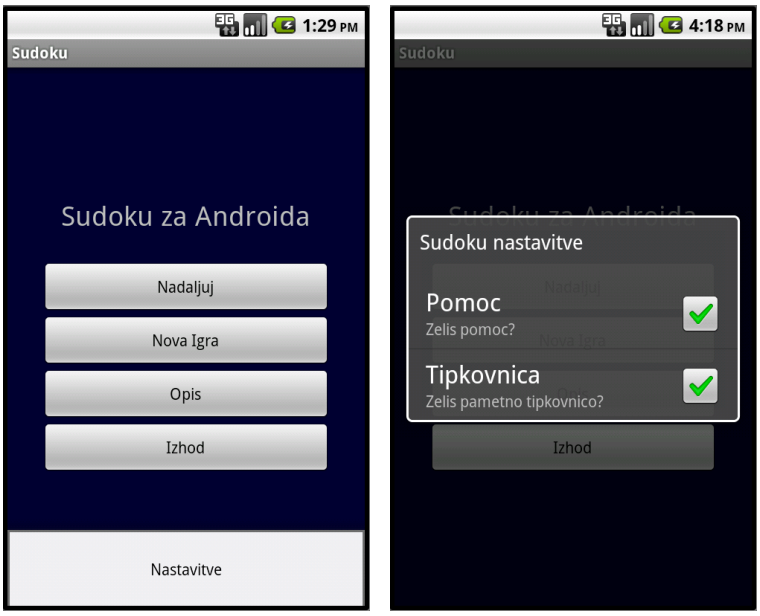
Slika 4.4: Kvadratna mreža 9x9.

izrisovanja je v metodi onDraw(), ki ob vsaki spremembi v igri od začetka izriše vse komponente.

4.4 Funkcionalnosti v igri Sudoku

4.4.1 Meni nastavitev

Ker niso vsi uporabniki vešči v igri Sudoku, smo se odločili, da jim malce pomagamo pri reševanju. V naslednjem podpoglavju bo prikazano, kako to izgleda. Vsak pametni telefon oz. tablični računalnik z operacijskim sistemom Android ima tipko »MENU«(sl. meni). Ta meni je lahko fizična tipka ali pa je programski. Po pritisku na gumb »MENU«, se nam prikaže dodaten gumb, kot je prikazano na sliki 4.5a. Ko kliku na gumb »Nastavitve«, se nam prikažeta možnosti pomoči, ki sta prikazani na sliki 4.5b. Njuni funkcionalnosti bosta opisani v naslednjih podpoglavjih.



(a) Navadna tipkovnica. (b) Pametna tipkovnica.

Slika 4.5: Klic in možnosti nastavitev.

4.4.2 Težavnostne stopnje

Vsaka sudoku igra je vedno označena z neko težavnostno stopnjo, ki je odvisna od vnaprej danih števil ter njihovih pozicij. Odločili smo se, da bo

naša igra imela 3 stopnje. Imenovali smo jih lahko, srednje ter težko. Vse težavnostne stopnje smo vnaprej zakodirali, kajti generiranje težavnostih stopenj je zelo kompleksno. Poleg tega bi potrebovali generator števil za vsako težavnost. V kodi 4.7 prikažemo primer lahke zahtevnosti. Ta ima 33 vnaprej danih števil, ki so enakomerno porazdeljene po mreži [3].

Koda 4.7: Definiranje lahke zahtevnosti - del datoteke Igra.java .

```
private final String lahko1 =
                    "040000060" +
                    "006090200" +
                    "100786004" +
                    "600509008" +
                    "058070930" +
                    "900308007" +
                    "800937005" +
                    "007050300" +
                    "010000040";
```

Sedaj lahko ta niz spremenimo v število, ker ga potrebuje logika igre. Za potrebo sprotnega shranjevanja smo naredili obraten postopek. Obe metodi sta navedeni v kodi 4.8.

Koda 4.8: Transformacija v polje števil in obratno - del datoteke Igra.java .

```
static protected int[] vStevilo(String string){
        int[] zah=new int[string.length()];
        for(int i = 0;i<zah.length;i++){
                zah[i]=string.charAt(i)-'0';
        }
        return zah;
}
static private String vString(int[] zah){
        StringBuilder buffer=new StringBuilder();
        for(int element:zah){
                buffer.append(element);
        }
        return buffer.toString();
}
```

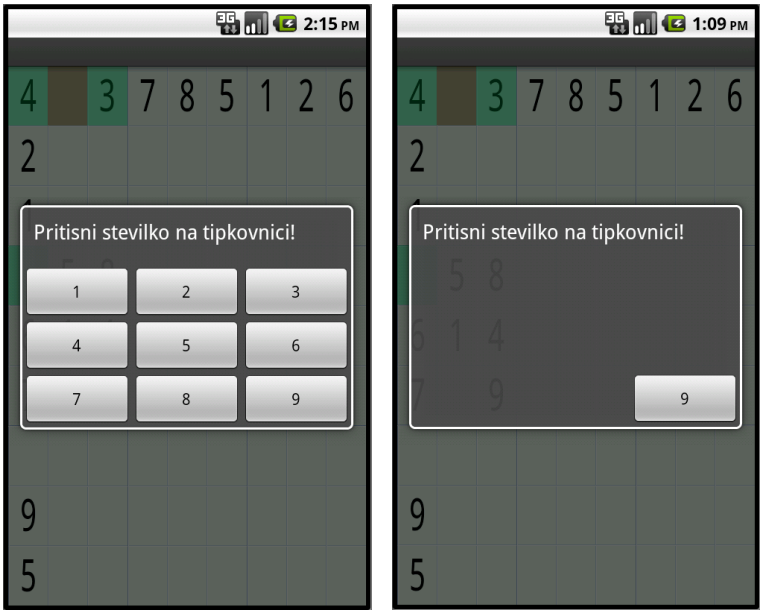
4.4.3 Pomoč na tipkovnici

Za pomoč uporabnikom smo implemetirali pametno tipkovnico. Algoritem pametne tipkovnice je prikazan v kodi 4.9.

Koda 4.9: Pametna tipkovnica - del datoteke Tipkovnica.java .

```
for(int element : uporabljeneVrednosti){
if((element != 0)&&(Pomoc.getTipkovnica(getContext())))
        tipke[element-1].setVisibility(View.INVISIBLE)
}
```

Le-ta nam pomaga, da ni možno narediti napake. To dosežemo tako, da so tipke s številkami, ki so že uporabljene v trenutni vrstici, stolpcu ali mini-mreži, uporabniku nevidne. Razlika med tipkovnicama je vidna na slikah 4.6a ter 4.6b. Izgled tipkovnice je definiran v jeziku XML. Njeno obnašanje pa je



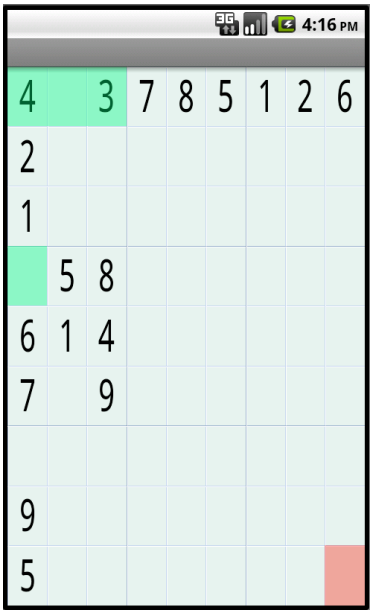
(a) Navadna tipkovnica. (b) Pametna tipkovnica.

Slika 4.6: Razlika med navadno in pametno tipkovnico.

predpisano v javanski kodi, naprimer skrite številke.

4.4.4 Implementacija pomoči z algoritmom eliminacije

V naslednjem podpoglavju je opisano delovanje tega algoritma. Je preprost, zato ga bomo uporabili za pomoč uporabnikom, ki imajo težave pri reševanju. Vsakič, ko v naboru trenutne celice ostane samo še ena vrednost, bomo to



Slika 4.7: Pomoč eliminacije.

celico obarvali z zeleno barvo. Izrisovanje pomoči je opisano v kodi 4.10. Spodnji desno je viden trenutno izbrani kvadratek. Obarvan je s svetlo rdečo barvo. Obe omenjeni barvi sta vidni na sliki 4.7, definirani pa sta v colors.xml, del katere lahko vidimo v kodi 4.3.

Koda 4.10: Izris pomoci - del datoteke sudokuMreza.java .

```
int proste=igra.getNabor(i, j).length;
if(proste<2){
        getRect(i, j, r);
        pomoc.setColor(getResources().getColor(R.color.
            celica_pomoc));
        canvas.drawRect(r,pomoc);
}
```

4.4.5 Eliminacija Stolpca,Vrstice in 3 krat 3 Kvadrata

Večina Sudoku iger se lahko reši s procesom eliminacije. Na primer, če je osem od devetih polj v vrstici že določenih, potem bo zadnje seveda zapolnjeno s tisto, ki še ni bila uporabljena. Ker pa je tak Sudoku preveč preprost, nam ponudniki teh iger dajejo bolj kompleksne kombinacije. Včasih so te kombinacije pretežke. Zato si je pametno pomagati z različnimi algoritmi. Eden izmed teh je algoritem eliminacija stolpca,vrstice in 3 krat 3 Kvadrata. Definiral ga je Wei-Meng Lee v svoji knjigi »Programming Sudoku«. Diagram poteka omenjenega algoritma je prikazan na sliki 4.8 [5].

Gradniki diagrama poteka:

i=1,j=1 Postavimo se v celico (1,1).

j manjši od 10? Ali je j manjsi od 10?

j=1,i++ Skoči na celico skrajno levo ter se pomakni 1 mesto navzdol.

i manjši od 10? Ali je i manjsi od 10?

j++ Pomakni se za eno celico desno.

Prazna? Je v celici že kakšna vrednost?

Nastavi nabor Nastavi nabor na [1,2,3,4,5,6,7,8,9]

Operacij stolpec Preveri stolpec trenutne celice in spremeni nabor.

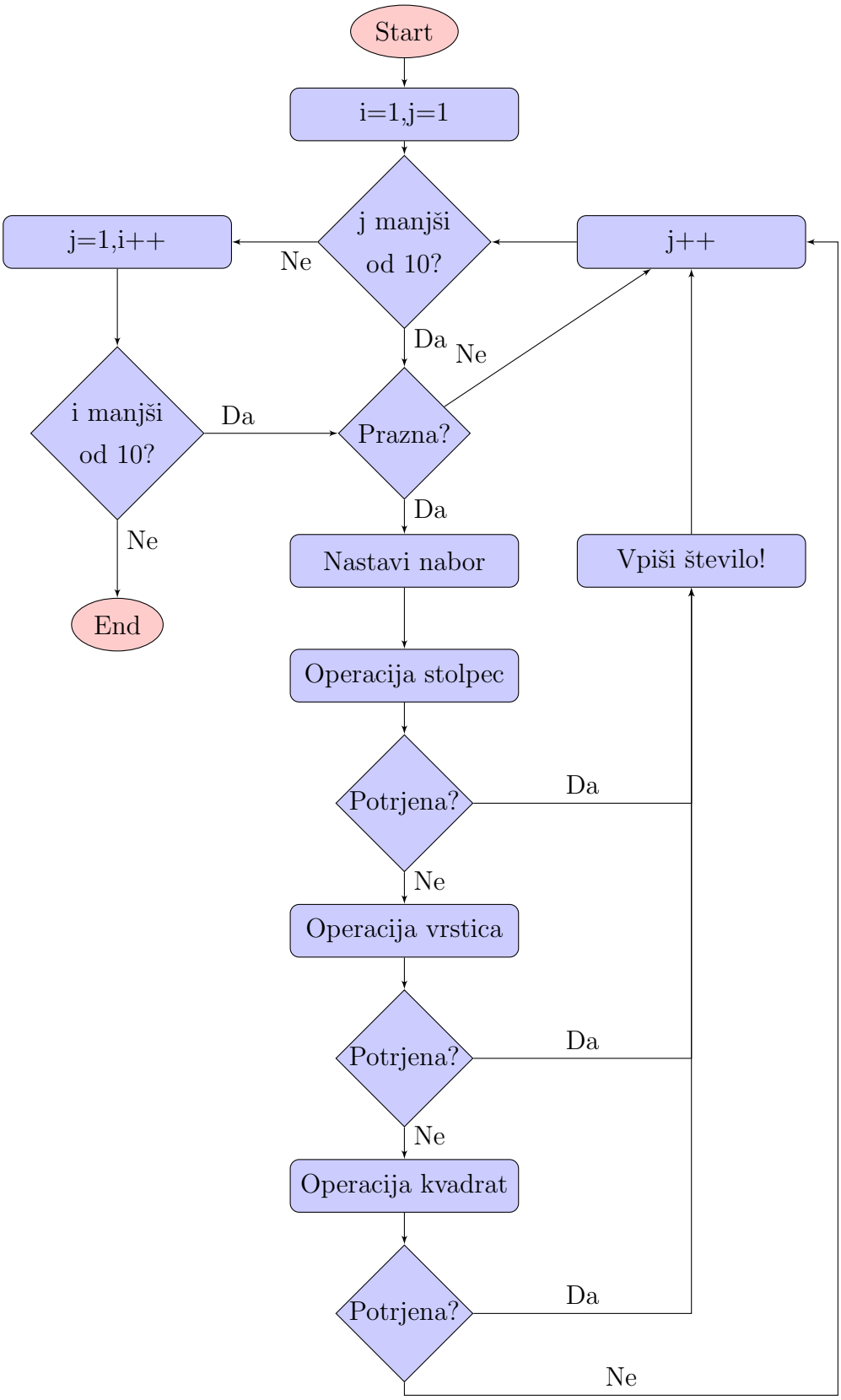
Operacij vrstica Preveri vrstico trenutne celice in spremeni nabor.

Operacij kvadrat Preveri kvadrat trenutne celice in spremeni nabor.

Potrjena? V naboru ostala samo ena vrednost?

Vpiši število! Vpiši število.

Start/End Začni/ Končaj.



Slika 4.8: Diagram poteka algoritma.

Pa se lotimo reševanja nekega primera. Koraki so prikazani na sliki 4.9.

1. Korak (slika 4.9a): Začnemo prebirati vsako celico od leve proti desni, od zgoraj navzdol in prva prazna, na katero naletimo, je na poziciji (1,2).

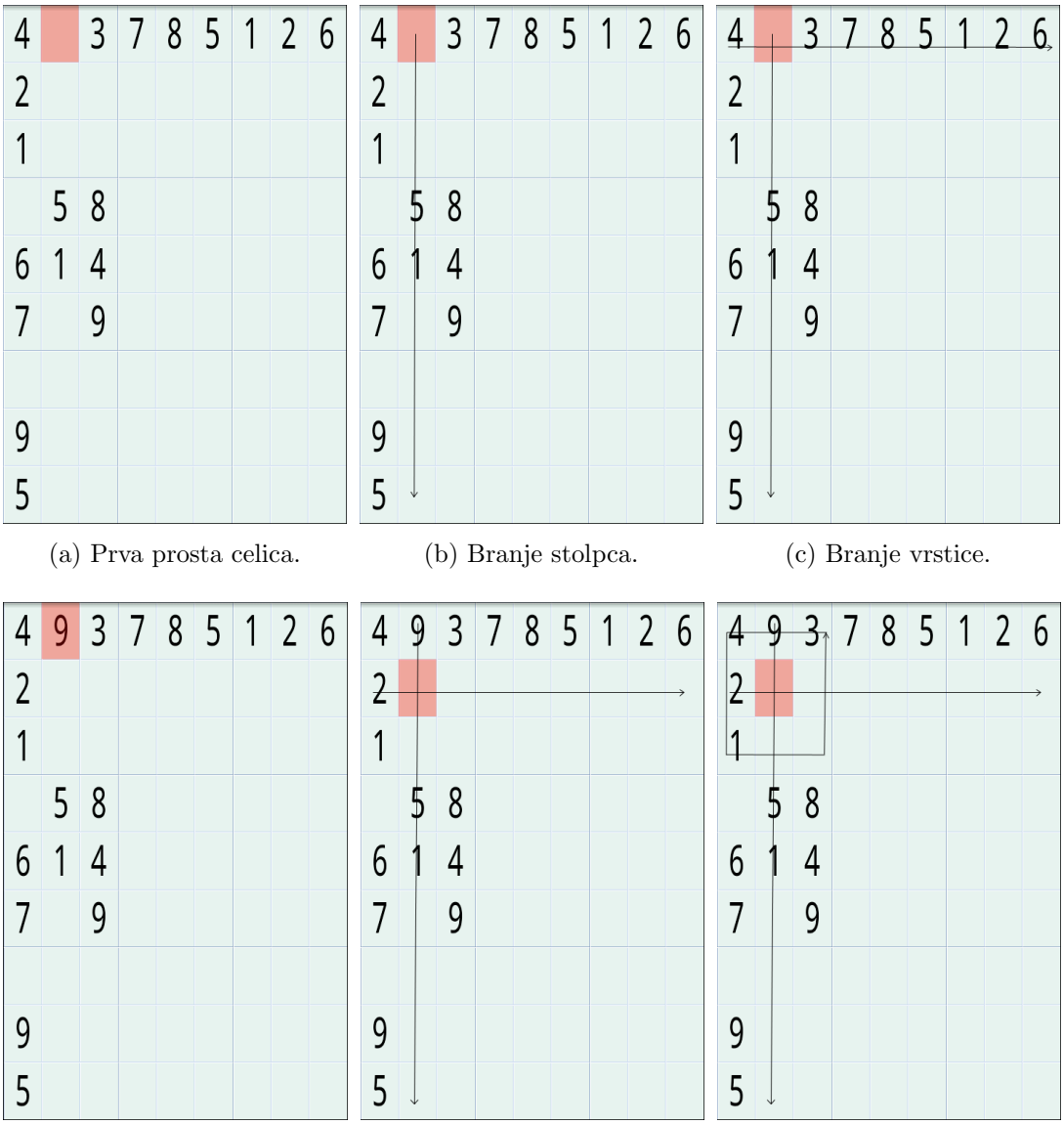
2. Korak (slika 4.9b): Pri branju stolpca trenutne celice so možne vrednosti, ki so ostale 2,3,4,6,7,8 in 9.

3. Korak (slika 4.9c): Več kot očitno, ko preberemo vrstico trenutne celice, je vrednost celice (1,2) 9.

4. Korak (slika 4.9d): Ker je edini ostali nabor stevilka 9, jo sedaj zapišemo v celico (1,2).

5. Korak (slika 4.9e): Nadaljujemo z branjem celic. Naslednja celica je (2,2). Po branju stoplca ter vrstice nam ostanejo vrednosti 3, 4, 6, 7 in 8.

6. Korak (slika 4.9f): Ker je v naboru še vedno več kot ena vrednost, nadaljujemo z branjem kvadrata. V kvadratu so že vrednosti 1, 2, 3, 4 in 9. Zato nabor trenutne celice zmanjšamo na 6, 7 ter 8. Torej ta celica nima potrjene vredosti zato nadajujemo z (2,3) in tako naprej.



(a) Prva prosta celica. (b) Branje stolpca. (c) Branje vrstice.

(d) Vpis števila 9. (e) Druga prosta celica. (f) Branje kvadrata.

Slika 4.9: Koraki algoritma eliminacije.

4.4.6 Zmoglijivost algoritma

Zmogljivost smo izmerili s pomočjo opcije dnevniškega izpisa (ang. LogCat) v razvojnem okolju Eclipse. Za testiranje lahko uporabimo emulator ali pa resnično napravo, npr. pametni telefon ali tablični računalnik. Za testiranje na resnični napravi jo je potrebno povezati z računalnikom preko vmesnika USB (ang. universal serial bus), nato pa jo nastaviti v razhročevalni način preko USB (ang. USB debug mode). Po namestitvi vseh gonilnikov se je začelo testiranje. Začnemo z dnevniškim izpisom pred klicem funkcije, ki računa nabor števil. Sledi izpis po izračunu nabora pri [1][1], konča pa se z izpisom po izračunu nabora [9][9]. Testiranje smo opravili na tabličnem računalniku Archos G9 101 z operacijskim sistemom Android 4.0.4 Icecream Sandwich. Archos G9 101 uporablja dvo jedrni 1 GHz OMAP procesor četrte generacije. OMAP so procesorji znamke Texas Instruments, ki so osnovani na ARM9 arhitekturi. Omogočajo večopravilnost, ki je bila v našem testiranju izrednega pomena, saj ves čas v ozadju tečejo tudi ostali procesi. Če ne bi imeli večopravilnosti, bi bili rezultati lahko drugačni zaradi alokacije virov kakšnemu procesu v ozadju. Rezultat testiranja, ki je izpisano v tabeli 4.1,

Čas	Napis
03-24 16:28:28.173	zacetek
03-24 16:28:28.173	nabor[1][1] = 2357
03-24 16:28:28.283	nabor[9][9] = 269

Tabela 4.1: Dnevniških izpisi.

je 110 milisekund. Ker je 1 test premajhen vzorec, smo nadaljevali s testiranjem. Vsakič, ko je bilo v neko celico vpisano število, je algoritem ponovno izračunal nabore v celicah. Za dokončanje igre je bilo potrebno vpisati še 47 števil, kar pomeni, da je algoritem še 47 krat izračunal stanje. Povprečje po 47 izračunih na tabličnem računalniku je bilo 106,27 milisekund. V celotni igri je algoritem izračunaval nabore 4994 milisekunde.

Poglavje 5

Zaključek

Ker operacijskega sistema Android ne uporabljajo samo mobilni telefoni, je bilo potrebno aplikacijo narediti primerno za rabo na tabličnih računalnikih. Aplikacija nudi poleg pokončne tudi podporo za ležečo orientacijo zaslona.

Naša aplikacija s privzetimi ikonami operacijskega sistema Android zavzema le 32 kilobajtov spominskega prostora. Na Androidovem trgu pa lahko najdemo tudi aplikacije Sudoku, ki zavzamejo več megabajtov prostora. To velikost lahko pripišemo glasbenim datotekam, različnim slikam ter animacijam. Prednost aplikacij, ki zasedajo malo prostora je v ceni prenosa, saj vsi uporabniki pametnih telefonov ter tabličnih računalnikov nimajo brezplačnega interneta. V širši regiji tudi povsod ni zagotovljen širokopasovni internet. 99.6 odstotka prebivalstva Slovenije je pokrito s signalom GSM. Ta uporablja tehnologijo EDGE (ang. Enhanced Data rates for Global Evolution) za prenos podatkov. Najvišja hitrost podatkov je 236,8 kb/s, kar je 29,6 kilobajtov na sekundo. Na hitrost vplivajo: stopnja nadgradnje omrežja, uporabljena naprava, razgibanost reliefa, naravne in umetne ovire, vreme in ozelenitev. Prenašanje nekaj megabajtov podatkov lahko postane dolgotrajen in drag postopek. V primeru, da je prenos podatkov tudi drag, pa se lahko vprašamo o smislenosti »brezplačnih« aplikacij. Za primer smo vzeli ceno prenosa pri mobilnem operaterju Mobitel. Izbrali smo si nezakupljen

podatkovni prenos, ki stane 0,00042 evro/kB oz. 0,42 evro/MB. To pomeni, da bi uporabnika prenos naše aplikacije stal 1,3 centa[26, 27].

Svetla stran večine Sudoku aplikacij je podpora starejšim operacijskim sistemom Android. Temna stran pa je posodabljanje aplikacij za trende, ki jih narekujejo novejši operacijski sistemi Android. Primer tega je Action Bar, ki je podprt od Honeycomb 3.0 oz. aplikacijskega programskega vmesnika 11 naprej (ang. Application programming interface - API). Res pa je, da imata Honeycomb ter IceCream Sandwich trenutno izredno nizek tržni delež. Večina naprav je na verzijah 2.2 ter 2.3 in to dejstvo se ne bo spremenilo še nekaj časa. Naša aplikacija podpira verzijo 1.5 to je API 3 in naprej. To je po našem mnenju trenutno najbolj optimalna rešitev, saj tako zagotovimo, da bo aplikacija delujoča na 99.9 odstoka pametnih telefonih ter tabličnih računalikih z operacijskim sistemom Android [28].

Aplikacija je namenjena za kratkočasenje povprečnih uporabnikov, zato smo temu primerno zagotovili količino ter težavnost Sudoku ugank. Obstajajo aplikacije Sudoku s po nekaj tisoč različnih ugank, vendar je to po našem mnenju preveč za povprečnega uporabnika. Povprečje je 1 sudoku uganka na dan. Zato tudi večina dnevnih časopisov objavi le eno.

Končni rezultat diplomskega dela je torej delujoča Sudoku aplikacija, ki je namenjena najširši možni ciljni skupini uporabnikov. To so uporabniki pametnih telefonov in tabličnih računalnikov na osnovi operacijskega sistema Android od verzije 1.5 naprej. Ob tem predvidevamo, da imajo omogočen dostop do Google Play - Android Market.

Literatura

[1] G. Bonsiepe, *Interface - An Approach to Design*,Jan Van Eyck Akademie, 1999, str. 42,43.

[2] J. Bosak, T. Bray, "XML and the Second-Generation Web",*Scientific American*, zv. 280, št. 5, str. 89-93, maj 1999.

[3] A. Chrisholm, "The Little Book of sudoku", *Michael O'Mara books Limited*, zv. 3, str. 5-6, 2005.

[4] F.Jarvis, B. Felgenhauer, *Enumerating possible Sudoku grids*, Department of Computer Science TU Dresden, 2005, str. 6

[5] W.M. Lee, *Programming Sudoku*, str. 47-68, 2006

[6] P. Manghi, F. Simeoni, D. Lievens, R. C. H. Connor, "Hybrid applications over XML: integrating the procedural and declarative approaches", ACM, str. 9-14, 2002

[7] M. Michael, U. Pajer, *Sudoku 1 : [ure in ure zabave za vaše možgane]*, uvodne strani, 2006.

[8] B.A. Myers, "A brief history of human-computer interaction technology", *Magazine interactions*, zv. 5, št. 2, Marec/April 1998.

[9] W.J.Rayment, "Indepth Sudoku", Dostopno na: http://www.indepthinfo.com/articles/sudoku.shtml

[10] R.Rogers,*Android Application Development: Programming with the Google SDK*, O'Reilly Media, Inc., 2009, str. 10-17.

[11] Android (operacijski sistem). Dostopno na:
http://www.android.com

[12] Eclipse. Dostopno na:
http://www.eclipse.org

[13] Android SDK. Dostopno na:
http://developer.android.com/sdk/index.html

[14] A Detailed Look at the Build Process. Dostopno na:
http://developer.android.com/guide/developing/building/index.html

[15] Sudoku. Dostopno na:
http://www.nikoli.co.jp/en/puzzles/sudoku.html

[16] The History of Sudoku. Dostopno na:
http://www.articlegarden.com/Article/The-History-of-Sudoku/8180

[17] Sudoku - live! Dostopno na:
http://www.guardian.co.uk/culture/culturevultureblog/2005/jun/29/vuturessudoku

[18] G2, home of the discerning Sudoku addict. Dostopno na:
http://www.guardian.co.uk/theguardian/2005/may/13/features11.g2

[19] Brain Age: Train Your Brain in Minutes a Day! Dostopno na:
http://ds.ign.com/articles/702/702057p1.html

[20] Giant Sudoku puzzle's record bid. Dostopno na:
http://www.guardian.co.uk/media/2005/jun/30/advertising.broadcasting

[21] 1st World Sudoku Championship. Dostopno na:
http://www.wsc2006.com/eng/results.php

[22] Java programming language. Dostopno na:
http://www.oracle.com/us/technologies/java/index.html

[23] Extensible Markup Language (XML) 1.0 (Fifth Edition). Dostopno na: http://www.w3.org/TR/2008/REC-xml-20081126/

[24] Android Programming Custom 2D Graphics and Games. Dostopno na http://www3.ntu.edu.sg/home/ehchua/programming/

[25] public class Color. Dostopno na http://developer.android.com/reference/android/graphics/Color.html

[26] Pokritost in hitrost. Dostopno na lhttp://www.mobitel.si/Storitve/Info/Pokritost.aspx

[27] Paketni prenos podatkov. Dostopno na http://www.mobitel.si/storitve/paketni-prenos-podatkov.aspx

[28] Platform Versions. Dostopno na http://developer.android.com/resources/dashboard/platform-versions.html